

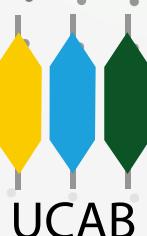


CoNCISa 2017

Sociedad Venezolana de Computación
18 al 20 de octubre 2017

Memorias de la Quinta Conferencia Nacional de
Computación, Informática y Sistemas

Universidad Católica Andrés Bello
Ciudad Guayana, Venezuela



Memorias de la Quinta Conferencia Nacional de Computación, Informática y Sistemas

CoNCISA 2017

**18 al 20 de octubre 2017
Universidad Católica Andrés Bello
Ciudad Guayana - Venezuela**

**Editor
Eric Gamess**



Depósito Legal: DC2017001587
ISBN: 978-980-7683-03-6

@ 2017 - Sociedad Venezolana de Computación
<http://www.svc.net.ve>

Memorias de la Quinta Conferencia de Computación, Informática y Sistemas (CoNCISa 2017).

Depósito Legal: DC2017001587
ISBN: 978-980-7683-03-6

Las contribuciones de CoNCISa 2017 han sido publicadas en la página web de la conferencia.
<http://www.concisa.net.ve/2017>

El diseño gráfico de la portada fue realizado por @CaracasDigital.

Pagad a todos lo que debéis:
al que tributo, tributo; al que impuesto, impuesto; al que respeto, respeto; al que honra, honra.
No debáis a nadie nada, sino el amaros unos a otros; porque el que ama al prójimo, ha cumplido la ley.
(Romanos 13:7-8)

Prólogo

La Sociedad Venezolana de Computación (SVC), con su firme propósito de promover la formación y el desarrollo científico-tecnológico en el área de informática en nuestro país, organiza eventos científicos y educativos de alto nivel como lo son la Escuela Venezolana de Informática (EVI) y la Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa).

EVI es un evento en el que expertos nacionales e internacionales dictan cursos especializados y pertenecientes a diferentes dominios de las ciencias computacionales. Estos cursos cubren temas de interés para la disciplina Informática y son complementarios a las materias y temas que figuran en los currículos de las carreras de computación, Informática y sistemas ofrecidas en el país.

CoNCISa es una conferencia amplia que abarca áreas relacionadas con las Ciencias de Computación, Informática y Sistemas, cuyo objetivo principal es iniciar, en algunos casos, y continuar en otros, con una serie de intercambios periódicos sobre experiencias académicas, tecnológicas y de investigación, para impulsar el desarrollo del área y crear lazos estrechos de cooperación a nivel nacional. En este sentido, también es el espacio para la presentación y discusión de trabajos, resultados y tecnologías, organizadas en sesiones plenarias y técnicas, foros y reuniones especializadas. La conferencia reviste su importancia a nivel nacional, pues recoge a científicos e investigadores a escala nacional en el área de computación, informática y sistemas.

Este año 2017, con el apoyo de las instituciones académicas Universidad Católica Andrés Bello (UCAB), Universidad Simón Bolívar (USB), Universidad Central de Venezuela (UCV), Universidad de Los Andes (ULA) y Universidad Marítima del Caribe (UMC), se han organizado EVI y CoNCISa 2017, en su quinta edición, en la Universidad Católica Andrés Bello de Ciudad Guayana del 16 al 20 de octubre de 2017.

En su quinta edición, CoNCISa continúa creciendo en tamaño e impacto con miras a convertirse en la primera referencia a nivel nacional por su solidez y compromiso científico. Este año se continúa con el concurso nacional de Trabajos Especiales de Grado, en su cuarta edición, con la idea de seguir motivando la investigación en ciencia y tecnología a través de la divulgación de trabajos desarrollados por jóvenes talentos egresados, a nivel nacional, de las carreras de computación, informática y sistemas. Así mismo, se organiza por segunda vez el concurso nacional de Proyectos Socio-Tecnológicos, dedicado a los trabajos de pregrado realizados en las instituciones que gestionan el Programa Nacional de Formación en Informática (PNFI). El objetivo de este concurso es promover, reconocer y divulgar el trabajo final de los estudiantes que hayan culminado recientemente cualquiera de los trayectos del PNFI.

CoNCISa 2017 incluye, al igual que en sus versiones previas, tres categorías de trabajos: (1) Trabajos de Investigación, (2) Trabajos en Tecnologías y Herramientas Computacionales y (3) Trabajos en Gestión de Tecnologías de Información. Estos trabajos tienen formato largo de máximo 12 páginas destinados a reportar investigaciones científicas y aplicadas culminadas o formato corto de máximo 5 páginas destinados a reportar avances de investigación o a generar discusión sobre temas de interés científico.

Prólogo

Se recibieron 21 trabajos en diferentes áreas, de los cuales se aceptaron 14 para presentación oral, que están incluidos en esta memoria. Todos los trabajos enviados fueron evaluados por al menos tres miembros del Comité de Programa, considerando adecuadamente los criterios de valoración establecidos. Al igual que en años anteriores, los autores de los mejores artículos presentados en CoNCISa serán invitados a publicar su trabajo en la Revista Venezolana de Computación, a cargo de la SVC.

La realización exitosa de esta conferencia corresponde al esfuerzo de reconocidos investigadores, profesores y estudiantes del área a nivel nacional e internacional, quienes contribuyeron con su tiempo y dedicación en las actividades de organización, evaluación y edición. Así, nuestro reconocimiento va a todos los autores que confiaron en CoNCISa como una conferencia merecedora de sus excelentes contribuciones, producto de sus trabajos de investigación; a los miembros de los comités de programa y organizador, conferencistas, expositores de tutoriales invitados; a los profesores y estudiantes, y en general, a todos aquellos que aportaron su valioso apoyo para llevar a buen término esta importante responsabilidad, manteniendo en alto la relevancia de nuestra conferencia nacional; a todos, muchas gracias.

Wilmer Pereira

Universidad Católica Andrés Bello

Junior Altamiranda

Universidad de Los Andes

Esmitt Ramírez

Universidad Central de Venezuela

Presidentes del Comité de Programa de CoNCISa 2017

La SVC (Sociedad Venezolana de Computación) está comprometida con el impulso de una nueva generación académica y profesional en nuestra área de saber para el desarrollo del país

Comité de Programa

Presidentes del Comité de Programa

Wilmer Pereira	Universidad Católica Andrés Bello – Venezuela
Junior Altamiranda	Universidad de Los Andes – Venezuela
Esmitt Ramírez	Universidad Central de Venezuela – Venezuela

Miembros del Comité de Programa

Judith Barrios	Universidad de Los Andes – Venezuela
Isabel Besembel	Universidad de Los Andes – Venezuela
Blai Bonet	Universidad Simón Bolívar – Venezuela
José Cadenas	Universidad Simón Bolívar – Venezuela
Jacinto Dávila	Universidad de Los Andes – Venezuela
Ernesto Exposito	Université de Pau et des Pays de l'Adour – Francia
Hendry Ferreira	Universidade Federal do Rio Grande – Brasil
Eric Gamess	Universidad Central de Venezuela – Venezuela
Anibal Ghanem	Universidad Bolivariana de Venezuela – Venezuela
Ana Leguizamo	Universidad Central de Venezuela – Venezuela
Ricardo Monascal	Universidad Simón Bolívar – Venezuela
Haydemar Núñez	Universidad Central de Venezuela – Venezuela
Dinarle Ortega	Universidad de Carabobo – Venezuela
Andres Sanoja	Universidad Central de Venezuela – Venezuela
Eugenio Scalise	Universidad Central de Venezuela – Venezuela
Leonid Tineo	Universidad Simón Bolívar – Venezuela

Comité Organizador

Presidente del Comité Organizador

Cora Urdaneta

Universidad Católica Andrés Bello

Miembros del Comité Organizador

Livia Borjas	IUT Federico Rivero Palacio – Venezuela
Nataly Carmona	Universidad Marítima del Caribe – Venezuela
David Coronado	Universidad Simón Bolívar – Venezuela
Eric Gamess	Universidad Central de Venezuela – Venezuela
Yosly Hernández	Universidad Central de Venezuela – Venezuela
Rosseline Rodríguez	Universidad Simón Bolívar – Venezuela
Leonid Tineo	Universidad Simón Bolívar – Venezuela
Milad Zaghab	Universidad Católica Andrés Bello – Venezuela

IV Concurso Nacional de Trabajos Especial de Grado

El IV Concurso Nacional de Trabajos Especial de Grado (pregrado) es un evento promovido por la Sociedad Venezolana de Computación (SVC) que se realizó en el marco de la V Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa 2017). El objetivo del concurso es promover el trabajo final de carrera de estudiantes que hayan terminado recientemente sus estudios en computación, informática y sistemas.

El concurso fue presidido por el profesor Junior Altamiranda (ULA). En esta cuarta versión del concurso, se invitaron a varias universidades nacionales a postular su mejor TEG defendido entre el 01 de julio de 2016 y el 31 de julio de 2017, en las carreras antes mencionadas. Se recibieron postulaciones de un total de siete TEGs, provenientes de distintas universidades del país.

Las evaluaciones de todos los TEGs participantes en el concurso estuvieron todas cercanas y en niveles muy altos, ocupando los tres primeros lugares los siguientes:

1. “Sistema Inteligente de Transporte Distribuido basado en Redes Mesh” por Ronald Oribio, de la Universidad Católica Andrés Bello, Caracas
2. “Reconocimiento de Emociones en un Conductor de Vehículo Usando el Paradigma de Crónicas” por Kristell Aguilar, de la Universidad de Los Andes, Mérida
3. “Asistente Virtual para Entrenamiento de Ciclismo Estático” por Manuel Lozada, de la Universidad Yacambú, Cabudare.

Los TEGs premiados fueron expuestos por sus autores en una sesión especial de CoNCISa 2017. A continuación se presenta un resumen de cada uno de estos trabajos.

IV Concurso Nacional de Trabajos Especial de Grado: Primer Premio

Título: Sistema Inteligente de Transporte Distribuido basado en Redes Mesh.

Autor: Ronald Oribio.

Tutor: Valeria León.

Afiliación: Universidad Católica Andrés Bello.

Resumen: Uno de los problemas del crecimiento urbano es la congestión de tránsito. Según la Cámara Automotriz Venezolana, para el año 2009, la velocidad promedio de un automóvil en el Área Metropolitana de Caracas (AMC) oscilaba entre 11 y 15 kilómetros por hora. El recorrido promedio era de 45 minutos, siendo el viaje más corto de 10 minutos y los más largos de hasta tres horas. Para el 2016, de 186 ciudades, Caracas se encontraba en la posición 25 entre las ciudades con peor congestión de tránsito del mundo, según el índice de satisfacción del conductor publicado por Waze Mobile. La congestión ha alcanzado tales niveles que una reducción del tráfico de tan sólo media hora diaria tendría un valor para la ciudad del orden de Bs. 1 billón por año, aproximadamente US\$2 millardos al tipo de cambio oficial expuesto por el Sistema Complementario de Administración de Divisas (SICAD I).

En respuesta, los gobiernos suelen intentar aplicar diversas medidas pero, según un estudio realizado por la Universidad de Texas, mantener informados a los conductores sobre las condiciones de tránsito puede reducir la congestión mucho más que otras medidas. Partiendo de esta idea, se propuso como solución desarrollar un sistema inteligente de transporte (SIT) descentralizado y de bajo coste, que asista a los conductores en hallar la ruta menos congestionada, con el fin de disminuir el tiempo de recorrido desde su punto de partida hasta su destino.

Inspirado en el forraje de las hormigas y utilizando una metodología basada en Scrum y Prototipado Evolutivo, se desarrolló un dispositivo inteligente fundado en la plataforma Raspberry Pi B+ que cuenta con WiFi y GPS, el cual es instalado en los vehículos. Al desplegarse, estos forman una red Mesh, que sirve para compartir información sobre el estado de la congestión, que luego es comunicado al conductor. Para ello, se evaluaron, diseñaron e implementaron los componentes de hardware y software necesarios.

Utilizando modelos de simulación por computadora y pruebas de campo bajo condiciones reales, se demostró el funcionamiento del sistema y su impacto sobre la congestión, reduciéndola en un 8.05%. Como resultado del trabajo, se logró demostrar que es posible implementar un sistema inteligente de transporte distribuido de bajo costo. Se obtuvo como producto final un prototipo que demuestra las capacidades, limitaciones y, más importante aún, el potencial de este tipo de soluciones.

Palabras Clave: Sistemas Inteligentes de Transporte; Redes Mesh; Optimización por Colonia de Hormigas.

IV Concurso Nacional de Trabajos Especial de Grado: Segundo Premio

Título: Reconocimiento de Emociones en un Conductor de Vehículo Usando el Paradigma de Crónicas.

Autor: Kristell Aguilar.

Tutor: Jose Aguilar.

Afiliación: Universidad de los Andes.

Resumen: Uno de los tópicos de investigación en el dominio automotriz tiene que ver con los sistemas de asistencia al conductor (ADAS). Con la popularidad de estos sistemas, y el interés actual prestado a la interacción hombre-máquina, el problema de adaptación de los ADAS a las características de cada conductor se hace importante. Para que los ADAS logren adaptarse a los conductores, es necesario que cuenten con sistemas adaptativos que puedan considerar las características internas de cada ser humano, tales como, sus estados afectivos y emocionales. Este trabajo plantea como objetivo principal, el diseño de un mecanismo para el reconocimiento de las emociones del conductor de un vehículo, basado en el paradigma de crónicas. Para ello, se propone un modelo jerárquico del estilo de conducción, compuesto de tres niveles, uno para describir el estado emocional, otro para el estado del conductor y, finalmente, otro para el estilo de conducción. Cada nivel se define por distintos tipos de descriptores, requieren de un enfoque multimodal (sonido, visión, etc.) para ser percibidos, y están relacionados entre ellos, porque unos son descriptores de otros. Como se mencionó antes, se utiliza para reconocer los patrones de cada nivel del modelo jerárquico del estilo de conducción, el paradigma de lógica temporal llamado crónica, el cual modela patrones caracterizados por eventos observables, con relaciones temporales entre ellos. Además, en el trabajo se desarrolla un algoritmo de aprendizaje de crónicas, se usa el lenguaje CQL para la descripción de las crónicas descubiertas, y se implementa la base de crónicas en el bus de servicios OpenESB. Finalmente, se compara el paradigma con otras técnicas de reconocimiento, una basada en lógica difusa y la otra basada en un algoritmo que modela el funcionamiento del neocórte x humano, llamado Ar2P. Se analizan las técnicas, evaluando sus capacidades para razonar sobre el conductor, para adaptarse al conductor, o para ser usado en la Internet de las Cosas (IoT). Como resultados tenemos, que el algoritmo de aprendizaje es capaz de aprender crónicas adaptadas a cada conductor, y que se puede modelar una misma situación (por ejemplo, una dada emoción) con distintas crónicas, expresando la diversidad de contexto en las que una misma situación puede ocurrir. El algoritmo de aprendizaje de crónicas propuesto realiza el descubrimiento de crónicas, y es flexible, ya que puede ajustarse a nuevos descriptores y nuevas situaciones que surjan. Con respecto a la capacidad de razonamiento, las crónicas permiten reconocer situaciones de diferentes maneras: la misma situación con diferentes patrones, situaciones caracterizadas por patrones atómicos (por ejemplo, solo emociones), o situaciones más complejas (descritas por patrones compuesto por patrones atómicos). Finalmente, la comunicación en crónicas es a través de la transmisión de eventos que describen un cambio de valor de un descriptor, o información relevante reconocido en un conductor (una emoción, un estado o un estilo), generados por los distintos descriptores, o por las propias crónicas reconocidas.

Palabras Clave: Sistemas de Asistencia al Conductor; Patrón Jerárquico de Estilo de Conducción; Reconocimiento de Patrones; Reconocimiento de Emociones; Crónicas; Inteligencia Artificial; Computación Afectiva.

IV Concurso Nacional de Trabajos Especial de Grado: Tercer Premio

Título: Asistente Virtual para Entrenamiento de Ciclismo Estático.

Autor: Manuel Lozada.

Tutor: Leoncio Gómez.

Afiliación: Universidad Yacambú

Resumen: Se desarrolló un asistente virtual para entrenamientos de ciclismo estático, el cual cumple la función de recopilar información de cada práctica realizada por un ciclista. Asimismo, éste es capaz de proporcionar a sus usuarios una nueva forma de realizar este deporte, al poder utilizar de forma estática sus bicicletas que normalmente utilizan para rodaje en vías. De este modo, el principal objetivo del sistema electrónico es la recolección de las señales generadas en la práctica, para a su vez ser procesadas y obtener variables tales como velocidad, distancia, tiempo, velocidad promedio y calorías quemadas aproximadamente, donde son posteriormente enviadas al equipo de visualización. La adquisición de los datos es proporcionada por la implementación de una serie de sensores que se encargan de captar las señales para proporcionar resultados, que son analizados por la unidad central de procesamiento.

El proyecto realizado aporta de gran manera a la sociedad, debido a que promueve el deporte en la vida diaria de las personas que no poseen de mucho tiempo para realizar entrenamientos deportivos. Con esto, se está impulsando a una sociedad con un estilo de vida saludable, de forma que se puedan evitar futuras enfermedades como lo es la obesidad, que podría a su vez traer otros tipos de problemas consecuentes tanto físicos como psicológicos. De este modo, se está previniendo este tipo de problemas tanto al usuario que adquiera este asistente como su entorno familiar, que redundaría también en lo económico, por tratarse de que al evadir estas enfermedades se estaría evitando realizar grandes gastos por tratamientos médicos.

Aunado a esto, el dispositivo que se expone promueve la seguridad de la sociedad, debido a que permite realizar los entrenamientos en lugares seguros como el hogar. En lo académico, sirve de base para posteriores investigaciones que busquen añadir otro tipo de variables a controlar y elaborar análisis más profundos a los datos obtenidos como podrían ser oxigenación, temperatura, entre otros.

Palabras Clave: Ciclismo; Bicicleta; Asistente Virtual; Sistema de Adquisición de Datos.

II Concurso Nacional de Proyectos Socio-Tecnológicos

El II Concurso Nacional de Proyectos Socio-Technológicos es un evento promovido por la Sociedad Venezolana de Computación (SVC) que se realizó en el marco de la V Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa 2017). El objetivo del concurso es promover, reconocer y divulgar el trabajo final de los estudiantes que hayan culminado recientemente cualquiera de los trayectos del PNFI (Programa Nacional de Formación en Informática).

El concurso fue presidido por el profesor Junior Altamiranda (ULA). En esta segunda versión del concurso, se invitaron a varios institutos nacionales a postular su mejor PST defendido entre el 01 de julio de 2016 y el 31 de julio de 2017. Se recibieron postulaciones de un total de ocho PSTs.

Las evaluaciones de todos los PSTs participantes en el concurso estuvieron todas cercanas y en niveles muy altos, ocupando los dos primeros lugares los siguientes:

1. “Aplicación Web para la Automatización del Proceso de la Recuperación de Vehículos del Cuerpo de Policía Bolivariana del Estado Zulia” por Andreina Rosales, Yoselyn Parra, Willis Barriga y Alejandro García, del Instituto Universitario de Tecnología de Maracaibo, Maracaibo
2. “Sistema de Información en Apoyo a la Toma de Decisiones para la Gestión de Dotación y Mantenimiento de Comunidades Escolares Adscriptas a la Oficina de FEDE, Cumaná, Estado-Sucre” por Loreanny Osorio, Juan Rivero y Enrique Rodriguez, de la Universidad Politécnica Territorial del Oeste de Sucre “Clodosbaldo Russián”, Cumaná.

Los PSTs premiados fueron expuestos por sus autores en una sesión especial de CoNCISa 2017. A continuación se presenta un resumen de cada uno de estos trabajos.

II Concurso Nacional de Proyectos Socio-Tecnológicos: Primer Premio

Título: Aplicación Web para la Automatización del Proceso de la Recuperación de Vehículos del Cuerpo de Policía Bolivariana del Estado Zulia.

Autores: Andreina Rosales, Yoselyn Parra, Willis Barriga, Alejandro García.

Tutores: Alonso Huerta, Mary Suarez.

Afiliación: Instituto Universitario de Tecnología de Maracaibo.

Resumen: Actualmente los organismos policiales llevan el control de cada uno de sus procesos internamente, en tal sentido, lo mismo ocurre con la recuperación de vehículos donde la información es manejada por cada ente encargado de la recuperación donde se puede observar retardo en tiempos de respuesta a los distintos usuarios. Al momento de realizar una consulta de algún dato de un vehículo recuperado, solo se obtiene como respuesta la manejada internamente lo cual dificulta o retrasa la información requerida al momento de la consulta y si la solicitud no aparece registrada en dicha entidad el propietario del automóvil debe acudir a otros Centros Policiales en búsqueda de la procedencia del mismo, lo que se puede traducir en pérdida de tiempo además de un manejo inaccesible de la información por falta de una centralización en las bases de datos.

Así mismo, se propone desarrollar una aplicación web para la automatización del proceso de la recuperación de vehículos del Cuerpo de Policía Bolivariana del Estado Zulia, que permita mejorar el mismo registrando de forma centralizada los datos correspondientes a este, obteniendo como beneficio el manejo de la información de forma organizada, a fin de mejorar significativamente los procesos de registro, consulta y manejo de la información del mismo creando a su vez bases para la estandarización de la información entre cada cuerpo policial fomentando en si la interoperabilidad entre las distintas organizaciones involucradas.

Para ello, es necesario considerar que para lograr el desarrollo del mismo, es de vital importancia diagnosticar la situación actual del proceso de recuperación de vehículos que maneja el Cuerpo de Policía Bolivariana del Estado Zulia. De igual manera, determinar los requerimientos funcionales y no funcionales de la aplicación web, así mismo, diseñar y construir la aplicación según los requerimientos y necesidades de los usuarios para culminar con la verificación de la misma a través de pruebas correspondientes.

Siguiendo el orden de ideas, se pueden mencionar como resultado de la realización de dicha investigación la interoperabilidad entre las organizaciones involucradas, ofreciendo seguridad y transparencia en el manejo de la información, al igual que confidencialidad e integridad en los datos, en tal sentido, se ofrece una aplicación 100% operacional y funcional ahorrando el tiempo de respuesta y de trabajo.

Palabras Clave: Aplicación Web; Servicios; Interoperabilidad.

II Concurso Nacional de Proyectos Socio-Tecnológicos: Segundo Premio

Título: Sistema de Información en Apoyo a la Toma de Decisiones para la Gestión de Dotación y Mantenimiento de Comunidades Escolares Adscriptas a la Oficina de FEDE, Cumaná, Estado-Sucre.

Autores: Loreanny Osorio, Juan Rivero, Enrique Rodriguez.

Tutores: Leomarys Semidey.

Afilación: Universidad Politécnica Territorial del Oeste de Sucre “Clodosbaldo Russián”.

Resumen: La Fundación de Edificaciones y Dotaciones Educativas (FEDE) es una institución pública que tiene como finalidad dar respuestas a la problemática de la planta física educativa a escala nacional, atendiendo de forma sistemática las etapas correspondientes a la construcción, dotación, rehabilitación y mantenimiento del conjunto escolar. Por ello, el presente proyecto socio-tecnológico tuvo como objetivo el desarrollo de un Sistema de Información en apoyo a la toma de decisiones para la gestión de dotación y mantenimiento de comunidades escolares adscriptas a la Oficina de FEDE Cumaná, Estado-Sucre y de esta forma mejorar y agilizar la toma de decisiones en el proceso administrativo que se lleva a cabo a la hora de realizar el mantenimiento o dotación de algún plantel perteneciente a FEDE Cumaná. Este estudio corresponde a una investigación de tipo proyecto factible ya que se desarrolla una propuesta viable para solucionar los requerimientos y necesidades de la Institución. Se implementó un diseño de investigación de campo y descriptiva empleando como elementos centrales para la recolección de información la observación y encuestas al personal que labora en FEDE Cumaná. Para el desarrollo del sistema, se utilizó la Metodología de Diseño Hipermedia Orientada a Objeto (OOHDM). Con este proyecto fue beneficiada la oficina FEDE Cumaná, ya que el sistema en apoyo a la toma de decisiones para la gestión de dotación y mantenimiento de los planteles educativos adscritos a la misma, arroja una serie de indicadores que sirven como señal de alerta para encaminar la actuación de la institución conduciendo a un mayor rendimiento y efectividad en sus operaciones.

Palabras Clave: FEDE Cumaná; Metodología OOHDM; Indicadores.

Tabla de Contenido

Prólogo	iii-iv
Comité de Programa	v-v
Comité Organizador	vi-vi
Concurso Nacional TEGs	vii-x
Concurso Nacional de PSTs	xi-xiii
 1. Sesión de Investigación – Artículos Largos	
1.1. Hacia Aplicaciones Web con Términos Difusos	2-12
José Labbad, Rosseline Rodríguez, Leonid Tineo	
1.2. Derivación del Sistema Moodle a Partir de una Arquitectura de Referencia para el Aprendizaje Electrónico	13-24
Francisca Losavio, Yuly Esteves	
1.3. Análisis Comparativo de Técnicas de Modelado de Metas: Un Enfoque Dirigido por la Planificación Estratégica del Negocio	25-36
Mayela Delgado, Alfredo Matteo, Francisca Losavio	
1.4. Juegos de Entrenamiento Mental bajo un Ambiente de Realidad Virtual Inmersiva	37-48
Daniel Sam, Esmitt Ramírez	
1.5. Filtrado 2D de Imágenes Médicas Basado en Filtros Myriad Ponderados Recursivos	49-59
Liliana Bautista, Juan Ramírez	
1.6. Lineamientos para el Despliegue de Redes SDN/OpenFlow	60-72
Gustavo Pereira, Eric Gamess	
1.7. EAW: Evaluador de Criterios de Accesibilidad Web para Pautas Relacionadas con Discapacidad Visual y Discapacidad Motora	73-81
Yusneyi Carballo, María Acosta, Ronald Aguilera	
1.8. Propuesta para la Gestión de Proyectos Socio Tecnológicos del Programa Nacional de Formación en Informática	82-93
Iris Albaran, Liliana Silva, Colombia Amezquita, Marbella Castañeda	
1.9. Propiedades Algebraicas y Decidibilidad del Transformador de Predicados wp sobre la Teoría de Conjuntos	94-105
Federico Flaviani	

Tabla de Contenido

2. Sesión de Investigación – Artículos Cortos	
2.1. Una Propuesta para el Ordenamiento de Datos Bimodales	107-112
José Cadenas, Soraya Carrasquel, David Coronado, Rosseline Rodríguez, Leonid Tineo	
3. Sesión de Tecnologías y Herramientas Computacionales - Artículos	
3.1. Metodología para el Desarrollo de Aplicaciones Web Utilizando Datos Enlazados	114-122
Taniana Rodríguez, Ricardo Dos Santos, Jose Aguilar	
3.2. Multiobjective Optimization Model using Genetic Algorithms for Oil Wells based on Electro Submersible Pumps	123-127
Edgar Camargo, Egner Aceros, Lice Guzmán, Hector Lozada, Jose Aguilar	
3.3. Diseño del Modelado Matemático para el Diagnóstico de Pozos de Extracción por Bombeo de Cavidades Progresivas	128-133
Edgar Camargo, Lice Guzmán, Egner Aceros, Mariuska Briceño	
3.4. Algoritmo Adaptativo para la Estimación por Series de Tiempo Usando Árboles de Decisión	134-140
Alfredo Naime	
Índice de Autores	142-142



Sesión de Investigación

Artículos Largos

Hacia Aplicaciones Web con Términos Difusos

José Ángel Labbad
Universidad Metropolitana
Caracas, Venezuela
joseangellabbad@gmail.com

Rosseline Rodríguez
*Departamento de Computación y Tecnología
de la Información*
Universidad Simón Bolívar
Caracas, Venezuela
crodrig@usb.ve

Leonid Tineo
*Departamento de Computación y Tecnología
de la Información*
Universidad Simón Bolívar
Caracas, Venezuela
leonid@usb.ve

Resumen— En la actualidad, casi cualquier ámbito de la vida cuenta con una aplicación Web relacionada que dispone de formularios de búsqueda. Lamentablemente, estas aplicaciones tienen problemas de rigidez propios del uso subyacente de la lógica booleana, los cuales se pueden superar gracias a la existencia de lenguajes de consulta a bases de datos, extendidos con lógica difusa. Así, los formularios de búsqueda pueden enriquecerse usando términos difusos en los controles Web. En este trabajo se hace una propuesta metodológica para abordar la migración de una aplicación Web existente hacia una nueva que haga uso de formularios Web con términos difusos, superando los problemas de rigidez de las aplicaciones tradicionales. Para ello, se realizó un estudio de opinión a programadores y usuarios Web para conocer si los programadores deseaban utilizar términos difusos en sus desarrollos, y si los usuarios se inclinaban a realizar búsquedas en formularios Web que proveyeran de estos términos para expresar sus preferencias. Posteriormente, se diseña un mecanismo sistemático que permitiera analizar los controles Web existentes en los formularios de búsqueda de una aplicación para incorporarles términos difusos en sus opciones de entrada de datos. Se evaluaron las diferentes extensiones con lógica difusa de los SGBD que existen actualmente, en base a criterios como completitud, disponibilidad, visibilidad, calidad, soporte, portabilidad y documentación. Se seleccionó SQLfi como la que satisface la mayoría de estos criterios. Finalmente, se plantea una propuesta metodológica para la migración de un sitio Web con el fin de que sus formularios de búsqueda usen términos difusos presentes en el lenguaje natural.

Palabras Clave— *formularios Web; términos difusos; controles Web; lenguaje natural; migración*

I. INTRODUCCIÓN

En la actualidad los usuarios de aplicaciones Web demandan mayor flexibilidad y adaptación a sus requisitos o preferencias. La mayoría de estas aplicaciones cuentan con formularios de búsqueda para dar acceso a los datos que ellas gestionan, los cuales en muchas oportunidades presentan rigidez en la adaptación a las demandas de los usuarios.

Para flexibilizar la gestión de datos y búsquedas a través de formularios Web [1], se ha propuesto la lógica difusa [2] como herramienta para la especificación de términos del lenguaje natural. Esta lógica [2] fue definida en base a la teoría de conjuntos difusos [3], propuesta por Zadeh, como mecanismo para dar tratamiento matemático y computacional a los términos vagos del lenguaje natural. Es por ello que se han hecho estudios dirigidos a definir lenguajes que permitan realizar consultas difusas a bases de datos relacionales [4].

Estos lenguajes son extensiones al lenguaje estándar para consultas a bases de datos, llamado SQL (*Structured Query Language*) [5], los cuales se han incorporado en los Sistemas Gestores de Bases de Datos (SGBD) más usados [6][7][8][9]. Algunas de estos SGBD extendidos están disponibles para ser utilizados libremente, sin embargo, no se observa que se haya masificado su uso a nivel de aplicaciones Web. Por otro lado, algunas de las extensiones de SQL con lógica difusa cuentan con implementaciones en distintos SGBD, sea en el código fuente o mediante una capa lógica. Sin embargo, aún no existe un número significativo de aplicaciones que aprovechen las ventajas de trabajar con lógica difusa usando un SGBD.

En este trabajo, se muestra el resultado de un estudio que valida la factibilidad de usar términos difusos en formularios Web, por parte de programadores y usuarios. Además, la principal contribución de este trabajo es una propuesta sistemática y metodológica que permite transformar los formularios de búsqueda existentes dentro de una aplicación Web operativa a una versión modificada que permita la gestión de términos lingüísticos vagos en dicho formularios usando lógica difusa.

En tal sentido el trabajo se estructura como sigue. En la Sección II se plantean los antecedentes de la investigación. En la Sección III, se exponen los conceptos que fundamentan la investigación. En la Sección IV, se muestra un estudio de opinión realizado a programadores y usuarios de aplicaciones Web sobre la posibilidad de realizar búsquedas online usando términos difusos. En la Sección V, se explica el proceso metodológico a seguir para incorporar términos difusos a formularios Web. En la Sección VI, se propone la metodología de migración de formularios Web tradicionales para que incluyan términos difusos. Finalmente, se presentan las conclusiones y trabajos futuros.

II. ANTECEDENTES

A pesar del reciente surgimiento de tendencias NO-SQL [10], hoy en día, el lenguaje de consultas a bases de datos más utilizado a nivel mundial sigue siendo SQL [11]. Gracias a su uso masivo en diversas áreas de aplicación, la perspectiva es que SQL siga siendo ampliamente usado en el futuro.

La incorporación de lógica difusa como mecanismo para flexibilizar los lenguajes de consultas a base de datos se ha venido estudiando por varios años. Las propuestas más completas son SQLf [7] y FSQL [8]. Usando SQLf como lenguaje de consulta, en [12] se desarrollaron aplicaciones basadas en lógica difusa para expresar las preferencias de

usuario: un sistema de evaluación de cursos docentes, un sistema de promoción de personal, y un sistema de compra y venta de vehículos. Estas aplicaciones fueron implementadas sobre una extensión en capa lógica del SGBD Oracle denominada SQLfi [12].

En base a esa experiencia y otras previas, en [13] se definieron elementos gramaticales y características que determinan cuando una aplicación posee requisitos difusos, es decir, aquellos que poseen términos lingüísticos vagos. En este trabajo se definen parámetros que ayudan a establecer cuándo una aplicación podría aprovechar el uso de la lógica difusa dentro de un SGBD. Por otro lado, se dan lineamientos para que los desarrolladores puedan identificar, en los requisitos escritos en lenguaje natural, los términos de la lógica difusa que luego serán usados en la definición de consultas a la base de datos.

Posteriormente, en [14] se estudió la incorporación de los beneficios de la lógica difusa en las metodologías de desarrollo de software a fin de expresar requisitos que involucran preferencias de usuario. En este trabajo, se añade al ciclo de vida de desarrollo de software, en las fases de análisis, diseño e implementación, los aspectos necesarios para que las aplicaciones usen consultas difusas sobre bases de datos.

En [15] se propuso un método formal para implementar requisitos difusos dentro de sistemas de software que soporten consultas difusas. De tal forma que, utiliza el cálculo relacional de tuplas para especificar formalmente las consultas difusas, además, provee reglas que permiten traducir una especificación formal a una consulta escrita en el lenguaje SQLf.

Por otro lado, en [16] se propone un perfil UML para representar requisitos difusos a base de datos relacionales. Este perfil hace uso de OCL (*Object Constraint Language*) extendido con condiciones difusas, a fin de completar los diferentes artefactos de la notación UML con requisitos difusos expresados formalmente, lo cual facilita el proceso de traducción a un lenguaje de consultas como SQLf. En [17] se define un algoritmo de traducción de requisitos difusos en OCL a consultas difusas en SQLf. Este algoritmo es de fácil implementación sobre un SGBD difuso, como SQLfi [12] o PostgreSQLf [6].

Los trabajos anteriores [16][17] fueron considerados en la construcción de un *framework* de persistencia [18] que permite el desarrollo de aplicaciones con consultas difusas, el cual es una extensión de Spring, *framework* muy usado actualmente en la comunidad de desarrolladores. Esta extensión puede ser incorporada dentro de ambientes de trabajo, así como, ser integrada con librerías, para el desarrollo de aplicaciones Java.

A pesar que los trabajos aquí mencionados muestran el creciente interés del uso de la lógica difusa como mecanismo para incorporar flexibilidad en diversos sistemas de software sobre bases de datos, ninguno de esos trabajos ha sido enfocado directamente a los formularios de búsqueda presentes en las aplicaciones Web, las cuales tienen mucho impacto en la actualidad. Además, la incorporación de información imprecisa e incierta en el modelo de base de datos sigue siendo un tema importante de investigación debido a que existen datos imprecisos e inciertos en muchas de las aplicaciones del mundo

real [19]. Por tal razón, es razonable pensar que incorporar términos difusos en los formularios de búsqueda como funcionalidad agregada dentro de las aplicaciones Web puede resultar de impacto en el presente.

III. MARCO TEÓRICO

En esta sección se describe la teoría de los Conjuntos Difusos que es la base matemática sobre la cual se define la lógica difusa, de manera que permita especificar términos del lenguaje natural. Además, se da un panorama general de los elementos presentes en los formularios de búsqueda en la Web, conocidos como controles Web.

A. Conjuntos Difusos

El lenguaje natural que usamos a diario está lleno de imprecisión. Muy rara vez se habla de temperaturas exactas como “38 grados”, más bien se emplean frases como “está caliente”, “hace frío” o “tiene fiebre alta”. Si se opine sobre la edad de una persona es común decir que es “joven”, “adulta” o “anciana” en lugar de un valor preciso. Al comentar sobre distancias recorridas se indica si algún lugar está “lejos” o “cerca”. Para enfatizar algo se usan adverbios dentro de frases imprecisas como “es muy joven” o “está extremadamente lejos”. En cuanto a cantidades, es común usar términos como “pocos”, “la mayoría” o “muchos”. Estos términos del lenguaje natural son mayormente adjetivos o adverbios que se caracterizan por ser impresos, vagos o difusos [13].

Una herramienta matemática que permite modelar estos términos es la Teoría de Conjuntos Difusos [3]. Un **conjunto difuso** permite membresía gradual. En los conjuntos clásicos, se dice si un elemento pertenece o no al conjunto. En los conjuntos difusos, la pertenencia de un elemento x viene dada por un grado de membresía $\mu(x)$ cuyo valor está en el intervalo $[0,1]$. Entonces, si el grado es cero el elemento está “completamente excluido”, y si el grado es 1, el elemento está “completamente incluido”. En todos los demás casos la pertenencia es gradual. Al subconjunto de los elementos que están completamente incluidos se les conoce como **núcleo**, a los que no están completamente excluidos (grado de pertenencia mayor que 0), pero tampoco son completamente incluidos se les conoce como **borde**. El **soporte** es la unión del núcleo y el borde.

Para especificar un conjunto difuso, se puede usar una notación trapezoidal de la forma (x_1, x_2, x_3, x_4) , donde $x_1 \leq x_2 \leq x_3 \leq x_4$ señalan los puntos de inflexión del trapecio. También, se puede especificar un conjunto difuso con una notación por extensión $\{\mu(x_1)/x_1, \dots, \mu(x_n)/x_n\}$, donde cada elemento x_i del dominio se acompaña de su grado de membresía $\mu(x_i)$. Estos grados son colocados según la preferencia del usuario.

B. Términos Difusos

Fundamentada en los conjuntos difusos, la lógica difusa permite expresar condiciones con términos difusos, que pueden usarse al realizar consultas o búsquedas sobre un grupo de datos. En esta lógica los valores de verdad son representados por números reales comprendidos en el intervalo $[0,1]$. Así el cero (0) significa completamente falso y el uno (1) significa

completamente cierto, y los demás valores del intervalo representan el grado de verdad obtenido para una proposición que se está evaluando.

El operador lógico negación se interpreta como el complemento de conjuntos, la conjunción como la intersección y la disyunción como la unión. El grado de membresía de un elemento x en el complemento del conjunto difuso F se calcula como $1-\mu_F(x)$. El grado de membresía para el conjunto intersección se calcula mediante una norma triangular, que es operador binario, cerrado en $[0,1]$, conmutativo, con elemento neutro uno (1). A cada norma triangular le corresponde una co-norma, que es el concepto dual, la cual se usa para el cálculo de la unión. Usualmente se adopta la norma triangular mínimo (\min) y su co-norma máximo (\max).

Las expresiones en lógica difusa se construyen utilizando términos lingüísticos, tales como predicados, modificadores, comparadores, conectores y cuantificadores [20].

Los **predicados** [13] representan adjetivos del lenguaje natural, que corresponden a los componentes atómicos de la lógica difusa, los cuales pueden ser definidos con un conjunto difuso en tres formas diferentes: una función trapezoidal, una expresión con rango en el intervalo $[0,1]$ y por extensión indicando para cada valor del dominio del predicado su respectivo grado de verdad. Por ejemplo, los adjetivos “joven”, “adulto” y “anciano” para la edad, pueden ser definidos de forma trapezoidal como se observa en la Figura 1. En este caso, el usuario especifica el predicado “joven” con el trapecio $(0, 0, 20, 40)$, “adulto” con el trapecio $(20, 40, 60, 80)$, y “anciano” con el trapecio $(40, 60, 100, 100)$.

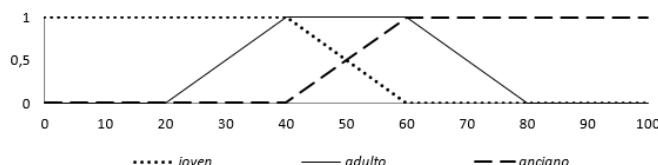


Figura 1. Definición Trapezoidal de los Términos Difusos *Joven*, *Adulto* y *Anciano*

Otro ejemplo, donde se usa una definición por extensión, sería el predicado “agradable”, relacionado al universo de los colores = {amarillo, verde, rojo, azul, negro}, puede representarse con el conjunto difuso *agradable* = $\{0.4/\text{amarillo}, 1/\text{verde}, 0.1/\text{rojo}, 0.7/\text{azul}, 0.3/\text{negro}\}$. Un ejemplo, que usa una expresión con rango en el intervalo $[0,1]$ es el predicado “alto” definido con la expresión $x/250$, siendo x un valor en el rango $0 .. 250$.

Los **modificadores** [13] son adverbios que sirven para expresar una propiedad adicional sobre un predicado, las cuales intensifican, relajan, desplazan o invierten el concepto que representa el predicado. Se pueden definir de tres formas: a través de una potencia a la cual será elevado el valor resultante de un predicado difuso, a través de un valor de traslación que indica lo que le será sumado o restado (según sea el caso) al valor de entrada de un predicado y con una función de norma. Por ejemplo, el modificador “extremadamente” pudiera representarse con la potencia al cubo del valor de la función de membresía, y el modificador “muy” como una traslación negativa.

Los **comparadores** [13] permiten definir una comparación para dos argumentos recibidos. Los términos lingüísticos que los representan son adjetivos de grado comparativo o adverbios. Se pueden definir de dos formas: trapezoidal, si se especifica una expresión con dos variables, cuyo resultado luego de ser evaluada cae en el conjunto difuso definido por un trapecio; o por medio de una relación difusa, donde se especifica para cada par de valores su valor de verdad. Por ejemplo, se podría definir el comparador “cercano”, para los valores en el intervalo $[-30,30]$, con la expresión $x-y$; donde el resultado obtenido se evalúa en la función de membresía representada por el trapecio $(-20, -1, 1, 20)$. Un segundo ejemplo es definir el comparador “similar” para colores, usando la relación difusa $\{(negro,gris)/0.5, (azul,gris)/0.7, (blanco,gris)/0.5, (azul,negro)/0.8, (azul,blanco)/0.2\}$. Aquí se indica que el negro es similar a gris en un grado de 0.5, mientras que el azul es similar a gris en un grado de 0.7.

Los **conectores** [13] son operadores lógicos que puede definir el usuario, para combinar dos condiciones difusas. Por ejemplo, la negación, la conjunción y la disyunción clásicas se pueden extender de forma difusa, preservando su correspondencia con los operadores de conjunto: complemento, intersección y unión. Así, el $x \Rightarrow y$ puede definirse con la expresión $\max(1-x, y)$.

Los **cuantificadores** [13] permiten describir cantidades difusas. Son representados por adverbios superlativos o cuantitativos, o también, con frases imprecisas que expresan cantidad. Pueden definirse de forma absoluta o proporcional dependiendo de los valores facilitados, usándose en ambos casos la forma trapezoidal. Los cuantificadores clásicos, existencial y universal, se mantienen como cuantificadores difusos. Por ejemplo, un cuantificador absoluto podría ser la frase “porlomenos3” representado por la función trapezoidal $(3, 8, \infty, \infty)$. Un cuantificador relativo podría ser “muchos” representado por el trapecio $(0.75, 0.90, 1, 1)$.

C. Formularios Web

Para interactuar con un sitio o página Web están las interfaces conocidas como formularios Web. Tienen muchos usos, desde funciones tan simples como casillas de búsqueda, suscripciones en listas de correos, libros de visitas o encuestas, así como sistemas de comercio en línea. Los formularios reúnen información del usuario a través de controles como botones, campos de texto o menús deslizantes, pero no procesan los datos. Los controles Web más usados son (ver Figura 2):

- **Campos de entrada de texto:** sirven para introducir una sola palabra, una línea de texto, una contraseña con caracteres ocultos, direcciones de email, enlaces a recursos o sitios Web, y áreas de texto multilínea, con control deslizable.
- **Casillas de verificación:** interruptores que pueden ser activados y desactivados por el usuario, para preguntas multiselección en el caso de aceptar más de una respuesta.

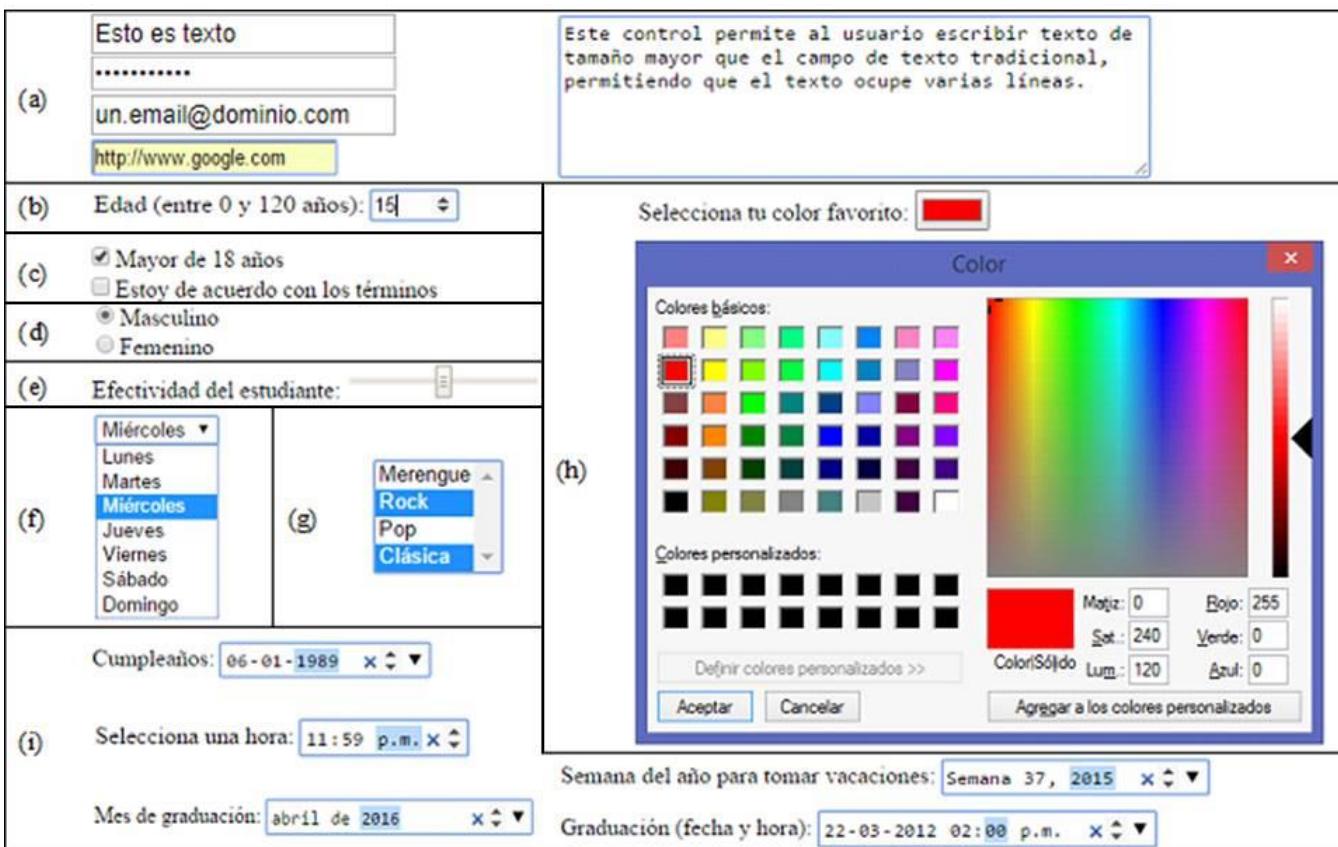


Figura 2. Tipos de Controles Web más usados: (a) Campos de Entrada de Texto, (b) Número, (c) Casillas de Verificación, (d) Botón de Opción, (e) Rango, (f) Menú Desplegable, (g) Menú Deslizante, (h) Color, (i) Tiempo

- Botón de opción:** interruptor que sólo puede activarse o desactivarse un control al mismo tiempo mientras todos los demás quedan desactivados.
- Número:** Campos de entrada que contienen un valor numérico, pueden aplicar restricciones.
- Rango:** para registrar valores que deben estar delimitados en un rango definido.
- Menús desplegables:** sólo puede seleccionarse un elemento cada vez.
- Menús deslizantes:** permite que los usuarios seleccionen más de una opción de la lista.
- Color:** Permite al usuario seleccionar un color de la escala RGB.
- Tiempo:** Permite indicar valores del tiempo como fecha, mes, año, semana, hora específica del día.

IV. CONTROLES WEB CON TÉRMINOS DIFUSOS

Muchas aplicaciones utilizan controles Web en las interfaces con los usuarios, los cuales no fueron concebidos para usar lógica difusa desde su diseño. Por tal motivo se realizó un estudio de las posibilidades de conversión de un sistema que ya funciona con lógica precisa a uno que funcione

con lógica difusa. Los detalles de este estudio y sus resultados se describen en [1].

A manera de resumen se mencionarán algunos aspectos. Primero, existen controles Web con entradas precisas que no pueden ser transformados a difusos: campos de entrada de textos con nombres, contraseñas, dirección email o url y texto multilíneas. Además de los cuales no suelen usarse en formularios de búsqueda (i.e. contraseñas o texto multilínea).

Campos de entrada numéricos, casillas de verificación, botones de opción, menús desplegables, menús deslizantes en la mayoría de los casos pueden asociarse a predicados difusos, representables con un menú desplegable o con casillas de verificación para múltiples selecciones. Las casillas de verificación con múltiples opciones, en algunos casos, también pueden transformarse a un cuantificador difuso

Los controles relacionados con fechas y tiempo también pueden representarse con predicados difusos (como cercano y lejano) para búsquedas difusas. La transformación de números también puede ser realizada con un comparador difuso.

Los controles tipo rango, donde se especifica un valor dentro de un dominio definido, pueden representarse con un comparador difuso especificado sobre el mismo dominio. Para los colores se puede agregar la funcionalidad de búsquedas con colores parecidos definiendo un predicado difuso que permita evaluar lo cercano que es un color de otro.

A todos los controles convertidos, se les puede aumentar la intensidad a través de modificadores difusos “muy” o “extremadamente”, si aporta algo al usuario. Estos se pueden agregar con casillas de verificación o botones de opción. Además, se puede especificar en las búsquedas el grado mínimo de verdad que deben satisfacer todos los resultados presentes en la misma. A esto se le conoce como calibración. Hay algunos controles equivalentes, en esos casos las decisiones se toman por aspectos relacionados con la interfaz.

V. ESTUDIO DE OPINIÓN

En el caso de los comercios Web, el impacto de una función de búsqueda efectiva puede ser muy importante. Los clientes que visitan estos sitios, ya saben lo que necesitan y por lo tanto van directamente a la sección de búsqueda. Esta sección muy probablemente es su primer contacto con el sitio Web, por lo que se debe asegurar que los usuarios no reciban resultados equivocados o mal clasificados, sino que realmente reciban lo que están buscando. De lo contrario, la frustración puede llevarlos a un sitio Web diferente.

En muchos casos esto ocurre cuando se hacen búsquedas precisas. Los formularios al tener controles Web que exigen entradas precisas (el valor exacto de una edad, precio, día, distancia, temperatura, etc.) puede ocasionar que las respuestas de las búsquedas sean incorrectas, incompletas o insatisfactorias. De allí surge la propuesta agregar términos difusos a los formularios Web.

Para analizar el impacto y el deseo que tendrían los programadores y usuarios Web de incluir lógica difusa en los formularios de búsqueda, se hizo una encuesta a 249 personas, de 24 países diferentes siendo el grupo más representativo el de Venezuela. Por ello, se analizaron estas 182 respuestas correspondientes a un 43% de usuarios y un 57% de programadores Web, a fin de ver si el proyecto era interesante. La información en común recabada fue edad, sexo, nivel educativo y si es programador o usuario Web. A los programadores se les consultó los SGBD que usaban y si estaban dispuestos a usar lógica difusa en sus desarrollos. A los usuarios se les consultó si al realizar búsquedas en formularios Web habían tenido respuestas negativas dándose cuenta luego que lo buscado si estaba. Además, se le consultó si estarían dispuestos a realizar búsquedas usando términos del lenguaje natural.

En los resultados recabados, un 95% de los encuestados estaban dispuestos a usar términos difusos y un 90% de los programadores estaban dispuestos a incluirlos en sus desarrollos. Ante la pregunta de si les ha ocurrido que utilizando formularios de búsqueda no encuentran lo que necesitan y posteriormente se dieron cuenta de que sí estaba, 94% respondió afirmativamente, concluyendo que los formularios presentan problemas para reflejar las preferencias de los usuarios.

En cuanto a los factores edad, sexo y nivel de educación alcanzado, se observó que no marcan una tendencia claramente diferenciada en ninguno de los casos. Por lo cual las descripciones y tendencias facilitadas en el análisis descriptivo de los datos se pueden considerar generalizadas, de forma independiente a estos factores mencionados.

Sobre la pregunta realizada a los programadores Web, referente a los SGBD preferidos para desarrollar, se observó que, aunque hay una tendencia actual de usar NO-SQL, las respuestas permiten concluir que el mercado sigue dominado el por el lenguaje SQL, con un 90% de preferencia.

VI. SELECCIÓN DE LA EXTENSIÓN CON LÓGICA DIFUSA

Algunos esfuerzos se han realizado para dar mayor flexibilidad al lenguaje estándar de bases de datos SQL, incorporando elementos de datos y condiciones de consultas basados en los conjuntos difusos. FSQL y SQLf son las extensiones más completas existentes para la incorporación de conjuntos difusos en SQL. Estas dos propuestas tienen enfoques complementarios. SQLf fue propuesta por Bosc y Pivert [7], se enfoca en la extensión de las expresiones de consulta. FSQL, propuesta por Galindo [8] y otros colaboradores, se centra en la extensión de los datos. Muchos trabajos de investigación y desarrollo se han realizado a partir de estas dos propuestas. SQLf es la extensión que permite expresar la mayor cantidad de términos difusos. Varias implementaciones de estas extensiones de SQL han sido realizadas.

SQLfi [12] es un sistema de consultas difusas que recibe instrucciones en el lenguaje SQLf, las cuales pueden ser: definiciones de términos difusos (predicados, modificadores, comparadores, conectores o cuantificadores), definiciones de objetos de datos (tipos, tablas, vistas, aserciones, procedimientos almacenados o funciones), manipulación de datos (inserción, eliminación, actualización), instrucciones transaccionales (commit o rollback) o consultas difusas. Si bien SQLfi fue desarrollado sobre Oracle 9i, su diseño se hizo pensando en la portabilidad del sistema, por lo que se implementó en Java, usando el estándar de conexión JDBC, como una capa lógica que se encarga del procesamiento de las nuevas funcionalidades.

PostgreSQLSQL [21] es una extensión de PostgreSQL para el procesamiento de consultas difusas que usa la estrategia de acoplamiento fuerte en su implementación. El *parser* del SGBD fue extendido para el reconocimiento de términos difusos y condiciones difusas. Asimismo, el catalogo y las estructuras abstractas de representación, de manera que son reconocidos por el optimizador, el ejecutor y las funciones de comandos. El uso de acoplamiento fuerte provee mayor escalabilidad y mejor desempeño. Por otro lado, esta complejidad lo hace no portable y con un desarrollo de gran esfuerzo, lo cual ha dificultado su actualización a las últimas versiones del SGBD y completar la implementación de todos los términos difusos.

FSQL [8] posee un servidor desarrollado en Oracle para una base de datos difusa, siguiendo el modelo teórico GEFRED que permite atributos difusos, los cuales almacenan etiquetas lingüísticas, en las tablas de una base de datos. Además, este servidor provee la facilidad de realizar consultas flexibles sobre atributos precisos o difusos. También, el uso y definición de constantes, operadores, comparadores y cuantificadores difusos.

SQLf_j [22] es un sistema que permite consultas difusas a los SGBD relacionales: MySQL y PostgreSQL. Las consultas

se especifican en SQLf. SQLf_j funciona como un traductor, de consultas difusas a consultas SQL estándar, las cuales se ejecutan en el SGBD relacional. SQLf_j fue escrito en Java con el parser de SQLf creado con yacc y flex, el cual fue reemplazado por un analizador orientado a objetos programado en SableCC.

La metodología desarrollada debía apoyarse en una de las extensiones existentes de SQL que maneje lógica difusa, se decidió establecer algunas características que permitieran evaluar tales extensiones después de realizar un proceso exhaustivo de investigación. Las características se describen a continuación y la presencia en las extensiones con lógica difusa de estas características se muestra, como un cuadro comparativo, en la Tabla I.

Tabla I. PRESENCIA DE CARACTERÍSTICAS DESEABLES EN LAS EXTENSIONES CON LÓGICA DIFUSA DE SGBD EXISTENTES

Característica	SGBD con lógica difusa			
	SQLfi	PostgreSQLSQLf	FSQL	SQLf_j
Completitud	Sí	Sí	No	Desconocido
Disponibilidad	Sí	Sí	Sí	Sí
Visibilidad	Baja	No visible	Alta	Alta
Calidad	Sí	No	No	No
Soporte	No	No	No	No
Portabilidad	6 SGBD	No	2 SGBD	2 SGBD
Documentación	Instalación y usuarios	Instalación y usuarios	Instalación y usuarios	Instalación
TOTAL	5	3	3	3

- **Completitud:** se refiere a si la extensión maneja todos los términos difusos definidos por Zadeh [22], así como, posee suficiente capacidad expresiva para que sea provechoso y aumente las posibilidades de realizar una migración exitosa.
- **Disponibilidad:** si la extensión está disponible en la

Web para su descarga.

- **Visibilidad:** si la extensión es fácilmente ubicable en la Web para su descarga. Relacionada con la anterior, ya que el producto podría estar disponible, pero con muchas dificultades para ubicarlo.
- **Calidad:** si la extensión cuenta con documentación sobre pruebas de calidad, en términos de rendimiento y correcto funcionamiento de los términos difusos implementados en la misma. Todos las extensiones son desarrollos experimentales, por lo cual es importante saber si tienen un control de calidad que permita su utilización en un ambiente de producción.
- **Soporte:** si existen personas u organizaciones con disposición de responder correos electrónicos aclarando dudas sobre la extensión en cuestión.
- **Portabilidad:** si la extensión puede funcionar con múltiples SGBD, y no está atada a uno específico.
- **Documentación:** si la extensión cuenta con documentación para los desarrolladores de software y a nivel de infraestructura para su instalación.

La Tabla I muestra que la extensión SQL que cumple con la mayor cantidad de criterios deseados es SQLf, por tal razón se selecciona como la extensión a ser utilizada en la metodología.

VII. PROYECTO DE MIGRACIÓN

En esta sección se describe en detalle la propuesta metodológica denominada Proyecto de Migración, cuyo objetivo es transformar un sitio Web con formularios de búsqueda que utilicen lógica clásica para que gestione términos difusos. Se explican los roles a considerar durante el proceso de migración, las diferentes etapas, la documentación y algunos aspectos relacionados a la calidad y viabilidad de la migración. La Figura 3 resume los diferentes componentes del proyecto de migración que se describen a continuación.

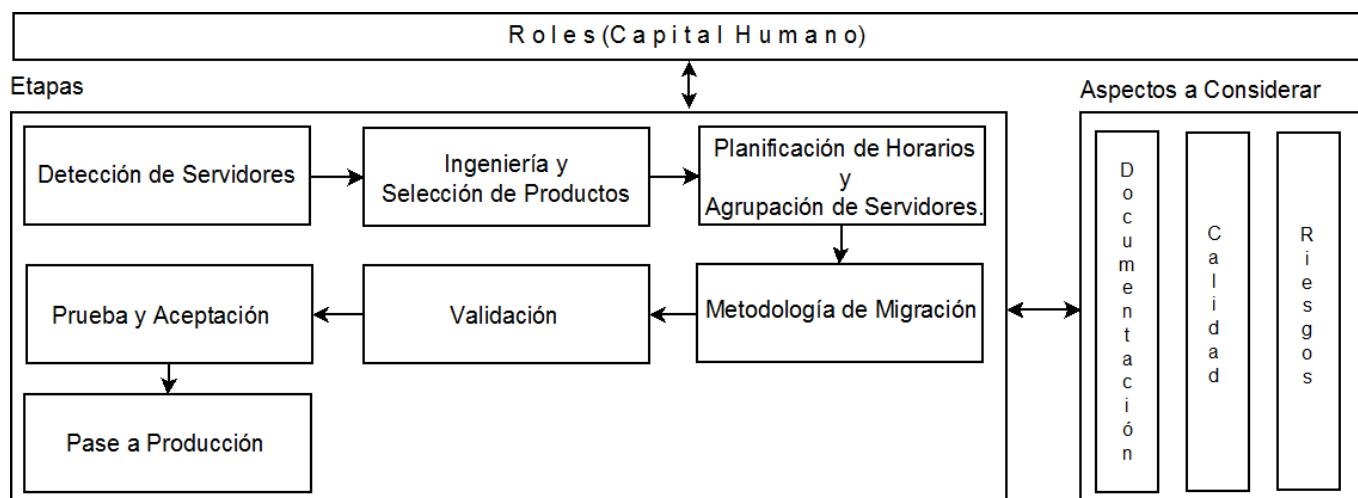


Figura 3. Esquema Resumen del Proyecto de Migración

A. Roles

En los proyectos de tecnologías de información, particularmente en proyectos de desarrollo o actualización de software, existen diferentes roles a lo largo de la planificación y ejecución de los mismos. En ocasiones, estos roles son cubiertos por una misma persona o grupo de trabajo.

Según Pressman [23], todo proyecto de software requiere de varios participantes que deben organizarse por roles. Tomando como base la propuesta de Pressman [23] se definen los roles para la planificación y ejecución del proyecto de migración de formularios Web.

- **Gerente del proyecto:** encargado de gestionar la planificación del proyecto en términos de tiempo, ejecución, costo y resultados esperados.
- **Especialista en infraestructura TI:** experto en instalaciones de SGBD, configuraciones de sistemas operativos y conocimientos en redes de computadoras.
- **Programador SQL:** persona conocecedora de las consultas utilizadas en los formularios de búsqueda a migrar, capacitada en programación en el lenguaje SQL, dispuesta a entrenarse en la extensión SQLfi.
- **Programador Web:** personal capacitado en programación, en el lenguaje utilizado en el desarrollo funcional del sitio Web a migrar.
- **Probador de software / encargado de la calidad:** persona con habilidades para la realización de pruebas y detección de posibles fallas o bugs en los desarrollos de software, específicamente en sitios Web.
- **Diseñador Web:** encargado(s) del aspecto visual del sitio Web, preferiblemente quien participó en el diseño inicial.
- **Clientes:** conocecedores expertos del modelo de negocio y la funcionalidad del sitio Web. Son los que dan las indicaciones de los cambios a realizar en el sitio Web para incorporarle lógica difusa.
- **Usuarios Finales:** grupo de personas que utilizarán el sitio Web una vez migrado, quienes participan en las pruebas de migración. Sólo deben saber navegar en internet y utilizar un formulario de búsqueda. Se sugiere que este grupo sea lo más numeroso posible, y con diversidad en términos de localización, edad, sexo, nivel de conocimiento, entre otros factores.

Adicionalmente, todos los roles deben contar con habilidades de comunicación efectiva y asertiva, así como de trabajo en equipo. A lo largo de todo el proceso será necesario el intercambio de ideas, así como la generación de propuestas.

B. Etapas

En base a la propuesta Martínez [24], la cual afirma que un proceso de migración de sistemas incluye seis componentes principales, se describen a continuación cada una de etapas con las adaptaciones correspondientes a la propuesta metodológica presentada en este trabajo.

1. **Detección de servidores.** Corresponde a determinar los requisitos de diseño relacionados a los productos o las soluciones que se utilizarán en el proyecto de migración. Para el caso de migración a formularios Web con lógica difusa, el requisito a considerar es que el servidor de SQLfi utilizado sea compatible con la arquitectura actual del sitio Web.
2. **Ingeniería y selección de productos.** Se identifica y establece toda la información básica para determinar los sistemas que se deben migrar. Para el caso de migración a formularios Web con lógica difusa se proponen dos actividades concretas:
 - **Entrenamiento:** Los roles profesionales, el gerente de proyecto y los clientes deben comprender el funcionamiento y las capacidades de los términos difusos provistos por SQLfi. En el caso de los clientes, deben recibir capacitación en lo que son términos difusos y lo que son capaces de hacer, para ayudar a proponer los cambios a realizar. El programador SQL y programador Web, requieren el mismo conocimiento que el experto del negocio y adicionalmente requieren entrenamiento en el lenguaje SQLf.
 - **Selección de controles a migrar:** Los clientes deben analizar uno a uno los controles existentes en los formularios de búsqueda, considerando todas las propuestas y tomando la decisión de migrar o no de forma individual. En algunos casos puede resultar que existe más de una migración viable para un determinado control. Es en este momento cuando el diseñador Web evalúa ambas propuestas, y opta por la que sea más amigable para los usuarios finales y más acorde con el diseño del sitio Web en cuestión.
- Una vez que se han diseñado los controles Web difusos, es necesario organizar o coordinar un proyecto de migración de formularios con términos precisos a formularios con términos difusos.
3. **Planificación de horarios y agrupación de servidores.** Se determina el estado futuro de cada servidor según su configuración actual y se agrupan por grupo de coherencia. Algunos servidores tienen interdependencias que requieren que se los migre de forma conjunta para eliminar la interrupción de servicios esenciales para los usuarios finales. Una vez que se hayan establecido los grupos de coherencia se hacen los cambios por olas de migraciones. Estos grupos sientan las bases para un plan de proyectos real, conformar el cronograma y los hitos de la migración.
4. **Metodología de migración.** Corresponde a los procesos que conforman la propuesta de este trabajo de investigación. Las actividades propuestas son:
 - **Instalación e integración:** El especialista de infraestructura TI instala y deja operativa la extensión SQLfi sobre el SGBD utilizado por el sitio Web a migrar. Esto se realiza en un ambiente de pruebas lo más similar posible al ambiente de producción. Luego, el programador de la aplicación Web redirecciona todas las consultas de los formularios de búsqueda

hacia SQLfi. Posteriormente, el probador de software verifica que los formularios sigan operando de la forma habitual y sin ninguna incidencia.

- **Definición de términos difusos:** El cliente, junto al programador SQL y programador Web definen los nuevos términos difusos usando SQLfi. Se hacen consultas de prueba sobre este SGBD para verificar que todos los términos están funcionando según lo esperado y acorde a su definición.
- **Definición de consultas SQL:** El programador SQL precisa las consultas apropiadas para las nuevas búsquedas a realizar, en todas sus combinaciones, a fin de que el sitio Web sea modificado por el programador Web. La meta es que los formularios de búsqueda envíen las consultas para ser ejecutadas por SQLfi.

5. Validación. Conjunto de tareas que aseguran que el software que se construye sigue los requerimientos del cliente. En esta etapa se realizan pruebas con el objeto de descubrir y prevenir problemas futuros. También, se establecen planes sobre mitigación de riesgos, que consideren los siguientes aspectos:

- Cuando las consultas no funcionen como se espera, no se hacen suficientes pruebas o el ambiente de producción y de prueba no son completamente equivalentes. Se recomienda estar preparado para regresar a la versión anterior del sitio Web.
- Cuando los usuarios no comprendan el funcionamiento del nuevo sistema de búsqueda, debido a un mal diseño de los términos difusos, o de los nuevos controles Web, o no se hicieron suficientes pruebas con múltiples usuarios externos al proyecto. Se sugiere regresar a la versión anterior del sitio Web.

6. Prueba y aceptación. Se asegura que todos los sistemas funcionen en su totalidad antes de entrar en producción. Para ello, se realizan pruebas de aceptación y un plan de recuperación. En esta etapa el probador de software se encarga de revisar exhaustivamente que el sitio Web no presenta errores en los formularios de búsqueda. También se valida el nivel semántico, es decir, si los resultados son los esperados y que se excluye cualquier otro. Luego de estas verificaciones, se integra a los usuarios finales en el proceso de prueba, así como, en conocer su opinión sobre los cambios realizados. Si éstos no resultan adecuados y agradables, se regresa a la etapa de selección de controles a migrar a fin de tomar nuevas decisiones en base a la experiencia reportada por los usuarios.

7. Pase a producción. Una vez culminadas todas las etapas anteriores se deben realizar los cambios en el ambiente de producción, y validar su funcionamiento como se hizo en el ambiente de pruebas. Algunas recomendaciones a considerar durante este procedimiento:

- Realizar los cambios en un horario donde el sitio Web tenga el menor tráfico posible.
- Informar a los usuarios del cambio a realizar mediante avisos en el sitio Web.

- Durante el proceso debe estar disponible todo el equipo de trabajo del proyecto para canalizar cualquier inconveniente y solventarlo en la forma más rápida posible.

Desde el punto de vista técnico, se sugiere realizar los cambios en el siguiente orden:

- a. Instalar la extensión SQLfi del SGBD usado por el sitio Web en el ambiente de producción.
- b. Probar el sitio Web original sobre la extensión, a fin de verificar que los formularios de búsqueda continúan funcionando. Esto es posible pues SQLfi tiene la capacidad de ejecutar consultas clásicas, por lo cual el sitio Web debería operar con total normalidad. Si hay algún problema, revisar el paso anterior.
- c. Si todo funciona apropiadamente, se procede a migrar los formularios de búsqueda con los nuevos términos difusos, y se indica a los usuarios el cambio realizado.

En la Tabla II, se resumen las etapas del proyecto de migración y los actores que participan en cada una ellas.

C. Documentación

Durante el proyecto de migración se deben elaborar los siguientes informes a fin de documentar todo el proceso realizado:

- **Cambios en infraestructura de TI:** el especialista en infraestructura de TI reporta todos los cambios a nivel de puertos, las rutas utilizadas para nuevas instalaciones, y cualquier otro cambio pertinente que haya aplicado.
- **Cambios en la arquitectura de la aplicación:** se detalla la arquitectura anterior, los cambios a realizar, y la nueva arquitectura del sitio Web producto de la incorporación de consultas sobre la extensión SQLfi.
- **Términos difusos:** se debe reportar todos los términos difusos definidos, su significado, sus valores, su concepción original y motivación. Además, a nivel técnico, se debe especificar los nombres y dominios que les fueron asignados dentro de SQLfi.
- **Pruebas de calidad:** se deben describir en detalle todas las pruebas realizadas durante la etapa seis, Pruebas y Validación, así como los resultados obtenidos.

Se recomienda que cada documento sea elaborado una vez terminada la etapa correspondiente al mismo. Luego que el sitio Web esté en el ambiente de producción, con resultados satisfactorios, se debe revisar toda la documentación nuevamente para verificar que cualquier cambio ocurrido ha sido reportado apropiadamente. Finalmente, se debe realizar cualquier documentación adicional que aporte valor al proyecto.

D. Calidad

Se propone como enfoque para comprobar la calidad del proyecto comisionar a un grupo de personas que se encarguen de revisar el software, su documentación y la supervisión de los procesos utilizados en el desarrollo. Este equipo verifica

Tabla II. ETAPAS DEL PROYECTO DE MIGRACIÓN Y LOS ACTORES QUE PARTICIPAN EN CADA UNA

Rol \ Actividad	Detección de Servidores	Ingeniería y Selección de Productos	Planificación de Horarios	Metodología de Migración	Validación	Pruebas y Aceptación	Pase a Producción
Gerente del Proyecto	✓	✓	✓	✓	✓	✓	✓
Especialista en Infraestructura TI	✓		✓	✓			
Programador SQL	✓	✓		✓			✓
Programador Web	✓	✓		✓			✓
Probador de Software					✓	✓	✓
Diseñador Web		✓				✓	
Clientes		✓		✓		✓	
Usuarios Finales						✓	

que se sigan los estándares seleccionados para el proyecto en los tres aspectos mencionados. Cualquier estándar que no haya sido cumplido debe ser atendido por el gerente del proyecto.

Además, se sugiere realizar las siguientes verificaciones de calidad:

- A nivel de infraestructura de TI, todos los cambios indicados en la documentación coinciden con el ambiente final de ejecución del sitio Web.
- Los términos difusos escogidos tienen una sola interpretación en lenguaje natural.
- Los usuarios finales seleccionados para las pruebas semánticas y de usabilidad representan apropiadamente todo el universo de usuarios del sitio Web.
- Se realizaron todas las pruebas propuestas con rigurosidad, sin omitir ningún caso.
- La documentación de los términos difusos coincide con lo que está operando en el ambiente de producción.

Estas verificaciones deben realizarse durante las etapas del proyecto que lo ameriten, buscando siempre que el resultado sea lo más satisfactorio posible. Por otro lado, no se excluye cualquier validación adicional que se desee realizar.

E. Viabilidad de la Migración

Para determinar si un sitio Web es factible de ser migrado a trabajar con lógica difusa en sus formularios de búsqueda, se debe contar con algunos requisitos:

- Disponer de capital humano suficiente para cubrir todos los roles descritos.
- Llevar a cabo las dos primeras etapas del proyecto de migración a fin de determinar si al menos un control Web es seleccionado para ser migrado.
- El equipo de trabajo está dispuesto a asumir los riesgos asociados al proyecto.

Se deben considerar todos los riesgos posibles, y planificar para cada uno de ellos una estrategia de mitigación y una de contingencia.

VIII. CONCLUSIONES Y TRABAJOS FUTUROS

En casi cualquier ámbito de la vida se pueden conseguir aplicaciones Web, las cuales usualmente están dotadas de formularios que permiten la realización de búsquedas para satisfacer requerimientos de los usuarios. De manera que resulta de gran interés el tratar con este tipo de formularios, considerando su extensión usando términos lingüísticos vagos.

En este artículo se han relacionado algunos de los varios trabajos previos que se han hecho en cuanto a la incorporación de lógica difusa en SQL y la consecuente extensión de los SGBD. Como resultado de esos trabajos se cuenta con SGBD que proveen mayor expresividad para búsquedas basadas en preferencias del usuario, los cuales están disponibles para ser utilizadas libremente, pero no se ha masificado su uso en aplicaciones Web. También se han mostrado los pocos esfuerzos previos en el uso metodológico de la incorporación de SQL extendido difuso. Estos trabajos relacionados no

habían tenido en cuenta la extensión de un formulario de búsqueda Web existente para agregarle términos difusos.

Se expuso en este artículo un marco conceptual resumido de la investigación realizada que involucra la teoría de conjuntos difusos, los términos lingüísticos y los controles web. La teoría de conjuntos difusos es una herramienta para el tratamiento matemático y computacional de términos vagos del lenguaje natural, la cual da soporte a una lógica gradual, conocida como lógica difusa. Los términos en esa lógica son palabras del lenguaje natural, las cuales se clasifican en: predicados, modificadores, comparadores, conectores y cuantificadores. Por otro lado, los formularios Web permiten el uso de diferentes tipos de controles: campos de texto, casillas de verificación, botón de opción, números, rangos, ingreso de fechas, menú desplegable, menú deslizante y selector de colores.

En un trabajo previo se ha reportado la incorporación de términos difusos en estos controles, a fin de dar mayor flexibilidad a los formularios Web de búsqueda. De tal forma que se provee de un mecanismo sistemático, que permite extender formularios de búsquedas existentes, que no fueron concebidos originalmente con lógica difusa, para que se facilite la expresión de preferencias de usuarios al involucrar términos difusos. La mayoría de los controles Web existentes puede incorporar términos difusos, por lo que se dispone de una guía que puede ser usada por cualquier programador Web que quiera flexibilizar las búsquedas en una aplicación Web existente, a fin de lograr mayor satisfacción del usuario.

A los fines de validar la aceptación que tendría el desarrollo y uso de aplicaciones Web incluyan controles Web con términos difusos en sus formularios de búsqueda, se realizó un estudio de opinión. Con base a este estudio, se puede concluir que los programadores Web tienen el deseo de utilizar herramientas de lógica difusa. Por otro lado, la encuesta deja ver que actualmente los usuarios de aplicaciones Web demandan mayor flexibilidad y adaptación a sus requisitos o preferencias. Más aún, en muchas oportunidades los usuarios quedan insatisfechos con los formularios de búsqueda presentes en aplicaciones Web debido a su rigidez. De manera que el estudio muestra tanto que los usuarios están deseando este tipo de términos en sus formularios de búsqueda Web, así como los programadores están dispuestos a incorporarlos en sus desarrollos. El soporte para estos desarrollos serían los SGBD basados en SQL extendido mediante la aplicación de la teoría de conjuntos difusos. La encuesta muestra que, a pesar de la nueva tecnología NO-SQL, los programadores prefieren actualmente el uso de SGBD basados en SQL.

Antes de proponer el proceso de migración, se analizaron las diferentes extensiones del lenguaje SQL que se implementan sobre los diferentes SGBD existentes, las cuales usan conjuntos difusos en los elementos de datos y las condiciones de consulta. De esta forma, se incorpora la lógica difusa en los SGBD como mecanismo para flexibilizar las consultas y expresar preferencias de usuario. Se estudiaron cuatro extensiones distintas: SQLfi, PostgreSQL, FSQL y SQLf_j. Además, se propusieron ciertas características que permitían evaluar la más conveniente para el proyecto de migración. Como resultado de esta investigación, se concluyó

que SQLfi es la extensión con lógica difusa que posee mayor cantidad de características deseables: completitud, disponibilidad, visibilidad, calidad, soporte, portabilidad y documentación. Por tal motivo, se seleccionó SQLfi para ser utilizada en la propuesta aquí presentada.

El principal aporte de este trabajo es una propuesta metodológica que permite realizar la migración de sitios Web existentes para que sus formularios de búsqueda incorporen términos difusos. Se definen los roles de los participantes en el Proyecto de Migración, estos son: Gerente del proyecto, Especialista en infraestructura TI, Programador SQL, Programador Web, Probador de software o encargado de la calidad, Diseñador Web, Clientes, Usuarios Finales. Además, se describen las etapas por las que debe pasar este proyecto: Detección de servidores, Ingeniería y selección de productos, Planificación de horarios y agrupación de servidores, Metodología de migración, Validación, Prueba y aceptación, Pase a producción. Estas etapas son similares a otros proyectos de migración, aunque requieren aspectos distintivos al objetivo del trabajo aquí presentado. En particular, en la etapa de la Metodología de migración es necesario instalar e integrar la extensión SQLfi al SGBD usado por el sitio Web a migrar, establecer las definiciones de los términos difusos y especificar las nuevas consultas que gestionarán estos términos. Finalmente, se dan sugerencias para garantizar la calidad y operatividad del sitio Web migrado, así como, recomendaciones para que el proyecto sea viable.

Como trabajos futuros se espera realizar experiencias prácticas de migración de aplicaciones Web existentes usando la propuesta metodológica aquí presentada. Así mismo, se procederá con el análisis estadístico de estas experiencias a fin de observar los beneficios obtenidos de los sitios Web migrados. Sería interesante evaluar el esfuerzo que debe realizar un programador al tratar de incorporar términos difusos durante la migración de un sitio Web. La expectativa de los autores es que este esfuerzo no sea significativo.

Por otro lado, se espera hacer nuevas versiones de la metodología que abarquen aplicaciones de escritorio y aplicaciones móviles.

AGRADECIMIENTOS

Agradecemos Aquél que nos da fe y valor para emprender proyectos hacia lo desconocido: “Por la fe Abraham, siendo llamado, obedeció para salir al lugar que había de recibir como herencia; y salió sin saber a dónde iba” (Hebreos 11:8).

REFERENCIAS

- [1] J. A. Labbad, R. Rodríguez, y L. Tineo, “Formularios Web con Lógica Difusa”. Revista Colombiana de Computación, unpublished, 2017
- [2] L. A. Zadeh, “Fuzzy Logic - a Personal Perspective”, Fuzzy Sets and Systems, Vol. 281, Issue C, December, 2015, pp. 4-20.
- [3] L. A. Zadeh, “Fuzzy Sets”. Information Control, Vol. 8, Issue 3, June, 1965, pp. 338-353.
- [4] O. Pivert, and P. Bosc, Fuzzy Preference Queries to Relational Databases, Imperial College Press, 2012.
- [5] ISO/IEC. Information Technology — Database Languages — SQL — Part 2: Foundation (SQL/Foundation), ISO/IEC 9075-2:2011 (en). 2011.

- [6] A. Aguilera, L. Borjas, R. Rodríguez, L. Tineo, "Experiences on Fuzzy DBMS: Implementation and Use", Actas de la XXXIX Conferencia Latinoamericana en Informática, Naiguatá, Venezuela, octubre, 2013, pp. 478-485.
- [7] P. Bosc, O. Pivert, "SQLf: A Relational Database Language for Fuzzy Querying", IEEE Transactions on Fuzzy Systems, Vol. 3, Issue 1, February, 1995, pp. 1-17.
- [8] J. Galindo, FSQ (Fuzzy SQL) A Fuzzy Query Language. Universidad de Málaga. Málaga, España. 2008. <http://www.lcc.uma.es/~ppgg/FSQL/>.
- [9] S. Carrasquel, A. Gyomrey, S. Moreau, R. Rodríguez, B. Stornelli, C. Timaury y L. Tineo, "Extensión de MariaDB para Ordenamiento y Agrupamiento Difuso", Novática. Revista de la Asociación de Técnicos de Informática, Nº 229, julio-septiembre, 2014, pp. 92-97.
- [10] E. Lai, "No To SQL? Anti-database Movement Gains Steam". Computerworld, 2009. http://www.computerworld.com/s/article/9135086/No_to_SQL_Anti_database_movement_gains_steam.html.
- [11] MetalByte, "¿Cuáles son los Lenguajes de Programación más Demandados en la Empresa?", muylinux, 2013. <http://www.muylinux.com/2013/04/18/cuales-son-los-lenguajes-de-programacion-mas-demandados-en-la-empresa>.
- [12] M. Goncalves, L. Tineo, "SQLf y Sus Aplicaciones". Revista Avances en Sistemas e Informática, Vol. 5, No. 2, Mayo, 2008, pp. 33-40.
- [13] R. Rodríguez, L. Tineo, "Elementos Gramaticales y Características que Determinan Aplicaciones con Requerimientos Difusos". Revista Tekhné, Vol. 12, Enero, 2009, pp. 50-64.
- [14] M. Goncalves, R. Rodríguez, L. Tineo, "Incorporando Consultas Difusas en el Desarrollo de Software", Revista Avances en Sistemas e Informática, Vol. 6, No. 2, Noviembre, 2009. pp. 87-101.
- [15] M. Goncalves, R. Rodríguez, L. Tineo, "Formal Method to Implement Fuzzy Requirements". DYNA, Revista de la Facultad de Minas, Vol. 173, No. 2. Enero, 2012. pp 15-24.
- [16] R. Rodríguez, M. Goncalves, "Perfil UML para el Modelado Visual de Requisitos Difusos". In: *Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento*, Vol. 6, No. 3, septiembre-diciembre, pp. 29-46, 2009.
- [17] R. Rodríguez, M. Goncalves, "Implementación de Requisitos Difusos en Sistemas Orientados a Datos utilizando el Lenguaje OCL y Lógica Difusa", In: *Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento*, Vol. 8, No. 1, pp. 31-54, Enero-Abril, 2011.
- [18] A. Aguilera, M. Goncalves, R. Rodriguez, "Framework for Fuzzy Application Development", Actas de la XXXVIII Conferencia Latinoamericana en Informática, Medellín, Colombia, octubre, 2012, pp. 478-485.
- [19] L. Yan, Z.M. Ma, "Modeling Fuzzy Information in Fuzzy Extended Entity-Relationship Model and Fuzzy Relational Databases", Journal of Intelligent & Fuzzy Systems, Vol. 27. Issue 4, July, 2014. pp. 1881-1896.
- [20] L. A. Zadeh, "PRUF – A Meaning Representation Language for Natural Languages". *Int. J. Man-Machine Studies*, Vol. 10, January, 1978, pp. 395-460.
- [21] A. Aguilera, J. Cadenas, L. Tineo, "Fuzzy Querying Capability at Core of a RDBMS", in Advanced Database Query Systems: Techniques, Applications and Technologies, L. Yan and Z. Ma, Eds. IGI Global. New York, USA, 2011, pp. 160-184.
- [22] P. Kalinowski, "SQLf_J", Poznan University of Technology, Polonia. 2006. http://calypso.cs.put.poznan.pl/~sqlf_j/en/index.php?.
- [23] R. Pressman, Ingeniería del Software: Un Enfoque Práctico. México D.F., McGraw Hill, 2010.
- [24] H. Martínez, Ocho Pasos para el Éxito en Migraciones. *The GMB journal*. 2013. <https://es.scribd.com/document/246408796/Pasos-Para-El-Exito-en-Migraciones-GBM-Journal>.

Derivación del Sistema Moodle a Partir de una Arquitectura de Referencia para el Aprendizaje Electrónico

Francisca Losavio

Laboratorio MoST, Escuela de Computación
Facultad de Ciencias, UCV
Caracas, Venezuela
francislosavio@gmail.com

Yuly Esteves González

Departamento de Ciencias Naturales y Matemática
UPEL-IPMJMSM
Caracas, Venezuela
yulyesteves@gmail.com

Resumen— La derivación de productos o sistemas de software concretos a partir de una Arquitectura de Referencia (AR) es un proceso complejo que requiere “instanciar” o configurar una arquitectura abstracta, representada por una familia de sistemas similares, para un dominio específico, en este caso el dominio del aprendizaje electrónico (e-learning). El objetivo de este trabajo es ofrecer lineamientos de cómo realizar esta instanciación para obtener una configuración arquitectónica de un producto concreto a partir de una AR ya creada en trabajos anteriores y conforme al estándar IEEE-LTSA para el dominio e-learning. El producto concreto cuya arquitectura se quiere derivar es el conocido sistema abierto de gestión de aprendizaje Moodle, muy utilizado en la práctica docente.

Palabras clave—aprendizaje electrónico; arquitectura de referencia; derivación de productos; configuración de una arquitectura de referencia; Moodle.

I. INTRODUCCIÓN

La presente investigación trata de la derivación de productos de software específicos o “concretos” a parir de una AR para el dominio del *aprendizaje electrónico* o del inglés *e-learning* (ambos términos se utilizarán indistintamente en este trabajo). Muchas son las definiciones dadas para una AR en la literatura; en nuestro contexto se considera como una arquitectura de software [1] genérica e instanciable, definida a un alto nivel de abstracción; es el artefacto principal en el contexto industrial de *Líneas de Productos de Software (LPS)* [2][3][4][5], la cual se propone como una solución industrial al desarrollo altamente basado en la reutilización para reducir costos de desarrollo y tiempo de mercadeo. La AR se reutiliza como plantilla o marco de referencia, para derivar una familia de productos o sistemas de software similares de la LPS en un determinado dominio. En nuestro contexto la AR es un eje central de la *plataforma* de aprendizaje electrónico, sobre la cual se articula el sistema de software completo. Un *dominio* se define como el conjunto mínimo de propiedades que definen una familia de problemas para los cuales se requieren soluciones computacionales [6].

En esta contribución se muestra cómo utilizar la AR construida en [7][8] para derivar la arquitectura de un producto concreto de e-learning. El proceso de derivación de productos a partir de una AR es un proceso complejo que contempla chequeos de consistencias y trazabilidad entre los

componentes arquitectónicos y los elementos de software de bajo nivel, como son los módulos de código, para generar un producto ejecutable; mucha investigación se ha realizado en este campo, [9][10][11] son ejemplo de algunas de ellas.

El objetivo del presente trabajo es mostrar cómo configurar o “instanciar” la AR para derivar una arquitectura de software de un producto específico en el dominio e-learning, cubriendo así una deficiencia encontrada en los trabajos [7][8], en los cuales no se mostró cómo se realizaba la configuración. La AR se obtiene en el primer ciclo de vida de *Ingeniería del Dominio* inspirado en el enfoque LPS [2][3][5]; la *configuración* es una fase inicial en el segundo ciclo de vida de la *Ingeniería de la Aplicación*; no se realizará aquí la derivación de código ni la verificación de consistencia de la configuración obtenida; suponemos que esta es adecuada porque la AR obtenida es conforme al estándar IEEE-LTSA [12]; esta conformidad es una práctica usual realizada para “validar” las AR para e-learning [7]. La configuración arquitectónica de la AR contempla componentes arquitectónicos funcionales, que capturan objetivos del negocio obtenidos a partir de la especificación de procesos de negocios y componentes no funcionales que realizan propiedades de calidad requeridas por los componentes funcionales, para que estas puedan responder adecuadamente a las tareas para las cuales fueron diseñadas, es decir cumplir con la adecuación funcional [14]. De esta manera la calidad global de la AR y de las configuraciones de allí derivadas es garantizada.

La configuración arquitectónica del producto concreto que se va a derivar a partir de la AR es el sistema *Moodle (Modular Object Oriented Dynamic Learning Environment)* [14], conocido sistema abierto por ser de amplio uso en la práctica docente actual de e-learning, ahora también considerado un Entorno Virtual de Aprendizaje o “*Virtual Learning Environment (VLE)*” por su adaptación a la arquitectura SOA¹ basada en servicios [10].

Este artículo se estructura, además de esta *Introducción* y de la *Conclusión*, en tres secciones: *Contexto*, donde se presentan trabajos relacionales y algunas bases conceptuales;

¹ Service Oriented Architecture

una sección dedicada a Moodle, donde se describe este sistema de aprendizaje electrónico de amplio uso práctico y la sección de *Configuración de la AR*, donde se presenta la AR considerada y se dan lineamientos para efectuar la configuración y así obtener una arquitectura del sistema Moodle.

II. CONTEXTO

A. Trabajos Relacionados con la Configuración de Arquitecturas de Productos Concretos a Partir de una AR

Muchos son los trabajos relacionados con el tema de la derivación de productos concretos pertenecientes a una familia de productos similares, como en el caso de las *Líneas de Productos de Software (LPS)* [2][3][5] y en particular con la fase de configuración. Se centra en un artefacto principal, la AR, que es el esquema instanciable para la derivación de productos concretos miembros de la familia de productos de la LPS. En la búsqueda sistemática de la literatura realizada se consideró la pregunta de investigación “*¿Cómo realizar la derivación de productos de software a partir de una arquitectura de referencia en LPS?*” y la expresión de búsqueda fue “*líneas de productos de software + configuración + arquitectura de referencia*”, en inglés “*software product line*” + “*configuration*” + “*reference architecture*”, en fuentes académicas y sociales como Google, Google Scholar, Research gate, etc. Muchos resultados fueron arrojados sobre métodos de desarrollo de LPS que incluyen etapas de configuración, pero eran generalmente relacionados con el diseño de marcos de referencia o conocidos también en inglés como “*frameworks*”, que no precisaban modelos o ejemplos específicos de configuración de productos directamente de la AR. Solo tres trabajos fueron considerados relevantes para el tópico de esta investigación; a continuación se presenta su discusión.

Los autores del artículo [9] desarrollaron el método *PuLSE (Product Line Software Engineering)* para el desarrollo completo de la LPS en un contexto empresarial. La instanciación o configuración de la AR para productos específicos se realiza en la fase de PuLSE denominada Usage, validando un solo producto respecto a los requisitos del cliente. Implementan un modelo de configuración, donde las decisiones específicas son recolectadas y resueltas durante la instanciación; este modelo extiende un modelo de decisión; sin embargo los mecanismos o motores de razonamiento no son especificados. PuLSE es el resultado de un proceso ascendente o “bottom-up”, igual que el seguido en uno de nuestros trabajos previos para construir una AR [15], que se basa en el conocimiento existente del dominio y en la experiencia obtenida en la industria; no se utiliza el enfoque seguido en [10] para la construcción de la AR.

En [11] se presentan resultados sobre la generación de configuraciones de productos directamente de la AR, siguiendo los enfoques ontológicos y el desarrollo dirigido por modelos, conocido en inglés como *MDD (Model Driven Development)*. Una ontología escrita en Protegé² es utilizada para capturar el conocimiento del dominio, restricciones y relaciones entre elementos arquitecturales. Los requisitos del

cliente son introducidos como peticiones o “queries” a la ontología y traducidos en reglas para generar el árbol de características, el cual es traducido a un lenguaje de descripción arquitectónico o en inglés “*Architecture Description Language (ADL)*” y así generar la AR; otro proceso complejo basado en transformaciones de modelos permite generar una instancia de la AR para el producto concreto, la cual es traducida a otro ADL. En nuestro caso no se ha utilizado enfoque ontológico y la AR construida en [8], siguiendo un enfoque ascendente o “bottom-up” para el desarrollo de LPS inspirado en [15], es producida a partir de la unión de configuraciones de AR ya existentes para e-learning; nuestra AR es adecuada en [8] a la IEEE-LTSA, que es un estándar para este dominio.

Un proceso de configuración y derivación a partir de la AR denominado ASSPRO es definido en [11]. La AR también ha sido construida siguiendo un enfoque “bottom-up”, a partir de productos existentes del mercado en el dominio de los Sistemas de Información Integrados de Salud. Respecto al presente trabajo, la AR que consideramos para e-learning ha sido también obtenida por un proceso bottom-up, pero los productos existentes considerados no fueron sistemas específicos de e-learning del mercado, sino arquitecturas de referencia para e-learning, como fue mencionado anteriormente, lo cual permitió la adecuación de nuestra AR al estándar IEEE-LTSA [8].

B. Aprendizaje Electrónico

Las innovaciones desarrolladas a finales del siglo XX y que han marcado el transcurso del siglo XXI, han redimensionado las estrategias de aprendizaje abierto y a distancia. Esta modalidad de estudio cada día tiene más aceptación “estimulado, en parte, por el creciente interés de educadores y tutores en las nuevas tecnologías vinculadas a internet y otras plataformas multimedia, y en parte debido al creciente consenso sobre la necesidad de apoyar las formas tradicionales de educación, valiéndose de medios más innovadores” [16].

En este sentido, cuando se habla de educación a distancia mediada por el computador, se hace referencia al aprendizaje electrónico el cual permite crear ambientes de aprendizaje interactivos, eficientes, accesibles y distribuidos que pueden ser clasificados según los medios tecnológicos de los que hacen uso [17]:

- El CBT (*Computer Based Training*) o CAI (*Computer Assisted Instruction*), aprendizaje basado en computador o instrucción asistida por computador, fue implantado en múltiples instituciones educativas y organizaciones. Estaba basado en la lectura e incorporaba mecanismos de realimentación pregunta-respuesta, convirtiendo al alumno en un ente más activo dentro de su propio proceso formativo.
- El IBT (*Internet Based Training*) fue el siguiente paso evolutivo de los sistemas de aprendizaje basados en computador, CBT. Con la llegada de las redes de comunicación, los contenidos podían llegar a sus destinatarios a través de Internet o de intranet.
- El WBT (*Web Based Training*) consiste en el aprendizaje haciendo uso de la Web, a través de la cual

² <http://protege.stanford.edu>

se reciben los contenidos. En este último tipo se encuentra el Campus Virtual.

En consecuencia, puede decirse que el e-learning es una forma de utilizar la tecnología para mediar el proceso de enseñanza y de aprendizaje, utilizando modalidades abiertas, a distancia y ambientes colaborativos, capaz de llegar a un gran número de personas en el mundo, sin perder la capacidad de interacción entre profesores y estudiantes. Los trabajos realizados en [18][19][20][21] muestran que, en general los sistemas de software que implementan aplicaciones computarizadas para e-learning son sistemas de información en estilo 3-capas clásico, cuya comunicación entre capas y distribución se rige por un modelo Cliente/Servidor; actualmente también se considera una arquitectura basada en servicios (SOA, por sus siglas en inglés), aprovechando los servicios de red. En este punto, hay que hacer mención especial al m-learning (*mobile learning*) o aprendizaje electrónico móvil, una evolución del e-learning que posibilita el aprovechamiento de las ventajas de las tecnologías móviles como soporte al proceso de aprendizaje [22]. Estas soluciones permiten acceder a las tecnologías de la información cuando y donde el usuario lo necesite, facilitando la posibilidad de implementar innovadores modos de dar enseñar y de aprender [23].

C. Características de Calidad

Para garantizar un desarrollo evolutivo del software, en nuestro enfoque, es importante identificar los requisitos de calidad relativos al dominio, en este caso e-learning. Algunos resultados obtenidos en [18][24] respecto a los sistemas de software educativos en general, sirven como base para analizar las características de calidad prioritarias, de acuerdo con el tipo de sistema educativo, entre los cuales están presentes los sistemas *e-learning*, junto con las soluciones arquitecturales requeridas por estos sistemas; este enfoque fue seguido para el desarrollo completo de la AR [7][8]. Esta información, junto con las características de calidad que puedan derivarse de los objetivos y reglas del negocio especificadas en el modelo del negocio, permiten precisar un modelo de calidad para el dominio de los sistemas *e-learning* y expresarlo mediante una adaptación o instanciación del modelo de calidad estándar ISO/IEC 25010 [13], en la Figura 1 se muestra una adaptación de este modelo al dominio e-learning.

Un *modelo de calidad del producto* según [13], está compuesto por ocho características de alto nivel de abstracción, que se subdividen jerárquicamente en subcaracterísticas y así sucesivamente en atributos o elementos medibles. Se refieren a propiedades *inherentes estáticas* (como usabilidad, disponibilidad, etc. y a las inherentes dinámicas (como eficiencia, fiabilidad, etc.) del sistema informático. También se habla de propiedades *asignadas*, que no son de calidad, como costo, tiempo de mercadeo, etc., pero influyen en las decisiones durante la construcción del software. El modelo de calidad del producto se utiliza durante el proceso de desarrollo y cuando el sistema está en prueba, pero aún no entregado al usuario final. El modelo es aplicable a los productos de software y en general a los sistemas informáticos, generalmente complejos, que también

involucran componentes de software [13]. También se define un *modelo de calidad en uso* que se relaciona con el resultado de la interacción con los usuarios finales, cuando un sistema de software se emplea en un contexto particular de uso, pero no será utilizado en este trabajo que se centra en la etapa de configuración de la AR.

Las características de calidad definidas son relevantes para todos los productos de software y sistemas informáticos. El modelo de calidad del producto, adaptado al *dominio e-learning* se muestra en la Figura 1. En nuestro caso nos encontramos en una etapa del desarrollo donde aún no se ha construido el producto, por lo tanto nuestras métricas de calidad consisten en expresiones booleanas “si/no”; ellas indican la presencia/ausencia del mecanismo, solución arquitectural o componente que satisface o “realiza” dicha calidad [7][8][15], la cual es generalmente requerida por otro componente que representa un requisito funcional. En [24] se presenta un modelo de calidad detallado especificado por [13] para los sistemas de Gestión de Aprendizaje o *LMS* (*Learning Management System*) para e-learning.

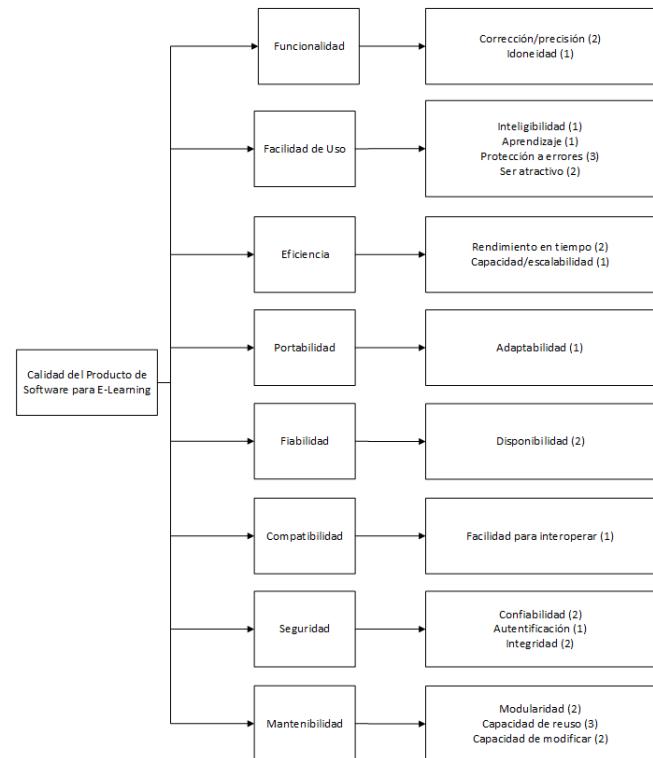


Figura 1. EL-DQM: Modelo de Calidad ISO/IEC 25010 [13] Adaptado al Dominio E-Learning [8], con Prioridades ($1 \leq p \leq 3$, 1 Máxima Prioridad)

III. EL SISTEMA MOODLE

Moodle [14] es un sistema de software abierto u “open-source” que contiene un ambiente de desarrollo o plataforma para e-learning, se basa en el lenguaje PHP con características de orientación a objetos y sigue un estilo arquitectural híbrido clásico de los sistemas de información de 3-capas, basado en eventos y el modelo Cliente/Servidor para la comunicación y distribución; el Servidor maneja la capa de procesos para la realización de las funcionalidades principales y otras operaciones y el cliente presenta la Interfaz Usuario (IU) en

html, como una puerta o “gate” para permitir al navegador o obtener del servidor las peticiones del usuario. El código de las funcionalidades es transparente para el navegador y es tratado en el servidor; los elementos de la IU son botones, menús en cascada, etc., que permiten el acceso a las funcionalidades residentes en el servidor que constituye la capa de proceso. Moodle se origina como un *CMS (Contents Management System)* y es ahora considerado un LMS, es un sistema de soporte a los procesos de aprendizaje y la mínima unidad de instrucción que maneja es el curso en sí mismo, además de las funciones administrativas. Un LMS podría, sin embargo, también crear contenidos de aprendizaje como un Sistema de Gestión de Contenidos de Aprendizaje o *LCMS (Learning Contents Management System)*, integrando funcionalidades de los CMS. Sin embargo el umbral entre LMS y LCMS no siempre es claro.

En algunos casos un LMS es simplemente una herramienta para distribuir el contenido del curso, que debe ser luego creado en un LCMS.

En [24] Moodle se considera aún un sistema LMS de gestión de cursos de código abierto, bajo la Licencia Pública General de GNU. Esto significa que Moodle tiene derechos de autor, con libertades adicionales. Se tiene autorización para copiar, usar y modificar Moodle, siempre que el usuario se comprometa a proporcionar públicamente la fuente a otros usuarios, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado.

Es compatible con otros formatos para la interoperabilidad de objetos de aprendizaje (SCORM³ que agrupa LOM⁴ e IMS⁵, entre otros). Puede ser instalado en cualquier ordenador que pueda ejecutar PHP, y puede soportar una base de datos relacional MySQL. Es altamente portable, se ejecuta sin modificaciones en Unix, GNU/Linux, OpenSolaris, FreeBSD, Windows, Mac OS X, NetWare y otros sistemas que soportan PHP, incluyendo la mayoría de proveedores de alojamiento Web. Es una aplicación Web gratuita que los educadores pueden utilizar para crear sitios de aprendizaje efectivo en línea o como complemento del aprendizaje presencial. Moodle permite una amplia gama de modos de enseñanza. Es un sistema organizado como un núcleo de aplicaciones, tales como: asignaciones, pruebas, calendario, temas, curso y actividades. Puede ser utilizado para generar contenido de manera básica o avanzada (por ejemplo páginas Web) o evaluación. Es utilizado por una gran variedad de instituciones educativas y no educativas y por educadores independientes.

IV. DERIVACIÓN DE LA ARQUITECTURA DEL SISTEMA MOODLE A PARTIR DE LA ARQUITECTURA DE REFERENCIA

Los sistemas e-learning deben estar soportados por una arquitectura que cumpla con las siguientes características arquitecturales de calidad (sintetizadas en [20]): *abierta* y

³ Sharable Content Object Reference Model

⁴ Learning Object Metadata, XML

⁵ Global Learning Consortium, implementa LOM

adaptable (*modificable-extensible-flexible* y basada en estándares), *escalable* (en cuanto a *usos de recursos*, memoria, espacio de almacenamiento, etc.), *global* (*portable* para ser usada en cualquier plataforma en el mundo), *integrada* (*interoperable* respecto a otros sistemas y formatos de datos proporcionados por diferentes proveedores de soluciones).

En este sentido, las últimas propuestas arquitecturales para sistemas e-learning apuntan al diseño de aplicaciones basadas en servicios. Un *servicio* es un tipo de componente de software reutilizable que proporciona a otros componentes o aplicaciones, un conjunto de funciones u operaciones que se invocan a través de una interfaz de programación. SOA facilita el desarrollo de nuevas aplicaciones reutilizables que resuelven problemas de integración, a través de aplicaciones independientes y distribuidas de manera que desde la red pueda accederse a las funcionalidades del sistema o aplicación, las cuales se ofrecen como servicios. Con respecto a nuestra AR, los servicios son considerados también como componentes arquitecturales.

A. Arquitectura de Referencia para E-Learning

Una AR describe la esencia de la arquitectura de software de una familia de sistemas similares con sus aspectos más significativos y relevantes [2][3][4][20][21]. El propósito de una arquitectura de referencia es proporcionar una guía para la derivación de sistemas concretos reutilizando la AR como un esquema instanciable. La Figura 2 muestra la vista lógica [25] en UML 2.0⁶ de la AR obtenida en [8].

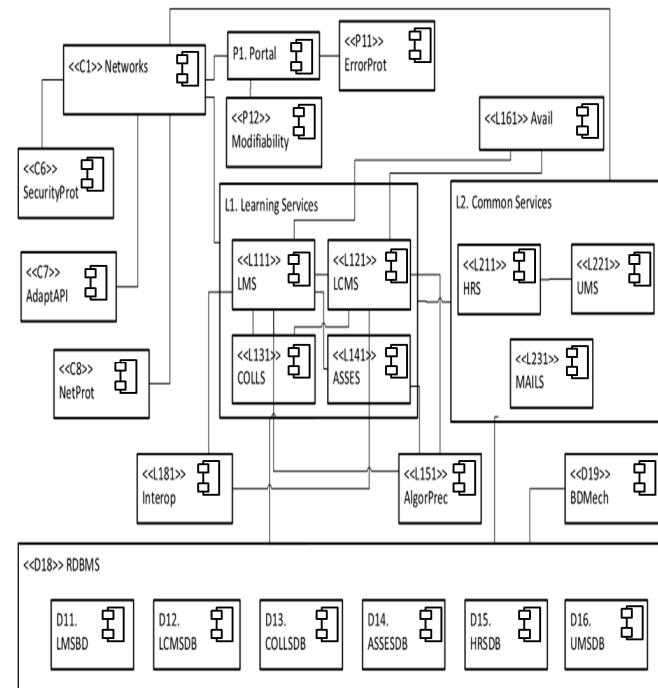


Figura 2. AR para Sistemas E-Learning [8]

Las funcionalidades básicas identificadas a partir de la especificación de los procesos de negocios [8][9] fueron los siguientes:

⁶ Unified Modeling Language, <http://www.omg.org/spec/UML>

- Crear, operar y administrar actividades de aprendizaje en línea => sistemas para gestión y creación de contenidos LMS, LCMS (Componentes de Servicios de Aprendizaje)
- Crear y proporcionar preguntas y pruebas para la evaluación del aprendizaje del estudiante => sistemas de evaluación ASSES (Componente de Servicios de Aprendizaje)
- Soportar la colaboración entre los usuarios => sistemas colaborativos COLLS (Componente de Servicios de Aprendizaje)
- Organizar recursos humanos y financieros => sistemas de Recursos Humanos HRS (Componente de Servicios Comunes)
- Administrar experiencias de aprendizaje virtual y distribuido vía Internet a estudiantes geográficamente distantes => sistemas de gestión de mails/navegador, gestión de usuarios MAILS, UMS (Componentes de Servicios Comunes e Infraestructura)

Por lo tanto, estos componentes funcionales forman parte de nuestra AR para e-learning. Los componentes que se muestran en la Figura 2 como estereotipos UML <<nombre del componente>> son denominados *puntos de variación* [3] o componentes que representan conjuntos de *variantes*, que serán instanciados en el momento de derivar la configuración del producto concreto; constituyen el *modelo de variabilidad* de la AR [3]. Nótese que en la Figura 2, los nombres de las componentes son numerados secuencialmente de acuerdo a la capa en donde están situados, P: Presentación, L: Lógica o Proceso, C: Comunicación, y D: Datos; se han mantenido los nombres en inglés para efectos de divulgación internacional [8]. También, se incluyen los componentes no funcionales, derivados de las propiedades de calidad.

Los componentes de la AR que se muestran en la Tabla I, son de alto nivel de abstracción y son articulados de acuerdo al estilo arquitectónico considerado para el dominio e-learning, es decir un estilo híbrido basado en eventos/capas, bajo el modelo cliente/servidor para distribución y comunicación, típico para sistemas o aplicaciones Web, de acuerdo a una arquitectura SOA; siguen generalmente una plataforma tipo LAMP⁷.

TABLA I. ADAPTACIÓN DE LOS COMPONENTES DE LA AR CON SUS CONECTORES [21]

Componentes	Conectores
P1. Portal	P1RC1, P1RC2
<<P11. ErrorProt>>	P11RP1
<<P12. Modifiability>>	P12RP1
L1. Learning Services	L1RL2, LIRC1
<<L111 LMS>>	L111RL1, L111RL12, L111RL131, L111RL141,
<<L121 LCMS>>	L121RL1, L121RL111
<<L151 AlgorPrec>>	L151RL111, L151RL121, L151RL141
<<L161 Avail>>	L161RL111, L161RL121
L16 LMS copy	L16RL111
L17 LCMS copy	L17RL121
<<L181 Interop>>	L181RL111, L181RL121
L18 AICC	L18RL181
L19 SCROM	L19RL181

⁷ Linux sistema operativo, Apache servidor web; MySQL/MariaDB gestor de bases de datos; Perl, PHP o Python lenguajes de programación

Componentes	Conectores
<<L131 COLLS>>	L131RL111, L131RL1
<<L141 ASSES>>	L141RL111, L141RL1
L2. Common Services	L2RC1
<<L211 HRS>>	L211RL221, L211RL2
<<L221 UMS>>	L221RL2
<<L231.MAILS>>	L231RL2
<<D18 RDBMS>>	D18RL1, D18RL2
D11 LMSDB	D11RD18, D11RL111
D12 LCMSDB	D12RD18, D12RL121
D13 COLLSDB	D13RD18, D13RL131
D14 ASSESSDB	D14RD18, D14RL141
D15 HRSDB	D15RD18, D15RL211
D16 UMSDB	D16RD18, D16RL221
<<D19 DBMech>>	D19RD18
<<C1 Networks>>	C1RP1, C1RC11, C1RC12
C11. Internet	C11RP1, C11RL1, C11RL2
C12. Satellite	C12RP1, C12RL1, C12RL2
<<C6SecurityProt>>	C6RC1
<<C7AdaptAPI>>	C7RC1
C4 JDBC	C4RC7
C5 ODBC	C5RC7
<<C8NetProt>>	C8RC1

En la Tabla I, Se denotan por *aRb* a los conectores entre dos componentes *a* y *b*, considerando AR un grafo no dirigido conexo; por ser simétricas las relaciones entre los conectores, se coloca solo una de ellas [15]. Un componente posee subcomponentes “hijos”, las cuales por defecto, siempre están relacionadas con el componente “padre”. Estas relaciones son representadas gráficamente por líneas rectas (lados del grafo) en UML en las Figuras 2 y 4.

Modelado de la Variabilidad para la AR

Los componentes que conforman cada punto de variación no aparecen gráficamente como componentes de la AR en la Figura 1 para facilitar su legibilidad, sin embargo las variantes más importantes se muestran en la Tabla I, con sus conectores y en la Tabla II, con las propiedades de calidad requeridas y las posibles soluciones arquitecturales o mecanismos que las realizan. Los componentes que no son variantes, son los componentes “obligatorios” que conforman el núcleo de componentes comunes que se consideraron al conformar la AR tomando en cuenta los requisitos funcionales básicos que se obtuvieron del análisis del dominio. Estos componentes pertenecerán siempre a la configuración de cualquier producto concreto que se derive de la AR. También pueden haber componentes opcionales. Su elección depende de la visión a largo plazo que se tenga al construir la AR. Es decir, que el arquitecto debe considerar futuras funcionalidades y prever los rápidos avances tecnológicos; esto le permitirá incluir componentesopcionales y nuevas variantes y/o nuevos puntos de variación, para que la AR tenga realmente un carácter evolutivo y pueda perdurar en el tiempo. Por otra parte, debemos aclarar que las restricciones impuestas por la plataforma son generalmente requisitos no funcionales que corresponden en general a propiedades de calidad requeridas por los componentes funcionales para su funcionamiento adecuado.

La Tabla II muestra el modelo de calidad extendido o *Extended Quality Model (EQM)* adaptado de [8], donde se describen los componentes comunes obligatorias u opcionales de la AR, es decir que pueden estar presentes en cualquier configuración arquitectural de un producto concreto que de

ella se derive y los componentes que son puntos de variación y sus variantes. Además se especifican las propiedades de calidad requeridas por cada componente y las soluciones que éstas ofrecen y las posibles restricciones entre componentes; además se dan sus prioridades, que son importantes en el momento de elegir una configuración concreta de producto. Se incluían componentes no funcionales, derivados de las propiedades de calidad.

Los componentes que se han introducido para satisfacer propiedades de calidad requerida por algún componente funcional ya existente, garantizan la trazabilidad entre los requisitos funcionales y no funcionales.

Los puntos de variación, <<C1>> Networks que agrupa las variantes C11, C12; <<C6>> SecurityProt agrupa los mecanismos para la seguridad, <<C7>> AdaptabAPI agrupa los mecanismos para la portabilidad variantes C4. JDBC y C5. ODBC, <<C8>> NetProt agrupa otros protocolos de red para tratar tiempo de respuesta, disponibilidad y capacidad en la transmisión, y <<D18>> RDBMS que agrupa como variantes sistemas manejadores de bases de datos o “Data Base Management Systems (DBMS)” relacionales para los respectivos servicios de aprendizaje, y <<D19>> DBMech que agrupa como variantes mecanismos propios a los DBMS para satisfacer las propiedades de calidad de capacidad y disponibilidad (por ejemplo bases de datos espejo o espejos con replicación), interoperabilidad e integridad; a efectos de legibilidad y abreviar la presentación; estas soluciones se dejaron agrupadas en un único punto de variación <<D19 DBMech>>.

Todos los subsistemas de L1 y L2 son renombrados como estereotipos por desempeñar las mismas tareas, <<L11>> LMS, <<L12>> LCMS, <<L13>> COLLS, <<L14>> ASSES, <<L21>> HRS, <<L22>> UMS, <<L23>>> MAILS; <<L16>> Avail agrupa L16. LMS copy y L17. LCMS copy, <<L15>> AlgorPrec agrupa algoritmos de cálculo, <<L18>> Interop agrupa L18. AICC y L19. SCROM; finalmente <<P11>> ErrorProt agrupa variantes de sistemas de auto corrección.

Adecuación de la AR a la IEEE-LTSA

Ahora bien, una AR para e-learning debe adaptarse al estándar IEEE-LTSA [8][12], el cual describe una arquitectura en cinco capas de alto nivel para sistemas de aprendizaje basados en tecnología de información.

La IEEE-LTSA especifica una solución en términos de una perspectiva de información tecnológica y es por lo tanto independiente respecto a aspectos pedagógicos, de contenido, culturales y de plataforma [12] (ver Figura 3).

La capa 3 de esta arquitectura es obligatoria para este estándar y debe estar presente en cualquier AR para e-learning que se diseñe; identifica los componentes de la arquitectura (funcionalidades principales que deben estar presentes) en cuatro procesos: D: Delivery (Entrega), LE: Learning Entity (Entidad de Aprendizaje), E: Evaluation (Evaluación) y C: Coach (Monitoreo); y dos almacenamientos de aprendizaje: Recursos y Registros.

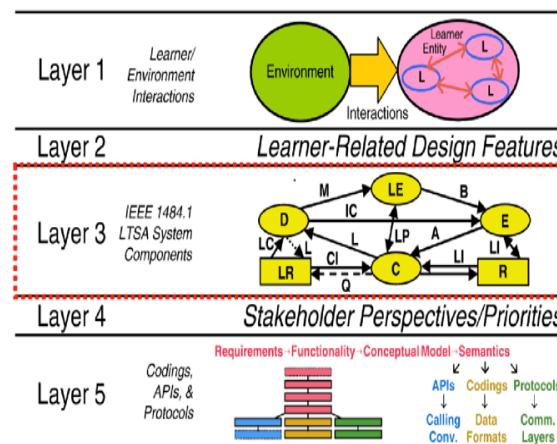


Figura 3. LTSA: IEEE P1484-1 Standard [12]

Las relaciones son M: Multimedia, B; Behavior (Comportamiento); IC: Interaction Context (Contexto de Interacción), LP: Learning Parameters (Parámetros de Aprendizaje), L: Locator (Localizador), A: Assessment (Evaluación), LC: Learning Content (Contenidos), CI: Catalog Info (Información del Catálogo), Q: Query

TABLA II. TABLA EQM (EXTENDED QUALITY MODEL) – ADAPTADA DE [8]

Componentes	Puntos de Variación/ Variantes	Descripción	Propiedad de Calidad con prioridad en 0		Restricciones
			Requiere/Proporciona indican las relaciones entre los componentes	Componentes que Proporcionan	
P1. Portal	No	Es la componente Interfaz Usuario (UI) a través de la cual se tiene acceso al sistema	- inteligibilidad (1), - aprendizaje (1), - ser atractiva (2), - protección de errores (3) - confiabilidad (2), autenticación (1) - integridad (2)	P1: técnicas de diseño de páginas, no se evalúan a nivel arquitectural P11: mecanismo corrector C1: SOAP/HTTP/ LDAP/HTTPS/Web service	CC: Componente Común funcional, obligatoria
<<P11. AutoCorrector>> <<P12	Si (*)	Provee autocorrección al introducir texto	Servicio Web	Directo del Browser, Editor HTML	Opcional

TABLA II. TABLA EQM (EXTENDED QUALITY MODEL) – ADAPTADA DE [8]

Componentes	Puntos de Variación/ Variantes	Descripción	Propiedad de Calidad con prioridad en 0		Restricciones
			Requiere	Proporciona indican las relaciones entre los componentes	
Modifiability>>	Sí (*)	Provee separación entre UI y la capa de proceso	- modificabilidad	Plataforma PHP - MVC ⁸	
L1. Learning Services	No	Agrupa sistemas que ofrecen servicios para el proceso de aprendizaje	Por cada sistema componente:	Proporcionadas para cada sistema componente:	CC funcionales opcionales u obligatorios
<<L111. LMS>>	Sí (*)	Sistemas de gestión de aprendizaje: permite administrar usuarios, recursos, contenidos (importar/exportar) y actividades de formación, administrar acceso, controlar el proceso de aprendizaje, generar informes, administrar servicios de comunicación; no incluyen autoría; variantes: SumTotal, Saba, OLAT, Sakai CLE, ATutor, Moodle, Chamilo, Canvas LMS, e-dceo	- idoneidad (1), - corrección-precisión (2), - rendimiento en tiempo (2) - capacidad-escalabilidad (1) - adaptabilidad (1) - disponibilidad-persistencia (2) - interoperabilidad (1)	L1: por construcción L151: AlgorPrec C1: HTTP/SOAP/FTP Messaging/SOAP/RPC/ Web service API ODBC/JDBC, Google drive L1: LMS copy (espejo, espejo con replicación) L1: AICC/SCROM, Google drive	CC funcional, obligatorio; CC funcional, obligatorio;
<<L121. LCMS>>	Sí (*)	Sistemas de autoría: creación de contenidos; permite a los autores registrar, ensamblar, administrar y publicar/exportar contenidos para ser entregados vía Web; Variantes: Evolution, ForceTen	- idoneidad (1), - corrección-precisión (2), - comportamiento en tiempo (2) - capacidad-escalabilidad (1) - adaptabilidad (1) - disponibilidad-persistencia (2) - interoperabilidad (1) - modificabilidad (2), - reuso (3)	L1: por construcción L151: AlgorPrec C1: HTTP/SOAP /Messaging/SOAP/RPC API ODBC/JDBC L1: LCMS copy (espejo, espejo con replicación) L1: AICC/SCROM L1: por construcción L1: por construcción	CC funcional, obligatorio variantes: Evolution, ForceTen, Eduslide por ser sistema crítico
<<L131. COLLS>>	Sí (*)	Sistema colaborativo: proporciona funcionalidades para crear y administrar sesiones colaborativas incluyendo comunicación síncrona (vide-conferencias, salón virtual) y asíncrona (blog, wiki, grupos, chats) variantes: Alfresco, Asana, Box.net, Clearspace, Drupal, Google Drive, Huddle	- comportamiento en tiempo (2), - disponibilidad-persistencia (2), - adaptabilidad (1), - capacidad-escalabilidad (1) - autenticación (1), - confidencialidad (2) - integridad (2)	C1: protocolos HTTP/SOAP T.120/H.323 - nube LDAP/HTTPS/Web service	CC funcional, obligatorio
<<L141. ASSES>>	Sí (*)	Sistema de evaluación: permite a los autores crear “surveys”, evaluación formativa y sumativa , exportar una evaluación vía AICC o SCORM, para ser entregada vía Web variantes: módulos de sistemas LMS	- idoneidad (1), - corrección-precisión (2), - comportamiento en tiempo (2) - capacidad-escalabilidad (1) - adaptabilidad (1) - disponibilidad-persistencia (2) - autenticación (1), - confidencialidad (2) - integridad (2)	L1: por construcción L151: Computation modules C1: HTTP/SOAP/ Messaging/RPC API ODBC/JDBC LDAP/HTTPS/Web service	CC funcional, obligatorio
<<L151. AlgorPrec >>	Sí (*)	Garantiza la precisión en los sistemas que requieren cómputos variantes: pueden haber muchos algoritmos diferentes	Componente	Módulos	Obligatorios
<<L161. Avail>>	Sí	Garantiza la disponibilidad en caso de falla	Subsistema	Componentes	Opcional
L16. LMS copy		Garantiza la disponibilidad en caso de falla	Subsistema	- Espejo, espejo con replicación	Opcional
L17. LCMS copy	Sí	Garantiza la disponibilidad en caso de falla	Subsistema	- Espejo, espejo con replicación	Opcional
<<L181. Interop>>		Garantiza importación/exportación	API		
L18. AICC	Sí	Garantiza importación/exportación	API		Obligatorio
L19. SCROM	Sí	Garantiza importación/exportación	API		Obligatorio

⁸ Model, View, Controller, Gang Of Four (GOF)

TABLA II. TABLA EQM (EXTENDED QUALITY MODEL) – ADAPTADA DE [8]

Componentes	Puntos de Variación/ Variantes	Descripción	Propiedad de Calidad con prioridad en 0		Restricciones
			Requiere/Proporciona indican las relaciones entre los componentes	Componentes que Proporcionan	
L2. Common Services	No	Agrupa sistemas de Recursos Humanos (RH) que ofrecen servicios a los que todos pueden acceder	Por cada sistema componente:	Proporcionadas para cada sistema componente:	CC funcionales, obligatorios
<<L211. HRS>>	Sí (*)	Sistemas de recursos humanos: permite administrar el perfil del usuario final incluyendo habilidades, competencias y tipo de trabajo; crear y mantener los registros de planes personales de desarrollo del usuario Variantes: los servicios de aprendizaje que se utilicen, LMS o LCMS, determinan cuales son las funcionalidades de RH que requieren variantes: Orange HRM, Centrifugo, SimpleHRM	- disponibilidad-persistencia (2), - capacidad-escalabilidad (1) - adaptabilidad (1), - autenticación (1), - confidencialidad (2)	C1: Messaging/SOAP/FTP/ API ODBC/JDBC LDAP/HTTPS, Web Service	CC funcional, obligatorio
<<L221. UMS>>	Sí (*)	Sistemas de gestión de usuarios; administra usuarios, grupos y roles a partir de todos los componentes Involucrados en la solución; maneja el registro y cuenta del usuario, la autorización y la autenticación variantes: Apache Syncope, OpenIAM	- disponibilidad-persistencia (2), - capacidad-escalabilidad (1) - adaptabilidad (1), - autenticación (1), - confidencialidad (2)	C1: Messaging/ SOAP/FTP API ODBC / JDBC LDAP/HTTPS, Web Service	CC funcional, obligatorio
<<L231. MAIIS>>	Sí (*)	Sistema e-mail/navegador: es responsable de enviar, recuperar y reenviar e-mails a los diferentes componentes involucrados en la solución. variantes: gmail, hotmail, yahoo, cantv.net, Firefox, Safari, Google chrome,	- comportamiento en tiempo (2), - disponibilidad-persistencia (2), - adaptabilidad (1) - capacidad-escalabilidad (1) - autenticación (1), - confidencialidad (2)	C1: SOAP/ HTTP/Messaging, /FTP/ T.120/H.323 Web services LDAP, HTTPS, Web service	CC funcional, obligatorio
<<C1. Networks>>	No	Agrupa los tipos de redes	Requeridas por cada componente:	Proporcionadas para cada componente:	CC funcionales, obligatorios
C11. Internet	Sí	Conjunto descentralizado de redes de comunicación interconectadas, basadas en protocolos TCP/IP, que hace ver las redes físicas heterogéneas que la componen como una red lógica única de alcance mundial.	- comportamiento en tiempo (2), - disponibilidad-persistencia (2), - adaptabilidad (1), - capacidad-escalabilidad (1) - autenticación (1), - confidencialidad (2) - integridad (2)	C4. Network protocols: SOAP/HTTP/FTP/T.120/H.323/Messaging (Publisher-Subscriber) LDAP/HTTPS/Web Services	CC, obligatorio
C12. Satelllite	Sí	Es un método de conexión a redes de comunicación (ej. Internet) utilizando como medio de enlace un satélite.	- comportamiento en tiempo (2), - disponibilidad-persistencia (2), - adaptabilidad (1), - capacidad-escalabilidad (1) - autenticación (1), - confidencialidad (2) - integridad (2)	C4. Network protocols: SOAP/HTTP/FTP/T.120/H.323/Messaging (Publisher-Subscriber) C2. ODBC C3. JDBC C5. Security protocols: LDAP/HTTPS/Web Services	CC funcional, opcional Messaging es implementado por un canal Publisher-Subscriber
<<C7. AdaptAPI>>	Sí (*)	Garantiza portabilidad	API, mecanismo		
C2. ODBC	Sí	Garantiza portabilidad	API	Mecanismo	obligatoria
C3. JDBC	Sí	Garantiza portabilidad	API	Mecanismo	obligatoria
<<C8. Network protocols>>	Sí (*)	Garantizan propiedades de la comunicación variantes: múltiples, dependen del mercado	Protocolos	Mecanismos, servicios Web	obligatorias
<<C6. Security protocols>>	Sí (*)	Garantizan las propiedades para el control de acceso al sistema y la consistencia de mensajes	Protocolos	Mecanismos, servicios Web, sistemas	obligatorias variante: múltiples dependen del mercado
<<D18. RDBMS>>	Sí (*)	Sistemas Gestores de Bases de	- capacidad-	D1. módulos, mecanismos propios de las	CC funcional,

TABLA II. TABLA EQM (EXTENDED QUALITY MODEL) – ADAPTADA DE [8]

Componentes	Puntos de Variación/ Variantes	Descripción	Propiedad de Calidad con prioridad en ()		Restricciones
			Requiere	Proporciona	
D19. DB mechanisms	Sí (*)	Datos; las diferentes bases de datos utilizadas por los subsistemas son consideradas un almacenamiento persistente compartido de datos: D11. LMSDB, D12. LCMSDB, D13. CALLSDB, D14. ASSESDB, D15. HRSDB, D16. UMSDB variantes: Oracle, MySQL, PostgreSQL, Firebird Garantizan propiedades de los RDBMS variantes: específicas para cada RDBMS	escalabilidad (1) - disponibilidad-persistencia (2) - interoperabilidad(1) - integridad (2) - adaptabilidad (1)	BD C1: API ODBC/JDBC	obligatorios; cada base de datos es obligatoria. Solo BD relacionales son consideradas;

(*) indica varios sistemas diferentes; cada uno es una variante

Para realizar esta adecuación, en [8] se consideró la proposición de [18], cuya AR integra los “frameworks” UKeU [20] y WbIS [21] y que además se adecúa a la IEEE-LTSA (Figura 3) [12]; se consideraron tres subsistemas o componentes principales:

- *Infraestructura & Servicios Comunes o Infrastructure & Common Services*: Portal (UI), Servicios Comunes (UMS: User Management System, MAIIS: Mail, HRS: Human Resources), Base de Datos (DBMS: Data Base Management Systems).
- *Servicios para el aprendizaje o e-learning services*: LMS, LCMS, sistema de evaluación (ASSES: Assessment System), sistema colaborativo (COLLS: Collaborative System), sistema de planificación (no fue considerado en esta AC como sistema aislado, es incluido con frecuencia en los sistemas ASSES).
- *Recursos para el aprendizaje o e-learning resources*: material en línea, material impreso, CD, DVD; estos aspectos se manejan en los sistemas LMS y LCMS.
- *Recursos Humanos o Human Resources*: Usuarios (estudiante, instructor o docente, administrador de la comunicación en línea, administrador del entrenamiento, autor, administrador del sistema), empresa u organización cliente, equipo de desarrollo, equipo de mantenimiento y soporte. Estos aspectos se consideran en los sistemas UMS y HRS.

La Tabla III muestra la adecuación general de los componentes de nuestra AR con el estándar IEEE-LTSA.

TABLA III. ADECUACIÓN AL ESTÁNDAR IEEE-LTSA DE LA AR PARA E-LEARNING [8]

AR LTSA capa 3	Infraestructura Técnica	Recursos Humanos	Recursos de Aprendizaje
L: Learning Entity	Portal, LMS, LCMS, ASSES, UMS	Estudiante	Material de aprendizaje
D: Delivery	LMS, LCMS	Estudiante, Instructor/Autor	LMSDB, LCMSDB
E: Evaluation	ASSES, UMS	Instructor	ASSESDB, UMSDB
C: Coach	ASSES, COLLS,	Instructor,	HRSDB, COLLSDB,

AR LTSA capa 3	Infraestructura Técnica	Recursos Humanos	Recursos de Aprendizaje
LR: Learning Resources R: Register	HRS, MAIIS DBMS	Supervisor Administrador	ASSESDB LMSDB, LCMSDB, COLLSDB

B. Derivación de la Configuración Arquitectural de Moodle a Partir de la AR

La Figura 4 presenta la arquitectura del sistema Moodle la cual se obtuvo de acuerdo al proceso de instanciación que se detalla a continuación, tomando en cuenta las capas del estilo arquitectural del dominio e-learning; esta figura muestra la arquitectura de Moodle [1] como una vista lógica [23] de componentes y conectores de alto nivel de abstracción y no una vista de despliegue donde aparecen las máquinas o dispositivos (servidores) donde se ejecuta el software; la vista de despliegue no se muestra en este trabajo por estar fuera del alcance de los objetivos iniciales, sin embargo puede ser derivada a partir de la vista lógica considerando el modelo Cliente/Servidor propio del estilo arquitectural del dominio e-learning. Nótese que los roles de los conectores o asociaciones entre componentes son todos Proporciona/Requiere (ver Figura 5) y no los hemos especificados para hacer más legible el diagrama.

Guía para el Proceso la Instanciación de la AR

Se procede con las siguientes actividades para elegir en la Tabla EQM (Tabla II) los componentes de la AR que formarán parte de la arquitectura del sistema concreto:

1. Si el componente es común y obligatorio, se coloca tal cual en la configuración arquitectural concreta del sistema, si no:
2. Si el componente es opcional, se debe determinar si se coloca o no en la configuración concreta; el arquitecto o ingeniero de software debe tomar esta decisión
3. Si el componente es un punto de variación, se deben estudiar sus variantes:
 - *Selección de la variante*: es la parte más delicada del proceso de instanciación, en la cual también

interviene la experticia del arquitecto de software; deben examinarse las diferentes alternativas propuestas en la Tabla EQM como variantes y tomar en cuenta las propiedades de calidad prioritarias que se quieren favorecer, como por ejemplo, para *interoperabilidad* se puede tener AICC o SCROM

- La variante seleccionada se coloca como componente de la configuración arquitectural concreta del sistema
4. Los conectores entre componentes comunes y variantes de la AR se mantienen

Justificación de la Selección de los Componentes

- *Capa de Presentación:*

Está constituida por un componente interfaz usuario que contempla la variante *Páginas Web Dinámicas*, la cual posee facilidades de corrección automática mediante un *Editor HTML* que corre directamente en el navegador, respondiendo al requisito de calidad *usabilidad* en cuanto a la protección de errores tipográficos; los aspectos de amigabilidad y estética de la interfaz usuario no son considerados aquí ya que dependen del diseño de las páginas y no de la arquitectura en sí misma.

La *modificabilidad* de la IU respecto a la capa de proceso está resuelta por la plataforma de desarrollo PHP, componente *Plataforma PHP*, la cual permite implementar un patrón tipo MVC para realizar esta propiedad, que implica la separación entre la capas IU y la presentación para facilitar las modificaciones de la IU respecto a los rápidos cambios tecnológicos.

- *Capa de Proceso:*

Contempla el componente común obligatorio *L1. Moodle E-learning Services*, que contiene como subcomponentes al sistema Moodle *LMS* con el core de funcionalidades propias al sistema; se ofrecen como variantes para Moodle *LMS* los servicios colaborativos de *Google Drive* y de *Asignaciones*. Para satisfacer el requisito prioritario de calidad de *interoperabilidad* se ha seleccionado el formato estándar *SCROM* para objetos de aprendizaje y en cuanto a la *corrección-precisión* requerida por el core de Moodle y las Asignaciones se tienen *módulos específicos de cálculo*, algoritmos considerados con ese propósito. Por otra parte, el core de Moodle utiliza el componente común *L2. Common Services*; en particular para el sistema de recursos humanos se instanció el subcomponente *HRS* en Moodle *HRS* y como sistema de navegador/mail se consideró *Google Chrome/gmail* por ser de uso frecuente en el ambiente académico.

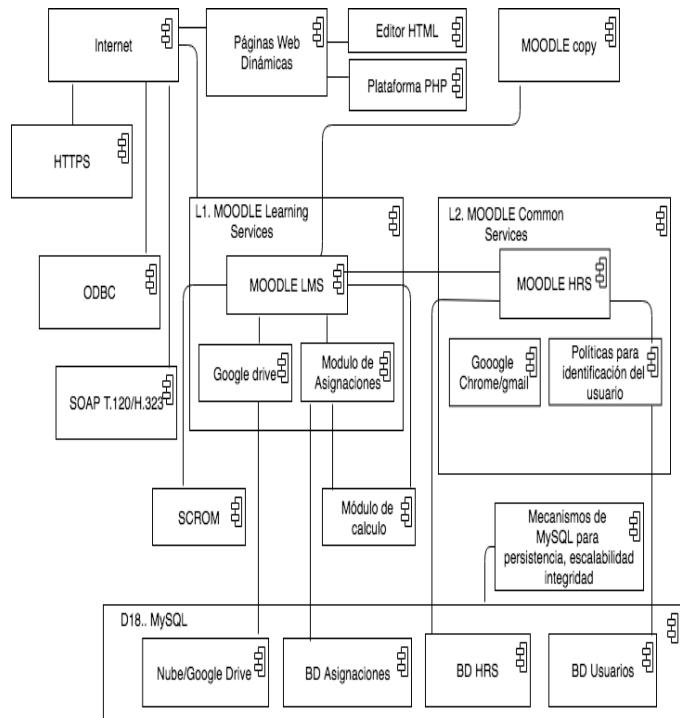


Figura 4. Configuración Arquitectural Concreta del Sistema Moodle Derivada Instanciando la AR

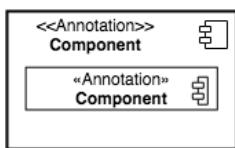
- *Capa de Datos:*

En <<D18. RDBMS>> se colocó como variante la base de datos relacional *MySQL*, que aparece en la mayoría de las versiones de Moodle estudiadas. En cuanto a las bases de datos específicas a los subsistemas considerados en Moodle, se colocaron las bases de datos específicas relativas a los servicios de aprendizaje de Moodle: *BD Asignaciones* y *Nube/Google Drive* para los servicios colaborativos en las prácticas docentes de aulas virtuales, etc.; en cuanto a los servicios comunes se instanciaron las bases de datos *BD HRS*, *BD Usuario* específicas a Moodle para el manejo de recursos humanos y de políticas de identificación de usuario respectivamente.

- *Capa de Comunicación:*

Se seleccionó *Internet* como red, variante de <<CI. Network>> por ser la de uso más común; para la seguridad en cuanto al paso de mensaje se seleccionó *HTTPS*, para la portabilidad también a nivel de bases de datos, la *API ODBC* que puede ser utilizada por MySQL y como protocolo de red por excelencia en el caso de Moodle, se seleccionó *SOAP*. La capa de comunicación es la que puede garantizar el uso de servicios Web y también la disponibilidad de todo el sistema; debe notarse que si la conexión a Internet falla, el sistema no estará disponible.

La Figura 5 ilustra la notación usada en los diagramas UML 2.0 de las Figuras 2 y 4.



Indica un componente de la arquitectura; un componente puede contener sub-componentes; el estereotipo <<Annotation>> indica el nombre o comentario que se quiera colocar para identificar el componente. Los componentes arquitecturales y sus conectores (asociaciones o relaciones) son especificados a un alto nivel de abstracción, no deben confundirse con las clases que se representan en los diagrama de clases

Es un conector que indica la relación entre componentes; puede ser más complejo, indicando interfaces, roles, quien proporciona la información —○— o quien requiere la información; en nuestro caso dejamos la linea recta a efectos de dar mayor legibilidad al diagrama; según el caso ella indica que un componente requiere los servicios de otro componente, el cual los proporciona (ver Figuras 3, 4 y Tabla II). En el caso de los diagramas UML 2.0 de las arquitecturas presentadas, todos los roles son Proporciona/Requiere

Figura 5. Notación UML 2.0 para Representar Arquitecturas de Software en Términos de Componentes y Conectores

V. CONCLUSIÓN

Una AR representa una familia de sistemas similares y se utiliza para derivar un producto concreto sobre pedido o del inglés “*product on-demand*”. En otras palabras la ventaja de una AR es de utilizarla como plantilla instanciable para satisfacer los requisitos de un sistema concreto, pedido por un cliente. Por lo tanto el cliente especificará sus requisitos y el equipo de configuración de la AR debe chequear que satisfaga en lo posible y correctamente dichos requisitos. En caso de no ser así, se tienen dos opciones, o modificar la AR para incluir los requisitos faltantes, o adecuar lo que se tiene, explicando al cliente lo que no puede obtener. Debe observarse que el objetivo es obtener un sistema de software concreto ejecutable, realizando los enlaces correspondientes entre los componentes arquitecturales abstractos y el código representado por módulos reutilizables. Sin embargo, no todas estas actividades están automatizadas aún, por lo que se están realizando muchas investigaciones al respecto.

Este trabajo muestra una guía para realizar una configuración “sencilla” por instanciación de una AR, considerando una configuración arquitectural para el sistema Moodle como producto concreto. El proceso de configuración no se había presentado en los trabajos anteriores [7][8] que dieron origen a esta AR para e-learning. Nuestra contribución principal es la obtención de la configuración arquitectural de Moodle que se muestra en UML 2.0 en la Figura 4. Es claro que para realizar una derivación completa de un producto, los módulos de los componentes de la arquitectura “concreta” de Moodle deben ser enlazados con el código correspondiente; pero esta fase está fuera del alcance de este trabajo. Debe notarse que la AR considerada para e-learning, desarrollada en [7][8] se ha realizado considerando los aspectos de calidad en etapas tempranas y estos se reflejan al realizar la instanciación, con lo cual se puede afirmar la calidad de la configuración arquitectónica del sistema concreto producido mediante esta AR, en este caso Moodle. Como trabajo futuro se espera realizar la especificación del proceso completo de derivación del producto, considerando Moodle como un caso de estudio. La automatización completa del proceso de configuración no es posible, debido a la intervención necesaria de la experticia del arquitecto o ingeniero de software, sin

embargo el diseño de una herramienta para la automatización parcial de este proceso está prevista. Además, se espera considerar la sugerencia de explorar aplicaciones en los sistemas cyber-físicos y el “ubiquitous learning”; por ejemplo, aquellos basados en la tecnología de sensores distribuidos vía internet “internet of things (IoT)”, haciendo énfasis en sus implicaciones en procesos de aprendizaje electrónico.

AGRADECIMIENTO

Al Postgrado en Ciencias de la Computación, Facultad de Ciencias, Universidad Central de Venezuela y al Vicerrectorado de Investigación y Postgrado de la Universidad Pedagógica Experimental Libertador, por el apoyo a las investigadoras para la realización de la investigación.

A los evaluadores por sus enriquecedoras observaciones y sugerencias que contribuyeron a mejorar la contribución.

REFERENCIAS

- [1] M. Shaw and D. Garlan, “Software Architecture. Perspectives of an Emerging Discipline”, 1996.
- [2] ISO/IEC NP 26550, “Software and Systems Engineering – Reference Model for Software and Systems Product Lines”. ISO/IEC JTC1/SC7 WG4, 2013.
- [3] K. Pohl, G. Bockle, G. & F. Van Der Linden, “Software Product Line Engineering: Foundations, Principles, and Techniques”. Springer, 2005.
- [4] E. Nakagawa, P. Antonio, and M. Becker, “Reference Architecture and Product Line Architecture: a Subtle but Critical Difference”, Crancovic V., pp. 207-211, Springer-Verlag, Berlin, Heidelberg, 2011.
- [5] P. Clements and L. Northrop, “SPL: Practices and Patterns”, 3rd ed. Readings, MA, Addison Wesley, 2001.
- [6] E. Bérard, “Testing of Object-Oriented Software. Proceedings of the Eighth International Conference on Technology of Object Oriented Languages and Systems”. USA: Prentice-Hall, Inc. 1992.
- [7] Y. Esteves, “Marco de Referencia Arquitectónico en el Dominio del Aprendizaje Electrónico”. Tesis Doctoral: Facultad de Ciencias, Escuela de Computación, UCV. Caracas, Marzo 2017.
- [8] Y. Esteves, F. Losavio, “Modelado del Negocio Dirigido por la Calidad para una Arquitectura de Referencia en el Dominio del Aprendizaje Electrónico”. RACCIS, Vol. 6, No. 1, pp. 59-76, 2016.
- [9] R. Perrott, “HPCC”, pp. 776–784, Springer-Verlag Berlin Heidelberg, 2007.
- [10] H. A. Duran-Limon, F. Castillo-Barrera, R. Lopez-Herrejón, “Towards an Ontology-Based Approach for Deriving Product Architectures”, 15th Intern. Software Product Line Conference SPLC’11, Vol. 2, Article No. 29, ACM, Munich, Germany, New York, NY, USA, August 2011.
- [11] F. Losavio, O. Ordaz, and H. Márquez, “Assessment for Quality Product Derivation from a Software Product Line Reference Architecture”. RACCIS, Vol. 5, No. 2, pp. 48-59, 2015.
- [12] IEEE, “Draft Standard for Learning Technology – Learning Technology Systems Architecture”, Technical Report IEEE-Std Draft-P1484-1, 2002.
- [13] ISO/IEC 25010, “Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE). Systems and Software Quality Models”, ISO/IEC JTC1/SC7/WG6. Ginebra, 2011.
- [14] Moodle, “The Moodle Project”. <https://moodle.org>, Julio 2017.
- [15] F. Losavio, O. Ordaz, and V. Esteller, “Refactoring-Based Design of Reference Architecture”. RACCIS, Vol. 5, No. 1, pp. 32-48, 2015.

- [16] UNESCO, "Aprendizaje Abierto y a Distancia. Consideraciones sobre Tendencias, Políticas y Estrategias". Ediciones Trilce. Uruguay, 2002.
- [17] J. Boneu, "Plataformas Abiertas de E-learning para el Soporte de Contenidos Educativos Abiertos". Revista de Universidad y Sociedad del Conocimiento, 2007.
- [18] J. Habraken, "Reference Architecture for E-Learning Solutions", Master Thesis, Open University Faculty Computer Science, January 2008.
- [19] Cisco, "Model of an E-Learning Solution Architecture for the Enterprise". Cisco Systems, 2001. http://www.cisco.com/warp/public/10/wwtraining/elearning/learn/whitepaper_docs/solution_architecture_wp.pdf.
- [20] UK e-Universities Worldwide, "Principles and Practice in E-Learning Platform Architecture", UKeU, 2002.
- [21] S. Retalis and P. Avgeriou, "Modelling Web-Based Instructional Systems", Journal of Information Technology Education, Vol. 1, No. 1, 2002.
- [22] M. A. Conde, F. J. García, "mLearning, de Camino Hacia el uLearning", en Avances en Informática y Automática. Salamanca. ISBN: 978-84-612-1283-5, pp. 11-20, 2007.
- [23] N. Hellers, "Aprendizaje Portátil, la Revolución que se Viene. E-Learning América Latina". http://www.elearningamericalatina.com/edicion/junio1_2004/na_1.php, 2010.
- [24] J. J. Maldonado, "Modelo de Calidad de un LMS", CEDIA – Proyecto de Objetos de Aprendizaje II, URI: <http://repositorio.cedia.org.ec/handle/123456789/1002>
- [25] P. Krutchen, "Architectural Blueprints — The “4+1” View Model of Software Architecture", IEEE Software Vol. 12, No. 6, pp. 42-50, November 1995.

Análisis Comparativo de Técnicas de Modelado de Metas: Un Enfoque Dirigido por la Planificación Estratégica del Negocio

Mayela Delgado

Departamento de Computación
Universidad de Carabobo
Valencia, Venezuela
mdelgadoh@uc.edu.ve

Alfredo Matteo

Centro ISYS, Escuela de Computación,
Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela
alfredojoze.matteo@gmail.com

Francisca Losavio

Centro ISYS, Escuela de Computación,
Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela
francislosavio@gmail.com

Resumen—La orientación a metas ha hecho importantes aportes a la Ingeniería de Requisitos. Las metas refinan la visión global del negocio, exponen el valor de un sistema/software y proporcionan la fundamentación para su desarrollo. No obstante, las bondades obtenidas de la Ingeniería de Requisitos Orientada a Metas no garantizan que los sistemas de software estén alineados con la estrategia del negocio, ni que contribuyan efectivamente a la proposición de valor. Como un medio para mejorar esta situación se plantea la utilización de la Planificación Estratégica en el direccionamiento de la Ingeniería de Requisitos Orientada a Metas, considerando que en este proceso se establecen las metas de más alto nivel y las estrategias del negocio. Dada la existencia de un grupo diverso de técnicas orientadas a metas, el objetivo de esta investigación es evaluar las técnicas de mayor difusión con el fin de determinar cuál satisface en mayor grado los requisitos de un modelo de proceso de Ingeniería de Requisitos dirigido por la Planificación Estratégica. La evaluación realizada se basó en el Análisis de Características propuesto por Kitchenham. El grupo de características establecido persigue examinar la capacidad de las técnicas para: (1) el modelado de los conceptos básicos de la Planificación Estratégica, y (2) la representación de metas en diferentes niveles de abstracción y sus interrelaciones.

Palabras Clave—Orientación a Metas, Ingeniería de Requisitos, Ingeniería de Requisitos Orientada a Metas, Planificación Estratégica

I. INTRODUCCIÓN

Uno de los mayores problemas concernientes al desarrollo de software está relacionado con la brecha de valor del software [1], lo que evidencia la necesidad de asegurar un aporte efectivo de los sistemas de software a las metas y estrategias del negocio. El estudio presentado en [1] destaca la necesidad de enfocar los sistemas de software, desde etapas tempranas del ciclo de desarrollo, hacia la generación de valor [2]. Tal situación demanda que en la Ingeniería de Requisitos (IR) [3] se vinculen los objetivos de los sistemas de software a las metas de alto nivel y estrategias del negocio, destacando su contribución a la proposición de valor [4]. Ésto indica que desde la IR se debe canalizar la alineación estratégica de los sistemas de software ([5], [6]).

Las técnicas tradicionales de la IR se concentran, primordialmente, en procesos y datos, y no capturan la fundamentación de los sistemas de software [7]. Ésto dificulta la comprensión de los requisitos dentro del contexto de intereses de alto nivel en el problema del dominio. Sin embargo, un paradigma más reciente como la IR Orientada a Metas [8] ofrece una solución a esta situación debido a que las metas refinan la visión global del negocio, exponen el valor de un sistema/software y proporcionan la fundamentación para su desarrollo ([8], [9]). Una meta es un fin general que debe ser alcanzado y es de naturaleza cualitativa [10]. En [9] se define una meta como una intención sin hacer referencia a las propiedades o uso del sistema de software.

No obstante, a pesar del aporte significativo de la IR Orientada a Metas (GORE – Goal-Oriented Requirements Engineering) [8], en el modelado no se puede asegurar el soporte efectivo a las metas de alto nivel y estrategias, lo cual incide negativamente en la alineación estratégica de los sistemas de software y en su contribución a la proposición de valor del negocio.

Ante esta problemática se propone la integración de GORE con la Planificación Estratégica (PE). Esta integración se basa en la utilización de los elementos fundamentales de la PE en el modelado de metas con el fin de direccionar el proceso de IR. La PE constituye un dominio extenso en el mundo de los negocios; sin embargo, para efectos de este trabajo lo importante es destacar que es un proceso del que se derivan las metas de más alto nivel y las estrategias del negocio.

La integración de la Ingeniería de Requisitos Orientada a Metas y la Planificación Estratégica persigue los siguientes objetivos:

- Vincular explícitamente las metas de un sistema/software con las metas y/o estrategias organizacionales. Este enlace facilita el análisis de contribución de una solución a metas y/o estrategias específicas.
- Soportar el análisis de las estrategias mostrando su descomposición en metas y cursos de acción más concretos. En este análisis también se incluye la evaluación de cursos de acción alternativos.

Adicionalmente, el uso de una sola notación para el modelado de estrategias y de metas facilita la comunicación con las partes interesadas (stakeholders) y la negociación en caso de conflictos.

Es necesario tomar en cuenta que la Ingeniería de Requisitos Orientada a Metas y la Planificación Estratégica son procesos independientes y difieren en sus objetivos primarios. La primera persigue generar el conjunto de requisitos que definirán las propiedades de un sistema/software. La segunda establece la dirección hacia donde se deben canalizar las actividades y procesos de una organización. Sin embargo, con la integración de ambos procesos se busca obtener un efecto sinérgico derivado de la complementariedad entre ellos. La Planificación Estratégica amplifica el alcance de la Ingeniería de Requisitos Orientada a Metas señalando las estrategias y metas organizacionales a las cuales debe aportar un sistema/software, lo cual contribuye a clarificar su rol dentro de la organización, evidenciando su colaboración en la creación de valor. Por su parte, el razonamiento aplicado en la Ingeniería de Requisitos Orientada a Metas apoya el análisis de las estrategias ayudando a concretar los cursos de acción y las metas establecidas en la Planificación Estratégica sin necesidad de entrar en detalles específicos relativos a su implementación.

Este trabajo forma parte de una investigación más amplia centrada en el desarrollo de un modelo de proceso de Ingeniería de Requisitos orientado a metas y dirigido por la Planificación Estratégica, que tiene por propósito: (1) enfocar los sistemas de software desde etapas tempranas del desarrollo hacia su alineación con las metas de más alto nivel y estrategias, (2) contribuir al modelado de las estrategias, y (3) representar en el modelado las interrelaciones entre las metas correspondientes a los niveles de dirección y gestión del negocio, y las metas de los sistemas de software.

El análisis y modelado de metas constituye la columna vertebral del proceso de Ingeniería de Requisitos mencionado previamente; de ahí que se consideró pertinente establecer como paso inicial la selección de una técnica de Ingeniería de Requisitos Orientada a Metas, la cual se integraría en el diseño del proceso, y en etapas ulteriores se adaptaría para extender el modelado de metas mediante la incorporación de los conceptos básicos de la Planificación Estratégica.

Dada la existencia de un grupo diverso de técnicas dentro de GORE, en esta investigación se plantea como objetivo principal la evaluación de las técnicas de mayor difusión en el campo de la Ingeniería del Software (IS), tal como lo exponen los estudios presentados en: [7], [8], [11] y [12], con el fin de determinar cuál de ellas ofrece mejor soporte al modelo de proceso de Ingeniería de Requisitos que se plantea desarrollar posteriormente.

En este trabajo se va a utilizar el término técnica para englobar las acepciones: técnica (propriamente dicha) y método. La primera acepción (técnica) hace referencia a las especificaciones de una notación utilizada para el modelado; y la segunda (método), alude a una notación sustentada en un proceso que dirige sistemáticamente el modelado.

Las técnicas seleccionadas para la evaluación fueron: KAOS (Knowledge Acquisition in autOmated Specification of

Software) [13], NFR Framework (Non-Functional Requirements Framework) [14], I* (I-Star) [15], GBRAM (Goal-based Requirements Analysis Method) ([16], [17]), Tropos [18], GRL (Goal-oriented Requirements Language) [19] y URN (User Requirements Notation) [20].

Este estudio aplica los principios del Análisis de Características propuesto por Kitcheham [21], el cual sugiere la definición de un grupo de características que deben evaluarse para determinar si una técnica, método, herramienta o producto de la IS se ajusta a un conjunto específico de necesidades. La evaluación de una técnica consistió en determinar para cada característica especificada el grado de conformidad. Al finalizar se obtuvo una puntuación global por técnica, donde el mayor valor corresponde a la que mejor se ajustó a los requisitos establecidos.

Esta investigación resulta de la extensión del trabajo presentado en [22], donde se efectúa la evaluación de técnicas de GORE considerando su capacidad para el modelado de metas en diferentes niveles de abstracción. Los niveles de abstracción considerados en [22] fueron: Corporativo, Negocio, Operativo, Sistema y Software. El nivel Corporativo se refiere a las metas relacionadas con la gestión global de una organización y/o las áreas de negocio en las que se participa o se desea participar. El nivel de Negocio incluye las metas relativas al aprovechamiento de las capacidades estratégicas y a la forma en que se compite o se espera competir en áreas específicas. El nivel Operativo comprende las metas vinculadas a la gestión de la organización, pueden estar asociadas al diseño de los procesos y/o a la definición de roles de los actores organizacionales. Las metas del nivel Sistema establecen el comportamiento del sistema y sus interacciones. En el nivel de Software, las metas especifican el funcionamiento del producto de software y la forma en que se espera ejecute los servicios requeridos.

El presente estudio amplía el alcance de la evaluación de las técnicas realizada en [22] incluyendo un grupo de características que permitan determinar la aplicabilidad de las técnicas en un proceso de IR que integre en el modelado los conceptos básicos de la PE.

Con respecto a la aplicación de la orientación a metas en el análisis de la alineación estratégica de sistemas de software y el modelado de las estrategias, existen varios estudios relacionados, cuyo desarrollo se basó en la utilización de alguna de las técnicas que se evalúan en esta investigación. En [23] se propone un marco de trabajo denominado B-SCP, el cual direcciona el modelado de la estrategia y la alineación de los requisitos del software. Se apoya en los constructos del lenguaje de modelado de metas I* [15]. El trabajo desarrollado en [24] se enfoca en el modelado de requisitos y su validación contra la estrategia. Se basa en la metodología Tropos [18]. Combina los modelos de metas con el Cuadro de Mando Integral [25], herramienta muy utilizada en la PE. La investigación realizada en [26] propone la construcción de un modelo de alineación estratégica basado en GRL [19].

En relación al análisis comparativo entre técnicas orientadas a metas, los investigadores reconocen que existen particularidades en cada técnica, por lo que los estudios se enfocan en determinar cuál es la técnica más apropiada para un

entorno específico de modelado. El trabajo presentado en [12] persigue la definición de lineamientos metodológicos para guiar la utilización de las diferentes técnicas según el contexto de aplicación. El análisis realizado en [27] está dirigido a diferentes metodologías de modelado basado en I* [15], examinando el soporte brindado por cada una a la construcción de los modelos. En [28] la evaluación se concentra en determinar la adecuación de las técnicas al modelado de sistemas colaborativos, subrayando la importancia de la selección apropiada del enfoque de metas a ser utilizado en un proceso de IR.

A continuación se indica la organización del contenido de este trabajo. En la Sección II se explican brevemente los fundamentos de cada una de las técnicas seleccionadas para la evaluación. La Sección III describe el conjunto de características que se van a evaluar. En la Sección IV se muestran los resultados de la evaluación de conformidad de las características para cada técnica. La Sección V contiene el análisis comparativo de los resultados obtenidos en la evaluación. Por último, la Sección VI presenta las conclusiones obtenidas luego de examinar los resultados.

II. VISIÓN GENERAL DE LAS TÉCNICAS DE GORE A EVALUAR

Cada una de las técnicas seleccionadas para la evaluación – KAOS, NFR Framework, I*, GBRAM, Tropos, GRL y URN – constituye un aporte significativo dentro del desarrollo de GORE.

KAOS es una técnica de modelado de metas que ha sido aplicada en múltiples proyectos académicos e industriales. Permite la generación de varios modelos complementarios que facilitan la trazabilidad entre las metas del sistema/software y los componentes arquitectónicos operacionales. Esta técnica, se ha ido enriqueciendo desde su origen con la incorporación de nuevos modelos y un conjunto de lineamientos y recomendaciones metodológicas ([7], [8], [9]).

El NFR Framework [14], representa el primer marco de trabajo concentrado en el modelado y análisis de los requisitos no funcionales (NFR – Non Functional Requirements). La relevancia del tratamiento de estos requisitos estriba en que condicionan y/o restringen el comportamiento del software, definiendo las propiedades de calidad requeridas en el sistema/software en forma global o en un conjunto específico de componentes, servicios y/o funciones ([8], [9], [14]). Los fundamentos del NFR Framework han aportado a otras técnicas orientadas a metas (I*, Tropos y GRL) mecanismos para evaluar y justificar alternativas de diseño considerando el impacto sobre los NFR relevantes a la situación que se modela ([7], [14]).

I* y sus derivadas, Tropos y GRL, representan las relaciones entre actores organizacionales, enfocándose en el análisis del contexto del sistema. Sus características han permitido aplicarlas en distintas áreas, tales como: IR, reingeniería de procesos de negocio, análisis de impacto organizacional y modelado de procesos de software ([7], [11], [15]).

GBRAM ([16], [17]), enfatiza el rol de las metas en la identificación, organización y justificación de requisitos.

GBRAM aporta un grupo de heurísticas que persiguen la obtención de un conjunto de metas que direccione la obtención y especificación de los requisitos del sistema/software ([11], [16], [17]).

URN [20] es un estándar internacional que integra metas y escenarios. Utiliza las bondades de GRL para el modelado de metas conjuntamente con una notación para escenarios, los Mapas de Casos de Uso, ofreciendo una notación visual integrada dirigida a mejorar las especificaciones de los requisitos ([20], [29]).

A continuación se presenta la concepción básica de cada técnica estudiada.

A. KAOS

KAOS es un método para educir, especificar y analizar metas, requisitos, escenarios y asignaciones de responsabilidades. Se basa en cinco vistas complementarias [8]:

- Vista Intencional. Refleja las metas funcionales y no funcionales del sistema y sus enlaces de contribución. Ayuda a identificar y comprender las razones que justifican un nuevo sistema. Incluye el análisis de riesgos.
- Vista Estructural. Esta vista se enfoca en los objetos conceptuales utilizados en el sistema actual y el sistema futuro. Un objeto conceptual puede representar una entidad, un agente, una asociación o un evento.
- Vista de Responsabilidad. Describe los agentes que forman el sistema, sus responsabilidades con respecto a las metas del sistema y sus interfaces para la comunicación.
- Vista Funcional. Se enfoca en los servicios que el sistema debe proporcionar con la finalidad de operacionalizar las metas funcionales.
- Vista de Comportamiento. Define los comportamientos requeridos para que el sistema satisfaga sus metas.

B. NFR Framework

El NFR Framework se enfoca en la identificación, análisis y representación de los NFR ([14], [30]). Un NFR no describe lo que el software hará sino cómo lo va a hacer, actúa restringiendo la funcionalidad. Los NFR son difíciles de probar dado que su cumplimiento no puede determinarse de manera taxativa por lo que, usualmente, se evalúan subjetivamente [14]. El NFR Framework presenta una propuesta sistemática para enfrentar las dificultades inherentes a los NFR y utilizarlos en el direccionamiento del proceso de desarrollo de software o parte de él [30].

C. I* (I-Star)

I* (i-estrella) fue desarrollado para modelar y razonar acerca de los entornos organizacionales y sus sistemas de información. Incluye un lenguaje gráfico para el modelado de las relaciones estratégicas entre actores organizacionales [15]. El término actor se refiere a cualquier unidad semi-autónoma

dentro de una organización, capaz de establecer relaciones con otros actores con un propósito definido [31]. En sus inicios esta técnica se orientó hacia las etapas tempranas de la IR [15]; también ha sido utilizada en la reingeniería de procesos de negocio y en el análisis de impacto en la organización [15].

D. GBRAM

GBRAM es un método enfocado en: (1) ayudar en la recopilación de las metas de la empresa y del software y (2) apoyar el proceso de descubrir, identificar, clasificar, refinar y elaborar las metas hasta la obtención de requisitos operacionales ([16], [17]). Su principal aporte es que provee heurísticas y lineamientos para identificar y construir metas [17]. Comprende un conjunto de actividades agrupadas en dos áreas principales: (1) el análisis de metas, que se enfoca en la identificación de las metas, su organización y clasificación; y (2) el refinamiento de metas, cuyo propósito es la obtención de los requisitos operacionales del sistema partiendo de las metas establecidas ([16], [17]).

E. Tropos

Tropos es un método para el desarrollo de software orientado a agentes y centrado en la arquitectura, basado en dos características claves [18]: (1) el uso de conceptos a nivel del conocimiento, tales como agente, meta, plan y otros, a lo largo del ciclo de desarrollo de software y (2) la asignación de un rol pivote al análisis y especificación de requisitos durante el análisis del ambiente y del sistema a desarrollar [32]. Para el modelado adopta la notación de I*. Comprende cinco fases [32]:

- Requisitos Tempranos. Durante esta fase se identifican las partes interesadas importantes y sus objetivos.
- Requisitos Tardíos. Se enfoca en la descripción del sistema futuro dentro de su ambiente operacional.
- Diseño Arquitectural. Se ocupa de la definición global de la arquitectura en términos de subsistemas interconectados por medio de flujos de datos y control.
- Diseño Detallado. Comprende la definición detallada de los componentes arquitectónicos en relación a las entradas, salidas, control y otra información relevante.
- Implementación. En esta fase las especificaciones obtenidas en el diseño detallado se transforman en una estructura para la implementación.

F. GRL

El Lenguaje de Requisitos Orientado a Metas (GRL) es un subconjunto de la Notación para Requisitos del Usuario (URN) utilizado para modelar y analizar requisitos, especialmente los NFR y atributos de calidad mediante la elaboración de grafos de metas ([20], [33], [34]). GRL combina elementos del NFR Framework y de I* para soportar el modelado de metas/agentes y el razonamiento en los modelos de metas.

G. URN

La Notación para Requisitos del Usuario (URN) es un lenguaje gráfico de modelado orientado a la educación, análisis,

especificación y validación de requisitos. Es el primer estándar internacional que direcciona explícitamente, en una forma gráfica y unificada, metas y escenarios y sus interrelaciones ([33], [34]). Enfoca el modelado de metas hacia el manejo de los NFR y atributos de calidad, y el modelado de escenarios hacia los requisitos operativos, los requisitos funcionales y el razonamiento sobre la arquitectura y el rendimiento [36]. Para el modelado de metas, URN utiliza GRL, y para los escenarios emplea los Mapas de Casos de Uso (UCM – Use Case Maps) [33].

GRL permite capturar las metas del negocio o del sistema, las diferentes alternativas para alcanzar estas metas y la fundamentación (rationale) que sustenta la selección de metas y alternativas [33].

Los UCM son una notación visual basada en escenarios (representados por casos de uso) para describir relaciones causales entre responsabilidades de uno o más casos de uso (una responsabilidad es alguna cosa que se debe ejecutar: operación, acción, tarea, función, etc.) [33].

III. CARACTERÍSTICAS A EVALUAR EN LAS TÉCNICAS DE GORE

Para evaluar las técnicas de GORE se van a aplicar los principios del Análisis de Características propuesto por Kitchenham [21], el cual establece la definición de un conjunto de características y/o propiedades que deben evaluarse para determinar si una técnica, método, herramienta o producto de la IS se ajusta a un conjunto específico de necesidades. El método específico de evaluación utilizado es denominado Exploración Cualitativa o “Qualitative Screening” [21]. Según Kitchenham, las características a ser evaluadas son compuestas porque se va a determinar el grado en que cada técnica las soporta mediante una escala ordinal. La citada autora indica que “cada característica compuesta debe estar acompañada de su importancia y una apropiada escala de valoración asociada con el grado de conformidad referente a una característica en particular” [21].

Kitchenham indica que la importancia de una característica se evalúa señalando si es obligatoria o solamente deseable. En este trabajo se definió la siguiente escala para medir la importancia de una característica: 2 – Obligatoria y 1 – Deseable. Con respecto al grado de conformidad, para cada característica se definió una escala específica de valoración.

A continuación se especifican las características a evaluar y la escala establecida para determinar el grado de conformidad.

A. Taxonomía de Metas

Se refiere al esquema utilizado para clasificar las metas y describir su comportamiento ([8], [17]), brinda soporte al proceso de refinamiento de metas. Para la medición del grado de conformidad de esta característica se definió la siguiente escala:

- 0: No soporta la característica.
- 1: Propone un esquema de taxonomía predefinido.
- 2: Además de proponer un esquema de taxonomía permite extender la taxonomía inicial y adaptarla a la situación en análisis.

B. Modelado de Metas Funcionales y Metas No Funcionales

Indica si el modelado permite representar y analizar metas funcionales y no funcionales. Las metas funcionales se refieren al comportamiento de la solución, y las no funcionales a las condiciones o cualidades necesarias para que la solución sea efectiva. La escala de conformidad establecida para esta característica es descrita a continuación:

- 0: No permite especificar si una meta es funcional o no funcional.
- 1: Reconoce esta clasificación, sin embargo, sólo permite el análisis y modelado de una sola clase de metas.
- 2: Permite el análisis y modelado de metas funcionales y metas no funcionales.

C. Recursos/Mecanismos para el Refinamiento de Metas

El refinamiento de metas permite obtener metas más concretas. Conduce al conjunto de requisitos que se deben satisfacer. También proporciona elementos para la trazabilidad, que faciliten la vinculación entre decisiones técnicas u operativas y los objetivos organizacionales. La escala de conformidad utilizada para evaluar esta característica es la siguiente:

- 0: No permite tratar con la diversidad, en naturaleza y alcance, de las diferentes metas que puedan estar relacionadas con la situación en análisis.
- 1: Proporciona recursos/mecanismos básicos para el refinamiento de metas. Es necesario recurrir a la utilización de otras técnicas de IR para: (1) complementar la descomposición de metas, y (2) vincular las decisiones de diseño y/o implementación y cada función del software con las metas establecidas.
- 2: Proporciona recursos/mecanismos robustos para el refinamiento de metas. Sin embargo, es necesario recurrir a la utilización de otras técnicas de IR para vincular las decisiones de diseño y/o implementación y cada función del software con las metas establecidas.
- 3: Proporciona recursos/mecanismos robustos para el refinamiento de metas. La técnica incluye la utilización de otras técnicas de IR para vincular las decisiones de diseño y/o implementación y las funciones de una solución con las metas establecidas.
- 4: Proporciona recursos/mecanismos robustos para el refinamiento de metas. El modelado con la técnica permite vincular las decisiones de diseño y/o implementación y las funciones de del software con las metas establecidas.

D. Capacidad de Representación del Refinamiento de Metas en Diferentes Niveles de Abstracción

En el análisis de metas pueden identificarse metas que se encuentren en diferentes niveles de abstracción, no sólo relacionadas con el alcance del sistema/software, sino también asociadas a los niveles de dirección y gestión de la organización. El manejo del nivel de abstracción facilita la trazabilidad y el análisis de contribución entre metas.

Con esta característica se pretende determinar la adecuación de una técnica para el modelado de metas en diferentes niveles de abstracción. En la evaluación se van a utilizar los niveles de abstracción definidos en [22], éstos son: Corporativo, Negocio, Operativo, Sistema y Software. Para una medición más precisa de esta característica se plantea dividirla en sub-características, cada una asociada a un nivel de abstracción.

La escala de conformidad utilizada para evaluar esta característica es la siguiente:

- 0: No soporta la representación de metas en el nivel de abstracción.
- 1: Soporta parcialmente la representación de metas en el nivel de abstracción.
- 2: Soporta la representación de metas en el nivel de abstracción pero no permite indicar explícitamente el nivel de abstracción de una meta.
- 3: Soporta la representación de metas en el nivel de abstracción permitiendo indicar explícitamente el nivel de abstracción de una meta.

Tomando en consideración que esta característica se dividió en sub-características, a cada una de ellas se le asignó un grado de importancia:

- | | |
|----------------------|------------------|
| • Nivel Corporativo: | 1 – Deseable. |
| • Nivel Negocio: | 1 – Deseable. |
| • Nivel Operativo: | 1 – Deseable. |
| • Nivel Sistema: | 2 – Obligatoria. |
| • Nivel Software: | 2 – Obligatoria. |

E. Mecanismos para el Modelado y Análisis de Obstáculos

El análisis de obstáculos contribuye a evaluar los factores de riesgo que pueden influir en que una meta no sea lograda/satisficha total o parcialmente. Para esta característica se estableció la siguiente escala de medición:

- 0: No soporta la característica.
- 1: Proporciona un soporte parcial al análisis de obstáculos.
- 2: Proporciona lineamientos y constructos para abordar el análisis de obstáculos.

F. Mecanismos para el Modelado y Análisis de Conflictos entre Metas

Se refiere a los elementos que permiten representar las interacciones negativas entre metas. La detección de conflictos implica que la satisfacción o logro de algunas metas afecta la satisfacción o logro de otras, lo cual conduce a establecer negociaciones entre las partes interesadas involucradas. Del análisis de conflictos se derivan cambios en el conjunto de metas que debe satisfacer la solución de software. La evaluación de conformidad de esta característica se efectúa aplicando la siguiente escala:

- 0: No soporta la característica.

- 1: Proporciona soporte parcial para el modelado y análisis de conflictos entre metas.
- 2: Proporciona soporte total para el modelado y análisis de conflictos entre metas, permitiendo representar alternativas para mitigar las interacciones negativas.

G. Mecanismos para el Modelado y Análisis de Interrelaciones entre Metas

Entre las metas pueden existir otras relaciones aparte de las derivadas por medio de la descomposición de metas o del análisis de conflictos. Tales relaciones permiten mostrar como una meta puede ser influenciada por otra en forma positiva o negativa, directa o indirectamente. La evaluación de conformidad de esta característica utiliza esta escala:

- 0: No soporta la característica.
- 1: Proporciona soporte parcial para el modelado y análisis de interrelaciones entre metas.
- 2: Proporciona soporte total para el modelado y análisis de interrelaciones entre metas, permitiendo representar la influencia de una meta en otra y definir el tipo de influencia.

H. Mecanismos para el Modelado y Análisis de Incumbencias Transversales

Una incumbencia transversal (IT) o “crosscutting concern” es un asunto/interés que en su implementación no puede ser encapsulado en un solo elemento y se distribuye en los elementos de otros asuntos/intereses (concerns) [35].

Esta característica se refiere a la forma en que la técnica, en forma directa o mediante extensiones, aborda la transversalidad de determinados comportamientos. El análisis de las IT favorece el reconocimiento de elementos (generalmente, propiedades de calidad) que influencian paralelamente distintas funciones; ésto puede incidir en la configuración de los componentes relacionados con la solución que se diseñe. Los mecanismos para el modelado de las IT ayudan al manejo de la complejidad en los grafos de metas. Para la medición de conformidad en esta característica se aplica la escala descrita a continuación:

- 0: No soporta la característica.
- 1: Existen extensiones que contribuyen parcialmente al modelado y análisis de las IT.
- 2: Existen extensiones que soportan ampliamente el modelado y análisis de las IT.
- 3: La técnica incluye el modelado y análisis de las IT.

I. Integración de los Modelos de Metas con Otras Técnicas de IR que Complementen la Especificación de los Requisitos

La técnica debe proporcionar lineamientos y/o mecanismos para integrar los modelos de metas con otros modelos derivados de técnicas complementarias. En la valoración de conformidad de esta característica se utiliza la siguiente escala:

- 0: No se propone la integración con otras técnicas.
- 1: Se definen lineamientos para integrar el modelado de metas con otras técnicas de modelado de IR complementarias.

- 2: Se definen lineamientos y mecanismos precisos para integrar el modelado de metas con otras técnicas de modelado de IR complementarias, haciendo énfasis en la trazabilidad entre modelos y definiendo explícitamente la contribución de cada enfoque de modelado utilizado.

J. Software para el Modelado

Indica si existe algún software que contribuya al modelado basado en la técnica y si el producto se encuentra en uso actualmente. La escala definida para medir el grado de conformidad es la siguiente:

- 0: No se ha desarrollado software para soportar el modelado con la técnica.
- 1: Existe un producto de software desarrollado pero no está en vigencia.
- 2: Existe un producto de software actualizado que soporta parcialmente el modelado con la técnica.
- 3: Existe un producto de software actualizado que soporta totalmente el modelado con la técnica.

K. Representación de Actores Organizacionales

Hace referencia al uso de constructos que representen y/o identifiquen no sólo a las metas sino a quienes (personas o unidades organizativas) las originan o están involucrados con su cumplimiento. El propósito es modelar las interacciones entre actores organizacionales y así reflejar la dinámica organizacional. La medición de conformidad aplica esta escala:

- 0: No soporta la característica.
- 1: Soporte parcial de la característica.
- 2: Permite representar los actores organizacionales vinculados a las metas establecidas.

L. Capacidad de Extensibilidad para Incorporar Conceptos Pertenecientes a la Planificación Estratégica en el Modelado de Metas

Como parte del requisito de integrar el análisis de metas y el análisis de la estrategia, se necesita que la técnica permita el modelado de algunos conceptos relativos a la PE, de ahí que se evalúe su capacidad para incorporar nuevos constructos y/o adaptar los existentes a nuevos requisitos en el modelado. Para esta característica la escala de conformidad establecida es la siguiente:

- 0: No soporta la característica.
- 1: La técnica ofrece limitadas opciones para incorporar nuevos constructos y/o adaptar los constructos existentes para el modelado de conceptos relativos a la PE.
- 2: La técnica ofrece opciones para incorporar nuevos constructos y/o adaptar los constructos existentes para el modelado de conceptos relativos a la PE, sin embargo, presenta algunas restricciones.
- 3: La técnica ofrece amplias opciones para incorporar nuevos constructos y/o adaptar los constructos existentes para el modelado de conceptos relativos a la PE.

M. Nivel de Estandarización

Se relaciona con los estudios y/o trabajos llevados a cabo para unificar y estandarizar conceptos y procedimientos referentes al modelado con la técnica. Para medir el grado de conformidad de la característica se definió la escala especificada a continuación:

- 0: No se ha definido un estándar basado en la técnica.
- 1: Existe un estándar basado en la técnica.

N. Proceso para Guiar el Modelado

Hace referencia a la existencia de un proceso que guíe la ejecución sistemática del análisis y modelado de metas. La valoración de esta característica se realiza por medio de la siguiente escala:

- 0: Sólo se dispone de algunos recursos metodológicos para el análisis y el modelado, tales como heurísticas, lineamientos, listas de chequeo, listas de recomendaciones, entre otros.
- 1: Posteriormente a la concepción de la técnica se han desarrollado diversos procesos complementarios para orientar el análisis y modelado.
- 2: La técnica dispone de un proceso establecido para guiar el análisis y modelado por sus autores y/o grupos interesados en compartir experiencias y unificar criterios para mejorar la práctica.

IV. EVALUACIÓN DE LAS TÉCNICAS DE GORE

El procedimiento ejecutado para la evaluación de una técnica es el siguiente:

- A cada característica se le asigna un valor que mide el grado en que se cumple lo establecido en su enunciado.
- El valor asignado a una característica define el grado de conformidad de la técnica y se multiplica por la magnitud que indica la importancia de la característica, la cual se determina según la siguiente escala: 2, si es “Obligatoria”, y 1, si es “Deseable”.
- El producto del grado de conformidad obtenido en una característica por la importancia de la misma, se totaliza generando una puntuación final.
- Se calcula el porcentaje de conformidad de la técnica considerando que la puntuación máxima que puede obtenerse es de 75. Esta magnitud resulta de calcular la puntuación final utilizando el valor máximo de la escala de medición de cada característica por la importancia definida para la misma.

La Tabla I muestra los resultados de la evaluación de las técnicas consideradas. Esta tabla incluye una columna que contiene la importancia de las características. En la columna “Max Conf” se muestra el valor máximo que se puede alcanzar en el grado de conformidad de una técnica para una característica determinada.

A continuación se presentan algunas observaciones derivadas de la evaluación de cada técnica.

A. KAOS

La mayor fortaleza de esta técnica es el soporte brindado al refinamiento de metas, basado en: el uso patrones, el uso de la taxonomía de metas, los medios para la trazabilidad entre modelos complementarios y los recursos metodológicos (heurísticas, prácticas, reglas, entre otros) [8]. En KAOS se construyen varios modelos: metas, riesgos u obstáculos, objetos, agentes, operación y comportamiento [8]. Clasifica las metas utilizando dos dimensiones: tipo y categoría. Según el tipo, una meta es de comportamiento (prescribe comportamientos intencionales) o softgoal (hace referencia a preferencias entre comportamientos alternativos). Según la categoría, una meta se clasifica como funcional (establece el propósito que soporta un servicio del sistema) o no funcional (cuando establece una calidad o restricción sobre la provisión o desarrollo de un servicio) [8]. KAOS no incluye un tratamiento directo de las incumbencias transversales. La extensión AspectKAOS presentada en [36] está dirigida sólo a los modelos de metas.

Esta técnica cuenta con varios productos de software para apoyar el modelado, entre los que destacan: (1) Objectiver [37], software bajo licencia que genera los modelos de metas, objetos, agentes y operación; (2) Dia Diagram Editor [38], software de código abierto; y (3) RE-Tools [39], también de código abierto, el cual permite representar el modelo de metas y el modelo de obstáculos aunque introduce algunos cambios en la simbología original planteada en la técnica.

B. NFR Framework

El NFR Framework utiliza el término meta para la representación de NFR, decisiones de diseño y argumentos aplicados en pro/contra de otras metas [14]. Las metas son modeladas como softgoals, metas cuya satisfacción no puede establecerse en forma taxativa o concreta. El análisis lleva a la construcción del Grafo de Interdependencia de Softgoals (SIG, Softgoal Interdependency Graph) ([14], [30]). Esta técnica considera tres tipos de metas: metas NFR, metas de operacionalización y metas de argumentación. El NFR Framework apoya el refinamiento de metas por medio del uso de: (1) catálogos de métodos de descomposición, (2) catálogos de métodos de operacionalización y (3) procedimiento de aplicación. El refinamiento está dirigido, únicamente, a los NFR, lo que implica que las metas asociadas a la funcionalidad son excluidas del análisis [30]. El modelado es soportado por la herramienta de código abierto RE-Tools [39].

En lo referente al tratamiento de las incumbencias transversales, el NFR Framework no aborda el asunto explícitamente, sin embargo, existen diversas propuestas dirigidas al manejo de este tipo de incumbencias como las presentadas en [40] y [41].

C. I (I-Star)*

I* maneja dos términos para referirse a una meta: meta (goal/hardgoal) y softgoal, dependiendo de si se hace referencia a una condición que se desea alcanzar y cuyo logro se pueda verificar en forma taxativa o no. En el primer caso, se utiliza el término meta y en el segundo, softgoal [15]. Las

metas no funcionales se representan como softgoals y las metas funcionales como metas (goals/hardgoals).

La técnica dispone de constructos para modelar los actores organizacionales, sus interrelaciones y el razonamiento interno de un actor referente a las alternativas para el logro/satisfacción de metas ([15], [31]).

En I* se construyen dos modelos [15]: (1) el Modelo de Dependencia Estratégica que muestra las relaciones externas entre actores, representando lo que los actores quieren unos de otros; y (2) el Modelo de Razonamiento Estratégico, el cual representa las relaciones internas dentro de un actor con el propósito de que se pueda soportar el razonamiento de los actores respecto a sus relaciones externas. El refinamiento en I* se apoya en: (1) desarrollo de rutinas (cursos de acción para lograr una intención de un actor); (2) uso de diferentes tipos de enlaces (medios-fin, descomposición y contribución); y (3) descomposición de actores (basada en el manejo de especializaciones del concepto Actor) ([15], [31]).

Para el tratamiento de las incumbencias transversales existen varias propuestas que extienden el alcance de la técnica, entre ellas destacan los trabajos presentados en [42], [43] y [44]. El sitio web istarwiki.org compila un conjunto de lineamientos y recomendaciones a ser considerados durante el modelado con I*, también presenta una lista de productos de software para apoyar el modelado [45].

D. GBRAM

GBRAM no plantea la construcción de un modelo gráfico, por lo tanto, no indica el uso de una notación particular para representar las metas y sus relaciones. Incluye el análisis de escenarios para obtener más detalles sobre los requisitos y validar el conjunto de metas identificado [16]. Propone la elaboración de una Topología de Metas (TM) para la representación de las características del Documento de Requisitos del Software (SRD, Software Requirements Document) expresada en la forma de una jerarquía [17].

En GBRAM la información se registra, principalmente, de manera textual en tablas y plantillas [17]. Sugiere la categorización de las metas considerando el tipo de la condición objetivo deseada, según lo cual las metas son: de logro (relativas a acciones que ocurren en el sistema) o de mantenimiento (asociadas a un estado continuo en el sistema). Recomienda el desarrollo de taxonomías auxiliares para las metas dependiendo del área de dominio del sistema ([16], [17]). La técnica no dispone de constructos para indicar cuando una meta es funcional o no funcional.

El refinamiento de metas se basa en: (1) soporte metodológico; (2) análisis de dependencias (precedencia, contrato y agente); (3) construcción de topología de metas; (4) análisis de escenarios, restricciones y obstáculos; y (5) esquemas de metas (detallan las operacionalizaciones) ([16], [17]). Aunque al inicio de su desarrollo el modelado se apoyaba en GBRAT (Goal Based Requirements Analysis Tool), herramienta orientada a la web, actualmente no se encuentra en vigencia [46].

E. Tropos

Tropos se basa en el modelado con I*, por lo tanto, utiliza los términos: meta (goal/hardgoal) y softgoal, definidos en esa técnica. Es un método para el desarrollo de software orientado a agentes, consta de cinco fases (descritas en el aparte E de la Sección II) [18]. No plantea el uso de modelos determinados, más bien define varias actividades de modelado desarrolladas en cada fase. Estas actividades utilizan artefactos específicos cuyo contenido depende de la fase en que se aplican [32]. Ésto significa que un mismo tipo de artefacto sirve para presentar varias vistas del sistema según la fase en ejecución.

Las actividades de modelado en Tropos son [32]: (1) modelado de actor, que consiste en identificar y analizar los actores del ambiente, y los actores y agentes del sistema; (2) modelado de dependencia, que comprende el estudio de dependencias entre actores; (3) modelado de meta, enfocado en el análisis de las metas de un actor; (4) modelado de plan, el cual complementa el modelado de meta, comprende la descomposición de planes en sub-planos; y (5) modelado de capacidad cuyo propósito es definir las capacidades de los componentes del sistema. El modelado de actor y el modelado de dependencia se representan mediante el diagrama de actor, equivalente al Modelo de Dependencia Estratégica de I*. El modelado de meta y el modelado de plan se muestran en el diagrama de meta, que se corresponde con el Modelo de Razonamiento Estratégico de I*. El modelado de capacidad se representa mediante los diagramas de capacidad y diagramas de plan. Estos diagramas se construyen mediante el uso de diagramas de actividad UML y diagramas de interacción AUML [47].

Esta técnica incluye mecanismos para derivar de las metas, los requisitos y las especificaciones de diseño e implementación a largo del ciclo de desarrollo. Además, sugiere cuáles técnicas adicionales de IR se podrían aplicar para apoyar la especificación de requisitos. Existen varias herramientas para soportar el modelado con Tropos, en el sitio web www.troposproject.org se presenta una lista de ellas. Las herramientas señaladas sólo permiten construir los diagramas de actor y de meta [48].

Con respecto al manejo de las incumbencias transversales, las propuestas desarrolladas para I* son extensibles a Tropos ([42], [43], [44]). Estos trabajos son aplicables en la construcción de los diagramas de actor y de meta. Sin embargo, no se ha planteado ninguna extensión a Tropos enfocada en integrar el manejo de las incumbencias transversales en todas las actividades de modelado que comprende esta técnica.

F. GRL

GRL como técnica derivada de I*, también representa las metas utilizando los términos: meta (goal/hardgoal) y softgoal [33]. GRL no prescribe modelos específicos; sin embargo, permite construir los modelos propuestos en las técnicas que le dieron origen, I* y NFR Framework: Modelo de Dependencia Estratégica (I*), Modelo de Razonamiento Estratégico (I*) y Grafo de Interdependencia de Softgoals (NFR Framework). Esta técnica ofrece mayor libertad de aplicación en lo que respecta a los constructos de modelado, por ello, permite

generar nuevos modelos que se ajusten a una situación particular, en [49] se ilustra este caso.

El modelado con GRL es soportado por la herramienta jUCMNav la cual a ajusta en un alto grado a las especificaciones de este lenguaje gráfico ([50], [51]). GRL tiene una gran capacidad de extensibilidad [33]. Además forma parte de URN [33] que es un estándar internacional. Con respecto al manejo de las incumbencias transversales, Mussbacher [50] presenta a AoGRL (Aspect-oriented GRL) una extensión que incorpora en el modelado GRL los conceptos básicos de la orientación a aspectos; también son aplicables a GRL las extensiones desarrolladas para I*, propuestas en [42], [43] y [44].

G. URN

URN combina conceptos y notaciones de modelado para metas e intenciones y escenarios; sin embargo, no especifica qué modelos deben desarrollarse [33]. URN comprende: (1) modelado de metas, el cual se basa en GRL (aparte F de esta Sección); y (2) modelado de escenarios, que utiliza los UCM para describir escenarios y arquitecturas. Los UCM se enfocan en flujos causales de comportamiento opcionalmente superpuestos sobre una estructura de componentes. Los UCM permiten describir la interacción causal de entidades

arquitecturales abstrayéndose de los detalles referentes a mensajes y datos.

URN soporta el refinamiento de metas por medio de [33]: (1) enlaces de descomposición y de contribución; (2) elaboración de escenarios; y (3) enlaces URN (URN Links) que ayudan a crear relaciones entre cualquier par de elementos de modelos URN.

En [50] se presenta AoURN (Aspect-oriented URN) una extensión para el manejo de incumbencias transversales en URN. AoURN comprende extensiones para GRL y UCM, AoGRL (Aspect-oriented GRL) y AoUCM (Aspect-oriented UCM), respectivamente.

La Universidad de Ottawa (Canadá) desarrolló jUCMNav, herramienta de código abierto para editar y analizar modelos URN: grafos GRL, mapas de casos de uso y cualquier otro modelo que se defina en base a los constructos de URN y sea compatible con los criterios adoptados en el desarrollo de la herramienta [51].

La capacidad de extensibilidad de URN es un atributo que otorga gran flexibilidad para el modelado en diferentes contextos. URN está definido como un estándar en el campo de la IS en la Recomendación ITU-T Z.151 [33].

TABLA I. EVALUACIÓN DE TÉCNICAS DE GORE – CUADRO COMPARATIVO

Evaluación de Técnicas de GORE – Cuadro Comparativo										
Característica		Importancia	Max Conf	Grado de Conformidad						
				KAOS	NFR Framework	I*	GBRAM	Tropos	GRL	URN
1	Taxonomía de metas	2	2	1	1	1	2	1	1	1
2	Modelado de metas funcionales y metas no funcionales	2	2	2	1	2	0	2	2	2
3	Recursos/Mecanismos para el refinamiento de metas	2	4	4	0	2	2	3	2	4
4	Capacidad de representación del refinamiento de metas en diferentes niveles de abstracción									
a.	Nivel Corporativo	1	3	0	0	2	1	2	2	2
b.	Nivel Negocio	1	3	0	0	2	1	2	2	2
c.	Nivel Operativo	1	3	2	1	2	2	2	2	2
d.	Nivel Sistema	2	3	2	1	2	2	2	2	2
e.	Nivel Software	2	3	2	1	1	2	2	1	2
5	Mecanismos para el modelado y análisis de obstáculos de metas	2	2	2	1	1	2	1	1	1
6	Mecanismos para el modelado y análisis de conflictos entre metas	2	2	1	2	2	2	2	2	2
7	Mecanismos para el modelado y análisis de interrelaciones entre metas	2	2	1	2	2	1	2	2	2
8	Mecanismos para el modelado y análisis de las incumbencias transversales	2	2	1	1	2	0	1	2	2
9	Integración de los modelos de metas con otras técnicas de la IR que complementen la especificación de los requisitos	2	2	2	0	0	2	1	0	3

Evaluación de Técnicas de GORE – Cuadro Comparativo										
	Característica	Importancia	Max Conf	Grado de Conformidad						
				KAOS	NFR Framework	I*	GBRAM	Tropos	GRL	URN
10	Software para el modelado	1	3	2	3	3	1	2	3	2
11	Representación de actores organizacionales	2	2	1	0	2	0	2	2	2
12	Capacidad de extensibilidad para incorporar conceptos pertenecientes a la Planificación Estratégica (PE) en el modelado de metas	2	3	1	0	3	0	3	3	3
13	Nivel de estandarización	1	1	0	0	0	0	0	1	1
14	Proceso para guiar el modelado	2	2	2	1	1	2	2	1	1
Puntuación final = Σ Grado-de-Conformidad _i * Importancia _i				48	26	51	39	56	52	63
% de conformidad				64,00	34,67	68,00	52,00	74,67	69,33	84,00

V. ANÁLISIS DE LOS RESULTADOS DE LA EVALUACIÓN DE LA TÉCNICAS DE GORE

Al analizar los resultados se puede advertir que KAOS, Tropos y URN proporcionan mayor apoyo en el refinamiento de metas (característica No. 3); estas técnicas complementan el análisis de metas con la aplicación de otros enfoques de IR, lo cual permite obtener mayor detalle de las necesidades de la organización y, de esta forma, validar las metas establecidas, asegurar la trazabilidad entre metas y decisiones de diseño y/o implementación, e identificar nuevas metas.

La técnica que obtuvo menor puntuación fue el NFR Framework, ésto se debe a que sólo direcciona el análisis de los requisitos no funcionales; sin embargo, esta técnica ha influenciado notablemente a las otras. Cabe señalar que KAOS e I* en sus versiones actuales han incorporado las estrategias de evaluación y procedimiento de etiquetado, definidos en el NFR Framework, para apoyar el análisis de contribución entre metas. También I*, Tropos, GRL y URN incluyen en su análisis de metas el método de razonamiento y análisis de contribución entre metas propuesto en el NFR Framework.

En cuanto al manejo de los niveles de abstracción de las metas (característica No. 4) I*, Tropos, GRL y URN, proporcionan mejor soporte para modelar metas de alto nivel de abstracción como las correspondientes a los niveles Corporativo y Negocio; ésto se debe a que sus conceptos básicos permiten representar elementos relacionados con la dinámica organizacional y no sólo los elementos asociados a los sistemas a desarrollar.

I*, Tropos, GRL y URN son técnicas que facilitan el modelado de la dinámica organizacional debido a que están dirigidas hacia el análisis de las intenciones y motivaciones de diferentes actores organizacionales y sus interrelaciones; también contribuyen a examinar el comportamiento de estos actores y razonar sobre la forma de alcanzar sus metas, lo cual proporciona los fundamentos para las soluciones que se plantean ante situaciones específicas.

El enfoque de I*, Tropos, GRL y URN es la clave en la consideración de que estas técnicas son las que ofrecen mayor viabilidad para integrar el modelado de metas y la PE y, de esta manera, cumplir los requisitos definidos al inicio de esta evaluación.

No obstante, URN es la técnica que obtuvo mejor puntuación en la evaluación de conformidad de las características consideradas relevantes para este estudio. Tomando en cuenta su porcentaje de conformidad (84,00 %), se concluye que es la técnica que mejor se ajusta a los requisitos establecidos.

A continuación se resumen las razones que justifican la selección de URN:

- Enfoque hacia el modelado organizacional (basado en I*) ([33], [34]).
- Soporte en un estándar internacional [33].
- Manejo explícito de los requisitos funcionales y de los requisitos no funcionales.
- Integración entre el modelado de metas y el de escenarios. La complementariedad entre el análisis de metas y el análisis de escenarios, proporciona mecanismos de refinamiento robustos y contribuye a la trazabilidad entre metas organizacionales y decisiones técnicas de diseño y/o implementación referidas a una solución determinada ([29], [34]).
- Capacidad de extensibilidad por medio de diferentes recursos: metadatos; enlaces URN (URN Links) de utilidad para apoyar: el refinamiento de metas y escenarios, la trazabilidad de requisitos, la composición de requisitos, entre otros propósitos; mecanismos para el desarrollo de perfiles; uso de restricciones OCL ([29], [33]).
- Disponibilidad de herramienta de software actualizada para soportar el modelado [51].

- Orientación al modelado basado en incumbencias (concerns). El metamodelo de URN incluye el concepto de incumbencia, el cual permite agrupar elementos de los modelos URN. El manejo de incumbencias mejora la modularización de los modelos de metas [33].

VI. CONCLUSIONES

La naturaleza del análisis de metas constituye un medio apropiado para enfocar los sistemas de software como instrumentos efectivos que soportan la proposición de valor del negocio y el éxito de las estrategias. Sin embargo, es notorio que el modelado de metas no es una tarea sencilla ([8], [17]) y se incrementa su complejidad con la introducción de nuevos elementos, como en el caso de los conceptos básicos de la PE. Por ello, se incluyeron en la evaluación características relacionadas con el soporte ofrecido para la organización y construcción de los modelos, tales como: (1) el manejo de las incumbencias transversales, (2) el nivel de estandarización de la práctica, y (3) la disponibilidad de un proceso para guiar el modelado.

La evaluación de las técnicas estuvo dirigida al estudio de los recursos y medios disponibles, pero también se enfocó en la capacidad de extensión, considerando la ampliación del alcance del modelado para incluir los conceptos básicos de la PE, dado que es el dominio al que pertenecen los asuntos relacionados con las estrategias organizacionales.

En los resultados de la evaluación se observó que:

- Las técnicas que utilizan constructos para la representación de los actores organizacionales proporcionan mejor soporte al modelado de la dinámica del negocio.
- Existen puntos de proximidad entre el análisis de metas y el análisis de las estrategias, de ahí que, en algunas técnicas, es factible la adición de nuevos constructos, derivados de los conceptos básicos de la PE, cuyo comportamiento es compatible con los fundamentos de estas técnicas.
- Las técnicas que se complementan con otros enfoques de modelado facilitan la trazabilidad entre la especificación operacional de los requisitos de una solución de software y las metas de alto nivel de la organización.
- El manejo de los niveles de abstracción favorece el análisis de metas apoyando el refinamiento y el estudio de la contribución entre metas. Lo anterior ayuda en el análisis de la alineación estratégica porque permite visualizar cómo las metas de un nivel apoyan a las de un nivel superior y cómo las metas de un nivel superior se descomponen en conjuntos de metas de mayor precisión.
- Los enfoques de las técnicas son producto de la evolución en el área, por ello se percibe, sobre todo en las versiones más actuales, la influencia recíproca entre ellas.

En la Tabla I se observa que el conjunto de técnicas conformado por I* y sus derivadas (Tropos y GRL) son las que mejor se ajustan a los requisitos establecidos.

Sin embargo, la mayor puntuación la obtuvo URN, conformada por GRL y UCM, la cual es un estándar internacional que integra el modelado de metas y escenarios. Su extensibilidad es un elemento clave debido a que permite expandir su aplicación hacia otras áreas donde se necesite ejecutar el análisis del negocio, estén relacionadas o no con proyectos de Ingeniería de Software.

Con base a los resultados de la evaluación se está trabajando en una extensión de URN para integrar en el modelado de metas los conceptos básicos de la PE. Esta extensión se utilizará para soportar un proceso de IR dirigido por la PE que contribuya a direccionar los productos de software hacia la creación de valor y asegure su alineación estratégica.

En lo que respecta a futuras investigaciones es importante abordar los siguientes temas: (1) aplicación del modelado con URN como herramienta para enfocar la alineación estratégica de productos de software; (2) validación de la efectividad de URN en el modelado de estrategias; y (3) evaluación de URN en otros dominios del análisis del negocio.

REFERENCIAS

- [1] S. Pass and B. Ronen, "Reducing the Software Value Gap", Communications of the ACM, vol. 57, no. 5, pp. 80-87, May 2014.
- [2] A. Blair, J. B. Lail, and S. Marshall, "Value Streams", Open Group Guide, Document Number G170, January, 2017.
- [3] P. Zave, "Classification of Research Efforts in Requirements Engineering", ACM Computing Surveys (CSUR), vol. 29, no. 4, pp. 315-321, December 1997.
- [4] G. Brown, "Value Chains, Value Streams, Value Nets, and Value Delivery Chains", BPTrends, April, 2009.
- [5] R. Kaplan and D. Norton, "Strategy Maps", Executive Books Summaries, vol. 26, no. 4 (2 parts) part 1, April, 2004.
- [6] H. Armitage and C. Scholey, "Using Strategy Maps to Drive Performance", Management Accounting Guideline published by The Society of Management Accountants of Canada (CMA-Canada), 2006.
- [7] A. Lapouchian, "Goal-oriented Requirements Engineering: An Overview of the Current Research", University of Toronto, June, 2005.
- [8] A. van Lamsweerde, "Requirements Engineering: From System Goals to UML Models to Software Specifications", England: John Wiley & Sons Inc., 2009.
- [9] K. Pohl, "Requirements Engineering: Fundamentals, Principles, and Techniques". Germany: Springer, 2010.
- [10] G. Johnson, K. Scholes, and R. Whittington, "Dirección Estratégica", 7th Edition, Pearson Educación, S. A., Spain, 2006.
- [11] E. Kavakli, "Goal-driven Requirements Engineering: Modelling and Guidance", Ph.D. thesis, University of Manchester, UK, 1999.
- [12] E. Kavakli and P. Loucopoulos, "Goal-driven Requirements Engineering: Evaluation of Current Methods", 8th International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, Austria, 16-17, June, 2003.
- [13] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed Requirements Acquisition", Selected Papers of the 6th International Workshop on Software Specification and Design, Elsevier Science Publishers, The Netherlands, pp. 3-50, 1993.
- [14] L. Chung, "Representation and Utilization of Non-functional Requirements for Information System Design", Proceedings of the 3rd Int. Conf. CAiSE '91, Trondheim, Norway, Berlin: Springer-Verlag, pp. 5-30, May, 1991.

- [15] E. Yu, "Modelling Strategic Relationships for Process Reengineering", Ph.D. thesis, University of Toronto, Canada, 1995.
- [16] A. Antón, "Goal-based Requirements Analysis", Proceedings of the 2nd International Conference on Requirements Engineering. IEEE Computer Society, Washington, DC, USA, pp. 136-144, 1996.
- [17] A. Antón, "Goal Identification and Refinement in the Specification of Software-based Information Systems", Ph.D. thesis, Georgia Institute of Technology, Georgia, USA, 1997.
- [18] J. Mylopoulos and J. Castro, "Tropos: A Framework for Requirements-driven Software Development", Information Systems Engineering: State of the Art and Research Themes, Lecture Notes in Computer Science, Springer-Verlag, June 2000.
- [19] L. Liu and E. Yu, "GRL - Goal-oriented Requirement Language", 2000. <http://www.cs.toronto.edu/km/GRL/>.
- [20] D. Amyot and G. Mussbacher, "URN: Towards a New Standard for the Visual Description of Requirements", 3rd SDL and MSC Workshop, U.K., LNCS 2599, pp. 21-37, June, 2002.
- [21] B. Kitchenham, "DESMET: A Method for Evaluating Software Engineering Methods and Tools", Technical Report TR96-09, Department of Computer Science, University of Keele, 1996.
- [22] M. Delgado, F. Losavio, and A. Matteo, "Goal-oriented Techniques and Methods: Goal Refinement and Levels of Abstraction", Proceedings of the 39th Latin American Computing Conference, Venezuela, October, 2013.
- [23] S. Bleistein, K. Cox, J. Verner, and K. Phalp, "B-SCP: A Requirements Analysis Framework for Validating Strategic Alignment of Organizational IT Based on Strategy, Context, and Process", Information and Software Technology, vol. 46, pp. 846-868, 2006.
- [24] A. Siena, A. Bonetti, and P. Giorgini, "Balanced Goalcards: Combining Goal Analysis and Balanced Scorecards", 3rd International Conference on Evaluation of Novel Approaches to Software Engineering, Portugal, May, 2008.
- [25] R. Kaplan and D. Norton, "Integrating Strategy Planning and Operational Execution", Balanced Scorecard Report, Harvard Business School Publishing, vol. 10, no. 3, May – June, 2008.
- [26] R. Ellis-Braithwaite, R. Lock, R. Dawson, and B. Haque, "Modelling the Strategic Alignment of Software Requirements using Goal Graphs", arXiv preprint arXiv:1211.6258, 2012.
- [27] G. Grau, C. Cares, X. Franch, and F. Navarrete, "A Comparative Analysis of I* Agent-Oriented Modelling Techniques", Proceedings of the 18th International Conference on Software Engineering and Knowledge Engineering, USA, 5-7 July, 2006.
- [28] M. Teruel, E. Navarro, V. López-Jaquero, F. Montero, and P. González, "Approaches to Modelling Requirements for Collaborative Systems", Technical Report No. DIAB-11-03-1. University of Castilla – La Mancha, March, 2011.
- [29] D. Amyot and G. Mussbacher, "User Requirements Notation: The First Ten Years, the Next Ten Years", Invited paper, Journal of Software (JSW), vol. 6, no. 5, pp. 747-768, Academy Publisher, May, 2011.
- [30] L. Chung, B. Nixon, and J. Mylopoulos, "Non-functional Requirements in Software Engineering", Kluwer Academic Publishers, 2000.
- [31] E. Yu and L. Liu, "Modelling Trust for System Design Using the I* Strategic Actors Framework", Proceedings of the Workshop on Deception, Fraud, and Trust in Agent Societies, Lecture Notes In Computer Science, vol. 2246, pp. 175-194, Springer-Verlag, London, 2001.
- [32] F. Giunchiglia, J. Mylopoulos, and A. Perini, "The Tropos Software Development Methodology", Agent-Oriented Software Engineering, pp. 162-173, Springer, Heidelberg, 2002.
- [33] International Telecommunication Union - Telecommunication Standardization Sector [ITU-T]. "User Requirements Notation (URN) – Language definition", Recommendation ITU-T Z.151, 2012.
- [34] D. Amyot, "Introduction to the User Requirements Notation", Computer Networks, vol. 42, no. 3, pp. 285-301, June, 2003.
- [35] K. van der Berg, J. Conejero, and R. Chitchyan, "AOSD Ontology 1.0 – Public Ontology of Aspect-orientation". AOSD-Europe-UT-01, no. D9, version 1.0, 2005.
- [36] A. Gil, "Integrating Early Aspects with Goal-Oriented Requirements Engineering: The Case of KAOS". MSc. thesis, Universidade Nova de Lisboa, Portugal, 2008.
- [37] Requirements Engineering and Specification Techniques for Information Technology, "A KAOS Tutorial", (n.d.). <http://www.objectiver.com>
- [38] The Dia Developers, (n.d.). <http://dia-installer.de/index.html.en>
- [39] RE-Tools, (n.d.). <http://www.utdallas.edu/~supakkul/tools/RE-Tools/>
- [40] G. Sousa, I. Silva, and J. Castro, "Adapting the NFR Framework to Aspect – Oriented Requirements Engineering", XVII Brazilian Symposium on Software Engineering, Brazil, October 2003.
- [41] I. Brito and A. Moreira, "Integrating the NFR Framework in a RE Model", Early-Aspects Workshop, 3rd International Conference on Aspect-Oriented Software Development, UK, March, 2004.
- [42] E. Spies, J. Rüger, and A. Moreira, "Using I* to Identify Candidate Aspects", Workshop in UML'04, Portugal, October, 2004.
- [43] F. Alencar, J. Castro, C. Monteiro, R. Ramos, and E. Santos, "Towards Aspectual I*", Proceedings of 3rd International I* Workshop, Brazil, February, 2008.
- [44] F. Alencar, J. Castro, M. Lucena, E. Santos, C. Silva, A. Aratijo, and A. Moreira, "Towards Modular I* Models", Proceedings of the 2010 ACM Symposium on Applied Computing. ACM, New York, NY, USA, 292-297, 2010.
- [45] IStarWiki. (n.d.). http://istarwiki.org/tiki-index.php?page=i*+Wiki+Home
- [46] A. Antón, E. Liang, and R. A. Rodenstein, "A Web-Based Requirements Analysis Tool", Proceedings of the 5th International Workshops on Enabling Technologies, IEEE Computer Society, USA, 238-243, 1996.
- [47] B. Bauer, J. P. Muller, and J. Odell, "Agent UML", Agent-Oriented Software Engineering, Paolo Ciancarini and Michael Wooldridge Eds., Springer, Berlin, pp. 91-103, 2001.
- [48] TroposProject. (n.d.). <http://www.troposproject.org/>
- [49] G. Mussbacher and D. Amyot, "On Modeling Interactions of Early Aspects with Goals", Proceedings of the 2009 ICSE Workshop on Aspect-Oriented Requirements Engineering and Architecture Design. IEEE Computer Society, USA, 14-19.
- [50] G. Mussbacher, "Aspect-oriented User Requirements Notation", Ph.D. thesis. University of Ottawa, Canada, November, 2010.
- [51] J. Roy, J. Kealey, and D. Amyot, "Towards Integrated Tool Support for the User Requirements Notation", R. Gotzhein, R. Reed (Eds.) SAM 2006: Language Profiles - Fifth Workshop on System Analysis and Modelling, Germany. LNCS 4320, 198-215, Springer. May, 2006.
- .

Juegos de Entrenamiento Mental bajo un Ambiente de Realidad Virtual Inmersiva

Daniel Sam* and Esmitt Ramírez†‡

*Escuela de Informática, Facultad de Ingeniería, Universidad Católica Andrés Bello, Venezuela 1020-A

†Centro de Computación Gráfica, Escuela de Computación, Universidad Central de Venezuela, Caracas 1040

‡Centre de Visió per Computador, Edifici O, Universitat Autònoma de Barcelona, Bellaterra, España 08193

sam.nyst@gmail.com, esmitt.ramirez@ciens.ucv.ve

Resumen—El mundo moderno está rodeado de información, siendo el cerebro el encargado de regular los procesos y funciones cognitivas en el medio externo para satisfacer las necesidades de un ser humano. A medida que se envejece, aumenta la probabilidad de disfunciones en facultades cognitivas, como memoria, atención, percepción, o aprendizaje de actividades nuevas. Por ello, es importante el ejercitarse el cerebro para entrenar las capacidades cognitivas (junto con una buena alimentación y control del estrés). Entre los ejercicios recomendados, se encuentran los basados en el entrenamiento cerebral empleando juegos para memorizar, visualizar, razonar, o aprender. En este artículo se presenta una solución computacional para el entrenamiento mental empleando juegos bajo un ambiente de realidad virtual inmersivo que permita involucrar completamente al individuo en los ejercicios, a través de interacción auditiva, motora y visual. Se incluyen tres juegos en una arquitectura basada en el uso de un casco virtual y un dispositivo de captura de los movimientos de la mano/dedos dentro de una escena con elementos 3D. Las pruebas realizadas en personas de edades entre 21-30 años, sirven de iniciación para incursionar en mayores estudios a mayor escala. Siendo la capacidad de impacto de nuestra solución medida a través del número de aciertos, fallos y tiempo de realización.

Palabras Clave—Entrenamiento Mental, Realidad Virtual, Capacidad Cognitiva, Juegos Serios, Escena 3D

I. INTRODUCCIÓN

El deterioro cognitivo es un problema o inconveniente que surge con el transcurso de los años a causa del envejecimiento natural de los seres humanos (inclusive desde los inicios de la adultez), abriendo la posibilidad de generar aspectos negativos. La estimulación cognitiva ayuda al cerebro a afrontar el deterioro y mejorar la capacidad de regenerar las neuronas que han sido afectadas.

Existen una gran cantidad de métodos de entrenamiento que permiten la oxigenación del cerebro. Desde sencillos juegos de memoria para desafiar a la mente en recordar ciertos patrones, hasta ilusiones ópticas o visuales que permiten a la mente procesar simplicidad, semejanza y equivalencia a una realidad objetiva dentro de una imagen. Además, que permite recrear un escenario (i.e. en espacio 2D o 3D) para resolver algún problema, lo cual se asemeja a cualquier problema cotidiano.

Actualmente empleando las nuevas tecnologías, es posible realizar métodos para el mejoramiento de las actividades cerebrales. Así, los *Brain Training* (Entrenamientos mentales) surgen como aplicaciones que permiten a los usuarios estimular o entrenar ciertas áreas cognitivas por medio de resolución de

problemas en un escenario en particular bajo el seguimiento de ciertas reglas. Así, existen varias plataformas que desarrollan estas aplicaciones por medio de un entorno web o móvil. Sin embargo, la mayoría de estas aplicaciones se ejecutan en un entorno en dos dimensiones, que limita la capacidad de orientación e interacción con el entorno y la percepción real de los objetos. Algunas de las limitaciones son impuestas por el dispositivo de hardware empleado, independientemente si es web o móvil.

Una de las plataformas que ha tomado auge en el mundo moderno son las basadas en la realidad virtual, debido a su fácil acceso y bajo coste. La realidad virtual es una tecnología que emplea un entorno generado por un computador para generar un mundo ficticio o virtual en tres dimensiones, con la idea de generar una inmersión total que permita una interacción real con el ambiente. Dicho ambiente puede representar un entorno real o totalmente ficticio, tal que un usuario pueda “vivir” dentro de este. De esta manera, si se cuenta con juegos que estimulen el cerebro a cerebro dentro de un ambiente de realidad virtual, se plantea resultados satisfactorios.

En este artículo se presenta una solución basada en juegos para el entrenamiento mental en un ambiente de realidad virtual, que permita una inmersión completa con el uso del hardware *Oculus Rift* como visor, y el dispositivo de control gestual *Leap Motion*. Dichos entrenamientos están orientados a la inteligencia espacial y permiten estimulaciones mentales específicas en base al objetivo de cada uno. Los usuarios pueden ver los resultados obtenidos en cada uno de los entrenamientos que hayan realizado, permitiendo generar un *feedback* en su desempeño. Al mismo tiempo, se plantea una interfaz gestual intuitiva basada en lenguaje natural.

Así, en la Sección II se describe los conceptos básicos asociados al entrenamiento mental. En la Sección III se muestran las principales características de la realidad virtual enfocadas en nuestra propuesta. La Sección IV explica en detalle la solución propuesta, y el diseño y consideraciones de cada uno de los juegos se muestra en la Sección V. La experimentación de nuestra propuesta se presenta en la Sección VI, y finalmente la Sección VII muestra las conclusiones y trabajos a futuro.

II. ENTRENAMIENTO MENTAL

El término neuróbica [1], [2] se emplea en el área de la Neurología para describir aquellos ejercicios mentales que

mantienen el cerebro alerta. Estos se plantean como una nueva forma de ejercicio cerebral, proyectado para mantener al cerebro ágil y saludable, creando nuevos y diferentes patrones de comportamiento, y de las actividades de las neuronas.

El entrenamiento mental se plantea como una serie de ejercicios diseñados y desarrollados específicamente para la plasticidad neuronal, que estimula diversas áreas cognitivas [3]. Esto influye en las capacidades que se presentan diariamente en las personas como lo son la percepción, el razonamiento, la memoria, la atención, entre otras. En la literatura, existe un debate para asegurar o refutar el efecto positivo de los juegos de entrenamiento mental [4], [5], [6]. Sin embargo, la activación de neuronas es clave [7], así como la excitación de diversas regiones del cerebro que contribuyen a la salud de una persona [8], [9]. Entonces, del mismo modo que se pueden entrenar las capacidades físicas también sucede con las áreas cognitivas, permitiendo una mayor estimulación cerebral y obteniendo resultados que pueden durar varios años si se realizan con una cierta frecuencia y bajo ciertos ambientes.

III. REALIDAD VIRTUAL

La realidad virtual es un término empleado para referirse a los ambientes 3D generados por computador que permiten a un usuario sumergirse en este, e interactuar con una realidad alternativa [10]. La realidad virtual se puede clasificar en inmersivo y no inmersivo: los inmersivos se asocian a un ambiente donde se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano; los no inmersivos emplean medios (e.g. Internet) donde se puede interactuar en tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales al computador.

La realidad virtual inmersiva se logra mediante una serie de impulsos que reciben las neuronas y transmiten al cerebro [11], debido al hardware que se requiere para estar totalmente sumergido en el ambiente tridimensional e intensificar la sensación de realidad.

Un usuario puede estar expuesto a una realidad virtual basada en sistemas *desktops* (aplicada en los videojuegos que permite la navegación en primera persona); a una realidad virtual en segunda persona (aplicada en simulaciones corporales); telepresencia (incluyendo la telerobótica y telemedicina); o sistemas de telepresencia que permiten manipular por medio de controles remotos, cámaras y elementos de retroalimentación, robots o dispositivos a largas distancias mientras se experimentan de forma virtual [12].

Hoy en día, al mencionar aplicaciones de realidad virtual, se hace referencia a aquella que se sustenta bajo la inmersión completa del usuario a un escenario computarizado por medio de dispositivos externos: Cascos o lentes asociados a cada ojo que permiten la salida de las imágenes, y guantes o controles manuales que permiten la interacción con el escenario. No obstante, existen también plataformas físicas el cual sujetan al usuario al nivel de la cintura, y estas permiten caminar sobre la superficie, compuesta por una cinta (similar a una caminadora

de ejercicios) y una serie de sensores indicando la dirección hacia donde se desplaza el usuario.

En la actualidad, la aplicación de la realidad virtual inmersiva ofrece ventajas para diversas áreas educativas, ya sea en la medicina, arquitectura, ingeniería, aviación, entre otras, donde se ofrece una práctica más completa de las habilidades fuera de un riesgo real y físico. Existen simuladores que permiten experimentar el comportamiento de un usuario ante algún evento en particular, por ejemplo, cómo reacciona la gente en una salida de emergencia ante un sismo, entre muchas otras posibilidades.

En la realidad virtual inmersiva, además de emplear dispositivos como cascos o lentes asociados a cada ojo que permiten la salida de las imágenes, o guantes y controles manuales que permiten la interacción con el escenario, existen también plataformas físicas que sujetan al usuario al nivel de la cintura, y que permiten caminar sobre su superficie (parecido a una caminadora de ejercicios con sensores de dirección). Particularmente en el área de Medicina, se ha empleado la realidad virtual inmersiva con el uso de cascos Oculus Rift [13], [14], [15], [16] con resultados exitosos. Basado en nuestras investigaciones, en Venezuela existen desarrollos basados en realidad virtual no inmersiva para el área de salud [17] o entretenimiento [18] empleando un Kinect. Sin embargo, estas no ofrecen una total integración del usuario debido a sus características propias de la no-inmersión.

El dispositivo Oculus Rift (desarrollado por la compañía Oculus VR) [19] consiste en un casco de realidad virtual (*HMD – Head Mounted Display*) que permite visualizar imágenes computarizadas, que son proyectadas sobre unas pequeñas pantallas (binocular). Existen dos pantallas, una para cada ojo, con el objetivo de englobar todo el campo de visión del usuario y sentirse inmerso completamente dentro de un ambiente virtual. Además, el Oculus Rift incluye auriculares para proveer sonido y crear una experiencia más envolvente.

Existen investigaciones de realidad virtual empleando el Oculus Rift para el entrenamiento cerebral, donde se tratan a pacientes con un cuadro clínico particular [20], [21] o se busca estudiar su impacto en la salud del paciente desde el punto de vista neurológico [22], [23]. Las aplicaciones de entrenamiento cerebral empleando ambientes inmersivos con el Oculus Rift son de amplio estudio en diversos centros de investigación. A continuación, planteamos nuestra propuesta basada en el uso de este HMD junto con un dispositivo de captura de los movimientos de la mano, para una interacción gestual.

IV. SOLUCIÓN PROPUESTA

La solución propuesta consta de una parte de hardware y software. Dado que se requiere que el usuario esté inmerso dentro de un ambiente virtual, debe existir una interacción de este con el ambiente. El ambiente virtual se basa en juegos virtuales mostrados en la Figura 1. La interacción la provee el dispositivo Leap Motion [24], que es un pequeño dispositivo que emplea cámaras infrarrojas monocromáticas e infrarrojas de LED, permitiendo capturar el movimiento de las manos y dedos a una distancia de 1 m. Es un dispositivo similar a

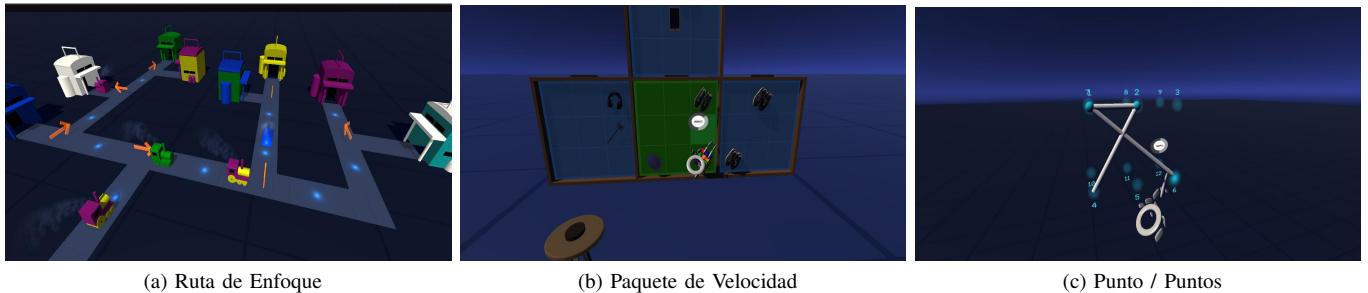


Figura 1. Muestra de un Instante en la Ejecución de los Juegos de Entrenamiento Planteados en Nuestra Propuesta

un ratón convencional de computador, pero sin contacto. Este dispositivo puede ser colocado en la parte frontal de un casco Oculus Rift [19], y así tener un ambiente de bajo costo para la creación de aplicaciones basadas en realidad virtual inmersiva.

En la Figura 2 se muestra la configuración tanto del casco Oculus Rift, como del periférico Leap Motion dentro de la solución propuesta. Nótese que el caso no restringe la movilidad libre de la cabeza del usuario, y el Leap Motion no restringe el movimiento total de los brazos, manos y dedos del usuario.



Figura 2. Configuración de Nuestra Propuesta, Empleando (a) un Casco Oculus Rift y el Leap Motion, (b) así como su Disposición con el Usuario

Ambos dispositivos se comunican con un computador empleando puertos USB. Una vez configurado los dispositivos de hardware, se construyó una arquitectura de software que permite la captura de la interacción de los dispositivos con el usuario, así como el despliegue de diversas escenas 3D para su navegación. El usuario, podrá navegar dentro de dichas escenas controlando e interactuando con el movimiento de la cabeza y el movimiento de las manos. Cada escena e interacción están asociados a un método de entrenamiento. Por otro lado, se mantiene un control de acceso de usuarios, para que cada uno pueda manejar diferentes niveles de entrenamiento, así como obtener sus estadísticas.

De esta manera, se tiene una solución de hardware y software para el entrenamiento mental basado en un juego serio [25], orientado a la inteligencia espacial bajo un ambiente de realidad virtual con el uso de dispositivos de bajo costo empleando movimientos gestuales. A continuación, se muestran los métodos de entrenamientos que son la base de nuestra solución.

A. Métodos de Entrenamiento

Los métodos de entrenamiento se componen principalmente por dos características fundamentales: la estimulación de ciertas áreas cognitivas y la interacción con el entorno. Dichas características fueron evaluadas en diferentes entrenamientos donde puedan adaptarse a un entorno de realidad virtual o aquellos que permitan mantener su función principal, pero cambiando el modo de interacción con el mismo. Por lo tanto, se seleccionaron y diseñaron tres entrenamientos:

Ruta de Enfoque: Consiste en la coincidencia de dos objetos del mismo color. Se genera una plataforma con una ruta preestablecida y una variedad de interruptores, dichos interruptores son controlados por el usuario e indican la dirección hacia donde debe de dirigirse un objeto móvil (ver Figura 1a). Los objetos estáticos serán colocados de manera aleatoria al final de cada ruta y a partir de ese momento comenzarán a circular los objetos móviles desde un único punto de salida donde se conecta con la ruta principal.

Paquete de Velocidad: Se basa en la colocación de un objeto dentro de un contenedor o paquete que contiene también otros objetos. Un contenedor o paquete está compuesto por varias partes o sub-paquetes, formando un plano en el espacio, cada sub-paquete contiene una variedad de objetos posicionados en un vector en particular (ver Figura 1b). El objetivo principal del usuario consiste en situar el objeto faltante en alguna posición del paquete sin que se superponga con los demás objetos al momento que cierre.

Punto / Puntos: Consiste en la unión de varios puntos con el fin de lograr una figura deseada, esta puede ser conocida o abstracta. Se le presenta al usuario una cierta cantidad de puntos simétricos y un modelo basado en dichos puntos, el usuario debe de lograr unir los puntos necesarios de manera lineal para completar la figura buscada (ver Figura 1c).

Las áreas cognitivas que presentan los entrenamientos previamente seleccionados consisten en: concentración, atención, agilidad o velocidad mental, modelación y memoria, como se muestra en la Figura 3. Basándose en dichas áreas y tomando en cuenta la adaptación de un entrenamiento a un entorno de realidad virtual, se diseñaron estimulaciones adicionales tales como la percepción, estimación y abstracción que podrán ser calculadas mediante un cierto tiempo de reacción que el usuario obtenga en los entrenamientos basándose en el comportamiento de cada uno.

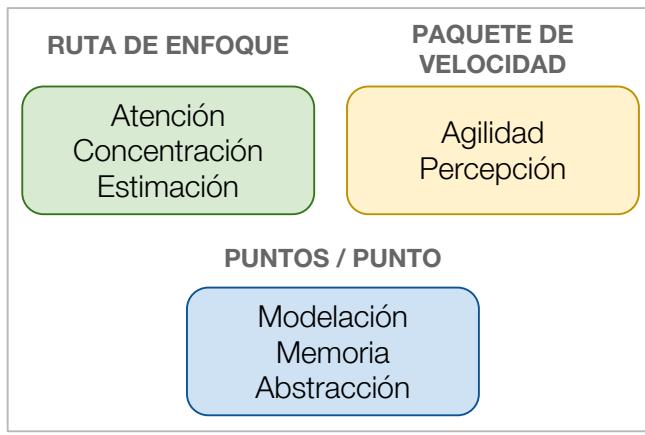


Figura 3. Áreas de Acción de cada Entrenamiento Propuesta

Los entrenamientos realizados son supervisados a través de un tiempo global para la realización del mismo. El tiempo varía según la dificultad y el tipo de entrenamiento definido. La definición operacional del grado de dificultad de la tarea es el tiempo de respuesta, ya que se supone que, a mayor grado de dificultad, mayor tiempo de procesamiento requerido para generar una respuesta apropiada.

Un ejemplo común para explicar este proceso, es el uso del tiempo para realizar un examen o evaluación a un estudiante. El cerebro del estudiante deberá procesar una información a partir de una pregunta, ya sea de desarrollo, y generar una respuesta en particular. No obstante, si la pregunta es de selección simple, el cerebro procesa la información utilizando un patrón para generar la respuesta que crea ser correcta. Para ambas preguntas, el cerebro se comporta y obtiene distintos tiempos de respuesta basándose en el tipo de pregunta que se le presente, suponiendo que el estudiante conoce efectivamente ambas respuestas. La velocidad de respuesta que el usuario obtenga en la aplicación permite a los algoritmos crear una estimación de resultados apropiados conforme a la complejidad del entrenamiento, generando valores aproximados de las estimulaciones realizadas.

Cabe destacar que, los resultados estimados por la aplicación no representan valores reales asociados a teorías sobre el proceso de estimulación que se adaptan al cerebro. Sin embargo, como se ha mencionado anteriormente, se sustentan bajo el principio básico del tiempo de respuesta que el cerebro presenta ante una tarea o actividad en particular.

B. Base de Datos

La base de datos se inicia con la necesidad de almacenar la información correspondiente para los entrenamientos: modelos, estadísticas, actividades e interpretaciones gestuales. Esto se realiza en un modelo convencional de entidad-relación. Los entrenamientos están compuestos por diferentes modelos que conforman una escena, algunos modelos pueden ser necesarios para la interpretación de gestos necesarios para la funcionalidad e interacción con la escena. La actividad corresponde al

tipo de estimulación que posee cada entrenamiento y utilizada para el cálculo de las estadísticas del usuario. Por último, los criterios de evaluación y resultados que son manejados por las estadísticas y dependerán del desempeño del usuario obtenido en cada uno de los entrenamientos.

Los usuarios, y debido a que la aplicación no está diseñada para un entorno en la red, sólo tendrán que ingresar un nombre de usuario con el fin de poder guardar resultados asociados a dicho nombre, similar a los juegos tradicionales de Microsoft Windows como Solitario o 3D Pinball.

C. Entidades 3D

La solución fue desarrollada con el motor de videojuegos Unity3d [26] que permite la utilización de ciertos recursos que facilitan el manejo de datos. Los objetos programables o *ScriptableObjects* son clases serializadas que permiten almacenar datos para compartir entre los componentes y evitar el acceso constante a base de datos. Los modelos utilizados en cada uno de los entrenamientos son almacenados en *ScriptableObjects* personalizados según el dominio de trabajo.

Los modelos fueron creados a partir de las funciones para posicionar, girar, expandir y agrupar objetos tridimensionales que ofrece Unity3d. Algunos objetos creados se muestran en la Figura 4. Algunos modelos fueron exportados directamente al entorno y por medio de archivos de tipo Object (.obj). Los modelos se clasifican en dos tipos: Estáticos y Dinámicos.



Figura 4. Modelos Creados del Tipo Estático o Dinámico

Los modelos estáticos pertenecen a la escena que representan y ambientan el entorno. Sin embargo, estos pueden ser creados en la escena sin componentes visuales manteniéndose como administradores de la escena. Los objetos dinámicos son aquellos que son inicializados a partir de un determinado momento, son almacenados en los *ScriptableObjects* y se utilizan para la interacción con el usuario a través de scripts de comportamiento.

De esta forma, se asociaron scripts a cada modelo dinámico perteneciente a una escena. Cada script se basa en un componente de comportamiento que permiten manipular el objeto a través de código, ofreciendo un intercambio de información entre los modelos a través de un determinado evento. Cada escena contiene un script administrador asignado a un único modelo. Esto permite manejar y controlar las

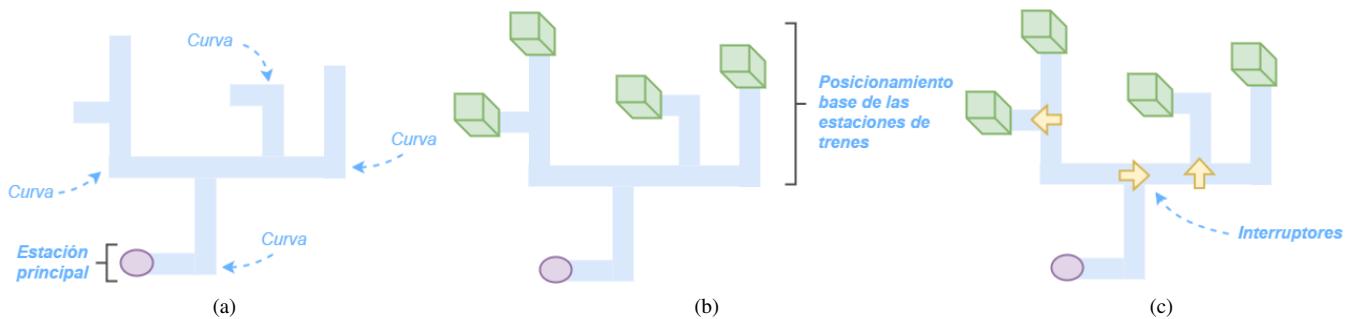


Figura 5. Esquema de las (a) Posiciones de las Rutas y Curvas, (b) las Inicialización de los Objetos, y (c) la Ubicación de los Interruptores

responsabilidades funcionales entre el comportamiento de cada uno de los modelos principales de la escena y la interacción con el usuario.

D. Leap Motion

A lo largo del desarrollo de los modelos generales, se mencionará la utilización de un controlador LeapHand para la ejecutar una cierta acción o tarea. Un controlador LeapHand es el medio por el cual el usuario interactúa con la aplicación que refleja la posición de las manos en el mundo virtual. El controlador es diseñado conforme a las exigencias que requiere un dominio en particular, es decir, está compuesto por distintos elementos y scripts de comportamiento para la captura de información. Se emplearon cuatro gestos principales a partir de los controladores asociados a la API del LeapHand descritas a continuación:

- **Toque:** Consiste en el uso del dedo índice del controlador para activar un objeto por medio de una colisión. Las colisiones son reconocidas por el objeto a través de etiquetas asignadas al componente del dedo índice. Cuando el objeto colisionado detecta un toque, se habilitan las variables físicas del objeto, permitiendo que este reaccione a la fuerza recibida por la colisión y pueda ejecutar una acción.
- **Doble Toque:** Presenta un comportamiento similar al gesto de Toque con la diferencia de que la etiqueta de colisión es asignada también al dedo medio del controlador. Es utilizado para acciones donde se requiera de mayor rapidez para realizar una acción.
- **Agarre:** Consiste en posicionar el controlador en una forma de agarre, es decir, la distancia entre la posición de punta de los dedos con respecto al punto centro de la palma sea menor que un umbral. Se utilizan vectores unitarios normalizados para obtener una distancia en el espacio. Para que ocurra el Agarre, el objeto candidato a ser agarrado debe estar identificado con una etiqueta que indique si es posible ser sostenido por el controlador.
- **Palma arriba:** Se basa en girar la palma del controlador hacia arriba, es decir, consiste en conocer el valor de la posición en un eje Y del vector director de la palma. Cuando este valor es igual a 0.5 unidades aproximadamente, se obtiene una palma girada unos 90°. Las

acciones se activan cuando la palma haya girado al menos hasta un valor aproximado de 0.6 unidades.

V. ENTRENAMIENTOS

Como se mencionó anteriormente, se han desarrollado tres métodos de entrenamiento cognitivos bajo un ambiente de realidad virtual: Ruta de Enfoque, Paquete de Velocidad y Puntos/Punto. Cada uno de estos, posee distintos niveles de dificultad: básico, medio y avanzado. A continuación, se mostrará más detalle de cada uno de estos.

A. Rutas de Enfoque

Para el diseño de este entrenamiento, se distribuyeron un conjunto de objetos 3D estáticos y dinámicos. Existirá un tren como objeto principal (dinámico) que debe pasar por diversas estaciones (estático). Así, cada tren iniciará en un único punto de salida con un color en particular, definido de forma pseudoaleatoria, a partir de este momento iniciará su recorrido por medio de las rutas hacia una estación, donde esta última iniciará a su vez con un color definido aleatoriamente. El usuario se encargará de interactuar con los botones asociados a los interruptores para realizar los cambios de dirección respectivos y guiar a cada uno de los trenes que circulan en la plataforma a la estación del color adecuado.

De acuerdo a la dificultad, ciertos objetos son instanciados y colocados dentro de la escena 3D. El posicionamiento final de las rutas y curvas se puede observar en la Figura 5a. Para realizar la creación de rutas y curvas, se cargan los distintos modelos y posicionarlos a partir de la estación principal, dicha estación indica el punto inicial de los trenes que circularán. Cuando la inicialización de las rutas culmina, se procede con el indicador de curvas para una ruta en específico, es decir, se colocan las curvas según el modelo de ruta.

Las estaciones se inicializan a partir del final de una ruta indicada, como se puede observar en la Figura 5b, se selecciona un color al azar y se asigna una posición en la escena (que se denomina plataforma). Una vez inicializada, se le asigna a cada modelo de estación su objeto lógico que permitirá generar las validaciones necesarias para la llegada de cada tren.

Para que el tren cumpla el objetivo de llegar a su destino (bases), existen una serie de interruptores que permiten cambiar la dirección del tren. Las posiciones de los interruptores

se inicializan de forma predeterminada basada en la dificultad seleccionada, para cada uno existe una dirección pseudoaleatoria a la propiedad del modelo. Seguidamente se inicializa y se asocia un botón encargado del cambio de dirección. El resultado final de este proceso de inicialización se muestra en la Figura 5c.

Por otro lado, el movimiento del tren consiste en el resultado de un vector de dirección por un escalar de velocidad v y un tiempo t , obteniendo como resultado el vector de posición que es utilizado por el controlador para trasladarse a sí mismo a través de las rutas. El valor de V permanecerá constante durante todo el recorrido del tren y, t es obtenido por medio de los cuadros por segundo del controlador. Esta ecuación se esquematiza en la Figura 6. La rotación del tren puede ejecutarse debido a una curva o mediante la dirección de un interruptor, siendo necesario que el tren reconozca cada uno de los casos.

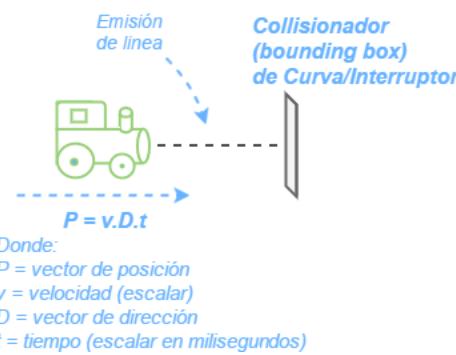


Figura 6. Movimiento y Emisión de Línea de un Tren

Para la rotación por curva se asignó una etiqueta de identificación a los modelos de curva que permiten al tren reconocer la colisión de un objeto en particular, una vez detectada la colisión, el controlador del tren obtiene la información de la curva indicando hacia donde debe de girar. La rotación por interruptor es muy similar la rotación por curva, con la diferencia que el controlador del tren, una vez detectada la colisión con un interruptor, activa parámetros que previenen que el tren gire nuevamente si este aún sigue colisionando con el mismo interruptor.

La llegada de cada tren, se valida a través de la presencia de una colisión continua con una estación. De esta manera, la colisión del tren persiste por un pequeño lapso del tiempo con el propósito de generar una sensación de entrada a dicha estación. Una vez transcurrido el tiempo se realiza la coincidencia de color y se almacena el resultado, en este punto el ciclo de vida del modelo de tren culmina y es eliminado de la plataforma.

Interacción: La interacción del usuario con el juego se basa en un panel que contiene cada uno de los botones asociados a un interruptor, representando todas las rutas y los interruptores en forma de miniatura. La configuración de sensibilidad del botón depende de la distancia con respecto a la posición del usuario, el tamaño y el peso físico del modelo.

Los indicadores de los botones consisten en una animación que permite identificar visualmente a cuál interruptor pertenece un botón, con el propósito de mejorar la experiencia de usuario y lograr la familiarización con el entorno al momento de ingresar al entrenamiento. La animación se basa en una flecha flotante sobre el interruptor y un efecto de luz sobre el botón asociado.

Empleando el dedo índice, se lanza un rayo que permite realizar la intersección entre una línea y el objeto que componen a los botones. Cuando una línea colisiona con un botón, se activan los indicadores del interruptor siempre y cuando la colisión persista con el mismo botón, de otra manera, los botones vuelven a su estado original (i.e. no seleccionados).

Puntuación y Estadísticas: La solución propuesta, tiene entre sus funciones la administración del tiempo de un entrenamiento, contando el tiempo de inicio y el tiempo total mostrando un panel visible para el usuario. El tiempo de inicio consiste en cinco segundos de preparación antes de la inicialización de trenes. De igual manera, se indica al momento de culminar el tiempo general, que consiste en tres minutos a partir del primer tren inicializado. El entrenamiento no culmina hasta que todos los trenes hayan llegado a su destino.

La puntuación se basa en un contador total de trenes y los aciertos que haya realizado el usuario. El contador se va incrementando a medida que vayan llegando trenes a una estación, donde a partir de cada llegada de los trenes se evalúan los datos obtenidos en todo su recorrido. Los datos para la evaluación estadística consisten en una variedad de criterios aplicados sobre el tren en un intervalo de tiempo de dos segundos antes de la colisión con un interruptor:

1. El tren llega a su respectiva estación y el usuario no haya realizado cambios de dirección del interruptor dentro del intervalo.
2. El tren llega a su respectiva estación y el usuario haya realizado uno o varios cambios de dirección del interruptor dentro del intervalo.
3. El tren no llega a su respectiva estación y el usuario no haya realizado cambios de dirección del interruptor dentro del intervalo.
4. El tren no llega a su respectiva estación y el usuario haya realizado uno o varios cambios de dirección del interruptor dentro del intervalo.

Estos datos son recopilados para su análisis y almacenamiento, y una vez culminado el entrenamiento se muestra al usuario los resultados finales del entrenamiento: nivel de atención, concentración y la puntuación general.

B. Paquete de Velocidad

En el diseño del entrenamiento Paquete de Velocidad, se plantea la selección de un tipo de paquete. Donde, un paquete es manipulado por una entidad encargada de efectuar la configuración de sus respectivas partes (o sub-paquetes), y proceder con distintos algoritmos de posicionamiento para los objetos estáticos en un espacio en particular. Una vez finalizado dicho proceso, se habilita un objeto manejable que

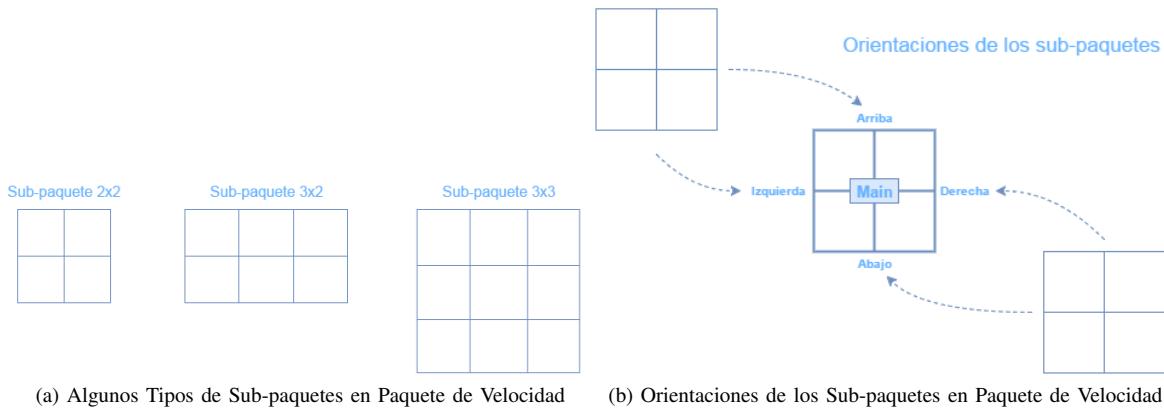


Figura 7. Organización y Estructuras de los Paquetes

consiste principalmente en el medio por el cual el usuario podrá interactuar con el escenario actual.

Un paquete se define como un modelo lógico compuesto por dos o más sub-paquetes físicos representados como una matriz plana en el espacio con un tamaño mínimo de 2×2 . Para que exista un paquete es necesario la existencia de al menos dos sub-paquetes con una orientación en particular, en la Figura 7a se muestran algunas configuraciones ejemplo de paquetes utilizados en el entrenamiento.

La configuración consiste en determinar el tipo o modelo lógico del paquete, asignar un sub-paquete principal y utilizar la orientación de éste modelo para realizar la inicialización, teniendo como resultado un paquete donde el usuario podrá colocar el objeto faltante en un algún espacio disponible. Se realizan las validaciones necesarias y se repite el proceso hasta la culminación del tiempo disponible.

La configuración de un paquete consiste en dos pasos: 1) elegir un sub-paquete y, 2) otorgar el rol de principal. El rol de principal es fundamental para el proceso de distribución e inicialización de objetos, y seguidamente la lectura de orientación del tipo de paquete previamente seleccionado. En la Figura 7b se puede observar las cuatro orientaciones de los sub-paquetes y dependiendo de la orientación que contenga el sub-paquete, el algoritmo de distribución e inicialización de objetos varía.

Por ejemplo, dado un paquete compuesto de tres sub-paquetes, si el sub-paquete con el rol principal tiene una orientación a la izquierda, se selecciona uno de los dos sub-paquetes restantes, donde será configurado e inicializado a la izquierda del principal, y para el último sub-paquete, dependiendo de la clase estrategia seleccionada al principio del proceso de creación. Entonces, dicho paquete será configurado e inicializado según la orientación del segundo sub-paquete.

La distribución e inicialización de objetos se basa en la creación de una lista lógica de objetos que son asignados al sub-paquete principal, y a partir de esta se selecciona una cierta cantidad de objetos para después ser reflejados en los sub-paquetes restantes. Primeramente, el sub-paquete principal inicializa una variedad de objetos y los descarta de la lista,

deshabilitando los espacios donde estos son inicializados con el fin de evitar superponer la posición con otro objeto. Luego, es enviada la lista con los objetos faltantes al paso de distribución. El algoritmo de distribución consiste básicamente en buscar una posición espejo a partir del sub-paquete precedente y reflejarlos al sub-paquete seleccionado, en caso de existir algún otro sub-paquete, el proceso se repite.

En la Figura 8 se muestra la ecuación empleada para reflejar un objeto a un sub-paquete adyacente. A nivel de comunicación entre los módulos internos de la solución, se requiere enviar como parámetros la orientación entre los dos sub-paquetes, el tamaño máximo del vector XY y la posición (x, y) actual del objeto sobre la matriz. Para una orientación Derecha-Izquierda se utiliza la variable X , mientras que para una orientación Arriba-Abajo se utiliza la ecuación con variable Y . Con la nueva posición espejo, el sub-paquete actual almacenará el objeto en su estructura de datos, y este procederá con su respectiva inicialización.

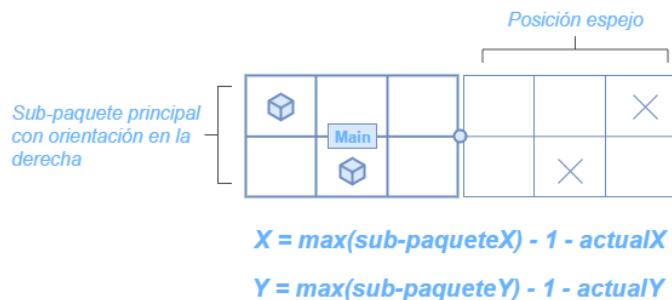


Figura 8. Ecuación para Obtener la Posición Espejo en Paquete de Velocidad

Interacción: En la interacción en este entrenamiento, además de las intersecciones rayo-botones, se requiere que el usuario pueda agarrar, arrastrar y soltar (i.e. *drag & drop*). Por ello, se deben considerar que un objeto tiene ciertas propiedades físicas básicas dentro de un ambiente virtual como el desplazamiento y la gravedad. A este tipo de objetos, dentro de la solución se les identifica como objetos manejables.

La manipulación de las restricciones consiste en anular cualquier tipo de fuerza física que pueda estar presente al

momento de retener el objeto, evitando comportamientos o efectos de rebote. Es importante destacar, que el controlador LeapHand ocasiona una serie de diversas colisiones contra el objeto en distintos ángulos, es decir, se generan distintas fuerzas en varias direcciones, produciendo el efecto de rebote en el controlador LeapHand que se busca eliminar.

El controlador LeapHand emite una línea a través de la palma, como se puede observar en la Figura 9. Si esta línea atraviesa la caja delimitadora (i.e. *bounding box*) que rodea al objeto, se procede con la anulación de las restricciones físicas del objeto. Cuando no existen restricciones en el objeto, el usuario puede realizar un gesto de agarre con el cual logrará sujetar el objeto y arrastrarlo a un punto en particular, el cálculo del gestor de agarre se basa en un algoritmo de distancia entre los dedos índice y medio contra el pulgar, obteniendo una posición particular utilizada como punto de agarre entre el objeto y el controlador LeapHand.

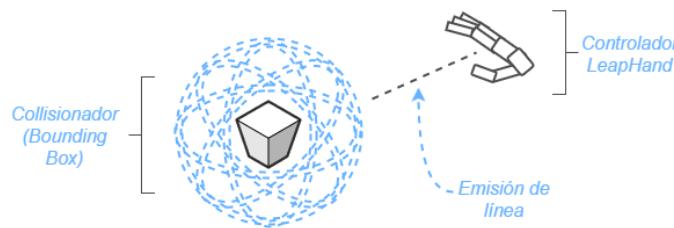


Figura 9. Área de Colisión y Línea de Emisión

Cuando un objeto manejable es colocado sobre algún espacio disponible, el controlador busca al sub-paquete más externo que posteriormente será eliminado. Así, la información se transfiere al sub-paquete anterior y se activa un método de animación que consiste en unir este último con el anterior, generando visualmente una transición en los objetos de ambos sub-paquetes. Este proceso se repite hasta llegar al sub-paquete principal, el cual adquiere, a este punto, todos los objetos inicializados con la excepción de que también estará el objeto colocado por el usuario. De esta forma, se podrá observar si este se ha sobrepujado sobre alguno de los objetos que pertenecen al paquete.

Por otra parte, se debe realizar una comprobación de la superposición de los objetos. Esta consiste en obtener la posición del espacio donde fue colocado el objeto y recorrer la lista de objetos existentes. Si la posición obtenida no existe en la lista, entonces es considerado un acierto y se obtiene la puntuación correspondiente al tipo de paquete, en caso contrario se considera un fallo. Cuando se coloca el objeto manejable, se inicializa un verificador con el propósito de generar un efecto amigable con el entorno indicándole al usuario si ha logrado un acierto o ha realizado un fallo en la resolución del problema, y a partir de este momento se repite todo el proceso de creación e inicialización de paquetes.

Puntuación y Estadísticas: El tiempo total del entrenamiento es de dos minutos exactos. Tal como el entrenamiento de Paquete de Velocidad, existe un tiempo de cinco segundos de preparación antes de la inicialización del primer paquete.

La puntuación consiste en un contador que se incrementa de acuerdo al paquete que se logre resolver, un contador para la cantidad de paquetes inicializados y un último contador correspondiente a los aciertos realizados. Como el entrenamiento tiene como finalidad estimular la agilidad mental, es indispensable llevar la suma del puntaje que se va obteniendo a medida que se resuelve cada paquete. Los datos para la evaluación estadística se basan en tres variables por cada tarea, estos se van recolectando a partir de la inicialización de un paquete:

- Tiempo de colocación: Calcula el tiempo de percepción y reacción del usuario inmediatamente después que es mostrado un paquete.
- Número de objetos: La cantidad de objetos inicializados y su respectiva distribución varía en la dificultad para la resolución de cada paquete.
- Tipo de paquete: Similar a la cantidad de objetos, el tipo de paquete (dimensión y número de sub-paquetes que lo conforman) genera distintas dificultades para la resolución, existiendo una puntuación diferente para cada uno de los tipos.

En este entrenamiento, se muestra al usuario su desempeño como agilidad, percepción y la puntuación general.

C. Punto / Puntos

En este entrenamiento, se deben construir líneas uniendo un conjunto de puntos siguiendo un patrón. Como estructura de datos, un conjunto de puntos seleccionados por el usuario se denomina modelo. Cuando un modelo es construido y asignado, el usuario interactuará con los puntos en la escena y podrá juntar o unir cada uno para formar líneas entre ellos. El usuario interactúa con los puntos de manera continua, es decir, si interactúa con un punto, este deberá trazar la línea hacia algún otro punto, sin embargo, tendrá la oportunidad de cancelar la selección del punto actual.

La construcción de un modelo consiste en crear una lista de puntos asociados entre sí que admite la repetición un punto más de una vez. Cada modelo seleccionado para el proceso de construcción permite hasta un cierto límite de asociaciones, el límite aumenta o disminuye según la dificultad seleccionada. En la Figura 10 se observa un ejemplo de la cantidad de puntos que pueden ser asociados a un modelo en particular.

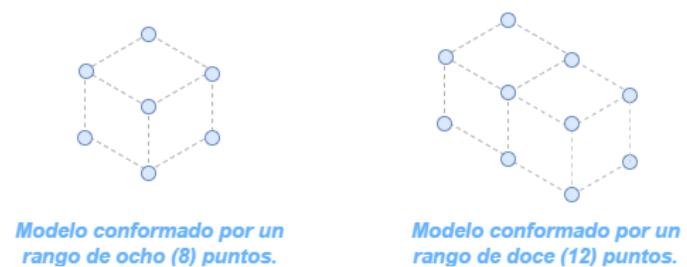


Figura 10. Cantidad de Puntos Asociados a un Modelo / Punto / Puntos

La construcción y creación de un modelo se hace de acuerdo a la selección del usuario en una dificultad previamente

seleccionada. Cada punto está numerado según el orden de creación, así, un modelo se conforma por la selección de un conjunto de puntos distribuidos simétricamente. Cada modelo contiene una lista de puntos asociados que componen una figura en el espacio.

Los puntos contienen una lista de puntos asociados, llamados puntos vecinos. Esta lista permite reconocer todos los puntos vecinos que contiene el modelo a partir de uno en particular. Cuando se define la cantidad de puntos que conforman a un modelo, se procede con la inicialización de las líneas a partir de un punto inicial y un punto final. La inicialización de líneas permite generar el modelo visual utilizado por el usuario como ejemplo y referencia del objetivo del entrenamiento.

El proceso de inicialización de líneas consiste en un bucle donde se recorre la lista de puntos correspondientes al modelo. A partir de un punto base de la lista, se inicializan las líneas correspondientes al par seleccionado. Una vez construido el modelo, se crea otra secuencia de puntos con el mismo orden de creación, enumeración y distribuidos para ser inicializados en la escena, asignándoles los valores de los puntos que posee el modelo creado. Este proceso es muy similar a la búsqueda por amplitud en un grafo (BFS – Breadth First Search).

La asignación de valores genera un apuntador lógico desde el punto en la escena con el punto que conforma al modelo, permitiendo manipular los datos y valores del punto mientras este permanezca en la escena del entrenamiento. La Figura 11 muestra una secuencia de modelación con tres puntos seleccionados.

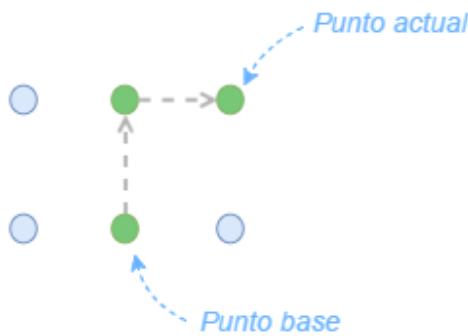


Figura 11. Cantidad de Puntos Asociados a un Modelo en Punto / Puntos

Interacción: La interacción en este entrenamiento se enfoca en dos comportamientos principales: colisión y gestor de rotación. El comportamiento para la colisión consiste en asignar componentes físicos al dedo índice para el reconocimiento de objetos específicos. Existen dos tipos de componentes físicos con distintas responsabilidades según la orientación de la mano. Para la mano izquierda, el componente asignado al dedo índice reaccionará a la colisión de un punto inicializado en la escena, permitiendo la creación de líneas para la construcción del modelo solicitado, a partir de esta interacción, el usuario puede modelar una figura. Para la mano derecha, el componente físico reaccionará cuando colisiona con un punto en la escena cuando el punto ya ha colisionado

con el componente de la izquierda. Esta última reacción es para anular el punto seleccionado por el usuario, permitiendo regresar a un paso anterior en la secuencia de modelación, o cancelar la secuencia.

Desde el punto de vista del usuario, las líneas representan el medio por el cual se podrá visualizar los modelos a construir. En cada cuadro de despliegue, una línea debe ser transformada adecuadamente (e.g. escalamiento, rotación) de acuerdo a la posición del controlador LeapHand. Del mismo modo, cuando existe una intersección entre el componente del controlador con un punto de la escena, se activan dichas transformaciones. De este modo, la configuración del vector de tamaño, o escalamiento, consiste en modificar constantemente su factor de escala mientras no esté contenido dentro del área de colisión del controlador LeapHand (ver Figura 12).

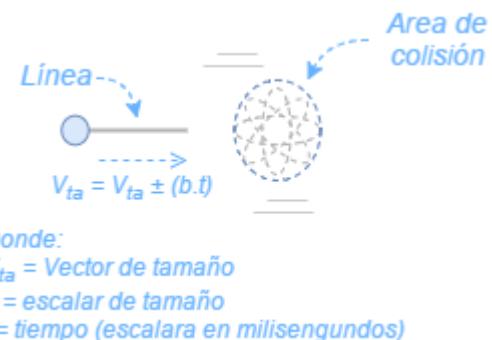


Figura 12. Comportamiento del Vector de Tamaño en Punto / Puntos

El vector de tamaño se va modificando constantemente mediante un tiempo t y un valor escalar de tamaño b . El funcionamiento tiene como propósito final, de manera visual, dejar la línea adherida al dedo índice del controlador LeapHand. Sin embargo, sin la configuración necesaria para el vector de rotación, este funcionamiento sólo aumentará o disminuirá de tamaño y no apuntará a la posición actual del controlador. La configuración del vector consiste en actualizar el vector director de una línea con respecto al dedo índice del controlador (i.e. vector normalizado).

Un modelo se compone por puntos, los cuales deben interactuar entre las intersecciones entre el componente LeapHand y la escena 3D donde residen los puntos. Cuando el evento de colisión se activa por medio del controlador LeapHand, existen tres comportamientos diferentes:

- Primera colisión: Se ejecuta cuando no ha iniciado la secuencia de puntos por parte del usuario, es decir, el primer punto seleccionado a partir de la construcción del modelo a completar.
- Colisión a partir de una secuencia: Se ejecuta cuando existe una secuencia, el usuario ha desplazado la línea a un punto en particular y este desea colocarlo. Si el usuario mantiene la colisión con el punto donde desea colocar la línea, se procede con incorporar un nuevo elemento a la secuencia principal y se realiza la validación de la unión del par de puntos.

- Colisión con el dedo índice del controlador derecho: Se ejecuta si existe una secuencia creada. Este comportamiento es utilizado para retornar a la secuencia anterior.

De este modo, la validación de un punto consiste en revisar y recorrer la lista del modelo base, y comparar el par de puntos unidos. Si un punto *A* contiene como vecino a un punto *B*, entonces la unión es correcta, en caso contrario se considera un error. Cuando el usuario decide culminar el modelo construido, se revisa la cantidad de errores obtenidos, en caso de no haber ningún error, se considera un acierto.

Puntuación y Estadísticas: El tiempo para este entrenamiento consiste en cinco minutos, dado que requiere una mayor interacción con el entorno. El puntaje consiste en registrar la cantidad de aciertos logrados junto con la cantidad total de modelos generados. La cantidad de revelaciones consiste en el llevar el control de todas las revelaciones que realice el usuario para completar un modelo. El valor obtenido de revelaciones es utilizado para realizar el cálculo de las estimulaciones del entrenamiento.

Para la evaluación, se considera el tiempo de realización del modelo y los aciertos. Sin embargo, estos se sustentan por la cantidad de revelaciones realizadas utilizando el valor obtenido. Un valor de revelaciones se considera negativo a partir de la segunda revelación, ya que el entrenamiento tiene como finalidad fomentar la capacidad de abstracción y memoria del usuario.

VI. EXPERIMENTACIÓN

La propuesta se compone principalmente en una aplicación basada en un entorno de realidad virtual, por lo que una medición del rendimiento es indispensable. Dicha medición se realizó desde un computador con procesador Intel *i7*, con tarjeta gráfica *NVIDIA GT540M*, bajo el sistema operativo *Windows7*. Para el dispositivo *Leap Motion* se utilizó la versión del *SDK Orion3.1.3* compatible para actuar desde un casco de realidad virtual. Se utilizó la versión *DK2* del *Oculus Rift* con el *SDK v0.6* compatible para tarjetas de video para portátiles. Por último, para el desarrollo de la aplicación, se utilizó *Unity3d* en su versión *5.1*, compatible con las especificaciones detalladas anteriormente. En la Figura 13 se muestra el equipo empleado para los experimentos.

Para la medición de los recursos computacionales empleados por nuestra solución, se empleó la herramienta Profiler de Unity [27], y ejecutándose 30 veces para obtener un promedio. El tiempo de procesamiento de la solución de software en un estado ocioso es de 1.55 ms con un porcentaje de distribución de tiempo balanceado. Sin embargo, los picos más altos de procesamiento se dan en presencia de una gran cantidad de objetos y la interacción del usuario. Así, se llega a alcanzar un tiempo de procesamiento de 32.05 ms con 88.1 % del tiempo consumido por el controlador *LeapHand*. Este comportamiento se debe a la recolección de basura que ocasiona la versión actual del *SDK* del dispositivo *Leap Motion*. Eventualmente se puede presenciar una caída de cps (cuadros por segundo) en la interacción con la aplicación.



Figura 13. Portátil, Dispositivos y Entorno de Desarrollo Empleado

Con respecto a la memoria empleada, debido a la gran cantidad de objetos (estáticos o dinámicos), el consumo de memoria es considerable. En promedio se emplea 0.62 Gb, variando de acuerdo a la cantidad de texturas empleadas en un cuadro visto por el usuario, las animaciones presentes, la cantidad de materiales, entre otros. Si se emplean versiones de menor resolución de los objetos, la cantidad de memoria se reducirá (e.g. algoritmos de reducción de polígonos, niveles de detalle).

Por otro lado, para la evaluación del software se realizó una encuesta a 6 personas con edades comprendidas entre 20-31 años, donde cada uno entrenó con los distintos juegos. Se evaluaron los siguientes criterios: diseño de objetos, jugabilidad y UX. El diseño de objetos representa a la correcta interpretación de los objetos 3D creados, con la representación conocida por el usuario de dicho objeto. La jugabilidad se refiere a la facilidad al realizar de forma satisfactoria el proceso completo de las actividades presentes en cada juego. Por último, UX (experiencia de usuario) indica la experiencia interna de cada persona al interactuar con cada aspecto del juego.

La encuesta se basa en una matriz de lado a lado para el análisis de datos, evaluando importancia/satisfacción (letra I para importancia, y S para satisfacción) para cada uno de los criterios conforme a su respectivo entrenamiento. La escala de varía entre 1-5, donde 1 se considera No Importante/Insatisfecho, y 5 como Importante/Muy Satisfecho. En la Tabla I se muestran los resultados de la encuesta del entrenamiento Ruta de Enfoque. Se puede observar que la jugabilidad es un factor que puede mejorarse en cuanto a la interacción y el posicionamiento del usuario con respecto a panel de control y escenario respectivamente. Sin embargo, el diseño de los objetos y la experiencia de usuario que presenta el entrenamiento es lo suficientemente estable para la satisfacción del usuario.



Figura 14. Interfaz de Usuario Intuitiva y Gestual Incorporada en Nuestra Solución

TABLA I
RESULTADO DE ENCUESTA PARA RUTA DE ENFOQUE

Diseño		Jugabilidad		UX	
I	S	I	S	I	S
4	5	4	4	5	4
4	4	4	4	3	4
5	4	5	4	5	4
3	4	3	3	3	4
5	5	5	3	5	5
4	4	5	5	4	4

TABLA III
RESULTADO DE ENCUESTA PARA PUNTO / PUNTOS

Diseño		Jugabilidad		UX	
I	S	I	S	I	S
4	3	4	3	4	4
4	2	4	4	4	4
5	3	4	2	4	3
5	3	4	3	4	4
5	2	5	3	5	4
5	3	5	3	4	5

La evaluación obtenida para el entrenamiento Paquete de Velocidad se muestra en la Tabla II, donde se hace énfasis en el diseño de los objetos y sus elementos. No obstante, la jugabilidad que presenta se mantiene entre la más valorada en comparación a los otros entrenamientos, teniendo en cuenta que, la experiencia de usuario es un balance entre el diseño y la jugabilidad del mismo.

TABLA II
RESULTADO DE ENCUESTA PARA PAQUETE DE VELOCIDAD

Diseño		Jugabilidad		UX	
I	S	I	S	I	S
5	2	5	5	4	4
5	1	4	5	5	3
5	3	5	4	4	4
4	3	5	5	5	4
5	3	5	5	4	3
4	4	4	4	5	5

Por último, el resultado para el entrenamiento Puntos / Punto se refleja en la Tabla III. Analizando la tabla se puede deducir que una jugabilidad ligeramente insatisfactoria. Según las opiniones recibidas (en retroalimentación obtenida posterior a la encuesta), esto se debe al nivel de dificultad que presenta el entrenamiento en cuanto a la interacción y la modelación de las figuras.

En cuanto a la interfaz de usuario, esta fue aceptada por todos los usuarios de forma satisfactoria. De hecho, no se requirió instrucciones extra para los usuarios antes de empe-

zar. En la Figura 14 se muestran un par de instantes dentro que la solución antes de empezar los juegos que consiste en la introducción de datos 14(a), y la selección de los juegos 14(b).

VII. CONCLUSIONES

El estímulo de las áreas cognitivas por medio de entrenamientos mentales a través de juegos adaptados a un ambiente de realidad virtual, se presentó en este artículo. Con la selección del juego, el usuario puede decidir qué áreas cognitivas desea estimular y/o visualizar el desempeño que ha obtenido con los distintos entrenamientos, interactuando por medio de una interfaz natural de usuario. Así, la inteligencia espacial permite al cerebro pensar de forma precisa y lógica la solución de un problema real o imaginario. La práctica constante de ejercicios, tareas o actividades relacionadas a las habilidades espaciales ayuda al cerebro a estimular aquellas áreas cognitivas utilizadas en este tipo de capacidad de pensamiento.

Los entrenamientos desarrollados basados en la inteligencia espacial permiten al usuario experimentar un entorno de realidad virtual de inmersión total a través de la interacción gestual. Cada dificultad que representa un entrenamiento, modifica las propiedades, los componentes y sus controladores ofreciendo un nuevo nivel de desenvolvimiento con el mismo. Del mismo modo, el proceso estadístico empleado, permite estimar los valores de las distintas estimulaciones cognitivas por medio del tiempo de respuesta que el usuario realice ante una tarea,

logrando obtener resultados apropiados y acertados para el control del mismo.

Los resultados obtenidos muestran que el consumo de recursos computacionales es aceptable, considerando el uso de un portátil de prestaciones media. Por otro lado, las encuestas realizadas permiten deducir una aceptación satisfactoria por parte de los usuarios. Sin embargo, es claro que las pruebas no son suficientes (cantidad, aspectos medidos, edades) para obtener resultados concluyentes del impacto del estudio; siendo un buen inicio a investigaciones futuras. Igualmente, podemos estimar que mejorar el hardware (e.g. tarjeta de video de alta generación) o incluir versiones nuevas del software (e.g. SDK de Oculus, versión 5.5+ de Unity3d), pueden influir positivamente en el rendimiento global de la solución.

En un futuro, se propone construir ambientes virtuales más realistas añadiendo elementos de iluminación global en las escenas como *ambient occlusion*, *color bleeding*, sombras suaves, entre otros. Igualmente, se propone añadir un módulo de “tutorial en vivo” que permita indicar al usuario los objetivos del entrenamiento justo antes de iniciarlos, es decir, un escenario similar al entrenamiento seleccionado que contenga una inteligencia artificial indicando las acciones que debe realizar.

REFERENCIAS

- [1] J. Gordon, “In Corpore Sano: Health and Humanities,” in *Proceedings of the Australian Academy of the Humanities Symposium*, Canberra, Australia, Noviembre 2005, pp. 75–81.
- [2] D. Ballard, “Improve Brain Health with Neurobics,” *Emergency Medicine News*, vol. 32, 2010.
- [3] S. Makin, “Brain training: Memory games,” *Nature*, p. S10–S11, Marzo 2016.
- [4] D. J. Simons, W. R. Boot, N. Charness, S. E. Gathercole, C. F. Chabris, D. Z. Hambrick, and E. A. L. Stine-Morrow, “Do ‘Brain-Training’ Programs Work?” *Psychological Science in the Public Interest*, vol. 17, no. 3, pp. 103–186, 2016.
- [5] J. A. McCabe, T. S. Redick, and R. W. Engle, “Brain-Training Pessimism, but Applied-Memory Optimism,” *Psychological Science in the Public Interest*, vol. 17, no. 3, pp. 187–191, 2016.
- [6] S. Fitzgerald, “Is Successful Brain Training Fake News?: Neurologists Parse Out the Messaging for Patients,” *Neurology Today*, vol. 17, pp. 8–18, 2017.
- [7] D. Bavelier and R. J. Davidson, “Brain Training: Games to do you Good,” *Nature*, p. 425–426, Febrero 2013.
- [8] T. Singer, “Training Mind and Heart: Effects of Mental Training on Mental and Physical Health, Brain, and Prosocial behavior, presentado en el 14th Dalai Lama,” Strasbourg, France, 2016, <https://goo.gl/UC6vtG>.
- [9] J. Buitenhof, J. Murre, and R. Ridderinkhof, “Brain Training in Progress: A Review of Trainability in Healthy Seniors,” *Frontiers in Human Neuroscience*, no. 183, p. 425–426, Julio 2017.
- [10] A. van Dam, A. Forsberg, D. Laidlaw, J. LaViola, and R. Simpson, “Immersive VR for Scientific Visualization: a Progress Report,” *IEEE Computer Graphics and Applications*, vol. 20, no. 6, pp. 26–52, Noviembre 2000.
- [11] D. Plass-Oude Bos, B. Reuderink, B. van de Laar, H. Gürkök, C. Mühl, M. Poel, A. Nijholt, and D. Heylen, *Brain-Computer Interfacing and Games*. London: Springer London, 2010, pp. 149–178.
- [12] A. Lécuyer, F. Lotte, R. B. Reilly, R. Leeb, M. Hirose, and M. Slater, “Brain-Computer Interfaces, Virtual Reality, and Videogames,” *Computer*, vol. 41, no. 10, pp. 66–72, Octubre 2008.
- [13] J. A. Juanes, J. J. Gómez, P. D. Peguero, J. G. Lagándara, and P. Ruisoto, “Analysis of the Oculus Rift Device As a Technological Resource in Medical Training Through Clinical Practice,” in *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2015, pp. 19–23.
- [14] X. Xu, K. B. Chen, J.-H. Lin, and R. G. Radwin, “The Accuracy of the Oculus Rift Virtual Reality Head-mounted Display During Cervical Spine Mobility Measurement,” *Journal of Biomechanics*, vol. 48, no. 4, pp. 721 – 724, 2015.
- [15] J. Bolton, M. Lambert, D. Lurette, and B. Unsworth, “PaperDude: A Virtual Reality Cycling Exergame,” in *CHI ’14 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’14. ACM, 2014, pp. 475–478.
- [16] N. F. Kleven, E. Prasolova-Førland, M. Fominykh, A. Hansen, G. Rasmussen, L. M. Sagberg, and F. Lindseth, “Training Nurses and Educating the Public using a Virtual Operating Room with Oculus Rift,” in *2014 International Conference on Virtual Systems Multimedia (VSMM)*, Diciembre 2014, pp. 206–213.
- [17] F. Moreno, J. Ojeda, E. Ramírez, C. Mena, O. Rodríguez, J. Rangel, and S. Alvarez, “Un Framework para la Rehabilitación Física en Miembros Superiores con Realidad Virtual,” in *Proceedings de la I Conferencia Nacional de Computación, Informática y Sistemas (CoNCISA 2013)*, Naiguatá, Venezuela, 2013, pp. 77–84.
- [18] A. Mayo and E. Ramírez, “DanceNet: Entorno Colaborativo para la Enseñanza de Coreografías empleando un Kinect,” in *Proceedings de la III Conferencia Nacional de Computación, Informática y Sistemas (CoNCISA 2015)*, Valencia, Venezuela, 2015, pp. 119–129.
- [19] Oculus VR, LLC, “Oculus Rift,” 2017, <http://www.oculus.com>.
- [20] H. Hoffman, W. Meyer, M. Ramirez, L. Roberts, E. Seibel, A. Barbara, S. Sharar, and P. David, “Feasibility of Articulated Arm Mounted Oculus Rift Virtual Reality Goggles for Adjunctive Pain Control During Occupational Therapy in Pediatric Burn Patients,” *Cyberpsychology, Behavior, and Social Networking*, vol. 17, pp. 397–401, Junio 2004.
- [21] M. Just, D. Stirling, M. Ros, F. Naghdy, and P. Stapley, “A Comparison of Upper Limb Movement Profiles when Reaching to Virtual and Real Targets using the Oculus Rift: Implications for Virtual-reality Enhanced Stroke Rehabilitation,” *Journal of Pain Management*, pp. 277–281, 2016.
- [22] D. Rose, B. Brooks, and A. Rizzo, “Virtual Reality in Brain Damage Rehabilitation: Review,” *CyberPsychology & Behavior*, vol. 8, no. 3, pp. 241–262, Junio 2005.
- [23] A. Lécuyer, F. Lotte, R. B. Reilly, R. Leeb, M. Hirose, and M. Slater, “Brain-Computer Interfaces, Virtual Reality, and Videogames,” *Computer*, vol. 41, no. 10, pp. 66–72, Octubre 2008.
- [24] Leap Motion, Inc., “Leap Motion,” 2017, www.leapmotion.com.
- [25] D. R. Michael and S. L. Chen, *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.
- [26] Unity, “Unity 3D,” 2017, <https://unity3d.com>.
- [27] Unity, “Profiler Window,” 2017, <https://docs.unity3d.com/Manual/ProfilerWindow.html>.

Filtrado 2D de Imágenes Médicas Basado en Filtros Myriad Ponderados Recursivos

Liliana Bautista

Postgrado en Ingeniería Biomédica
Universidad de Los Andes
Mérida, Venezuela
Email: liesbato@gmail.com

Juan Marcos Ramírez

Centro de Estudios en Microcomputación y
Sistemas Distribuidos (CEMISID)
Universidad de Los Andes
Mérida, Venezuela

Resumen—El problema de la remoción de ruido está presente en diversas aplicaciones de procesamiento de imágenes médicas. En general, se asume que el ruido que contamina esta clase de imágenes sigue una distribución gaussiana, donde los filtros desarrollados bajo esta suposición incorporan un efecto de borrosidad (*blurring*) que reduce el contraste en las imágenes. Por tanto, se hace necesaria la incorporación de técnicas de filtrado que se ajusten con mayor precisión a las características del ruido, preservando a su vez los contrastes de las imágenes. En este sentido, los filtros myriad ponderados recursivos han sido propuestos recientemente como estructuras robustas ante una amplia variedad de modelos de ruido, donde la recursividad inherente a estas estructuras favorece el diseño de filtros con respuestas muy cercanas a las operaciones de filtrado deseadas. Este trabajo propone la implementación de técnicas de filtrado 2D basadas en los filtros myriad ponderados recursivos para la remoción del ruido dependiente de las amplitudes de los datos, un tipo de contaminación que afecta frecuentemente a distintas modalidades de imágenes médicas. Estas técnicas de filtrado 2D aprovechan la recursividad de los filtros myriad ponderados recursivos con el objeto de diseñar estructuras que minimicen los efectos del ruido sin tener que recurrir a una etapa adicional de realce de contrastes. El rendimiento de estas técnicas de filtrado se evalúa, usando diferentes métricas de calidad de la imagen, en la remoción de ruido presente en bases de datos de tomografía computarizada multicorte y de resonancia magnética.

Palabras Clave—Filtrado 2D; filtros myriad ponderados recursivos; resonancia magnética; ruido Poisson; ruido riciano; tomografía computarizada.

I. INTRODUCCIÓN

El diagnóstico usando imágenes médicas consiste en una estrategia no invasiva que se instrumenta como herramienta de análisis para la detección temprana de enfermedades, la localización de lesiones y la planificación de intervenciones quirúrgicas. En este sentido, existen una amplia variedad de modalidades de imágenes médicas que han sido desarrolladas con el objeto de evaluar tanto la morfología de las estructuras anatómicas como el funcionamiento de los órganos de interés.

Un ejemplo de estas modalidades de imágenes médicas lo constituye la tomografía computarizada multicorte (MSCT, *multi-slice computed tomography*) que básicamente consiste en una tecnología de adquisición basada en rayos X que favorece la captura de información relacionada con la disposición 3D de las estructuras anatómicas de interés [1]. Además, a partir de la información suministrada por las imágenes de MSCT, se han desarrollado diversos métodos de segmentación 2D y 3D, cuyos modelos resultantes han sido usados en la estimación

de múltiples parámetros relacionados con el funcionamiento de los órganos bajo proyección [2], [3].

Por su parte, las imágenes de resonancia magnética (MRI, *magnetic resonance imaging*) es otra modalidad de imágenes médicas que obtiene información tanto de la anatomía como del funcionamiento de los órganos en estudio. En general, un escáner de MRI captura la información espacial de los órganos de interés, y en algunas ocasiones, el software integrado a los equipos de adquisición permite la visualización de modelos 3D de las estructuras anatómicas bajo observación. Además, la adquisición de MRI no involucra la proyección de rayos X, característica que la diferencia de los equipos de MSCT. No obstante, las modalidades de MRI y MSCT se consideran herramientas complementarias de diagnóstico, debido a que cada una de estas tecnologías presenta sus ventajas y limitaciones en diversas aplicaciones médicas [4], [5].

En general, las imágenes de MRI y MSCT son afectadas por ruido dependiente de las amplitudes de los datos. Esta clase de contaminación produce un tipo de distorsión en las imágenes que complica las tareas de discriminación de las distintas regiones de interés, dificultando a su vez, los procesos de segmentación 2D y 3D. Además, las técnicas tradicionales de tratamiento de imágenes degradan su desempeño ante este tipo de contaminación debido a que el proceso de adquisición no sigue un modelo de ruido aditivo independiente de las amplitudes de los píxeles de la imagen original. En el contexto de imágenes médicas, la distorsión que afecta a las imágenes de MSCT se modela usando la distribución de Poisson, mientras que las perturbaciones presentes en las imágenes de MRI se caracterizan usando la distribución de Rice [6]–[8].

En las últimas dos décadas se han propuesto múltiples métodos de remoción de ruido dependiente de las amplitudes de los datos. En concreto, estas técnicas aplican una operación de filtrado lineal sobre una versión transformada de la imagen, donde las perturbaciones en el nuevo dominio se asumen como muestras aditivas que obedecen a una distribución gaussiana [7], [9], [10]. Luego, la imagen de interés se obtiene a partir de la aplicación de la transformada inversa sobre la versión filtrada de la imagen transformada. No obstante, estos métodos imponen un efecto de difuminación (*blurring*) sobre la imagen de interés, requiriendo la aplicación de técnicas de realce de contrastes. Por consiguiente, se hace necesario la evaluación de otras técnicas de filtrado de imágenes, que minimicen el efecto del ruido, y que a su vez preserven los contrastes entre las distintas regiones que conforman la imagen.

Este trabajo propone la implementación de técnicas de filtrado 2D basada en los filtros myriad ponderados recursivos [11], [12] para la remoción de ruido dependiente de las amplitudes de los píxeles. En concreto, esta nueva familia de filtros recursivos considera tanto un subconjunto de muestras de entrada como un subconjunto de salidas previas en la estimación de la salida actual del filtro, y además, esta clase de filtros tiene la capacidad de rechazo de una amplia gama de modelos de ruido mediante el ajuste de un par de parámetros sintonizables. Así mismo, estos filtros han sido implementados con éxito en diversas aplicaciones de procesamiento de señales e imágenes, donde la recursividad inherente a estas estructuras favorece la recuperación de respuestas ajustadas a las salidas deseadas [11], [12].

La organización del artículo es la siguiente. En la Sección II se describe brevemente los modelos de ruido dependiente de las amplitudes de los píxeles, y en la Sección III se incorporan las métricas de calidad de la imagen usadas en la evaluación de las técnicas de filtrado. Los fundamentos de los filtros myriad ponderados recursivos, sus algoritmos de diseño y los aspectos prácticos para la implementación de las técnicas de filtrado 2D se desarrollan en la Sección IV. La Sección V presenta los resultados de la implementación de estas estructuras recursivas en el filtrado 2D de bases de datos sintéticas y reales de MSCT y MRI. Finalmente, la Sección VI expone las conclusiones derivadas del trabajo.

II. MODELOS DE RUIDO DEPENDIENTE DE LOS DATOS Y MÉTRICAS DE CALIDAD DE LA IMAGEN

A. Ruido Poisson

Los fotones de rayos X se proyectan sobre una superficie receptora siguiendo un patrón aleatorio, de manera que una región de esta superficie puede recibir más fotones que otra a pesar de que ambas regiones hayan sido expuestas a la misma intensidad de radiación. Este comportamiento aleatorio se conoce como *ruido cuántico*. En general, el ruido cuántico se describe usando el modelo estadístico de Poisson. Este modelo asume que cada píxel x de una imagen $f(x)$ se modela como una variable aleatoria que obedece a una distribución Poisson de parámetro $\lambda = x_0$, donde x_0 es el valor del píxel correspondiente de la imagen sin ruido. En concreto, la función de densidad de probabilidad de Poisson se define como

$$P(x = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (1)$$

donde λ es proporcional al número de fotones que incide en la superficie receptora durante el tiempo de exposición. La Figura 1 muestra las curvas de la función de densidad de probabilidad de la distribución Poisson para distintos valores del parámetro λ . Como se puede observar en la Figura 1, λ está relacionado tanto con la media como con la desviación estándar de la distribución. Por tanto, considerando que λ es el valor del píxel de la imagen original, el nivel de ruido que contamina a cada píxel será proporcional a su intensidad [7]. En otras

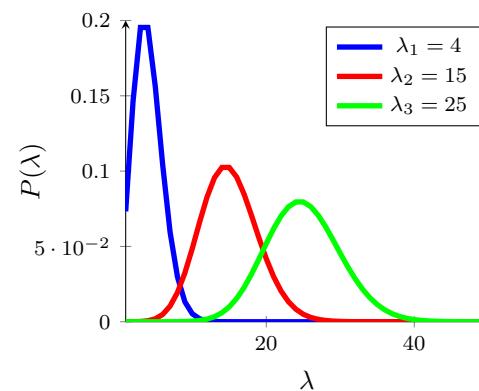


Figura 1. Función de Densidad de Probabilidad de Poisson para Distintos Valores de λ

palabras, el nivel de contaminación del ruido Poisson depende de la magnitud de los píxeles de la imagen sin ruido.

B. Ruido Riciano

El ruido riciano es un modelo de contaminación dependiente de los datos que afecta significativamente las imágenes de MRI de bajo contraste. En regiones de baja intensidad (oscuras) de la imagen, el ruido riciano tiende a seguir un modelo estadístico de Rayleigh, mientras que en las regiones de mayor intensidad el ruido se aproxima a una distribución gaussiana.

Específicamente, el ruido riciano se incorpora en las bases de datos de MRI de la siguiente manera. La información capturada por el equipo consiste en la transformada de Fourier de la imagen a visualizar, cuyos datos se almacenan en dos canales (canal real y canal imaginario). En general, las perturbaciones que afectan a cada canal son atribuidas a la contaminación inducida por los componentes electrónicos del equipo, que comúnmente se caracteriza como ruido aditivo blanco gaussiano. Luego, se determina la transformada inversa de Fourier de los datos adquiridos, donde la información de la transformada inversa se registra también en dos canales (canal real y canal imaginario). La contaminación en el dominio de la transformada inversa de Fourier también se modela como ruido aditivo blanco gaussiano. No obstante, la imagen a visualizar, se obtiene como la magnitud de la transformada inversa de Fourier, donde la intensidad de cada píxel se obtiene como la raíz cuadrada de la suma de los cuadrados de los píxeles ubicados en la posición correspondiente en cada canal. Esta operación que determina cada píxel de la imagen a visualizar hace que la contaminación resultante se aproxime con mayor precisión a un modelo estadístico dependiente de las amplitudes de los píxeles, que comúnmente se describe usando la distribución de Rice [7], [13].

III. MÉTRICAS DE CALIDAD DE LA IMAGEN

Las métricas de calidad de la imagen son un conjunto de operadores que se aplican sobre las imágenes procesadas con el propósito de valorar cuantitativamente el rendimiento de

las técnicas de tratamiento de imágenes. En general, teniendo en cuenta la disponibilidad de una imagen de referencia, que se considera libre de ruido, estas medidas de evaluación se clasifican en dos categorías: métricas de calidad de la imagen con referencia (FR, *Full Reference*) y métricas de calidad de la imagen ciegas o sin referencia (BR, *Blind Reference*) [14]. En las últimas décadas se han propuesto un número importante de operadores para la medición de la calidad de las imágenes, donde cada uno de estos operadores posee sus ventajas y limitaciones en términos de precisión, velocidad de cálculo y la aplicación a considerar. A continuación, se describen las métricas de calidad de la imagen usadas en la evaluación de las técnicas de filtrado basadas en los filtros myriad ponderados recursivos.

A. Métricas de Calidad de la Imagen con Referencia

Las métricas pertenecientes a esta categoría comparan la imagen de interés con respecto a una imagen de referencia que se considera libre de distorsión. En concreto, esta clase de métricas mide la diferencia entre las dos imágenes. Tales métodos son ampliamente usados en la cuantificación del ruido que afecta la imagen bajo evaluación, sin embargo, este conjunto de métodos presenta serias dificultades en la valoración del realce de las imágenes debido a que, en general, no se dispone de una imagen realizada de alta calidad [14].

1) *Error Medio Absoluto (MAE, Mean Absolute Error)*: se obtiene como el promedio de la diferencia absoluta entre la imagen de referencia y la imagen a evaluar, esto es [15]:

$$\text{MAE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_o(i,j) - I_r(i,j)|. \quad (2)$$

donde MN es el tamaño de la imagen, I_o es la imagen de referencia y I_r es la imagen bajo evaluación.

2) *Error Medio Cuadrático (MSE, Mean Squared Error)*: el MSE determina la potencia de la distorsión en la imagen bajo evaluación, en otras palabras, obtiene la potencia del error entre la imagen de referencia y la imagen procesada, dada por:

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_o(i,j) - I_r(i,j))^2. \quad (3)$$

3) *Relación Señal a Ruido Pico (PSNR, Peak Signal-to-Noise Ratio)*: es la métrica más usada para la evaluación del rendimiento de las técnicas de procesamiento de imágenes [14], [15]. En concreto, el PSNR está definido como la relación entre la potencia máxima posible de la imagen y la potencia de la distorsión [15], es decir:

$$\text{PSNR[dB]} = 10 \log_{10} \left(\frac{(P-1)^2}{\text{MSE}} \right), \quad (4)$$

donde P es el rango dinámico de valores de intensidades de los píxeles, por ejemplo, $P = 256$ para imágenes en escala de grises de 8 bits.

B. Métricas de Calidad de la Imagen Sin Referencia

Este conjunto de métricas se fundamenta en la idea de que el valor de un píxel depende fuertemente de los valores de los píxeles más cercanos, disminuyendo esta dependencia a medida que los píxeles se encuentren más distantes entre sí. Este conjunto de métricas son apropiadas cuando no se dispone de una imagen de referencia, situación que es muy común en aplicaciones de procesamiento de imágenes médicas. Además, las métricas ciegas apuntan hacia una mayor correlación del operador con las características de evaluación subjetiva del sistema de visión humana [16].

1) *Medida de Realce (EME, Measure of Enhancement)*: esta métrica valora cuantitativamente el realce de la imagen basada en los conceptos de percepción visual de Webers Low [17]. Específicamente, este operador promedia los mayores cambios de magnitud de las intensidades de los píxeles registrados en un conjunto de bloques de la imagen. En general, esta métrica es adecuada para evaluar el realce de imágenes con estructuras uniformes y regulares. La medida EME se define como:

$$\text{EME} = \frac{1}{k_1 k_2} \sum_{m=1}^{k_1} \sum_{l=1}^{k_2} 20 \ln \left(\frac{I_{max}^{l,m}}{I_{min}^{l,m}} \right), \quad (5)$$

donde la imagen filtrada I es dividida en $k_1 \times k_2$ bloques no solapados, además, I_{max} y I_{min} corresponden a las intensidad máxima y mínima de los píxeles en el bloque m, l de la imagen filtrada.

2) *Métrica de Realce por Entropía (EMEE, Measure of Enhancement by Entropy)*: en la tarea de evaluar el contraste de la imagen se adiciona la medida estadística conocida como entropía que permite conocer la distribución de los niveles de grises, de este modo, cuanto mayor es el número de niveles de grises que participan en la imagen, mayor es la entropía. Esta métrica se ajusta para imágenes que contienen objetos formados por regiones uniformes de diversos niveles de grises y se determina como:

$$\text{EMEE} = \frac{1}{k_1 k_2} \sum_{m=1}^{k_1} \sum_{l=1}^{k_2} \alpha \left(\frac{I_{max}^{l,m}}{I_{min}^{l,m}} \right)^\alpha \ln \left(\frac{I_{max}^{l,m}}{I_{min}^{l,m}} \right), \quad (6)$$

donde $\alpha \in (0, 1]$ es una constante que permite parametrizar la métrica EMEE. En particular, cuanto mayor es el valor de α mayor es el énfasis de la entropía, permitiendo de este modo el manejo de mayor aleatoriedad en la imagen filtrada [18].

IV. FILTROS MYRIAD PONDERADO RECURSIVOS

Los filtros myriad ponderados recursivos pertenecen a una clase general de estructuras de filtrado basadas en el operador no lineal muestra myriad. Más precisamente, la salida de este tipo de filtros toma en cuenta, además de un subconjunto de muestras de la señal de entrada, un subconjunto de salidas previas del filtro. En otras palabras, considere un subconjunto de muestras de entrada $\{x_i|_{i=-N}^N\}$, que para la aplicación específica de filtrado de imágenes se considerará como un grupo de píxeles procedente de una ventana de dimensiones

$\ell \times \ell$, donde $\ell^2 = 2N + 1$. Además, considere un subconjunto de salidas previas $\{y_j|_{j=1}^N\} = \{y[n-j]|_{j=1}^N\}$, que en conjunto con las muestras de la imagen de entrada se incluyen en una ventana de observación recursiva. Adicionalmente, considere un conjunto de coeficientes $\{g_i|_{i=-N}^N\}$ que pondera los niveles de confiabilidad de las muestras de entrada y un conjunto de ponderaciones $\{h_j|_{j=1}^N\}$, que controla la influencia de las salidas previas en la respuesta actual del filtro. En particular, la salida del filtro *myriad ponderado recursivo* (RWMy, recursive weighted myriad) está definida como la solución del problema de minimización [11], [12]:

$$\begin{aligned}\hat{\beta}_{K_1, K_2} &= \arg \min_{\beta} \{G(\beta)\} \\ &= \arg \min_{\beta} \left\{ \prod_{i=-N}^N [K_1^2 + |g_i|(\operatorname{sgn}(g_i)x_i - \beta)^2] \right. \\ &\quad \left. \dots \times \prod_{j=1}^N [K_2^2 + |h_j|(\operatorname{sgn}(h_j)y_j - \beta)^2] \right\}, \quad (7)\end{aligned}$$

donde K_1 y K_2 son parámetros sintonizables que controlan la robustez del filtro ante diversos tipos de ruido. Debido a que el operador logarítmico es una función monótona no decreciente, la Ecuación (7) puede reformularse como:

$$\begin{aligned}\hat{\beta}_{K_1, K_2} &= \arg \min_{\beta} \{\log(G(\beta))\} = \arg \min_{\beta} \{\mathcal{H}(\beta)\} \\ &= \arg \min_{\beta} \left\{ \sum_{i=-N}^N \log[K_1^2 + |g_i|(\operatorname{sgn}(g_i)x_i - \beta)^2] \right. \\ &\quad \left. \dots + \sum_{j=1}^N \log[K_2^2 + |h_j|(\operatorname{sgn}(h_j)y_j - \beta)^2] \right\}. \quad (8)\end{aligned}$$

El diagrama de bloques de una estructura no causal del filtro RWMy se muestra en la Figura 2. Como puede observarse en dicha figura, la salida del filtro RWMy depende de las muestras de entrada así como de las salidas previas del filtro. Con el propósito de determinar algunas propiedades de estos filtros recursivos, a continuación se desarrolla un análisis de la función costo objetivo $\mathcal{H}(\beta)$:

$$\begin{aligned}\mathcal{H}(\beta) &= \left\{ \sum_{i=-N}^N \log[K_1^2 + |g_i|(\operatorname{sgn}(g_i)x_i - \beta)^2] \right. \\ &\quad \left. \dots + \sum_{j=1}^N \log[K_2^2 + |h_j|(\operatorname{sgn}(h_j)y_j - \beta)^2] \right\}. \quad (9)\end{aligned}$$

Bajo una revisión detallada de (9), se puede notar que tanto las muestras de entrada $\{x_i|_{i=-N}^N\}$ como las salidas previas $\{y_j|_{j=1}^N\}$ están multiplicadas por los signos de sus coeficientes respectivos. En principio, esta característica permite el uso de estas estructuras recursivas no lineales en la implementación de operaciones de filtrado de frecuencia selectiva [11], [12]. No obstante, a partir de esta especificación considere el

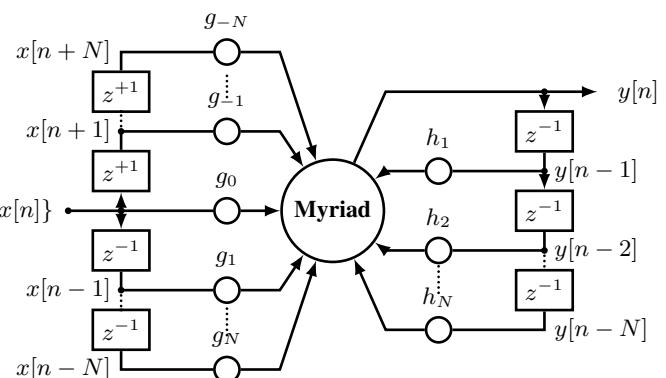


Figura 2. Estructura del Filtro Myriad Ponderado Recursivo

conjunto extendido de muestras multiplicadas por los signos de sus respectivos pesos, esto es,

$$\{s_m|_{m=1}^L\} = \{\operatorname{sgn}(g_{-N})x_{-N}, \dots, \operatorname{sgn}(g_N)x_N, \operatorname{sgn}(h_1)y_1, \dots, \operatorname{sgn}(h_N)y_N\}, \quad (10)$$

con $L = 3N + 1$. Además, permita denotar $\{s_{(m)}|_{m=1}^L\}$ como el orden estadístico del conjunto extendido de muestras multiplicadas por los signos de las ponderaciones, de modo que $s_{(1)} \leq s_{(2)} \leq \dots \leq s_{(L)}$, donde el elemento más pequeño y el más grande en el orden estadístico está dado por $s_{(1)} = \min\{s_m|_{m=1}^L\}$ y $s_{(L)} = \max\{s_m|_{m=1}^L\}$, respectivamente. En primera instancia, de la Ec. (9), se puede observar que el filtro RWMy tiene L parámetros independientes, que corresponden a las versiones normalizadas de las coeficientes del filtro, en otras palabras, $\left\{ \frac{g_{-N}}{K_1^2}, \dots, \frac{g_0}{K_1^2}, \dots, \frac{g_N}{K_1^2}, \frac{h_1}{K_2^2}, \dots, \frac{h_N}{K_2^2} \right\}$. Además, obsérve en la Ec. (8) que $\mathcal{H}(\beta) = \log(G(\beta))$, donde $G(\beta)$ es una función polinómica de β de orden $2L$ con derivadas bien definidas en todos sus órdenes. Por tanto, la derivada $\mathcal{G}'(\beta)$ es una función polinómica con no más que $(2L - 1)$ raíces reales y extremos locales. Por su parte, la derivada de la función costo $\mathcal{H}(\beta)$ está dada por:

$$\begin{aligned}\mathcal{H}'(\beta) &= \mathcal{G}'(\beta)/\mathcal{G}(\beta) = 2 \sum_{i=-N}^N \frac{|g_i|(\beta - \operatorname{sgn}(g_i)x_i)}{K_1^2 + |g_i|(\operatorname{sgn}(g_i)x_i - \beta)^2} \\ &\quad \dots + 2 \sum_{j=1}^N \frac{|h_j|(\beta - \operatorname{sgn}(h_j)y_j)}{K_2^2 + |h_j|(\operatorname{sgn}(h_j)y_j - \beta)^2}. \quad (11)\end{aligned}$$

Al revisar la Ec. (7), se puede notar que $\mathcal{G}(\beta) \neq 0$ para cualquier valor de β , por tanto, las raíces de la derivada de la función costo $\mathcal{H}'(\beta)$ y las raíces de $\mathcal{G}'(\beta)$ son idénticas. De esta forma, $\mathcal{H}'(\beta)$ tiene como máximo $(2L - 1)$ raíces reales y extremos locales, donde el valor de β que genera uno de los mínimos de la función costo corresponderá a la salida del filtro RWMy. Además, luego de una revisión en detalle de (11), se puede inferir que $\mathcal{H}'(\beta) \neq 0$ para $\beta < s_{(1)}$ y $\beta > s_{(L)}$. Por tanto, $\mathcal{H}'(\beta) = 0$, y en consecuencia la salida del filtro myriad ponderado recursivo, se alcanzará para valores de β dentro del rango de magnitudes delimitadas por el conjunto extendido de muestras signadas: $s_{(1)} \leq \hat{\beta}_{K_1, K_2} \leq s_{(L)}$.

El filtro RWMy tiene la particularidad de contar con dos parámetros sintonizables, K_1 y K_2 , que controlan la capacidad de rechazo del filtro ante diversas clases de ruido. De hecho, cuando $K_1, K_2 \rightarrow \infty$, el filtro RWMy se comporta como una versión normalizada del filtro IIR lineal, observándose un mejor desempeño cuando el ruido que contamina la señal de interés sigue un modelo estadístico gaussiano. A medida que disminuyen los valores de K_1 y K_2 , el filtro RWMy incrementa su robustez ante ruido de naturaleza impulsiva, y cuando $K_1, K_2 \rightarrow 0$, el filtro RWMy se comporta como un filtro de moda muestral recursivo.

A. Filtro Myriad Ponderado Recursivo Escalado

Una restricción del filtro RWMy es su incapacidad para amplificar las magnitudes de las muestras dentro de la ventana de observación, debido a que la salida del filtro está delimitada al intervalo $[s_{(1)}, s_{(L)}]$. Además, la respuesta del filtro depende de las salidas calculadas previamente, situación que puede generar un efecto de agrupamiento (*clustering*) alrededor de las salidas previas, y en consecuencia, inducir una atenuación adicional sobre la imagen filtrada. Con el objeto de superar esta limitación, Ramírez y Paredes proponen en [11] una versión escalada del filtro RWMy denominada *filtro recursivo myriad ponderado escalado (SRWMy, scaled recursive weighted myriad)* cuya definición viene dada por:

$$\tilde{\beta}_{K_1, K_2} = \left\{ \sum_{i=-N}^N |g_i| + \sum_{j=1}^N |h_j| \right\} \hat{\beta}_{K_1, K_2}, \quad (12)$$

donde $\hat{\beta}_{K_1, K_2}$ es la salida del filtro RWMy definida en (8). En consecuencia, la amplificación de la salida del filtro SRWMy se realiza simplemente incrementando la magnitud de las ponderaciones. Una vez la salida del filtro es amplificada, se retroalimenta en el operador myriad ponderado recursivo. Bajo una revisión detallada de (12), se puede observar que el término de escalamiento se reduce a la suma de las magnitudes de los coeficientes “desnormalizados” del filtro. Esto hace que la salida del filtro SRWMy dependa de $L+2$ parámetros, que corresponden a las L ponderaciones del filtro en conjunto con los parámetros de linealidad K_1 y K_2 .

B. Algoritmos Adaptativos

En esta sección, se describen los algoritmos adaptativos de diseño de las estructuras recursivas propuestas basadas en el operador myriad ponderado. Estos algoritmos buscan los mejores coeficientes de modo que un criterio de desempeño sea minimizado. Un criterio ampliamente usado en el diseño de filtros no lineales [19], [20] es el error medio absoluto (*MAE, Mean Absolute Error*). En otras palabras, considere un filtro recursivo myriad ponderado, cuyos coeficientes denotados por $\{g_i|_{i=-N}^N\}$ y $\{h_j|_{j=1}^N\}$ son las ponderaciones de las muestras de la imagen de entrada y los pesos de las salidas previas del filtro, respectivamente. El objetivo principal del algoritmo adaptativo consiste en encontrar los mejores coeficientes del filtro de manera que se minimice el error absoluto entre la imagen filtrada $y[n]$ y una imagen deseada $d[n]$. Por tanto,

bajo el criterio MAE, el objetivo es determinar los parámetros óptimos del filtro, tal que $J(g, h) = E\{|y[n] - d[n]|$ sea minimizado, donde $E\{\cdot\}$ denota el operador de esperanza matemática y $y[n]$ es la salida del filtro recursivo basado en el operador myriad ponderado.

Debido a que la minimización directa de la función costo $J(g, h)$ no conduce a una solución cerrada, se recurre al algoritmo de descenso escalonado (*steepest descent*) para determinar los coeficientes de interés. Este enfoque actualiza iterativamente los coeficientes del filtro en un intento por converger al mínimo global de la función costo $J(g, h)$. Fundamentalmente, los pesos del filtro son actualizados de modo iterativo de acuerdo a

$$\begin{aligned} g_i[n+1] &= g_i[n] + \mu \left[-\frac{\partial}{\partial g_i} J(g, h) \right], i = -N, \dots, N, \\ h_j[n+1] &= h_j[n] + \mu \left[-\frac{\partial}{\partial h_j} J(g, h) \right], j = 1, \dots, N, \end{aligned} \quad (13)$$

donde $g_i[n+1]$ y $h_j[n+1]$ son respectivamente, la actualización del peso no recursivo i -ésimo y el coeficiente recursivo j -ésimo, en la iteración $n+1$; μ es el tamaño de paso de la actualización y el gradiente de la función costo esta dado por

$$\begin{aligned} \frac{\partial}{\partial g_i} J(g, h) &= E \left\{ \text{sgn}(e[n]) \frac{\partial y}{\partial g_i} \right\}, i = -N, \dots, N, \\ \frac{\partial}{\partial h_j} J(g, h) &= E \left\{ \text{sgn}(e[n]) \frac{\partial y}{\partial h_j} \right\}, j = 1, \dots, N. \end{aligned} \quad (14)$$

En aplicaciones prácticas, no se cuenta con la información estadística de la salida del filtro, y por tanto, la esperanza matemática definida en (14) no puede ser evaluada. Esta restricción se resuelve recurriendo a estimaciones instantáneas del gradiente de la función costo, lo cual es una práctica muy común en el diseño de filtros adaptativos [20], [21], [22]. Así, los pesos del filtro son actualizados de la siguiente forma:

$$\begin{aligned} g_i[n+1] &= g_i[n] - \mu \left[\text{sgn}(e[n]) \frac{\partial y}{\partial g_i}[n] \right], i = -N, \dots, N, \\ h_j[n+1] &= h_j[n] - \mu \left[\text{sgn}(e[n]) \frac{\partial y}{\partial h_j}[n] \right], j = 1, \dots, N. \end{aligned} \quad (15)$$

Nótese que la estimación del gradiente requiere el cálculo de las derivadas $(\frac{\partial y}{\partial g_i}[n], \frac{\partial y}{\partial h_j}[n])$, en cada iteración. No obstante, la determinación de tales derivadas es una tarea extremadamente compleja debido a la retroalimentación presente en los filtros recursivos propuestos. Para solventar esta limitación, se sigue un enfoque denominado formulación de la ecuación de error [21], que asume las salidas previas del filtro $\{y[n-j]\}_{j=1}^N$ cercanas a las muestras de la imagen deseada $\{d[n-j]\}_{j=1}^N$. Por tanto, durante la etapa de determinación de los coeficientes, las salidas previas del filtro son reemplazadas por las componentes previas de la imagen deseada, es decir, $y[n-j] = d[n-j]$ para $j = 1, \dots, N$. La Figura 3 ilustra esquemáticamente la estructura de los algoritmos adaptativos bajo el enfoque de la formulación de la ecuación de error. El proceso para la determinación de las derivadas instantáneas

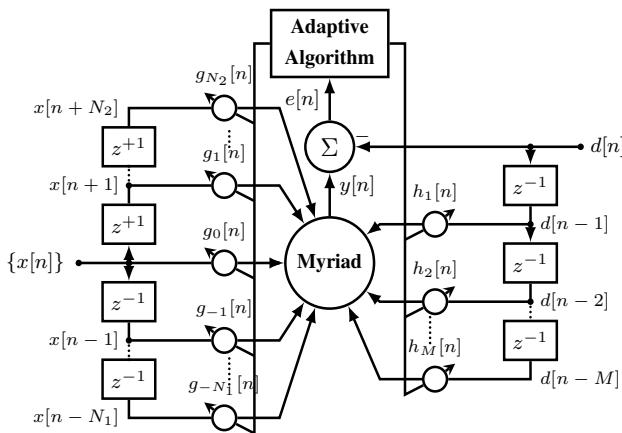


Figura 3. Estructura de los Algoritmos Adaptativos para el Cálculo de las Ponderaciones Óptimas de los Filtros Recursivos Propuestos

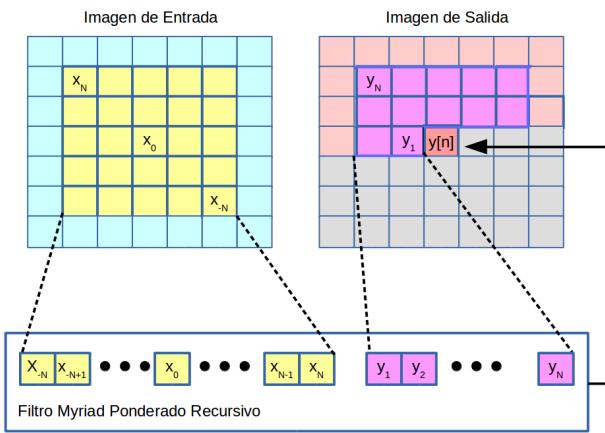


Figura 4. Esquema de Filtrado Recursivo en Imágenes

relacionadas con los filtros previamente descritos (el filtro RWMy y el filtro SRWMy) bajo el enfoque de la formulación de la ecuación de error, y en consecuencia las expresiones para la actualización iterativa de los parámetros de dichos filtros, se obtienen detalladamente en [11], [12].

C. Filtrado 2D Usando los Filtros Myriad Ponderados Recursivos

En este trabajo se adaptaron las estructuras de filtrado recursivas basadas en el operador myriad ponderado para la implementación de técnicas de filtrado 2D en imágenes médicas de MSCT y MRI. La Figura 4 muestra el esquema para la determinación la respuesta filtrada $y[n]$ usando una ventana deslizante de tamaño 5×5 . Como se puede observar en dicha figura, las muestras de entrada son obtenidas de un bloque de la imagen ruidosa de tamaño 5×5 , cuyo píxel central corresponde a la posición del píxel a determinar en la imagen de salida. Por su parte, las salidas previas son obtenidas de un conjunto de $\lfloor \frac{u^2}{2} \rfloor$ píxeles determinados previamente, donde $[u]$ es el número entero más grande menor a u . En

la Figura 4 se muestra el conjunto de salidas previas cuando la ventana de observación se desliza en dirección de las filas de la imagen. Finalmente, las muestras de entrada y las salidas previas son tomadas por el filtro recursivo myriad ponderado para la determinación de la salida del píxel $y[n]$.

V. RESULTADOS

Esta sección muestra los resultados de la aplicación de los filtros recursivos basados en el operador myriad ponderado en la remoción de ruido dependiente de las amplitudes de los píxeles, contaminación que comúnmente afecta a diversas modalidades de imágenes médicas. Específicamente, se evalúa el rendimiento de estas estructuras de filtrado recursivas en la remoción de ruido Poisson y ruido riciano, modelos de ruido comúnmente relacionados con bases de datos de MSCT y MRI, respectivamente.

A. Filtrado 2D de una Base de Datos Sintética de MSCT

Previo a la aplicación del proceso de filtrado, se determinan los coeficientes de los filtros recursivos con sus respectivos parámetros de linealidad. Para este fin, se implementan los algoritmos adaptativos descritos en la Sección IV-B, usando como imagen deseada $d[n]$ un bloque de 128×128 píxeles perteneciente a un corte axial de una base de datos sintética de MSCT. Específicamente, esta base de datos corresponde a un phantom numérico que consiste en un doble cono concéntrico libre de distorsiones, denominado *ground truth*. Además, los algoritmos adaptativos usan como imagen de entrada $x[n]$ una versión contaminada con ruido Poisson de la imagen deseada. Los parámetros K_1 , K_2 y μ_0 se seleccionan de modo los filtros diseñados que exhiban su mejor rendimiento.

A continuación, se implementa el proceso de filtrado 2D con las estructuras de filtrado recursivas previamente diseñadas sobre las 50 capas o secciones transversales que conforman la base de datos sintética de MSCT, donde cada capa consiste en una imagen de 256×256 píxeles y resolución de 12 bits. Las Figuras 5a y 5b muestran, respectivamente, la imagen sin ruido de una capa de la base de datos (Original), y la imagen de la misma capa contaminada con ruido Poisson (Noisy).

Con propósitos comparativos se implementan otras técnicas de filtrado 2D lineales y no lineales. En concreto, las Figuras 5c y 5d muestran, respectivamente, los resultados de aplicar el filtro de mediana ponderada (WMedian, *weighted median*) y el filtro de promedio ponderado (WMean, *weighted mean*) a la imagen correspondiente, donde los coeficientes de estos filtros se obtuvieron mediante la implementación del algoritmo de diseño desarrollado en [19]. Adicionalmente, la imagen filtrada por la técnica basada en la transformada de estabilización de la varianza (VST, *variance stabilization transform*), así como también la respuesta entregada por la técnica de umbralización dura (HT, *hard thresholding*) se muestran en las Figuras 5e y 5f, respectivamente [7].

Las imágenes resultantes de implementar el proceso de filtrado 2D usando el filtro RWMy y el filtro SRWMy se muestran en las Figuras 5g y 5h, respectivamente. Observe

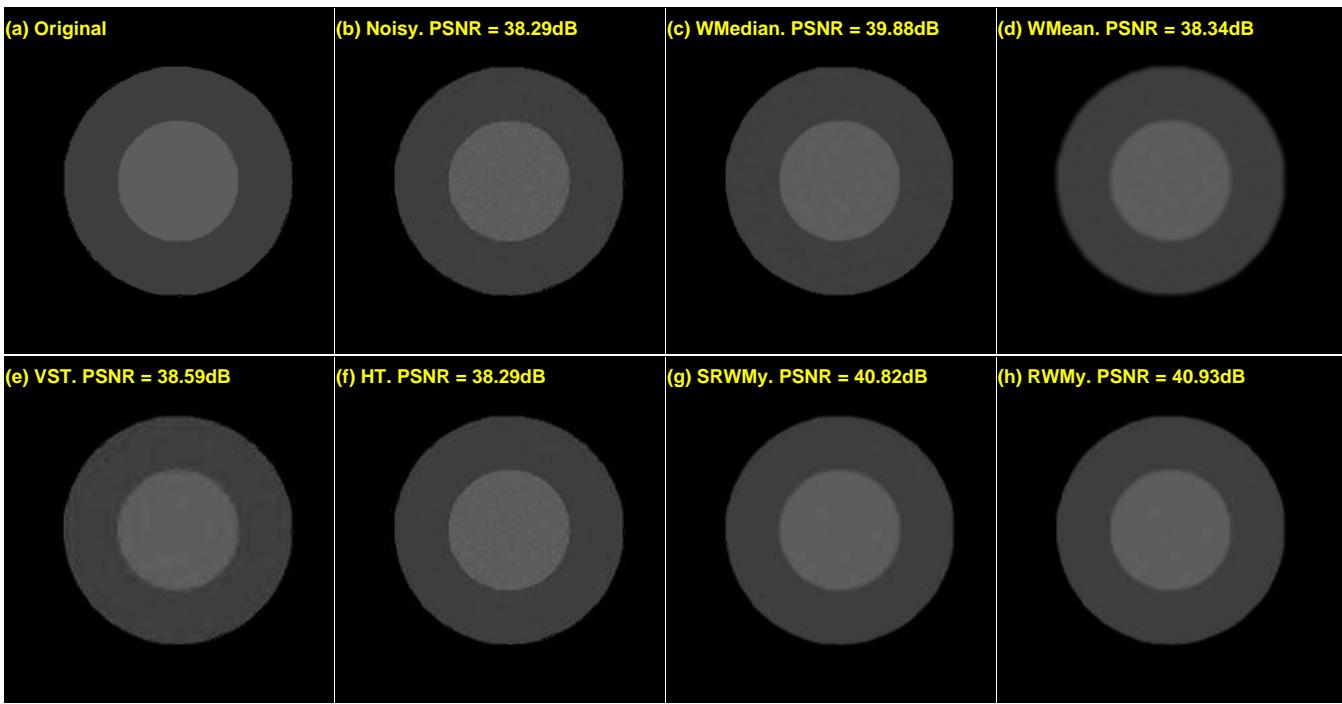


Figura 5. (a) Imagen original y (b) Imagen Contaminada con Ruido Poisson. Imágenes Filtradas con sus Respectivos Valores de PSNR en Decibeles Generados por (c) El Filtro WMedian, (d) El Filtro WMean, (e) La Transformada VST, (f) La Técnica HT, (g) El Filtro SRWMy y (h) El Filtro RWMy

que tanto la imagen contaminada como las imágenes filtradas tienen incorporados los valores de PSNR en decibeles (dB). A partir de una inspección visual de la Figura 5, puede observarse que el filtro WMean induce un efecto de difuminado que degrada el contraste en los bordes de la imagen. Así mismo, puede apreciarse que las imágenes filtradas usando los RWMy y SRWMy mimimizan el efecto del ruido Poisson preservando a su vez el contraste de los bordes de la imagen. Es de hacer notar que los filtros RWMy y SRWMy generan los mejores resultados de PSNR, obteniéndose una ganancia de desempeño de aproximadamente 2 dB con respecto a las otras técnicas de filtrado 2D.

Por su parte, la Tabla I muestra los valores promedio de diversas métricas de calidad de la imagen calculados a partir de la aplicación de las diferentes técnicas de filtrado sobre las 50 capas de la base de datos sintética de MSCT. Específicamente, cada valor de esta tabla se obtiene de aplicar el promedio sobre los resultados generados por las 50 imágenes filtradas. Es importante mencionar que las métricas EME y EMEE se determinan usando un bloque de tamaño 5×5 píxeles y estableciendo el parámetro $\alpha = 0.10$. Cabe destacar que, en la Tabla I, el mejor rendimiento para las distintas métricas se resalta en negrita y subrayado, mientras que el segundo mejor rendimiento se presenta sólo en negrita.

Como puede observarse en la Tabla I, el filtro RWMy ofrece el mejor rendimiento bajo las métricas de calidad de la imagen referenciadas (MSE y PSNR) y toma un segundo lugar en la métrica MAE siendo el filtro de mediana ponderada mejor en esta medida. Adicionalmente, el filtro SRWMy exhibe el segundo mejor desempeño en las métricas implementadas

Tabla I
VALORES PROMEDIO DE DISTINTAS MÉTRICAS DE CALIDAD DE LA IMAGEN OBTENIDOS DEL FILTRADO 2D DE LA BASE DE DATOS DE SINTÉTICA DE MSCT CONTAMINADA CON RUIDO POISSON

FILTRO	MAE	MSE	PSNR[dB]	EME	EMEE
WMean	12.27	2,560.31	38.35	1.42	0.01
WMedian	<u>7.16</u>	1,887.85	39.71	0.43	0.00
VST	11.91	2,537.30	38.43	6.12	0.04
HT	15.56	2,823.14	38.09	4.76	0.03
RWMy	<u>7.92</u>	<u>1,473.10</u>	<u>40.79</u>	<u>112.36</u>	<u>24.88</u>
SRWMy	8.04	<u>1,513.34</u>	<u>40.67</u>	<u>111.43</u>	<u>24.35</u>

exceptuando la métrica MAE, por tanto, las estructuras de filtrado recursivas basadas en el operador myriad ponderado pueden considerarse una alternativa eficiente para la remoción de ruido Poisson. Por otro lado, bajo las métricas EME y EMEE, los filtros RWMy y SRWMy muestran un desempeño ostensiblemente mejor comparado con las otras técnicas de filtrado. Esto indica que los filtros RWMy y SRWMy, a parte de minimizar los efectos del ruido Poisson, generan imágenes con estructuras mucho más uniformes con respecto a las imágenes procesadas por las otras técnicas de filtrado.

B. Filtrado 2D de un Volumen Real de MSCT

Posteriormente, se implementan diversas técnicas de filtrado 2D sobre una base de datos real de MSCT. En particular, la base de datos consta de 261 capas que contienen información 3D de la anatomía del corazón en un instante específico del ciclo cardíaco, donde cada sección transversal consiste en una imagen de 512×512 píxeles y resolución de 12 bits. Además, se utilizan los filtros diseñados con la base de datos sintética

Tabla II
 MEDIDAS DE CALIDAD DE LAS IMÁGENES OBTENIDAS DEL FILTRADO 2D
 SOBRE A LA BASE DE DATOS REAL DE MSCT

FILTRO	EME	EMEE
WMean	1.40 ± 0.09	0.01 ± 0.00
WMedian	0.83 ± 0.10	0.00 ± 0.00
VST	2.89 ± 0.14	0.02 ± 0.00
HT	2.39 ± 0.11	0.02 ± 0.00
RWMy	66.37 ± 0.08	76.73 ± 0.78
SRWMy	65.99 ± 0.08	74.03 ± 0.74

debido a que las proyecciones reales no cuentan con imágenes de referencia para la fase de entrenamiento.

Las técnicas de filtrado 2D se aplican sobre las secciones transversales de la base de datos real usando una ventana de observación de tamaño 5×5 píxeles. La Figura 6a muestra una capa de la base de datos real con su respectivo valor de EME. Por su parte, las Figuras 6b-g corresponden con las imágenes filtradas —con sus respectivos valores de la métrica EME— usando el filtro WMedian, el filtro WMean, la transformada VST, la técnica HT, el filtro SRWMy y el filtro RWMy. Observe que los filtros RWMy y SRWMy generan valores mucho mayores de la métrica EME con respecto a los valores entregados por las otras técnicas de filtrado, indicando que, bajo la perspectiva de estas métricas ciegas, los filtros recursivos basados en el operador myriad ponderado ofrecen imágenes con estructuras más uniformes con contrastes realzados.

Adicionalmente, con el objeto de evaluar cuantitativamente el rendimiento de los filtros, la Tabla II presenta los valores promedio (con sus respectivos valores de desviación estándar) de las métricas EME y EMEE. En particular, cada valor de esta tabla se obtiene de aplicar el promedio (o la desviación estándar) sobre los resultados generados por las 261 imágenes filtradas. Como puede observarse en la Tabla II, los valores más altos de las métricas EME y EMEE se obtienen con los filtros RWMy y SRWMy, los cuales se diferencian de manera significativa con respecto a los valores producidos por las técnicas de filtrado restantes. En consecuencia, los filtros RWMy y SRWMy exhiben un desempeño superior bajo estas métricas ciegas, que valoran tanto la uniformidad de las estructuras como el realce del contraste.

C. Filtrado 2D de una Base de Datos Sintética de MRI

Por otra parte, se ejecuta el proceso de entrenamiento de los filtros myriad ponderados recursivos para la remoción de ruido rítmico. En particular, se determinan los parámetros de los filtros implementando los algoritmos adaptativos descritos en IV-B, donde la imagen deseada $d[n]$ se obtiene de un bloque de tamaño 91×91 perteneciente a la sección transversal de una base de datos de sintética de MRI disponible en [23]. Así mismo, la imagen de entrada del algoritmo adaptativo se toma la misma región de la imagen deseada pero en su versión contaminada con ruido rítmico. Además, los valores de los parámetros de linealidad K_1 y K_2 y del tamaño de paso μ_0 se establecen de la misma forma a la realizada en el diseño de los filtros para la remoción de ruido Poisson en imágenes de MSCT.

Los filtros entrenados se implementan para el filtrado 2D de la base de datos real de MRI. Específicamente, se aplica el proceso de filtrado 2D sobre las 181 capas que pertenecen a la base de datos, donde cada capa consiste en una imagen de 181×217 píxeles y resolución de 12 bits. Es importante mencionar que tanto el entrenamiento como el proceso de filtrado 2D se implementaron usando las bases de datos contaminadas con distintos niveles de ruido rítmico (3 %, 5 %, 7 % y 9 %) con ventanas de observación de tamaño 3×3 y 5×5 píxeles.

En la Figura 7a, se visualiza la imagen sin ruido de una sección transversal de la base de datos sintética de MRI, donde la Figura 7b incluye una versión contaminada con ruido rítmico de 3 % de la misma sección transversal. Las respuestas del filtrado 2D con ventanas deslizantes de 3×3 píxeles usando el filtro WMedian, el filtro WMean, el filtro RWMy y el filtro SRWMy se muestran en las Figuras 7c-f, respectivamente. Nuevamente, los coeficientes de los filtros WMedian y WMean se obtuvieron mediante la implementación del algoritmo de entrenamiento reportado en [19]. Nótese que la imagen con mayor valor PSNR corresponde a la generada por el filtro SRWMy con una ganancia de rendimiento de más de 5 dB, donde, bajo una inspección visual de los resultados generados por este filtro, se observa la conservación de los bordes y la mejora en la definición de las estructuras que conforman la imagen. Además, el rendimiento superior del filtro SRWMy con respecto al filtro RWMy se atribuye principalmente a que la versión escalada amplifica su salida para ajustarse al rango dinámico de píxeles de la imagen.

Como se mencionó anteriormente, tanto la etapa de entrenamiento como el proceso de filtrado 2D se aplicó sobre bases de datos de MRI contaminadas a distintos niveles de ruido rítmico. Específicamente, se usaron las bases de datos contaminadas a 3 %, 5 %, 7 % y 9 %. La Figura 8 muestra, respectivamente, los valores promedio de (a) MSE, (b) MAE, (c) PSNR y (d) EME, entregados por distintas técnicas de filtrado 2D a diferentes niveles de ruido rítmico usando ventanas deslizantes de tamaño 3×3 . Más precisamente, las técnicas de filtrado consideradas en esta evaluación son el filtro WMedian, el filtro WMean, el filtro RWMy y el filtro SRWMy. Además, cada punto de las curvas se obtiene de aplicar el promedio a los resultados generados por las 181 imágenes filtradas usando la base de datos contaminada al nivel de ruido rítmico correspondiente.

Como puede observarse en la Figura 8, la versión escalada del filtro myriad ponderado recursivo exhibe un rendimiento notablemente superior comparado con el desempeño mostrado por los otros filtros bajo estudio en el contexto de las métricas de calidad de la imagen con referencia (el MSE, el MAE y el PSNR[dB]), donde, bajo la métrica PSNR, el filtro SRWMy exhibe una ganancia de aproximadamente 4 dB con respecto a los otros filtros 2D para el intervalo mostrado de niveles de ruido rítmico. Por consiguiente, el filtro SRWMy tiene un comportamiento notablemente superior en la remoción de ruido rítmico con respecto a las otras técnicas de filtrado 2D bajo estudio, minimizando el efecto de la contaminación

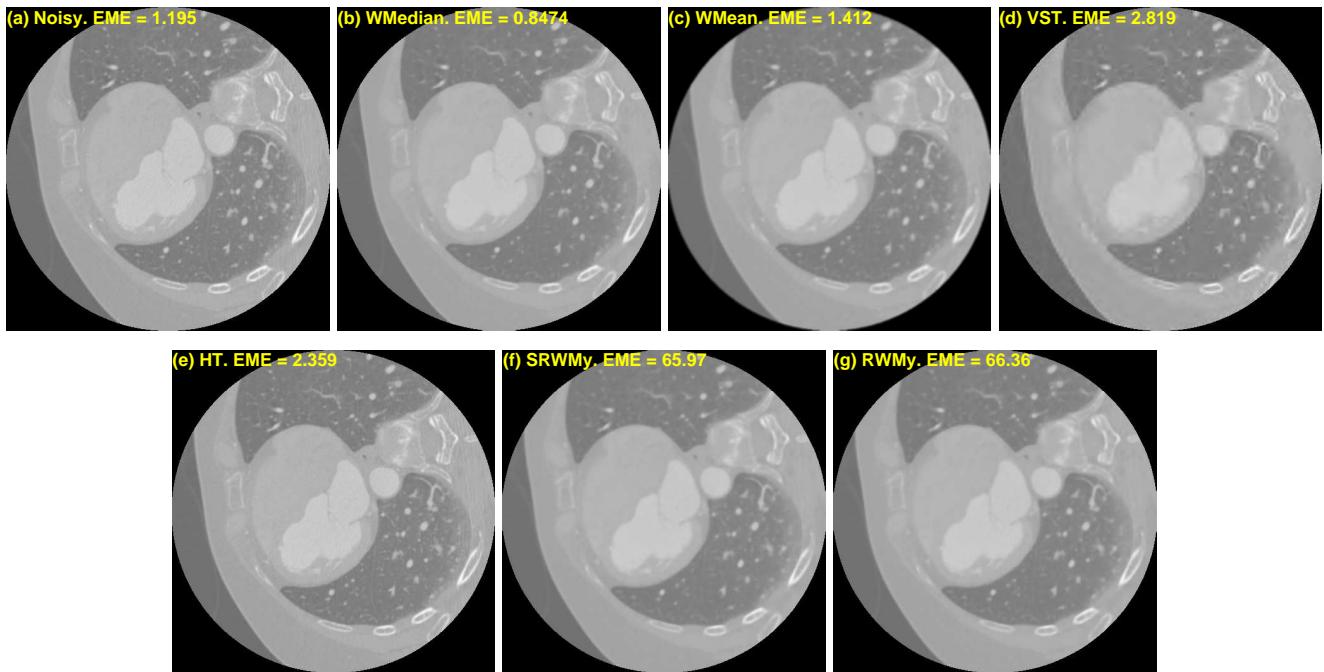


Figura 6. (a) Imagen Original de la Base de Datos Real de MSCT con su Respectivo Valor de EME. Imágenes Filtradas con su Respectivo Valor de EME usando (b) El Filtro WMedian, (c) El Filtro WMean, (d) La Transformada VST, (e) La Técnica HT, (f) El Filtro SRWMy y (g) El Filtro RWMy

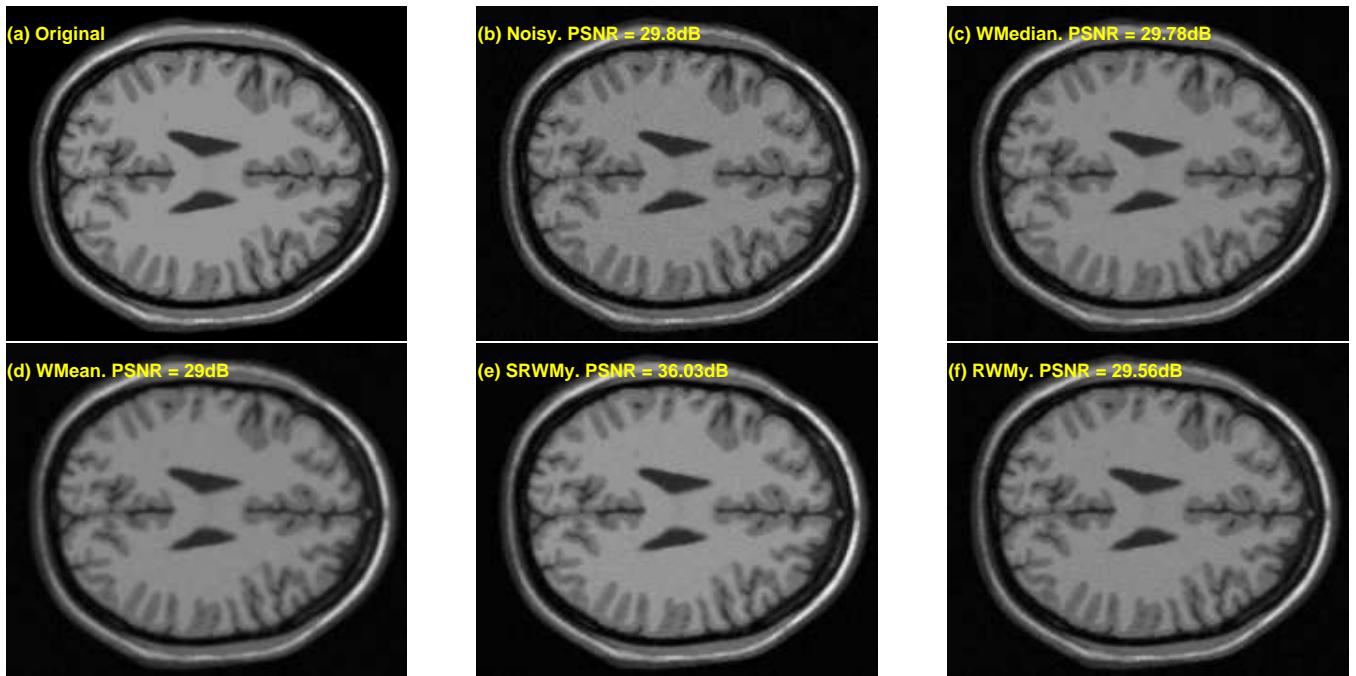


Figura 7. (a) Imagen Original y (b) Imagen Contaminada con Ruido Ricciano de 3 %. Imágenes Filtradas con su Respectivo Valor de PSNR en Decibeles Utilizando una Ventana de Observación 3×3 , Generadas por (c) El Filtro de Mediana Ponderada, (d) El Filtro de Promedio Ponderado, (e) El Filtro Myriad Ponderado Recursivo Escalado y (f) El Filtro Myriad Ponderado Recursivo

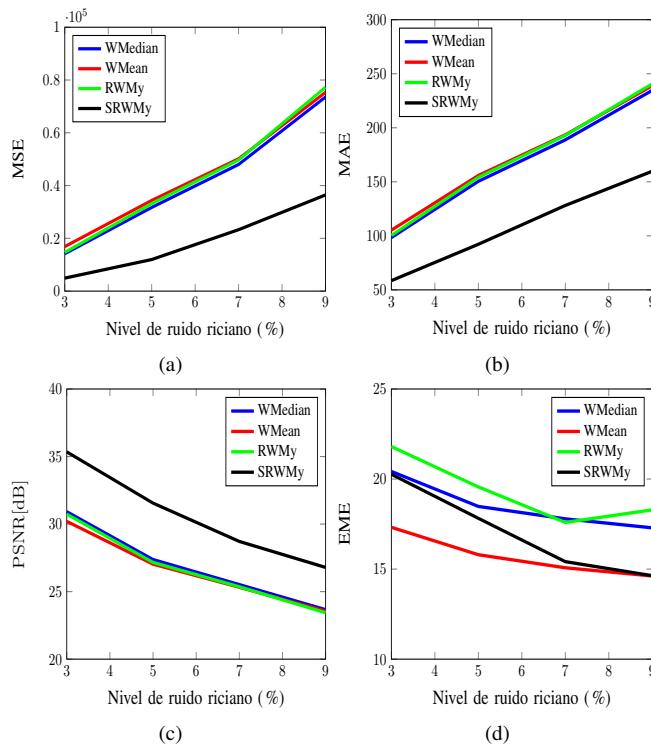


Figura 8. Valores Promedio de (a) MSE, (b) MAE, (c) PSNR y (d) EME. Generados por las Diversas Técnicas de Filtrado 2D a Diferentes Niveles de Ruido Riciano Usando Ventanas Deslizantes de 3×3

y ajustando su salida al rango dinámico de magnitudes de los píxeles de la imagen sin ruido. Por su parte, la versión normalizada del filtro myriad ponderado recursivo tiene un rendimiento competitivo con respecto a las otras técnicas de filtrado 2D. Este comportamiento se atribuye principalmente al hecho que este filtro no tiene la capacidad de amplificar su salida para ajustarse al rango de magnitudes de los píxeles de la imagen original.

Por otro lado, en el marco de la métrica EME, la Figura 8 muestra un rendimiento ligeramente superior del filtro RWMy para casi todo el intervalo de niveles de ruido bajo estudio. Esto último indica que la versión normalizada del filtro myriad ponderado recursivo minimiza efectivamente el efecto del ruido riciano, entregando estructuras con mayor uniformidad y mayor contraste. A diferencia de los resultados obtenidos con la base de datos real de MSCT, los valores de EME generados por los filtros RWMy y SRWMy no muestran una ganancia significativa con respecto a las otras técnicas de filtrado 2D debido a que las estructuras de las imágenes usadas de resonancia magnética no presentan regiones tan regulares como las exhibidas por las imágenes cardíacas de MSCT.

VI. CONCLUSIONES

En el presente trabajo se implementan técnicas de filtrado 2D basadas en los filtros myriad ponderados recursivos. En particular, se implementan estas técnicas de filtrado no lineal para la remoción de ruido dependiente de las amplitudes de los

píxeles, comúnmente presente en imágenes médicas de MSCT y MRI. Inicialmente, los filtros myriad ponderados recursivos se diseñaron a partir de la implementación de los algoritmos de optimización adaptativos correspondientes, donde la señal contaminada de ruido Poisson y la señal deseada se obtienen de una sección transversal de una base de datos sintética de MSCT. Además, las estructuras recursivas diseñadas se utilizaron en el filtrado 2D de todas las capas de la base de datos sintética de MSCT. Bajo diferentes métricas de calidad de la imagen, los filtros RWMy y SRWMy ofrecen los mejores rendimientos comparados con aquellos entregados por otras técnicas de filtrado 2D.

A continuación, los filtros entrenados con la base de datos sintética de MSCT se aplicaron para el filtrado 2D de una base de datos real de MSCT. Dado que no se dispone de una imagen de referencia, las imágenes filtradas se evaluaron bajo los operadores EME y EMEE, los cuales pertenecen a la categoría de métricas de calidad de la imagen sin referencia, donde el desempeño de los filtros RWMy y SRWMy superan notablemente el rendimiento de otras técnicas de filtrado 2D. En particular, se observa que los filtros myriad ponderados recursivos ofrecen su mejor rendimiento sobre imágenes con regiones regulares y uniformes, tales como las exhibidas por las imágenes cardíacas de MSCT.

Finalmente, los filtros RWMy y SRWMy se implementaron para la remoción de distintos niveles de ruido riciano que afecta a las imágenes de MRI. Más precisamente, para cada nivel de ruido riciano, se diseñaron los filtros myriad ponderados recursivos usando una sección transversal de una base de datos simulada de MRI. Adicionalmente, para cada nivel de ruido riciano, los filtros entrenados se implementaron en el filtrado 2D de todas las capas de la base de datos simulada de MRI, donde el filtro SRWMy mostró un rendimiento superior con respecto a las otras técnicas de filtrado bajo las métricas de calidad de la imagen con referencia (MSE, MAE y PSNR). Esto indica que el filtro SRWMy ofrece mayor eficiencia en la minimización de los efectos del ruido riciano, generando una imagen más cercana a la imagen original sin ruido.

REFERENCIAS

- [1] L. W. Goldman, "Principles of CT: Multislice CT," *Journal of nuclear medicine technology*, vol. 36, no. 2, pp. 57–68, 2008.
- [2] A. Bravo, J. Clemente, M. A. Vera, J. Avila, and R. Medina, "A Hybrid Boundary-Region Left Ventricle Segmentation in Computed Tomography," in *VISAPP* (2), 2010, pp. 107–114.
- [3] E. Velázquez, A. Bravo, and M. Vera, "Segmentación del Ventrículo Izquierdo en Imágenes Cardíacas Usando Técnicas No Paramétricas," in *Memorias del V Congreso Venezolano de Bioingeniería*, 2015, pp. 98–101.
- [4] P. Suetens, *Fundamentals of Medical Imaging*. Cambridge University Press, 2009.
- [5] B. M. Dale, M. A. Brown, and R. C. Semelka, *MRI: Basic Principles and Applications*. John Wiley & Sons, 2015.
- [6] P. Gravel, G. Beaudoin, and J. A. D. Guise, "A Method for Modeling Noise in Medical Images," *IEEE Transactions on Medical Imaging*, vol. 23, no. 10, pp. 1221–1232, October 2004.
- [7] A. N. P. Raj and T. Venkateswarlu, "Denoising of Poisson and Rician Noise from Medical Images using Variance Stabilization and Multiscale Transforms," *Computer Applications*, vol. 57, no. 21, pp. 24–31, November 2012.

- [8] I. Rodrigues, J. Sanches, and J. Bioucas-Dias, "Denoising of Medical Images Corrupted by Poisson Noise," in *15th IEEE International Conference on Image Processing, 2008. ICIP 2008*, 2008, pp. 1756–1759.
- [9] M. Makitalo and A. Foi, "Optimal Inversion of the Generalized Anscombe Transformation for Poisson-Gaussian Noise," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 91–103, January 2013.
- [10] V. P. Raj and T. Venkateswarlu, "Denoising of Magnetic Resonance and X-ray Images using Variance Stabilization and Patch Based Algorithms," *The International Journal of Multimedia & Its Applications*, vol. 4, no. 6, pp. 53–71, 2012.
- [11] J. M. Ramirez and J. L. Paredes, "Recursive Weighted Myriad Based Filters and their Optimizations," *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 4027–4039, August 2016.
- [12] J. M. Ramírez, "Técnicas Robustas de Tratamiento de Señales Basadas en Métodos de Estimación Estadística," Ph.D. dissertation, Universidad de Los Andes, Mérida, Venezuela, 2016.
- [13] N. Wiest-Daesslé, S. Prima, P. Coupé, S. Morrissey, and C. Barillot, "Rician Noise Removal by Non-local Means Filtering for Low Signal-to-Noise Ratio MRI: Applications to DT-MRI," *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2008*, pp. 171–179, 2008.
- [14] S. K. Nisha, "Image Quality Assessment Techniques," *Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 7, pp. 636–640, July 2013.
- [15] R. G. Jaya V. L, "IEM: A New Image Enhancement Metric for Contrast and Sharpness Measurements," *Computer Applications*, vol. 79, no. 9, pp. 1 – 8, October 2013.
- [16] P. Prajapati, Z. Narmawala, N. P. Darji, S. M. Moorthi, and R. Ramakrishnan, "Evaluation of Perceptual Contrast and Sharpness Measures for Meteorological Satellite Images," *Procedia Computer Science*, vol. 57, pp. 17 – 24, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).
- [17] S. S. Agaian, K. Panetta, and A. M. Grigoryan, "A New Measure of Image Enhancement," in *IASTED International Conference on Signal Processing & Communication*, 2000, pp. 19–22.
- [18] S. Gupta and R. Porwal, "Appropriate Contrast Enhancement Measures for Brain and Breast Cancer Images," *Biomedical Imaging*, vol. 2016, p. 8, March 2016.
- [19] G. R. Arce, "A General Weighted Median Filter Structure Admitting Negative Weights," *IEEE Transactions on Signal Processing*, vol. 46, no. 12, pp. 3195–3205, December 1998.
- [20] G. R. Arce and J. L. Paredes, "Recursive Weighted Median Filters Admitting Negative Weights and their Optimization," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 768–779, March 2000.
- [21] J. J. Shynk, "Adaptive IIR Filtering," *IEEE ASSP Magazine*, vol. 6, no. 2, pp. 4–21, April 1989.
- [22] S. Kalluri and G. R. Arce, "Robust Frequency-Selective Filtering using Weighted Myriad Filters Admitting Real-Valued Weights," *IEEE Transactions on Signal Processing*, vol. 49, no. 11, pp. 2721–2733, November 2001.
- [23] "Brainweb: Simulated Brain Database." <http://www.bic.mni.mcgill.ca>

Lineamientos para el Despliegue de Redes SDN/OpenFlow

Gustavo Pereira¹ y Eric Gamess^{1,2}

¹ Escuela de Computación, Universidad Central de Venezuela
Caracas, Venezuela

² Department of Mathematical, Computing, and Information Sciences, Jacksonville State University
Jacksonville, Alabama, USA
gustavo.pereira.salas@gmail.com, egamess@jsu.edu

Resumen— SDN [1] (Software Defined Networking) es una arquitectura de red emergente que separa el plano de control del plano de datos de los dispositivos de la red y coloca el plano de control en uno o varios servidores de control capaces de gestionar las reglas de reenvío de tráfico de todos los dispositivos de comunicación bajo su dominio. Este cambio de paradigma ofrece grandes beneficios en comparación con los métodos de red tradicionales debido a que el aprovisionamiento de políticas y servicios se efectúa en una sola entidad central, simplificando la administración, reduciendo los gastos de operación, acelerando el ciclo de innovación de nuevas tecnologías y ofreciendo la posibilidad de programar el comportamiento de la red. Esta investigación describe los principales componentes de una arquitectura SDN incluyendo componentes de hardware, software y protocolos y las consideraciones de diseño para el despliegue de redes SDN en el ámbito de campus empresariales. Las redes de campus empresariales están delimitadas a un conjunto de edificios o pisos de una edificación interconectados mediante redes Ethernet.

Palabras Clave—SDN, OpenFlow, Controladores SDN, Modelos de Despliegue SDN, Lineamientos Técnicos y Estratégicos, Fases para el Despliegue de Red SDN.

I. INTRODUCCIÓN

Un entorno de campus empresarial está conformado por un conjunto de edificios interconectados a través de medios de transmisión guiados y no guiados que enlazan los dispositivos de comunicación de la red y permiten el transporte del tráfico de los usuarios. En un campus empresarial tradicional el plano de control se encuentra distribuido en todos los dispositivos de la red y requiere la coordinación de los mismos para decidir cómo tratar los paquetes que ingresan a sus puertos. Esta naturaleza distribuida del plano de control obliga a los administradores de la red a configurar todos los dispositivos de red cada vez que se incorpora una nueva aplicación o política a la red, ralentizando el aprovisionamiento y la escalabilidad de servicios en ambientes con gran cantidad de dispositivos. Esta limitación ha impulsado a la comunidad científica a buscar nuevos enfoques y arquitecturas de redes alternativas que mejoren la agilidad, la flexibilidad y la escalabilidad en las redes de campus empresariales.

En un entorno SDN uno o varios controladores se encargan de operar y gestionar el tráfico de la red desde un nodo lógico central. El controlador SDN instruye a los dispositivos SDN las reglas que se deben aplicar a los paquetes que circulan en la red en base a las políticas de una organización.

El despliegue de redes SDN requiere la aplicación de las mejores prácticas y principios de diseño que satisfagan los requerimientos de diseño de las organizaciones. Las mejores prácticas resultan de experiencias propias y de una investigación documental que abarca las siguientes categorías: (1) artículos académicos que describen la arquitectura de red SDN y el protocolo de red OpenFlow, (2) documentación de implementaciones SDN en campus empresariales y (3) evaluaciones presentadas en artículos y notas técnicas.

II. TRABAJOS RELACIONADOS

Los estudios relacionados son de carácter general y se encuentran algunos casos de estudio de implementaciones SDN en campus empresariales. La Universidad de Stanford [2][3] efectuó en el año 2010 una migración de parte de su red a una red SDN OpenFlow. El estudio de la migración a SDN en la Universidad de Stanford se divide en dos secciones: la primera sección muestra la red de arranque, las premisas de diseño y las herramientas de monitoreo utilizadas durante la migración. La segunda sección describe las fases de migración en la red de producción de los edificios considerados en la implementación.

Skorupa y Fabbi [4] describen los modelos de despliegue SDN existentes dependiendo de las características de la red de una organización.

La selección del tipo de controladores SDN y los dispositivos SDN para un despliegue se obtiene a partir de la revisión de las especificaciones de los productos abiertos y de carácter comercial disponibles por sus desarrolladores.

Los aspectos generales de diseño se encuentran en publicaciones y estudios [5][6] y textos de referencia [7][8] incluyendo principios de escalabilidad, alta disponibilidad, gestión y seguridad.

III. SDN

SDN [9] es un nuevo enfoque en la programación de redes que consiste en la capacidad de inicializar, controlar, cambiar y gestionar el comportamiento de reenvío del tráfico de una red mediante APIs abiertas. En una red SDN se separan los planos de control y datos de los dispositivos de red y se desplaza el plano de control a una unidad central llamada controlador SDN. Un controlador SDN se encarga de definir y comunicar las reglas de reenvío de tráfico a los dispositivos SDN y abstraer la infraestructura de red y su topología a las aplicaciones. Bajo este modelo las aplicaciones consideran a la red como un solo switch lógico central que provee servicios de

conectividad a los usuarios y a las aplicaciones. Al separar los planos de datos y control, los switches de la red se convierten en dispositivos de reenvío simples y la lógica de control se implementa en un sistema operativo de red centralizado o NOS (Network Operating System). Con este nuevo enfoque se obtienen grandes beneficios: Aprovisionamiento de políticas simple, agilidad para reconfiguración e implementación de nuevos servicios y aceleración en la innovación de las redes [10].

A. Arquitectura de Red SDN

Una red SDN está conformada generalmente por tres capas: Capa de Aplicación, Capa de Plano de Control y Capa de Plano de Datos (ver Figura 1 tomada de [11]).

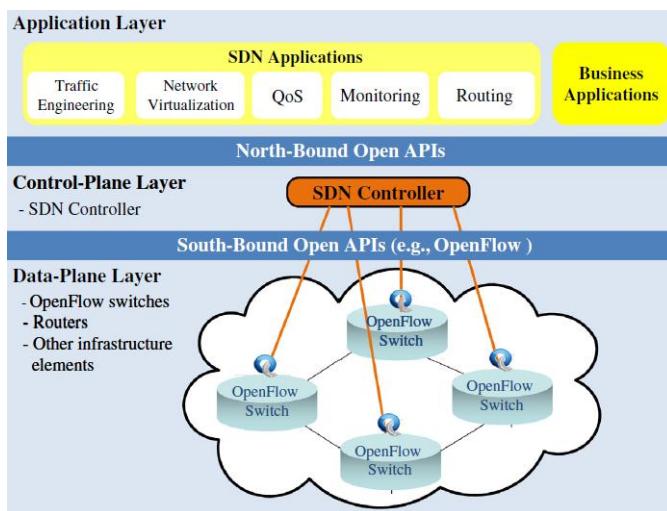


Figura 1. Representación Lógica de una Arquitectura de Red SDN

Capa de Aplicación: Está conformada por las aplicaciones del negocio y por las aplicaciones de servicios de red.

Capa del Plano de Control: Contiene los controladores SDN encargados de gobernar y dirigir la manera en que se transportan los datos en los dispositivos SDN. Se encuentra a cargo de todas las funciones complejas de enrutamiento, manejo de políticas, monitoreo y chequeos de seguridad de la red.

Capa del Plano de Datos: Está conformada por dispositivos SDN físicos y virtuales encargados de transportar datos en base a instrucciones recibidas por los controladores SDN de la red.

En una red SDN los controladores se comunican con las aplicaciones externas mediante APIs (Application Program Interface) Northbound abiertas y con los dispositivos SDN mediante APIs Southbound abiertas incluyendo el protocolo OpenFlow.

Una red SDN también puede ser vista como una composición de capas y sistemas (ver Figura 2 tomada de [12]). Algunas capas se encuentran presentes en todos los despliegues SDN incluyendo: controladores, infraestructura de red y APIs Southbound, mientras que otras son opcionales incluyendo APIs Northbound, aplicaciones, hipervisores y virtualización de redes.

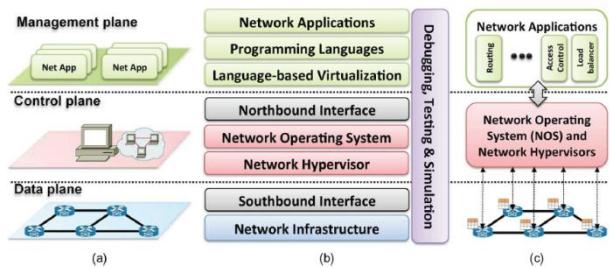


Figura 2. Arquitectura SDN basada en (a) Planos, (b) Capas y (c) Sistemas

Comenzando de abajo hacia arriba, se identifican los siguientes sistemas, planos y capas:

- Infraestructura de red: Está conformada por los dispositivos SDN físicos o virtuales que manipulan y reenvían paquetes en base a reglas definidas por un controlador. Un dispositivo de red contiene un plano de datos (Forwarding Plane) encargado de transportar y manipular campos de cabecera de paquetes y un plano operacional (Operational Plane) encargado de efectuar tareas administrativas relacionadas con su funcionamiento. Los dispositivos SDN pueden ser switches, routers o elementos de reenvío físicos o virtuales que soportan planos de datos y Southbound SDN, como OpenFlow [2].
- APIs Southbounds: Constituyen el API de comunicación que facilita la comunicación entre controladores y dispositivos SDN. Estas APIs pueden ser abiertas o propietarias. El controlador SDN cuenta con las siguientes opciones de APIs Southbound abiertas: OpenFlow [2], OVSDB [13] (Open vSwitch Database) y ForCES [14] (Forwarding and Control Element Separation). También existen las siguientes opciones de plugins: BGP [15] (Border Gateway Protocol), SNMP [16] (Simple Network Management Protocol) y NETCONF [17] (Network Configuration Protocol) entre otros. OpenFlow es el API Southbound estándar de la industria de las redes para entornos SDN. A través de OpenFlow un controlador puede crear, actualizar, modificar y eliminar entradas en las tablas de flujos de los dispositivos SDN y obtener información de estadísticas de estos.
- Plano de datos (Forwarding Plane): El plano de datos se encuentra en los dispositivos de red y se encarga de la manipulación y del reenvío de los paquetes e incluye, pero no está limitado a filtros, medidores, marcadores, y clasificadores.
- Hipervisor de Red: En un ambiente de virtualización de servidores, un hipervisor es una plataforma de software que permite correr varias VMs sobre un mismo dispositivo de cómputo. Un hipervisor cuenta con switches virtuales, o vSwitches, que permiten la comunicación entre VMs. En una red SDN basada en vSwitches, un controlador le comunica a los vSwitches las reglas de reenvío de tráfico asociadas a la comunicación entre las VMs. Los vSwitches disponen de mecanismos de túneles para comunicarse con VMs hospedadas en otros hipervisores externos.

- Controlador SDN o NOS (Network Operating System): Es un software que corre en un servidor de red para gestionar las reglas de reenvío de tráfico de los dispositivos SDN. Un controlador SDN posee en su núcleo abstracciones, servicios esenciales y APIs comunes para comunicarse con el resto de los elementos de la arquitectura (ver Figura 3 tomada de [12]). Entre los principales servicios de red ofrecidos se encuentran: (1) gestión de la topología, encargado de descubrir la topología de la red, (2) gestión de estadísticas, encargado de recopilar información de contadores del tráfico de la red, (3) gestión de notificaciones, encargado de gestionar la comunicación del plano de control con los elementos de la red, (4) gestión de dispositivos, encargado de configurar y gestionar los elementos de la infraestructura de red, (5) reenvío de caminos más cortos, encargado de seleccionar los mejores caminos hacia los destinos y (6) mecanismos de seguridad, encargado de proveer mecanismos de protección a la red.

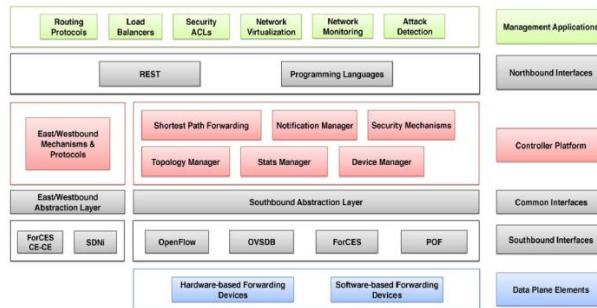


Figura 3. Componentes de un Controlador SDN

- Plano de Control: Es responsable de gestionar y dirigir el reenvío de tráfico de la red desde un nodo central.
- APIs Northbound: Presentan una interfaz común para el desarrollo de aplicaciones y permiten la interacción entre las aplicaciones externas y el controlador SDN. Actualmente los controladores de red ofrecen una variedad de APIs Northbound, como APIs RESTful [18][19], sistemas de archivos y lenguajes de programación.
- APIs Eastbound/Westbound: Se utilizan en esquemas de controladores SDN distribuidos para permitir la comunicación entre los controladores y compartir información de alcanzabilidad y control. Entre las funciones de estas interfaces se encuentran: (1) importar o exportar datos entre controladores, (2) proveer algoritmos para modelos de consistencia de datos y (3) proveer capacidades de monitoreo y notificaciones. Para que exista compatibilidad e interoperabilidad entre diferentes controladores, se requiere disponer de APIs Eastbound/Westbound estandarizadas, como SDNI [20], PCEP [21] y BGP [15].
- Virtualización de Red (Slicing): Consiste en la capacidad de compartir el plano de datos del hardware de red a múltiples redes lógicas, cada una con su propio direccionamiento y su propio mecanismo de reenvío. En una red SDN se puede virtualizar la capa de hardware de red en slices y asignar a cada slice recursos y espacios de direcciones. Un slice se define como una instancia de

una red virtual, y dos redes virtuales distintas sobre el mismo hardware físico se conocen como slices. La virtualización de una red SDN se puede implementar siguiendo un esquema de virtualización mediante proxys o siguiendo un esquema de virtualización basado en lenguajes. La virtualización de redes SDN mediante proxys se logra con la colocación de un controlador SDN especial entre los controladores y los dispositivos SDN de la red. El proxy actúa como un multiplexor de tráfico y como un gestor de la asignación de recursos a slices independientes. En la actualidad se encuentran los siguientes virtualizadores de red basados en proxies: FlowVisor [22], OpenVirteX [23], y AutoSlice [24]. FlowVisor [22] es un controlador OpenFlow de propósito especial que actúa como un proxy transparente entre switches OpenFlow y controladores OpenFlow ofreciendo una capa de virtualización de red basada en OpenFlow. En FlowVisor los slices pueden ser definidos por una combinación de puertos de switches Capa 1, direcciones Ethernet origen/destino Capa 2, direcciones IP origen/destino de Capa 3 y puertos TCP/UDP origen/destino o códigos/tipos ICMP Capa 4. En la Figura 4 se muestra una topología de red conformada por tres controladores OpenFlow, un controlador FlowVisor y seis switches OpenFlow. FlowVisor recibe todos los comandos OpenFlow de los controladores hacia los switches OpenFlow y las respuestas y notificaciones de estadísticas de los switches hacia los controladores respectivos garantizando el aislamiento de tráfico entre cada slice definido. En el ejemplo se puede observar como cada controlador tiene una vista particular y diferente de la misma red física. A través de FlowVisor un grupo de investigadores puede crear sus propias instancias de red lógicas corriendo sus propios protocolos de enrutamiento y ejecutarlas sobre una red real en paralelo con una red de producción y mantener el aislamiento y las velocidades de reenvío del hardware existente. La virtualización basada en lenguajes es el uso de lenguajes de propósito especial para virtualización incluyendo Pyretic [30] y Splendid [25].

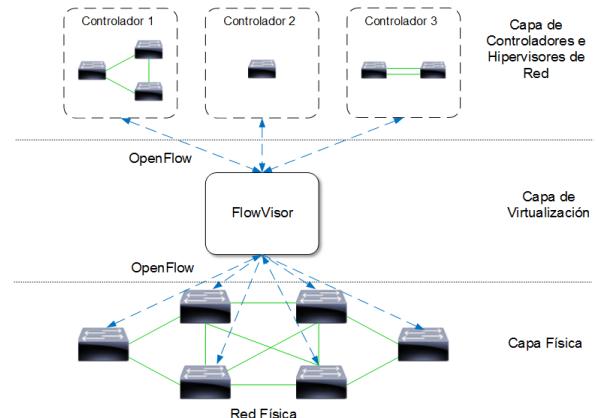


Figura 4. Topología de Red OpenFlow con FlowVisor

- Lenguajes de Programación: Se encuentran los lenguajes que pueden interactuar con el controlador SDN. Entre los más utilizados se encuentran: Python y Java y lenguajes

- de programación específicos para SDN, como Procera [26], FML [27] (Flow Based Management Language), Frenetic [28], NetCore [29] y Pyretic [30].
- Aplicaciones: Constituyen las aplicaciones del negocio de las organizaciones y aplicaciones externas de propósito especial como balanceadores de carga, sistemas de monitoreo y aplicaciones de QoS.
 - Plano de Gestión: El plano de gestión es responsable de las funciones de monitoreo, configuración y mantenimiento de los dispositivos de red. En SDN la gestión de la red se implementa mediante una jerarquía de capas de protocolos, modelos y datos (ver Figura 5 tomada de [32]). Los protocolos de gestión utilizan un lenguaje de modelo de datos como YANG [33] (Yet Another Next Generation) para configurar y obtener datos del estado de los dispositivos. Entre los protocolos de gestión de red SDN se encuentran NETCONF [17], RESTCONF [34], gRPC [35] y OF-CONFIG [36] (OpenFlow Management and Configuration Protocol).

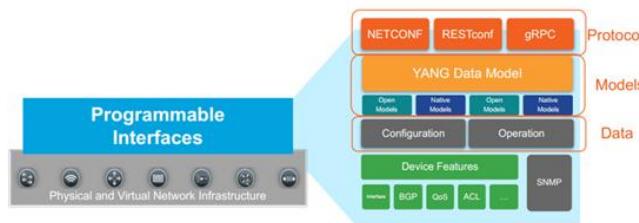


Figura 5. Arquitectura de Gestión basada en Modelos

- NETCONF es un protocolo de configuración de red estandarizado por el IETF en el RFC 6241 [17] que provee mecanismos simples para instalar, manipular, y eliminar la configuración de dispositivos de red. Sigue un paradigma solicitud-respuesta RPC (Remote Procedure Call) con codificación XML y transporte seguro SSH (Secure Shell). NETCONF define almacenes de datos que contienen la configuración de los dispositivos de red y operaciones CRUD (Create, Read, Update, Delete) que permiten recuperar, configurar, copiar y eliminar almacenes de datos. RESTCONF es un protocolo de configuración de red basado en HTTP estandarizado por el IETF en el RFC 8040 [34]. A través de RESTCONF las aplicaciones Web pueden acceder y modificar la configuración de un dispositivo de red siguiendo una arquitectura cliente/servidor basada en RPC con clientes y servidores RESTCONF. gRPC [35] es un framework para llamadas a procedimientos remotos RPC de código abierto y de alto desempeño liberado por la empresa Google para construir sistemas distribuidos masivos. OF-CONFIG [36] es un protocolo de gestión desarrollado por la ONF (Open Networking Foundation) para gestionar switches OpenFlow físicos y virtuales. Utiliza NETCONF para la administración, XML para la codificación y SSH para el transporte.

B. Modelos de Despliegue SDN

En la práctica se utilizan tres modelos de despliegue SDN [4]: (1) Modelo SDN basado en Dispositivos, (2) Modelo SDN Overlay, y (3) Modelo SDN Híbrido. El Modelo SDN basado en Dispositivos [4] se refiere a una red de switches físicos SDN que operan solo bajo las instrucciones de un

controlador SDN. Se implementa con rapidez en despliegues nuevos como un nuevo complejo de oficinas dentro de un campus. En la Figura 6 se muestra un despliegue SDN con 6 switches dirigidos por las instrucciones de un controlador SDN. El modelo SDN Overlay [4] está basado en la superposición de redes sobre una infraestructura de red física subyacente. En una red SDN Overlay, los nodos finales SDN son dispositivos virtuales que forman parte de hipervisores en un ambiente de virtualización de servidores.

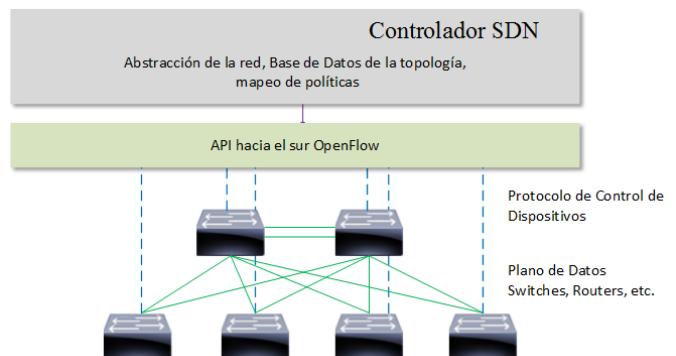


Figura 6. Ejemplo del Modelo SDN basado en Dispositivos

En este escenario el controlador controla el reenvío del tráfico de los switches lógicos que se encuentran definidos en los hipervisores y no altera la red física actual ni el plano de control distribuido de la red subyacente. Para crear la red virtual, los nodos SDN lógicos establecen túneles overlay entre sí a través de un protocolo de túneles: (1) VXLAN [37] (Virtual Extensible LAN), (2) NVGRE [38] (Network Virtualization using Generic Routing Encapsulation), o (3) STT [39] (Stateless Transport Tunneling). Los túneles overlay usualmente terminan en los switches virtuales dentro de los hipervisores o en dispositivos físicos que actúan como gateways hacia la red existente. En la Figura 7 se muestra un despliegue SDN Overlay conformado por un controlador SDN, tres vSwitches hospedados en tres hipervisores y una red tradicional subyacente de cinco switches.

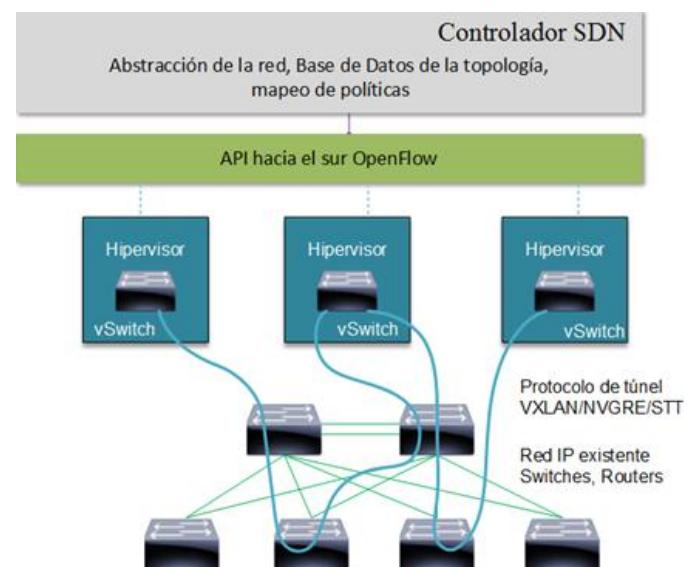


Figura 7. Modelo SDN Overlay

La red virtual consiste de switches lógicos interconectados por enlaces virtuales punto-a-punto. El controlador SDN se apoya en el protocolo hacia el sur OpenFlow para aprovisionar las reglas de reenvío a las tablas de flujos de los vSwitches presentes en los hipervisores de las VMs. El modelo se utiliza en escenarios donde se quiere hacer una implementación rápida de una solución SDN montada sobre una red IP existente. La limitante del modelo es la pérdida de visibilidad del tráfico en la red subyacente que restringe la aplicación de servicios diferenciados e ingeniería de tráfico y la complejidad de gestionar y operar ambas redes.

C. Modelo SDN Híbrido

El modelo SDN Híbrido está basado en la convivencia de tecnologías de redes tradicionales con tecnologías de red SDN en un mismo entorno. En este caso un gateway SDN corre tanto el modelo SDN Overlay, como el modelo SDN Basado en Dispositivos para esquemas híbridos de modelos SDN basado en dispositivos con modelos SDN Overlay. También existe la opción de disponer de un gateway SDN que corre protocolos legados tradicionales y protocolos SDN/OpenFlow. El gateway en un esquema híbrido se comunica con el controlador SDN OpenFlow y con los dispositivos de red Ethernet tradicionales y corre bajo ambos esquemas de red y protocolos.

IV. OPENFLOW

OpenFlow [40] es la primera interfaz estandarizada diseñada específicamente para SDN, brindando alto desempeño, y control de tráfico granular en dispositivos de múltiples fabricantes. OpenFlow es un protocolo SDN abierto que se utiliza para controlar el plano de datos de los dispositivos SDN desde un nodo central. Constituye la primera interfaz de comunicación estándar definida entre los planos de datos y control de una arquitectura SDN y permite el acceso directo y la manipulación del plano de datos de routers y switches, tanto físicos, como virtuales. El protocolo se implementa en ambos lados de la interfaz entre los dispositivos de la infraestructura de red y el software de control SDN. Utiliza el concepto de flujos para identificar el tráfico de la red basado en reglas estáticas predefinidas o en reglas programadas dinámicamente por el software de control SDN.

A. Arquitectura OpenFlow

La arquitectura de red OpenFlow consiste de tres conceptos básicos: (1) la red está soportada por switches con capacidad OpenFlow que conforman el plano de datos, (2) el plano de control consiste de uno o más controladores OpenFlow que definen y publican las reglas de reenvío al plano de datos de los switches OpenFlow y (3) el controlador y los switches OpenFlow se comunican entre sí mediante un canal de control seguro (ver Figura 8 tomada de [40]). En una arquitectura OpenFlow, el reenvío de los datos se efectúa en los switches de la red, y las decisiones de reenvío se hacen en un programa de software de un controlador externo implementado en un servidor que se comunica con los switches a través del protocolo OpenFlow. A través del protocolo OpenFlow, un controlador puede añadir, actualizar, y eliminar entradas de flujos en las tablas de flujos, de manera proactiva o de manera reactiva en respuesta a la llegada de paquetes,

modificando el comportamiento de reenvío del plano de datos de los switches.

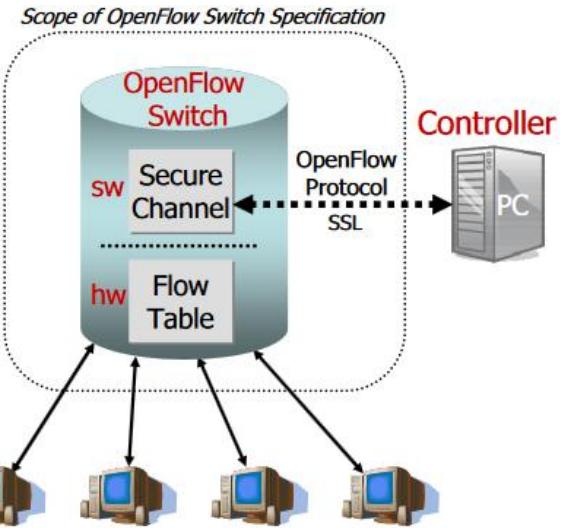


Figura 8. Arquitectura de Red OpenFlow

Dentro de un dispositivo OpenFlow, cuando un paquete ingresa a alguno de sus puertos se inicia un proceso de consulta en la tabla de flujos para encontrar una entrada de flujos que concuerde con el paquete. Las entradas de flujos son evaluadas en orden de prioridad y se utiliza la primera concordancia en la tabla. Si existe una concordancia, se ejecutan las acciones indicadas en la entrada de flujo y se actualizan los contadores de paquetes para la entrada de flujos correspondiente. En caso de no existir concordancia en alguna entrada de la tabla de flujos, el paquete se envía al controlador sobre el canal OpenFlow o se descarta el paquete.

Entre las instrucciones asociadas con cada entrada de flujo se encuentran: (1) reenviar el paquete a un puerto, (2) modificar los campos de cabecera del paquete y (3) descartar el paquete. Las entradas de flujos pudieran indicar enviar un paquete a un puerto físico, o a un puerto virtual definido por el switch, o a un puerto virtual reservado por la especificación OpenFlow. Los puertos virtuales reservados pudieran especificar acciones de reenvío genéricas como reenvío a un controlador, hacer una inundación de puertos, o reenviar los paquetes usando métodos no-OpenFlow, como el procesamiento de un switch Ethernet tradicional. Los puertos virtuales definidos por el switch permiten especificar grupos de agregación de enlaces, túneles o interfaces de loopback.

B. Ejemplo de Tabla de Flujos OpenFlow

Ingress Port	Src MAC	Dst MAC	Ether Type	VLAN ID	VLAN Priority	Src IPv4	Dst IPv4	IP Protocol	IP TOS	TCP/UDP Src	TCP/UDP Dst	Action	Priority	Counter
*	3c:07:54:*	*	Switching	*	*	*	*	*	*	*	*	Fwd Port 10	100	
*	*	*	Routing	*	*	*	192.168.1.*	*	*	*	*	Fwd Port 12	100	
Port1	*	*	Replication/SPAN	*	*	*	*	*	*	*	*	Fwd Port 14..24	100	
*	*	*	Firewall/Security	*	*	*	*	*	*	*	23	Drop	100	
*	*	*	Inspection	*	*	*	*	0x06	*	*	*	Controller	100	
*	00:01:E7:*	*	VLAN10	*	Combinations	*	*	*	*	*	80	Fwd Port 8	200	
*	*	*	Multi-action; NAT	*	*	*	192.168.1.*	*	*	*	80	Rewrite 10.1.2.3; Fwd Port 9	200	
	Local handling	*	*	*	*	10.*	*	*	*	*	*	Local	200	

Figura 9. Ejemplo de Tabla de Flujos OpenFlow

En la Figura 9 tomada de [41] se presenta un ejemplo de una tabla de flujos de un switch OpenFlow v1.0.

En este ejemplo, se presentan las siguientes reglas para todos los paquetes que ingresan al switch:

- Switching: Todos los paquetes cuya dirección MAC origen sea 3c:07:54:* son enviados al puerto 10.
- Routing: Todos los paquetes cuya dirección IP destino sea 192.168.1.* son enviados al puerto 12.
- Replicación/SPAN: Todos los paquetes que ingresen por el puerto 1 son replicados en los puertos que van desde el puerto 14 hasta el puerto 24.
- Firewall/Seguridad: Todos los paquetes con número de puerto TCP/UDP destino 23 son descartados.
- Inspección: Todos los paquetes con protocolo IP 0x06, que corresponden a segmentos TCP, son reenviados al controlador para análisis ulterior.
- Combinaciones: Todos los paquetes cuya dirección MAC origen sea 00:01:E7:* y que pertenezcan a la VLAN 10 y que tengan puerto TCP/UDP destino 80 son enviados al puerto 8.
- Multi-acción/NAT: Todos los paquetes cuya dirección IP destino sea 192.168.1.* y con puerto TCP/UDP destino 80 tendrán nueva dirección IP origen 10.1.2.3 y son enviados al puerto 9.
- Manejo local: Todos los paquetes cuya dirección IP destino sea 10.*.*.* son manejados localmente.

C. Tipos de Switches OpenFlow

Existen dos categorías de switches OpenFlow: switches Solo-OpenFlow (OpenFlow-only), cuyo procesamiento de paquetes está basado únicamente en el protocolo OpenFlow y switches OpenFlow-híbrido (OpenFlow-hybrid), que soportan tanto la operación OpenFlow como la operación de switching Ethernet tradicional.

D. Protocolo OpenFlow

El protocolo OpenFlow describe el intercambio de mensajes que tiene lugar entre un controlador OpenFlow y un switch OpenFlow. Generalmente el protocolo es implementado sobre SSL (Secure Socket Layer) o TLS (Transport Layer Security), para proveer un canal OpenFlow seguro o sobre un canal TCP. Existen tres tipos de mensajes entre el controlador y los switches OpenFlow: (1) Controller to Switch, son mensajes enviados por el controlador a un switch para solicitar notificaciones de capacidades, estados de los puertos y estadísticas de paquetes, o para modificar el estado del reenvío del switch, como añadir, eliminar o modificar entradas en las tablas de flujos, (2) Asynchronous, son mensajes enviados de manera asíncrona por los switches al controlador para denotar la llegada de un nuevo paquete, notificar el cambio de estado en el switch, o informar la ocurrencia de algún error y (3) Symmetric, son mensajes enviados en cualquier dirección sin una solicitud expresa, como los mensajes Hello que notifican la disponibilidad de un dispositivo y los mensajes Echo Request y Echo Reply que pueden ser utilizados para medir la latencia en el canal de comunicación entre el controlador y el switch.

E. Especificaciones OpenFlow

OpenFlow ha evolucionado desde su aparición a finales de 2007 hasta la actualidad. En la siguiente sección se describen las características resaltantes de cada versión mayor:

OpenFlow 1.0.0 [42]: Especificación donde cada switch tiene una sola tabla de flujos. La tabla de flujos consta de tres grupos de campos: Header Fields, Counters y Actions. El grupo Header Fields contiene 12 campos: Ingress Port o puerto de ingreso del paquete, Ethernet Source Address o dirección Ethernet origen del paquete, Ethernet Destination Address o dirección Ethernet destino del paquete, Ethernet Type que indica cual es el protocolo encapsulado en el payload del frame Ethernet, VLAN ID para la VLAN del paquete, VLAN priority CoS que es un campo de prioridad de 3 bits, IPv4 Source Address o dirección IPv4 origen del paquete, IPv4 Destination Address o dirección IPv4 destino del paquete, IPv4 Protocol Number o número de protocolo IPv4, IP ToS o campo Type of Service de IPv4, TCP/UDP Source o número de puerto TCP/UDP origen o tipo ICMP (Internet Control Message Protocol), TCP/UDP Destination o número de puerto TCP/UDP destino o tipo ICMP.

OpenFlow 1.1.0 [43]: Modifica la nomenclatura de los campos a Match Fields, Counters e Instructions. Amplía los campos de cabecera incluyendo etiquetas para el soporte de MPLS. Incorpora múltiples tablas de flujos e implementa un mecanismo pipeline OpenFlow, en el cual un paquete atraviesa varias tablas de flujos en serie y recibe un procesamiento particular en cada tabla. Incluye una tabla de grupos para brindar funcionalidad común a un conjunto de paquetes. Permite disminuir el campo TTL (Time-To-Live) en la cabecera IP (ver Figura 10).

OpenFlow 1.2.0 [44]: Incluye campos para el soporte de IPv6. Soporta el análisis de campos de paquetes adicionales a través de una estructura TLV (Type-Length-Value), referenciada como OXM (OpenFlow Extensible Match). Soporta canales redundantes desde un switch a múltiples controladores.

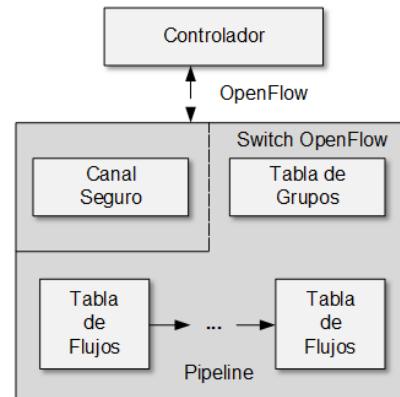


Figura 10. Componentes de un Switch OpenFlow 1.1.0

OpenFlow 1.3.0 [45]: Introduce nuevas funcionalidades para el soporte de funciones administrativas OAM (Operations Administration and Management) e incorpora la tabla especial de medidores al plano de datos del switch que recopilan estadísticas particulares como la tasa de bytes o paquetes de un flujo particular. Otra mejora importante es el soporte extendido

de múltiples controladores auxiliares, permitiendo implementar mecanismos de balanceo de cargas hacia los controladores.

OpenFlow 1.4.0 [46]: Agrega estructuras TLV para puertos, tablas y colas. Incluye el soporte de puertos ópticos. Adicionalmente, los controladores soportan el envío de mensajes en lotes.

OpenFlow 1.5.0 [47]: Introduce el mecanismo de Tablas de Egreso permitiendo que el procesamiento se realice en el contexto de puertos de salida.

V. CONTROLADORES SDN

En esta sección se analizan 3 controladores de código abierto disponibles en la industria: OpenDaylight [48], Floodlight [49] y Ryu [50]. Los criterios de selección de los controladores utilizados fueron: (1) tiempo en el mercado mayor a 3 años, (2) foco activo en su desarrollo, (3) soporte del API OpenFlow 1.3 o superior y (4) disponibilidad de documentación y soporte en línea con FAQs (Frequently Asked Questions).

A. OpenDaylight

La Tabla I resume las características principales del controlador OpenDaylight.

TABLA I. CARACTERÍSTICAS DE OPENDAYLIGHT

Concepto
Controlador de código abierto que sigue una arquitectura modular e incluye su propia máquina virtual Java, lo cual le permite ser desplegado en cualquier plataforma de hardware y software con soporte Java.
Software
Última versión y fecha de liberación: Sexta versión denominada Carbon, mayo, 2017. Lenguaje de Núcleo: Está basado en un framework modular basado en Java 1.7 y soporta el framework de programación OSGI (Open Specifications Group Initiative). APIs Northbound: Soporta REST bidireccional. Servicios y Orquestación de Aplicaciones de Red: Incluye aplicaciones del negocio y aplicaciones de control, aprovisionamiento y gestión de la red, aplicaciones de nube, aplicaciones de centros de datos, funciones de red y servicios de virtualización. Plataforma de Controladores: Incluye los siguientes módulos: gestión de topología (Topology Manager), gestión de estadísticas (Statistics Manager), gestión de switches (Switch Manager), gestión de reglas de reenvío (Forwarding Rules Manager) y rastreador de hosts (Host Tracker). Plugins de protocolos y APIs Southbound: OpenFlow 1.0, OpenFlow 1.3, OF-CONFIG, NETCONF, LISP, OVSDDB, BGP, CAPWAP y otros. Soporte de Comunicación C2C (Controlador-a-Controlador): Incluye la aplicación ODL-SDNi para proveer el establecimiento de la interfaz Este-Oeste (SDNi communication) entre múltiples controladores OpenDaylight.
Usabilidad
Documentación de instalación y documentación operativa disponible en la wiki del proyecto.

Implementación

El controlador OpenDaylight corre sobre una máquina virtual Java. Para mejores resultados, se recomienda una distribución Linux y Java 1.7.

Funcionalidades

Soporta OpenFlow 1.0 y 1.3. Soporta el framework OSGi y Java como lenguaje de desarrollo y soporta REST bidireccional empleando JSON y API DOM empleando XML. Incluye aplicaciones: OpenDOVE y VTN para implementar virtualización de red overlay. Integración con OpenStack a través de un API Neutron.

Licenciamiento

Licencia: Eclipse Public License (EPL-1.0).

Impacto Potencial

Popularidad: Alta. Tiene fuerte soporte de la industria y de la comunidad de código abierto.

B. Floodlight

La Tabla II resume las características principales del controlador Floodlight.

TABLA II. CARACTERÍSTICAS DE FLOODLIGHT

Concepto
Controlador OpenFlow extensible basado en Java con licencia Apache desarrollado por una comunidad abierta de programadores y patrocinado por Big Switch Networks y un conjunto de aplicaciones construidas en el tope del controlador.
Software
Última versión y fecha de liberación: Floodlight v1.2, febrero. 2016. Lenguaje de Núcleo: Java 1.7. APIs Northbound: REST con formato de datos JSON. Aplicaciones: Circuit Pusher, utiliza APIs REST Floodlight para crear un circuito bidireccional entre 2 puntos terminales IP. OpenStack Quantum Plugin, permite que el controlador Floodlight corra como una red de backend para el sistema operativo de nubes OpenStack utilizando el plugin Neutron. Virtual Network Filter, módulo para virtualizar redes basadas en direcciones MAC capa 2. Learning Switch, switch de aprendizaje capa 2 común. Firewall, implementa reglas ACL a switches OpenFlow. Hub, aplicación que siempre inunda un paquete de ingreso a todos los puertos activos de un switch. Load Balancer, balancea cargas de flujos TCP, UDP y ping. Servicios/Funciones del Core: Inventario de dispositivos (hosts). Administrador de la topología. Módulo para insertar flujos y grupos en la red OpenFlow (Static Entry Pusher). Monitoreo del desempeño del controlador. Capacidad de reenvío Dijkstra que permite interconectar islas de switches OpenFlow con switches no-OpenFlow. APIs Southbound: Solamente OpenFlow desde la v1.0 hasta la v1.4. Soporte de Comunicación C2C: Floodlight no provee soporte para ningún tipo de comunicación C2C, por lo cual no soporta alta disponibilidad, ni interfaces este-oeste, ni controladores jerárquicos.

Usabilidad	Sistema Operativo Soportado: Distribución GNU/Linux reciente. Dependencias en Tiempo de Ejecución: Paquetes Python (python-eventlet, python-routes, python-webob y python-paramiko).
Implementación	Se recomienda una distribución Linux Ubuntu 14.0.4 TLS Trusty Tahr o Ubuntu 16.04.1 LTS Xenial Xerus. Para instalar Floodlight en Linux se requiere un cliente Git, Python y la herramienta Apache Ant.
Funcionalidades	Versión OpenFlow Soportada: 1.0, 1.2, 1.3, 1.4, 1.5 y extensiones Nicira. APIs Southbound: NETCONF, OF-CONFIG, SNMP, OVSDB y OpenFlow. Integración con Sistemas de Gestión de Nube: Incluye el plugin OpenStack. Utiliza APIs REST. Servicios/Funciones del Core: Incluye analizadores y generadores de paquetes de diferentes protocolos de red. Incluye un plugin para integrarse al controlador de redes de nube OpenStack Neutron que soporta túneles overlay basados en GRE y configuraciones de VLANs.
Licenciamiento	Licencia: Apache 2.0.
Impacto Potencial	Licencia: Apache 2.0.

C. Ryu

La Tabla III resume las características principales del controlador Ryu.

TABLA III. CARACTERÍSTICAS DE RYU

Concepto	Ryu es un framework SDN basado en componentes con APIs bien definidas que facilitan la creación de aplicaciones de control y gestión de redes.
Software	Última versión y fecha de liberación: Ryu v4.8, noviembre. 2016. Lenguaje de Núcleo: Python 2.6+, greenlets, pseudo-mutihilo cooperativo (utilizando un sólo núcleo). Interfaces/Lenguaje de Aplicaciones: Python 2.6+ para las aplicaciones del controlador y REST para las aplicaciones externas. Arquitectura de Software: El framework SDN Ryu está conformado por un conjunto de componentes, librerías e interfaces para el intercambio de información. La capa de aplicaciones contempla las siguientes aplicaciones de red: switch capa 2, firewall, IDS (Snort), abstracciones de túneles GRE, VRRP (Virtual Router Redundancy Protocol) y los servicios de descubrimiento de la topología y manejo de estadísticas entre otros. La capa de control incluye el soporte de protocolos hacia el sur y los siguientes componentes principales: manejador de eventos, analizador de mensajes OpenFlow, gestor de memoria, gestor de aplicaciones, servicios de infraestructura y un conjunto de librerías que incluyen NETCONF, sFlow y NetFlow. Soporte de Comunicación C2C: Es capaz de soportar servicios de alta disponibilidad a través del componente Zookeeper.
Usabilidad	Documentación de la instalación y documentación operativa disponible.
Implementación	

La selección de un controlador para un despliegue SDN depende de factores como el tamaño de la red, el costo de adquisición, el desempeño de la red requerido por las aplicaciones, la confiabilidad, la facilidad de gestión, el nivel de escalabilidad y las APIs soportadas. Para ambientes de escala grande y mediana se puede optar por un conjunto de controladores OpenDaylight en alta disponibilidad en base a su robustez, escalabilidad y amplio soporte. Para ambientes de escala pequeña se puede considerar una implementación basada en un controlador Floodlight y para entornos de experimentación y desarrollo de nuevas tecnologías se pueden utilizar controladores Ryu.

En la industria existen alternativas de controladores SDN/OpenFlow comerciales incluyendo: (1) Big Network Controller [53], basado en el controlador Floodlight, (2) Brocade SDN Controller [54], distribución comercial de OpenDaylight por Brocade, (3) Cisco Open SDN Controller [55], distribución comercial de OpenDaylight por Cisco Systems, (4) Aruba VAN SDN Controller [56] y (5) NEC ProgrammableFlow PF6800 [57]. La selección del controlador comercial dependerá de factores como el costo de capital, el servicio técnico, la documentación, el apoyo durante la migración o implementación de la solución, la facilidad de la gestión, los APIs soportados para interactuar con el controlador y las aplicaciones embebidas en las propuestas de cada fabricante.

VI. FASES PARA EL DESPLIEGUE DE SDN

En esta sección se describen las fases para desplegar redes SDN basado en las mejores prácticas y recomendaciones recopiladas de los casos de estudio de SDN descritos por el Grupo de Trabajo para Migración hacia SDN de la ONF, específicamente considerando las pautas relacionadas con la

migración hacia SDN en el campus de la Universidad de Stanford [2][3].

La Universidad de Stanford migró dos edificios de su campus a SDN en el año 2010 [2]. La meta fue promulgar la tecnología SDN para facilitar la experimentación en redes de producción, comprender y verificar la tecnología SDN y contribuir con el desarrollo del protocolo OpenFlow. La implementación estuvo basada en un despliegue SDN Híbrido con switches OpenFlow híbridos de múltiples fabricantes con asignación de una VLAN OpenFlow y una VLAN legada. También se consideraron Access Points con switches de software de referencia OpenFlow basados en Linux. El plano de control se hizo con controladores abiertos disponibles para la fecha y se virtualizó la red con FlowVisor en slices para tráfico de datos de experimentación y slices para tráfico de datos de producción. Para garantizar el éxito de la migración se implementó una infraestructura de monitoreo que recopiló estadísticas del plano de control antes, durante y después de la migración.

En general un despliegue SDN consta de tres fases importantes: (1) Preparación, (2) Implementación y (3) Validación.

A. Preparación

La fase de preparación contempla (1) el análisis de los requerimientos del negocio, (2) el diseño de alto nivel de la solución, (3) la preparación de las pruebas de concepto, (4) el diseño de bajo nivel de la solución y (4) la planificación de la implementación.

Análisis de Requerimientos: Esta etapa consiste en la identificación de las metas y objetivos del negocio que se deben alcanzar con la nueva red y en el reconocimiento del estado actual de la red. Particularmente se debe recopilar y analizar la siguiente información: (1) metas del negocio, (2) caracterización de las aplicaciones del negocio y los servicios de red incluyendo tipo de aplicación, volumen, prioridades, requerimientos de ancho de banda y reglas de acceso, (3) topología de red incluyendo usuarios, sistemas, servidores, equipos de red, tipo y velocidad de enlaces de conexión, (4) identificación de segmentos de red incluyendo definiciones de VLANs, direccionamiento IP y protocolos de red, (5) identificación de políticas del negocio, políticas de seguridad, políticas de calidad de servicio de la red y normas de cumplimiento como regulaciones y (6) identificación de los sistemas de gestión de la red.

La evaluación de la red se puede hacer levantando la información con informes del cliente y/o corriendo herramientas de monitoreo y análisis de tráfico que permitan identificar métricas de utilización de la red identificando las aplicaciones, usuarios y servicios con mayor uso y volumen de solicitudes. Esta etapa también contempla la identificación de riesgos, cuellos de botella de la red o problemas de desempeño.

Diseño de Alto Nivel: Muestra un diagrama de alto nivel de la solución identificando los componentes de la arquitectura de red SDN de la solución incluyendo controladores, dispositivos de red, aplicaciones y sistemas de gestión. La selección de los controladores y los dispositivos de red se realizan en base a los resultados del análisis de los requerimientos y considera

aspectos como costos, facilidad de uso, desempeño, soporte y APIs soportadas. Se debe seleccionar el tipo de switch OpenFlow solo-OpenFlow/Híbrido o virtual, dependiendo del modelo de despliegue de la solución seleccionado.

Pruebas de Concepto: Consiste en la preparación de un banco de pruebas para determinar si la solución a implementar satisface las metas del negocio. La implementación del banco de pruebas se puede realizar mediante la utilización de herramientas de simulación SDN de código libre de prototipado rápido incluyendo las herramientas Mininet [58], VT-Mininet [59] y OFNet [60] o herramientas comerciales incluyendo EstiNet [61]. La herramienta OFNet permite hacer emulaciones de red gráficas y provee capacidades de depuración visual, monitoreo de desempeño y capacidades de generación de tráfico sintético que facilitan la depuración de la solución y la optimización del diseño de la solución.

Diseño de Bajo Nivel: Consiste en la definición de las reglas de control de tráfico que serán implementadas en el controlador SDN/OpenFlow en base a los requerimientos y las políticas del negocio.

Planificación: Se establece la línea de tiempo de las fases de ejecución del despliegue SDN y los entregables e informes de cada fase.

B. Implementación

La fase de implementación consiste en la migración de la red legada, los usuarios y servicios a la red SDN/OpenFlow de la solución. El esquema de migración dependerá del análisis de requerimientos donde se determinará el modelo de despliegue SDN. En el caso de una red nueva totalmente OpenFlow, se implementa el modelo SDN basado en dispositivos. En el caso de escenarios de migración de red legada a SDN/OpenFlow, se implementa un modelo SDN Híbrido.

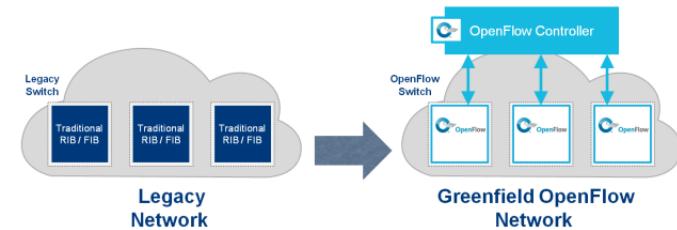


Figura 11. Despliegue Greenfield

Despliegue Greenfield: La ONF denota al modelo SDN basado en Dispositivos con el nombre de despliegue Greenfield (ver Figura 11). En este caso la red existente es actualizada para convertirse en OpenFlow y el control de los dispositivos de la red es reemplazado con un esquema de control de tráfico a través del controlador SDN. Los dispositivos SDN pueden ser dispositivos que aceptan un plugin a los cuales se le instala un agente OpenFlow o switches OpenFlow nativos. El despliegue de esta red consiste en la conexión de red entre el controlador y los dispositivos de red. A partir de este momento el controlador se encargará de hacer un descubrimiento de la red y publicará las reglas de las tablas de flujos a los dispositivos de red. Posteriormente se definen las reglas de filtrado y conectividad específicas en base a las políticas de

seguridad, calidad de servicio y políticas de red recopiladas durante la fase de preparación.

Despliegue Mixto (Mixed): La ONF denota el modelo SDN Híbrido con el nombre Despliegue Mixto. Este enfoque considera que se desplegarán nuevos dispositivos OpenFlow los cuales coexistirán con los dispositivos legados que operan bajo protocolos tradicionales distribuidos. En este caso el controlador SDN/OpenFlow y los dispositivos tradicionales intercambiarán información de enrutamiento entre sí a través de protocolos de red tradicionales.

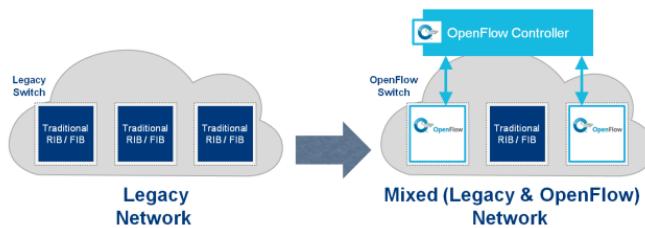


Figura 12. Despliegue Mixto

Enfoque de Migración: El enfoque de migración recomendado por la ONF para este escenario consiste en mover usuarios individuales y VLANs individuales a la red de control OpenFlow para gestionar el riesgo asociado en desplegar la nueva tecnología. Se consideran 4 fases fundamentales:

- Añadir soporte OpenFlow al hardware: Se hace una actualización del firmware a los dispositivos de red para soportar OpenFlow.
- Verificar el soporte OpenFlow en los dispositivos de red: Se verifica la funcionalidad OpenFlow incorporando una VLAN experimental gestionada por el controlador SDN.
- Migración de usuarios a la nueva red: Se crea una nueva red no-OpenFlow y se migran los usuarios a esta nueva red antes de utilizar OpenFlow para el tráfico de producción. Esta tarea involucra las siguientes actividades:
 - Añadir una nueva subred de producción.
 - Añadir/Mover usuarios gradualmente a la nueva subred.
 - Verificar la alcanzabilidad dentro de la nueva subred.
- Habilitar OpenFlow en la nueva subred: Una vez verificado que la nueva subred es funcional, se habilita el control OpenFlow a la subred configurando la regla correspondiente en el controlador.

C. Validación

La validación es una fase fundamental en el proceso de despliegue de la red ya que permite verificar el cumplimiento de los objetivos del diseño y de las metas del negocio. Para efectuar la validación, el despliegue debe contar con una infraestructura de monitoreo que permita recopilar, analizar y verificar el correcto funcionamiento y la alcanzabilidad, desempeño y estabilidad de la nueva red.

Infraestructura de Monitoreo: La infraestructura de monitoreo está conformada por un conjunto de herramientas de captura, almacenamiento y análisis de tráfico y por encuestas

de satisfacción al cliente. La infraestructura de monitoreo recopila información en 2 planos:

- **Plano de Control:** Se recopila información a nivel de flujos basado en mensajes de control entrantes *packet-in* y *flow-exp*. Otras estadísticas importantes a recopilar son la tasa de arribo de flujos *flow_arrival_rate* y los flujos activos *active_flows*.
- **Plano de Datos:** Se pueden implementar nodos de monitoreo dedicados a la recolección de estadísticas ejecutando comandos ping y wget entre sí que permitan recopilar información sobre la utilización de CPU del switch, el tiempo de establecimiento de flujos, RTT, retardos wget y tasa de pérdidas de paquetes.

Entre las herramientas de monitoreo convencionales se encuentran ping [62], tcpdump¹ y wget². También es conveniente disponer de herramientas más completas con interfaces gráficas y soporte de reportes como NetFlow [63], sFlow [64], JFlow [65] o herramientas de monitoreo especializadas que verifiquen el comportamiento de la red y la correctitud del funcionamiento OpenFlow. Algunas herramientas especiales disponibles son:

PayLess [66]: Es un framework de monitoreo basado en consultas para redes SDN que provee un API RESTful flexible para recopilar estadísticas a diferentes niveles de agregación, como flujos, paquetes y puertos. PayLess ejecuta la recopilación de información con alta precisión en tiempo real sin incurrir en un alto overhead de red.

OpenTM [67]: Es una arquitectura de monitoreo que lleva la traza de los flujos activos en una red OpenFlow. Adicionalmente obtiene la información de enrutamiento de la aplicación de enrutamiento del controlador y sondea periódicamente los contadores de cantidad de bytes y número de paquetes de los flujos activos en los switches a lo largo del camino de datos. Para reducir la sobrecarga en la red se pueden sondear de manera aleatoria un subconjunto de los switches seleccionados cuidadosamente para no afectar la precisión de las estadísticas recopiladas por la herramienta.

FlowSense [68]: Es una arquitectura de monitoreo que permite estimar el desempeño de una red OpenFlow a un bajo costo. Utiliza un método pasivo que captura y analiza el intercambio de los mensajes de control entre los switches y el controlador de una red OpenFlow asociados a cambios en el tráfico de la red, como ocurre con los mensajes *PacketIn* y *FlowRemoved*. FlowSense utiliza los mensajes *PacketIn* que notifican la llegada de un nuevo flujo y los mensajes *FlowRemoved* que notifican la expiración de un flujo, para estimar la utilización de un enlace por un flujo.

VII. LINEAMIENTOS GENERALES

El éxito de una implementación SDN se encuentra alineado a los siguientes lineamientos generales:

¹ <http://www.tcpdump.org>

² <https://www.gnu.org/software/wget>

A. Factibilidad Técnica y Comercial

La red SDN/OpenFlow de la solución debe satisfacer los requerimientos del negocio y el siguiente conjunto mínimo de objetivos generales para proveer factibilidad técnica:

- La disponibilidad de la red debe ser superior a un objetivo definido. Por ejemplo, en la red de la Universidad de Stanford, la disponibilidad esperada era superior a 99.9%.
- Debe existir un mecanismo de reversión a la red legada en caso de fallas durante la migración.
- El desempeño de la red debe estar cercano o ser superior al desempeño de la red legada.
- La experiencia de los usuarios no debe ser afectada durante la migración. La red objetivo debe ser programable mediante APIs abiertos y extensibles.
- La red objetivo debe ser utilizable y soportar actualizaciones automatizadas de software con mínima interrupción de servicio y debe tener mecanismos de reversión.
- La red objetivo debe ser interoperable con dispositivos de red de diferentes fabricantes.
- La red objetivo debe ser administrable con software, herramientas y simuladores disponibles.
- La red de arranque puede requerir la preparación y transformación a un estado intermedio seguro desde el cual se pueda proceder a una migración total de la solución.
- La red final debe ser validada con respecto a los requerimientos y expectativas del negocio.

Los costos de la solución deben respetar el presupuesto asignado al proyecto para satisfacer la factibilidad comercial.

B. Modelo de Despliegue

La selección del modelo de despliegue es un factor determinante en los costos y la gestión de riesgo de la solución. El modelo SDN basado en Dispositivos debe implementarse en despliegues Greenfield, mientras que los modelos SDN Overlay o SDN Híbridos deben implementarse en despliegues Mixtos.

C. Selección de los Componentes de Hardware y Software

La selección de los componentes de la solución dependerá de los costos y los requerimientos de criticidad de la solución. El tipo y modelo de controladores se determina en base a las cargas de tráfico actual, al crecimiento esperado de la red y a las funcionalidades soportadas. Para la selección de los controladores se toman en cuenta los siguientes criterios: (1) desempeño, (2) escalabilidad, (3) alta disponibilidad, (4) modularidad, (5) flexibilidad, (6) interoperabilidad, (7) soporte de delegación del control, (8) portabilidad de aplicaciones y (9) licencias.

D. Fases

Para el despliegue de una red SDN en un campus empresarial, se deben cumplir una serie de fases y un conjunto de consideraciones y lineamientos técnicos y estratégicos que garanticen el éxito de la implementación. En un despliegue SDN se deben cumplir las siguientes fases de ejecución: (1) Preparación, (2) Implementación y (3) Validación.

E. Validación

El éxito de un despliegue SDN depende del cumplimiento de la solución con las expectativas del negocio. Se deben implementar pruebas de validación en los planos de control y datos durante la migración para verificar que se cumplen los objetivos del negocio.

F. Disponibilidad

En una red SDN/OpenFlow, el controlador es el único responsable del funcionamiento de la red lo cual ocasiona la condición de un solo punto de falla, en la cual al fallar el controlador o al fallar su comunicación ocasiona un fallo general en la red. Para evitar esta condición se pueden implementar los siguientes mecanismos:

Redundancia de controladores: La plataforma de controladores debe ser redundante para continuar el funcionamiento de la configuración de políticas en caso de que falle el controlador principal a cargo de la red.

Datos de controladores en espejo: La base de datos utilizada por el controlador principal para efectuar la toma de decisiones debe ser replicada y estar disponible para su uso por otros controladores de la red, de tal forma, que los controladores puedan configurar los dispositivos de red de una manera consistente.

Redundancia de caminos hacia los controladores: Se deben establecer enlaces y caminos redundantes hacia los controladores para asegurar que exista al menos un camino de comunicación disponible entre un switch y un controlador.

G. Escalabilidad

La solución debe ser capaz de escalar con el crecimiento de usuarios, aplicaciones y dispositivos en la red. En caso de superarse la capacidad, se debe contar con un plan de acción para soportar las nuevas cargas de tráfico, incluyendo mecanismos de redundancia o segmentación de la red en dominios de control que compartan alcanzabilidad entre sus controladores.

H. Gestión

Para garantizar el éxito de la implementación de la solución se debe contar con una infraestructura de monitoreo y gestión robusta capaz de dar seguimiento y control a los aspectos de alcanzabilidad, desempeño, estabilidad y correctitud del funcionamiento SDN/OpenFlow.

I. Seguridad

Se deben proveer mecanismos de seguridad al tráfico de la red a través de la definición de reglas de filtrado en los controladores y en el desarrollo de programas de control que interactúen con el controlador y permitan al controlador manipular los caminos de flujos basado en el análisis de tráfico y en las estadísticas de la red.

J. Transferencia de Conocimientos

Para garantizar la operación de la nueva red es fundamental impartir el conocimiento del funcionamiento y operación de la red OpenFlow a los administradores de la red. Se debe hacer énfasis en los aspectos de configuración, actualización y validación de reglas en el controlador SDN, en la integración

con aplicaciones externas, en los mecanismos de redundancia de los controladores y en los aspectos de programación de los controladores de la red.

VIII. CONCLUSIONES Y TRABAJOS FUTUROS

SDN es una arquitectura de red emergente que desacopla el plano de control y el plano de datos de los dispositivos y centraliza la inteligencia y el control de la red en un controlador central. La centralización de la inteligencia en una entidad central provee grandes beneficios en términos de operación, control y gestión de la red. El controlador SDN dispone la visión global de la red lo cual facilita y acelera los tiempos de respuesta en la resolución de problemas en la red y permite a los administradores de red implementar optimizaciones y aprovisionar nuevos servicios de manera ágil y expedita.

El análisis de los resultados reportados en la investigación de la arquitectura SDN/OpenFlow, permitió fundamentar un conjunto de consideraciones para apoyar a los administradores de red en el despliegue de redes SDN. Entre los aspectos más relevantes a considerar se encuentran los siguientes:

- Una solución SDN debe estar alineada con los objetivos y metas del negocio para la cual fue diseñada.
- Invertir una mayor cantidad de esfuerzo durante las pruebas de concepto de la solución permite detectar fallas a tiempo y agilizar los tiempos de despliegue en etapas posteriores de este.
- Un mecanismo para garantizar el funcionamiento de una nueva red SDN radica en contar con una buena infraestructura de monitoreo de red que permita efectuar chequeos y controles durante todas las etapas de un despliegue SDN.

Como trabajo futuro, planificamos desarrollar una herramienta que medición de desempeño para controladores SDN. Esta herramienta daría el tiempo de respuesta de un controlador cuando surge un cambio en un switch y que este último debe comunicarse con el controlador para que le especifiquen las acciones a tomar. Similarmente, la herramienta podría inundar un controlador SDN de peticiones para estresarlo, y se reportaría la cantidad de peticiones que el controlador sería capaz de atender en condiciones de carga extrema. En este último caso, la herramienta se desarrollaría con una arquitectura distribuida de tal forma que se pueda someter una mayor carga de peticiones al controlador SDN.

REFERENCIAS

- [1] Open Networking Foundation. Software Defined Networking: The New Norm for Networks. ONF White Paper. Abril 2012.
- [2] Open Networking Foundation. Migration Use Cases and Methods. Migration Working Group. 2014.
- [3] M. Kobayashi, S. Seetharaman, G. Parulkar, et al. Maturing of OpenFlow and Software Defined Networking through Deployments. Computer Networks. Vol. 61, pp. 151-175. Marzo 2014.
- [4] J. Skorupa, M. Fabbi, et al. Ending the Confusion about Software-Defined Networking: A Taxonomy. Gartner Research, G00248592. Marzo 2013.
- [5] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie. A Survey on Software-Defined Networking. IEEE Communication Surveys & Tutorials, Vol. 17, No. 1, pp. 27-51, First Quarter 2015. 2015.
- [6] A. Lara, A. Kolasani, B. Ramamurthy. Network Innovation using OpenFlow: A Survey. IEEE Communications Survey & Tutorials. Agosto 2013.
- [7] P. Goransson, C. Black. Software Defined Networks, A Comprehensive Approach. Morgan Kaufmann. Mayo 2014.
- [8] P. Morreale, J. Anderson. Software Defined Networking Design and Deployment. CRC Press. 2015.
- [9] E. Haleplidis, K. Pentikousis, S. Denazis, J. Hadi Salim, D. Meyer, O. Koufopavlou. Software-Defined Networking (SDN): Layers and Architecture Terminology. RFC 7426. Internet Research Task Force. Enero 2015.
- [10] H. Kim, N. Feamster. Improving Network Management with Software Defined Networking. IEEE Communications Magazine. Vol. 51, No. 2, pp. 114-119. 2013.
- [11] A. Porras. Virtualization-enabled Adaptive Routing for QoS-aware Software-Defined (Master thesis). Universitat Politècnica de Catalunya, Spain, Diciembre 2014.
- [12] D. Kreutz, F. Ramos, et al. Software-Defined Networking: A Comprehensive Survey. Proceedings of the IEEE, Vol. 103, No. 1, pp. 14-76. Enero 2015.
- [13] B. Pfaff, B. Davie. The Open vSwitch Database Management Protocol. RFC 7047. Internet Research Task Force. Diciembre 2013.
- [14] A. Doria, J. Hadi, R. Hass, H. Khosravi, W. Wang, L. Dong, R. Gopal, J. Halpern. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810. Internet Research Task Force. Marzo 2010.
- [15] Y. Rekhter, T. Lid, S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271. Internet Research Task Force. Enero 2006.
- [16] J. Case, M. Fedor, M. Schoffstall, C. Davin. A Simple Network Management Protocol (SNMP). RFC 1098. Internet Research Task Force. Mayo 1990.
- [17] R. Enns, M. Bjorklund, J. Schoenwaelder, A. Bierman. Network Configuration Protocol (NETCONF). RFC 6241. Internet Research Task Force. Junio 2011.
- [18] L. Richardson, S. Ruby. RESTful Web Services. Web Services for the Real World. O'Reilly Media. Mayo 2007.
- [19] R. Fielding. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California. Irvine, CA, USA, 2000.
- [20] H. Yin, H. Xie, T. Tsou, D. Lopez, P. Aranda, R. Sidi. SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains. Internet Engineering Task Force, Internet Draft. Junio 2012.
- [21] JP. Vasseur, JL. Le Roux. Path Computation Element (PCE) Communication Protocol (PCEP). RFC 5440. Internet Engineering Task Force. Marzo 2009.
- [22] R. Sherwood, G. Gibb, K. Yap, et al. FlowVisor: A Network Virtualization Layer. Deutsche Telekom Inc. R&D Lab, Stanford University, Nicira Networks. Octubre 2009.
- [23] A. Al-Shabibi, M. De Leenheer, et al. OpenVirtex: Make Your Virtual SDNs Programmable. HotSDN '14: Proceedings of the third workshop on Hot topics in software defined networking. ACM. Agosto 2014.
- [24] Z. Bozakov, P. Papadimitriou. AutoSlice: Automated and Scalable Slicing for Software-defined Networks. CoNEXT Student '12: Proceedings of the 2012 ACM conference on CoNEXT student workshop. ACM. Diciembre 2012.
- [25] S. Gutz, A. Story, C. Schlesinger, et al. Splendid Isolation: a Slice Abstraction for Software-defined Networks. Proceedings of the first workshop on Hot topics in software defined networks. 2012.
- [26] H. Kim, N. Feamster. Improving Network Management with Software Defined Networking. IEEE Communications Magazine. Vol. 51, No. 2, pp. 114-119. 2013.
- [27] T. Hinrichs, N. Gude, M. Casado, et al. Practical Declarative Network Management. Proceedings of the 1st ACM workshop on Research on enterprise networking. Agosto 2009.
- [28] N. Foster et al. Frenetic: A Network Programming Language. Proceedings of the 16th ACM SIGPLAN international conference on Functional programming. Septiembre 2011.
- [29] C. Monsanto, N. Foster, R. Harrison, et al. A Compiler and Run-time System for Network Programming Languages. Proceedings of the

- 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages. Enero 2012.
- [30] J. Reich, C. Monsanto, N. Foster, et al. Modular SDN Programming with Pyretic. Usenix, The Advanced Computing Systems Association. Octubre 2013.
- [31] ITU. Management Framework for Open Systems Interconnection (OSI) For CCITT Applications. ITU Recommendation X.700. Septiembre 1992.
- [32] F. Maccioli. Network Automation with Ansible 2.1 and beyond [PowerPoint slides]. Cisco. Septiembre 2016.
- [33] M. Bjorklund. The YANG 1.1 Data Modeling Language. RFC 7950. Internet Research Task Force. Agosto 2016.
- [34] A. Bierman, M. Bjorklund, K. Watsen. RESTCONF Protocol. RFC 8040. Internet Research Task Force. Enero 2017.
- [35] gRPC. gRPC A High Performance, Open-Source Universal RPC Framework. <http://www.grpc.io>. 2015.
- [36] Open Networking Foundation. OF-CONFIG 1.2. OpenFlow Management and Configuration Protocol. ONF TS-016. 2014.
- [37] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, C. Wright. Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. RFC 7348. Internet Engineering Task Force. Agosto 2014.
- [38] P. Garg, Y. Wang. NVGRE: Network Virtualization using Generic Routing Encapsulation. RFC 7637. Internet Engineering Task Force. Septiembre 2015.
- [39] B. Davie, J. Gross. A Stateless Transport Tunneling Protocol for Network Virtualization (STT). Internet Engineering Task Force, Internet Draft. Abril 2016.
- [40] N. McKeown. OpenFlow: Enabling Innovation in Campus Networks. Stanford University. Marzo 2008.
- [41] J. Davis. Introduction to Software-Defined Networking (SDN) and Network Programmability. CiscoLive BRKSDN-1014 [Power Point Slides]. 2015.
- [42] Open Networking Foundation. OpenFlow Switch Specification, Version 1.0.0 Implemented (Wire Protocol 0x01). Diciembre 2009.
- [43] Open Networking Foundation. OpenFlow Switch Specification, Version 1.1.0 Implemented (Wire Protocol 0*02). Febrero 2011.
- [44] Open Networking Foundation. OpenFlow Switch Specification, Version 1.2.0 Implemented (Wire Protocol 0*02). Diciembre 2011.
- [45] Open Networking Foundation. OpenFlow Switch Specification, Version 1.3.0 Implemented (Wire Protocol 0*02). Junio 2012.
- [46] Open Networking Foundation. OpenFlow Switch Specification, Version 1.4.0 Implemented (Wire Protocol 0*02). Octubre 2013.
- [47] Open Networking Foundation. OpenFlow Switch Specification, Version 1.5.0 Implemented (Wire Protocol 0*02). Diciembre 2014.
- [48] Linux Foundation. OpenDaylight: Open Source SDN Platform. <https://www.opendaylight.org>. Mayo 2017.
- [49] Project Floodlight. Open Source Software for Building Software-Defined Networks. <http://www.projectfloodlight.org/floodlight>. 2017.
- [50] Ryu SDN Framework Community. Component-Based Software Defined Networking Framework. Build SDN Agilely <http://osrg.github.io/ryu>. 2017.
- [51] R. Khondoker, A. Zaalouk, et al. Feature-Based Comparison and Selection of Software Defined Networking (SDN) Controllers. Fraunhofer Institute for Secure Information Technology. 2013.
- [52] A. Shalimov, D. Zuikov, et al. Advanced Study of SDN/OpenFlow Controllers. In Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia, Moscow, Russian Federation. Octubre 2013.
- [53] Big Switch Networks. SDN Controller Unified Network Control Plane. <http://www.bigswitch.com/products/SDN-Controller>. 2012.
- [54] Brocade. Brocade SDN Controller Data Sheet. 2016.
- [55] Cisco. Cisco Open SDN Controller 1.2 Data Sheet. Octubre 2015.
- [56] HPE. Aruba VAN SDN Controller Software Data Sheet. 2017.
- [57] NEC. PF6800 ProgrammableFlow Controller. <https://www.necam.com/sdn/Software/SDNController>. 2017.
- [58] B. Lantz, B. Heller, N. McKeown. A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. In Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks. Octubre 2010.
- [59] J. Yan, D. Jin. VT-Mininet: Virtual-time-enabled Mininet for Scalable and Accurate Software-Define Network Emulation. ACM SIGCOMM Symposium on SDN Research 2015 (SOSR15), Santa Clara, CA, USA. Junio 2015.
- [60] B. Linkletter. OFNet SDN Network emulator. <http://www.brianlinkletter.com/ofnet-a-new-sdn-network-emulator>. Noviembre 2016.
- [61] S. Wang, C. Chou, C. Yang. EstiNet OpenFlow Network Simulator and Emulator, IEEE Communications Magazine, Vol. 51, No. 9. Septiembre 2013.
- [62] J. Postel. Internet Control Message Protocol. RFC 792. Internet Engineering Task Force. Septiembre 1981.
- [63] Cisco Systems. Introduction to Cisco IOS NetFlow - A Technical Overview. Mayo 2012.
- [64] sFlow. Traffic Monitoring using sFlow. 2003.
- [65] A. Myers. JFlow: Practical Mostly-Static Information Flow Control. In Proceedings of the 26th ACM Symposium on Principles of Programming Languages (POPL'99). San Antonio, TX, USA. Enero 1999.
- [66] S. Chowdhury, D. Cheriton, et al. PayLess: A Low Cost Network Monitoring Framework for Software Defined Networks. 2014 IEEE Network Operations and Management Symposium (NOMS). Mayo 2014.
- [67] A. Tootoonchian, M. Ghobadi, et al. OpenTM: Traffic Matrix Estimator for OpenFlow Networks. In Proceedings of the 11th international conference on Passive and active measurement (PAM'10). Zurich, Switzerland. Abril 2010.
- [68] C. Yu, C. Lumezanu, et al. Flowsense: Monitoring Network Utilization with Zero Measurement Cost. In Proceedings of the 14th International Conference on Passive and Active Measurement (PAM'13). Hong Kong, China. Marzo 2013.

EAW: Evaluador de Criterios de Accesibilidad Web para Pautas Relacionadas con Discapacidad Visual y Discapacidad Motora

Yusneyi Carballo Barrera, María Gabriela Acosta Vásquez y Ronald Aguilera González
Centro de Enseñanza Asistida por Computador, CENEAC, Escuela de Computación, Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela
yusneyi.carballo@ciens.ucv.ve, maria.gabriela.acosta.v@gmail.com, ronald.aguilera@gmail.com

Resumen - La accesibilidad Web indica la posibilidad de acceso a contenidos, servicios o productos disponibles en la WWW por parte de las personas, independientemente de su condición, discapacidad o contexto de uso. El Consorcio World Wide Web mediante la *Web Accessibility Initiative* (WAI) ha promovido el desarrollo de documentos con pautas, técnicas y recursos que ayudan a garantizar un grado de accesibilidad mayor, independientemente de la presencia de una discapacidad visual, auditiva, motriz, del lenguaje o cognitiva en el usuario, del hardware que posea o de su ubicación geográfica. Son ejemplos de estas pautas el *Web Content Accessibility Guidelines* (WCAG) y el *Evaluating Websites for Accessibility*. Garantizar la accesibilidad Web redonda en construir una sociedad donde la comunicación mediada por tecnologías no amplíe brechas. En esta línea, surge el interés en desarrollar recursos para apoyar la evaluación de accesibilidad Web en las investigaciones del CENEAC UCV en el área de Tecnologías Educativas, creándose la Herramienta para Verificación de Criterios de Accesibilidad en Sitios Web (HEVAC) y posteriormente el Evaluador de Criterios de Accesibilidad Web (EAW). Este artículo describe las mejoras incorporadas en EAW con el objetivo de verificar en sitios Web pautas relacionadas con la discapacidad visual y la discapacidad motriz, a partir del análisis de sentencias HTML y CSS, con el posterior despliegue de los resultados de la verificación, el grado de accesibilidad del recurso, los errores encontrados, el código dónde se presentan y recomendaciones para su corrección.

Palabras Clave - Accesibilidad; Discapacidad Visual; Discapacidad Motora; Pautas de Accesibilidad Web; W3C WAI; W3C WCAG; HEVAC; EAW.

I. INTRODUCCIÓN

La accesibilidad de los sitios Web está relacionada con la capacidad de acceso por parte de los usuarios, independientemente de las limitaciones que pueda presentar el individuo o por limitaciones que se deriven del contexto de uso o características técnicas del equipo. La accesibilidad Web procura un diseño que permite a las personas navegar, entender, percibir, interactuar y aprovechar contenidos, recursos y servicios, beneficiando no solo a las personas que presentan una discapacidad, sino también a los adultos mayores y personas que por condiciones temporales ven mermadas sus habilidades y capacidades para el uso de la tecnología [1].

La Organización Mundial de la Salud define **discapacidad** como una deficiencia, carencia o limitación en la capacidad de

realizar una actividad en la misma forma o grado que se considera normal para un ser humano, incluyendo las restricciones en la participación y las limitaciones en la ejecución de actividades, en aptitudes o conductas que se esperan de las personas o del cuerpo en conjunto [2].

Valdez define la **discapacidad visual** como la alteración del sistema visual o la deficiencia en la estructura o funcionamiento de los órganos visuales, lo cual, ocasiona dificultad en el desarrollo normal de las actividades cotidianas que requieran el uso de la visión [3], bien sea por ceguera total, visión reducida o baja visión. La deficiencia visual es aquella visión menor de 20/400, es decir, la persona requiere estar a 20 pies (aprox. 6m) del punto observado, en comparación a la necesidad de estar a 400 pies (aprox. 122m) de una persona con visión normal, considerando siempre el mejor ojo y con la mejor corrección. Se considera que existe ceguera parcial cuando la visión es menor de 20/200 (6m/61m) en el mejor ojo y con la mejor corrección o cuando independientemente de que su visión sea mejor, tiene un campo visual inferior a 20° [2].

La **discapacidad motora o motriz** se puede definir como el impedimento físico o dificultad para trasladarse, controlar o mover algún miembro inferior o superior del cuerpo, debido a que no lograron desarrollarse normalmente o sufrieron algún tipo de traumatismo. Entre las principales causas de alteraciones en el sistema motriz se encuentran enfermedades como la parálisis cerebral (incluyendo la cuadriplejía y hemiplejía), la distrofia muscular, esclerosis múltiple, espina bífida, artritis y la enfermedad de Parkinson [2][4].

Para desarrollar código Web accesible el Consorcio World Wide Web (W3C) publicó en el año 1999 a través de la *Web Accessibility Initiative*¹ (WAI, Iniciativa de Accesibilidad a la Web) las guías o lineamientos *Web Content Accessibility Guidelines 1.0* o **WCAG 1.0** [5]. En el año 2008 se publicaron las **WCAG 2.0**, la última versión hasta el momento [6]. También conocidas como **Pautas de Accesibilidad del Contenido en la Web**, fueron definidas para que los desarrolladores tengan a disposición una serie de criterios, técnicas y recursos que los ayuden a crear un diseño accesible y a evaluar el nivel de accesibilidad de sitios y contenidos Web.

Mientras que la WCAG 1.0 está conformada por catorce (14) pautas centradas en técnicas, la WCAG 2.0 se centra en cuatro (4) principios de nivel superior, alrededor de los cuales

¹ W3C-WAI, <http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

se organizan pautas específicas que explican cómo hacer accesibles los contenidos. Estos principios conforman el conjunto de propiedades POUR: *Perceivable, Operable, Understandable, Robust* (Perceptible, Operable, Comprensible y Robusto). Al estar centradas en principios, y no en pautas técnicas, los lineamientos de la WCAG 2.0 se mantienen vigentes aún con los cambios en las tecnologías y lenguajes de desarrollo, pasando a ser un estándar técnico internacional en el año 2012 [6][7][8].

La WAI también ha definido criterios relacionados con la evaluación de la accesibilidad, recopilando en el documento *Evaluating Websites for Accessibility* (Evaluación de Accesibilidad de Sitios Web) lineamientos y técnicas para evaluar en forma rápida algunos de los problemas de accesibilidad e indicando procedimientos generales para verificar el cumplimiento de las pautas [9].

Verificar los principios WCAG en un sitio Web se torna una tarea tediosa, pudiendo ser mucho más compleja en función de la cantidad de enlaces y recursos que lo conforman. Por ello, se recomienda incorporar las pautas desde las etapas tempranas del diseño y a lo largo del desarrollo de las aplicaciones, contenidos o recursos. Sin embargo, es un hecho común encontrar aplicaciones y contenidos Web que no cumplen con las pautas de accesibilidad, lo cual motiva la investigación en el área y el desarrollo de aplicaciones para verificarlas, detectar errores y suministrar recomendaciones que ayuden a mejorar la creación de recursos que puedan ser usados por todos.

Con este objetivo fueron desarrolladas las herramientas que se describen en este artículo, el cual está dividido en las secciones de Motivación y Antecedentes, Desarrollo de la Herramienta EAW, Uso de la Herramienta EAW, Resultados, Conclusiones y Referencias.

II. MOTIVACIÓN Y ANTECEDENTES

En la actualidad se cuenta con aplicaciones y sitios Web que en teoría pueden ser usados por todas las personas, pero ¿realmente todos pueden acceder y hacer uso de los contenidos disponibles en la Web? La respuesta a esta pregunta es no, especialmente para las personas con discapacidad o limitación.

Aunque algunos usuarios tengan la mejor disposición para navegar por la red y posean el hardware o software necesario, los estudios sobre evaluación de accesibilidad en sitios web indican que hay barreras que se deben superar [10][11]. A manera de ejemplo, Jackson-Sanborn, Odess-Harnish y Warren reportan que al evaluar 100 páginas principales o *index* de organismos federales sólo el 60% era accesible en 2002 [12]; mientras que Loiacono, McCoy y Chin reportan que al evaluar 417 sitios Web federales y de contratistas federales, sólo el 23% cumplían con los lineamientos de accesibilidad de la Sección 508 de la Ley de Rehabilitación o “Acta de los Americanos con Discapacidad” [13][14].

Estas barreras se incrementan además para las personas con discapacidad visual o discapacidad motriz, incluso por la necesidad del uso de un dispositivo tan común como el ratón para dirigir el acceso a la información en la pantalla y la navegabilidad. Cobra importancia entonces la investigación en

torno a la creación de herramientas que faciliten la verificación de pautas de accesibilidad e indiquen a los encargados del desarrollo, prueba o certificación de calidad, cuáles son las omisiones o errores presentes en el código de los sitios Web.

En relación a las herramientas para la evaluación de pautas de accesibilidad Web (WAET, *Web Accessibility Evaluation Tools*), éstas han evolucionado en su objetivo y alcance, desde listas de verificación de pautas, hasta herramientas que en la actualidad verifican en forma automatizada las páginas, detectando problemas de accesibilidad según necesidades de diferentes usuarios (diseñadores, desarrolladores, evaluadores, instancias de certificación, etc.) [8][15].

No todas las herramientas pueden realizar una evaluación automatizada del sitio Web completo o de todo su contenido, en algunos casos sólo pueden evaluar una página a la vez, otras se enfocan en elementos específicos en la página (p.e., menús, colores, textos, metadata de imágenes, tablas) cuyo mal diseño puede limitar el uso del recurso para las personas con discapacidad. En su mayoría, verifican criterios asociados a una de las dos normas mundialmente aceptadas, bien sea las pautas de la WCAG 2.0 para determinar la accesibilidad de la página según los niveles de conformidad y el cumplimiento de los cuatro principios, o trabajan verificando los dieciséis estándares de la norma Sección 508.

En el marco de esta investigación se realizó una revisión detallada de las pautas WCAG, en sus dos versiones. También un análisis comparativo de las herramientas WAVE (*Web Accessibility Evaluation Tool*, WebAIM), TAW (Test de Accesibilidad Web, Fundación CTIC) y HERA (Hojas de Estilo para la Revisión de la Accesibilidad, Fundación SIDAR) [16][17][18].

La **Herramienta para Verificación de Criterios de Accesibilidad en Sitios Web (HEVAC)** [19] se enfocó principalmente en la evaluación de pautas definidas en la WCAG 2.0 que apoyan la accesibilidad de personas con discapacidad visual, aunque algunas de estas pautas también apoyan la accesibilidad de contenidos para personas con discapacidad auditiva y motora. Específicamente se incorporaron en los algoritmos de comprobación 25 técnicas de la WCAG 2.0 asociadas a éxitos y fallos comunes generados con la tecnología HTML, algunas de las cuales se identifican a continuación por su nombre y nivel de conformidad, organizadas según el principio que involucran:

- Principio: Perceptible
 - Nivel A: Campo de texto sin nombre, Enlace sin aviso de nueva página, Campo de imagen sin texto alternativo, Botón de formulario sin texto, Página con elemento en movimiento, Página con elemento de parpadeo, Página con refrescamiento automático, Enlace sin destino de referencia, Enlace de imagen sin descripción, Imagen con texto alternativo vacío.
 - Nivel AA: Selección de formulario sin opciones.
- Principio: Operable
 - Nivel A: Elemento con accesibilidad vía ratón únicamente.

Nivel AA: Campo de texto sin orden de tabulación.

Nivel AAA: Área de imagen sin título, Conjunto de enlaces sin indexación tabulada.

- Principio: Comprensible

Nivel A: Botón de formulario sin valor, Campo del formulario sin etiqueta referenciada, Formulario sin botón de envío, Página sin indicador de lenguaje, Elementos de marcos sin título, Campo del formulario sin etiqueta referenciada.

Nivel AA: Tabla sin texto de resumen, Tabla sin demarcación de título, Tabla sin celdas de cabecera, Imagen sin enlace hacia su descripción, Celda de cabecera sin objetivo.

- Principio: Robusto

Nivel AA: Objeto o *plug-in* sin elemento de sustitución.

El Evaluador de Criterios de Accesibilidad Web (EAW) [4] también evalúa considerando los principios POUR y pautas que pueden apoyar la accesibilidad de personas con discapacidad visual incluidas en HEVAC, adicionando pautas para ayudar a personas con discapacidad motora, las cuales se identifican en el punto III.

En la Tabla I puede verse un cuadro comparativo con las características observadas en las herramientas de accesibilidad Web analizadas [4].

TABLA I. COMPARACIÓN DE LAS HERRAMIENTAS PARA EVALUACIÓN DE ACCESIBILIDAD TAW, HEVAC Y HERA

Característica	TAW	HEVAC	HERA
Sistemas operativos soportados	Windows, Mac OS, Linux, AIX, Solaris, HP-UX	Cualquier que ejecute un navegador	Cualquier que ejecute un navegador
Lenguaje de Programación	Java	Java	PHP
Estándar verificado	WCAG 1.0 WCAG 2.0	WCAG 2.0	WCAG 1.0
Trabaja en modo <i>Offline</i> (versión de escritorio)	Sí aplica	Sí aplica	Sí aplica
Trabaja en modo <i>Online</i>	Sí aplica	Sí aplica	Sí aplica
Idiomas	Español, inglés, gallego y catalán	Español	Español, alemán, portugués, catalán, inglés, francés, gallego, italiano, danés, rumano y serbio
Evaluación de múltiples páginas en un mismo ciclo de revisión	Sí lo permite	No lo permite	No lo permite
Reporte de resultados	Sí lo permite, los presenta en versión HTML	Sí lo permite, los presenta en versión HTML	Sí lo permite, disponibles para la descarga en formatos XHTML, RDF y PDF
Almacenamiento de resultados en base de datos	No aplica	No se dispone de información	Sí, por 7 días
Licencia	Propietario	Open Source	Open Source

La Figura 1 muestra un ejemplo del reporte generado por HEVAC para la evaluación de una página Web.

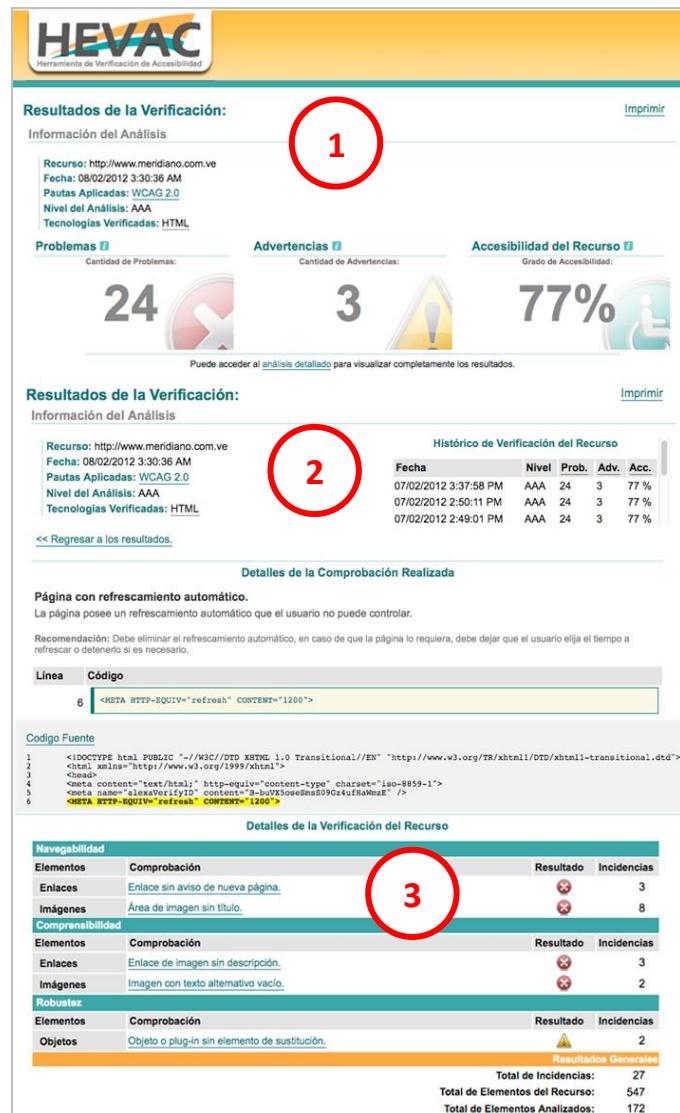


Figura 1. Reporte de Resultados de la Verificación de una Página Web Utilizando HEVAC

Inicialmente se identifica el recurso y se indica la cantidad de problemas detectados (pautas que siempre deberían cumplirse pero presentan problemas), las advertencias (pautas recomendadas u opcionales para mejorar la accesibilidad) y el grado de accesibilidad del recurso o grado de accesibilidad de la página Web.

En un segundo nivel de detalle, más orientado a diseñadores y desarrolladores de aplicaciones, se puede consultar los resultados detallados de la verificación, con la identificación de la página Web, histórico de evaluaciones, detalles de la comprobación, descripción de los problemas encontrados, recomendaciones para su corrección y la identificación de las líneas de código en donde se encuentran.

En un tercer nivel, se puede consultar el detalle de cada uno de los errores detectados, clasificados según el principio de

accesibilidad (navegabilidad, comprensibilidad, robustez), la incidencia o cantidad de apariciones en la página Web, y un resumen, con el total de elementos identificados en el recurso y el total de elementos evaluados, esta diferencia debido a que la herramienta HERA de la Fundación SIDAR solo verifica elementos en código HTML.

III. DESARRROLLO DE LA HERRAMIENTA EAW

Considerando la conveniencia de ampliar el alcance de HEVAC y de continuar con las investigaciones en el área de la accesibilidad Web, nos enfocamos en diseñar y desarrollar una nueva aplicación que incluyera funcionalidades no disponibles en HEVAC, creándose así el Evaluador de Criterios de Accesibilidad Web, EAW.

A. Objetivo y Alcance

El objetivo principal de EAW es indicar a diseñadores y desarrolladores Web el nivel de accesibilidad de los recursos evaluados, orientarlos en la solución de errores y ayudar a quienes asisten a los usuarios que presentan algún tipo de discapacidad visual o motriz a seleccionar las páginas Web con la mejor accesibilidad y usabilidad. Entre las mejoras incorporadas en la herramienta destacan:

- Considera criterios pertenecientes a pautas relacionadas con discapacidad visual y discapacidad motriz que pueden ser verificados en forma automática, a fin de apoyar los cuatro principios del estándar WCAG 2.0, es decir, crear contenidos perceptibles, operables, comprensibles y robustos.
- Evalúa código HTML y CSS.
- Almacena un historial de resultados para posteriores consultas, con la opción de actualizar una revisión ya existente o eliminarla. Esto facilita mantener un historial de páginas Web evaluadas.
- Los usuarios pueden agregar un comentario a cada evaluación, lo cual permite responder consultas relacionadas con la accesibilidad del recurso.
- Evalúa múltiples páginas Web en una sola revisión, partiendo del URL suministrado y sus enlaces de primer nivel.

A continuación se indican algunas de las pautas que fueron incorporadas a EAW para ampliar la evaluación de criterios de accesibilidad, especialmente relacionadas con el apoyo a personas con discapacidad motora:

- Principio: Perceptible

Pauta 1.3: Adaptabilidad, creación de contenido flexible que pueda presentarse de diversas maneras, sin perder parte de la información, ni estructura, al tener que adaptarse a otras modalidades y tecnologías.

- Principio: Operable

Pauta 2.1: Acceso por medio del teclado, permitiendo que toda funcionalidad pueda ser operable a través del mismo.

Pauta 2.2: Tiempo suficiente, considerando un tiempo promedio prudente para poder transmitir la información de manera efectiva, sean textos, audios o videos, y permitir interactuar con la aplicación.

- Principio: Comprensible

Pauta 3.2: Desarrollo de páginas Web que aparezcan y se manejen de manera predecible, incluyendo un adecuado manejo de foco, de la entrada de datos, una navegación consistente y cambios de petición solo a solicitud del usuario.

Pauta 3.3: Ayuda a los usuarios para evitar y corregir los errores.

- Principio: Robusto

Pauta 4.1: Compatibilidad, maximizándola para los agentes de usuario actuales y futuros, incluyendo los productos de apoyo y la tiflotecnología.

B. Arquitectura de la Aplicación y Proceso de Verificación

La aplicación fue desarrollada bajo arquitectura Modelo-Vista-Controlador. Consta de diez componentes principales, conformados por tres vistas (página de inicio o vista *index*, página de nueva evaluación o vista *create* y página de resultados o vista *show*), un controlador, cinco modelos (*webpage*, *child*, *comment*, *tag* y *css*), además de una base de datos MySQL, como puede observarse en el diagrama de componentes de la Figura 2 [4].

La interacción con la herramienta se realiza a través de un sitio Web desarrollado con páginas dinámicas en PHP, el cual, a partir de datos de entrada proporcionados por el usuario puede realizar una nueva evaluación de los criterios de accesibilidad o mostrar una evaluación previa guardada en el historial.

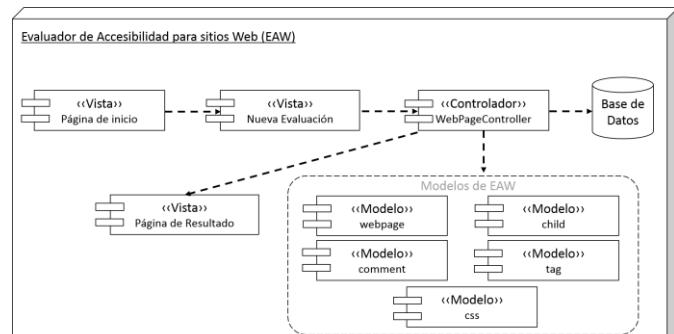


Figura 2. Diagrama de Componentes de EAW

Si el usuario elige la opción de realizar una nueva evaluación se le solicita el URL de la página Web, el nombre con el cual será identificada en el historial y el tipo de evaluación, pudiéndose elegir entre verificar por principios de accesibilidad (perceptible, comprensible, operable, robusto) o por niveles de conformidad (A-AA-AAA), el botón con signo de interrogación remite al usuario a una ayuda donde se le orienta sobre los principios de accesibilidad y niveles de conformidad, como puede observarse en la Figura 3 [4].

The figure consists of two screenshots of the EAW web application. The left screenshot shows a form titled 'Comienza una nueva evaluación' with fields for 'URL de la página que se desea evaluar', 'Nombre: (Indica un nombre para identificar el sitio evaluado)', and 'Tipo de evaluación:'. A note on the left says 'El usuario puede elegir si la evaluación la hará por principios de accesibilidad o por niveles de conformidad'. The right screenshot shows a dialog box for selecting an evaluation type ('Tipo de evaluación') with options: 'Evaluar por principio de accesibilidad' (selected) or 'Evaluar por nivel de conformidad'. Below it is another dialog for selecting a conformance level ('Selección uno o más principios de accesibilidad') with checkboxes for 'Perceptible', 'Comprensible', 'Operable', and 'Robusto'. The bottom right of the dialog has 'Cancel' and 'Aceptar' buttons.

Figura 3. Datos y Opciones para la Evaluación de una Página Web en EAW

El proceso de evaluación tiene una secuencia de eventos que inicia con el despliegue de la vista *index* (ver Figura 4 [4]), la cual muestra el historial de evaluaciones y la opción “Nueva evaluación”. Si el usuario elige evaluar un nuevo recurso, el enrutador recibe la petición, se comunica con el componente *WebPageController* (controlador) y se despliega al usuario la interfaz para indicar los datos de entrada del recurso (página Web) mediante la vista *create*.

Al enviar los datos, el método *store* los verifica, incluyendo que se suministre un URL válido, para lo cual se utiliza el método *validateURL* y el modelo *Webpage*, adicionalmente se verifica e informa si existen evaluaciones previas del recurso. Si no se ha registrado previamente una evaluación del recurso en el historial, se procede a crear una nueva entrada en la base de datos, utilizando los modelos *webpage*, *child* y *comment*.



Figura 4. Página Principal de EAW

El controlador suministra el URL verificado y bien formado al paquete *cURL* [20] a fin de crear un objeto *DOMDocument* de la página Web, considerando el URL suministrado como la raíz del árbol de enlaces [21][22], esto permitirá acceder al código de la página, principalmente su HTML y CSS. El controlador realiza peticiones a los modelos *tag* y *css* para ejecutar las evaluaciones de los criterios de accesibilidad en los elementos codificados en estos lenguajes de etiquetado según pautas específicas del WCAG 2.0.

Finalmente cada modelo suministra al controlador los resultados que son desplegados al usuario por la vista *show*. En la Figura 5 [4] se observa el diagrama de secuencia para la evaluación de un nuevo recurso por principio de accesibilidad.

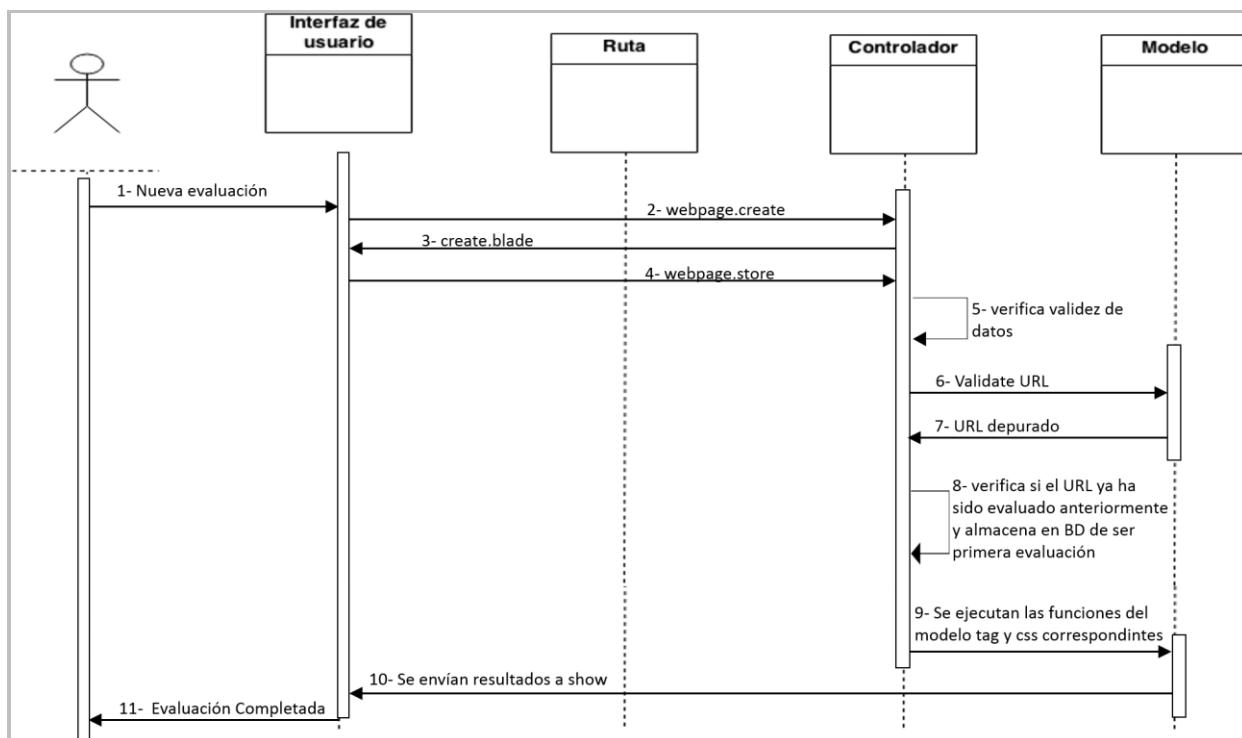


Figura 5. Diagrama de Secuencia para la Evaluación de un Recurso en EAW

El reporte de resultados de EAW se realiza con un despliegue progresivo de información, iniciando con la identificación de la página Web evaluada, una captura de pantalla de la página y sus enlaces de primer nivel, los botones de opciones, la accesibilidad global del recurso y la sección de comentarios.

En un segundo nivel de detalle, se presentan los resultados de cada elemento evaluado en la página agrupados por etiqueta HTML y una tabla con detalles más específicos como el número de línea en donde fue detectado el error, los atributos involucrados, el principio y si se aprueba o no el criterio de accesibilidad, como puede observarse en la Figura 6 [4].



Figura 6. Reporte de Resultados de la Evaluación de un Recurso en EAW

C. Método de Desarrollo y Tecnologías

Se utilizó una metodología de desarrollo *ad-hoc*, incorporando principios de Programación Extrema (XP) y utilizando algunos artefactos UML para la documentación en las fases de planeación y diseño, entre otros, bocetos de interfaces y diagramas de casos de uso (en lugar de historias de usuario); diagrama de componentes, diagramas de secuencia, diagrama entidad-relación, diagrama de navegación de la aplicación.

Para crear la herramienta se utilizaron las tecnologías Laravel como *framework* de desarrollo, PHP 5.4.3, HTML5, JavaScript, Bootstrap para el diseño de una interfaz

adaptativa con CSS3, MySQL como manejador de bases de datos, utilizando un servidor Web Apache. Para la gestión de la información de los URLs, comprobar la existencia del URL a evaluar, explorar el contenido de la página Web, obtener su árbol de enlaces y copiar el contenido del URL, se utilizó el paquete cURL de la librería libcurl. Compatible con PHP, cURL permite la conexión y comunicación con diversos servidores y protocolos, entre otros, http, https, ftp, gopher, telnet, dict, file y ldap [4][20].

En la Tabla II se muestra una comparativa entre las herramientas EAW, TAW y HEVAC, pudiéndose visualizar de manera más concreta características comunes y diferencias [4].

TABLA II. COMPARACIÓN DE LAS HERRAMIENTAS PARA EVALUACIÓN DE ACCESIBILIDAD TAW, HERA Y EAW

Características	TAW	HEVAC	EAW
Pautas	WCAG 1.0, WCAG 2.0, MobileOK	WCAG 2.0	WCAG 2.0
Tecnologías soportadas	HTML, CSS, JavaScript (parcialmente)	HTML	HTML, CSS
Técnicas HTML	Completa	Parcial, criterios de discapacidad visual	Parcial, criterios de discapacidad visual y motriz
Selección del nivel de conformidad (A-AA-AAA)	Sí	Sí	Sí
Selección de principios de accesibilidad	No	Sí	Sí
Clasificación de comprobaciones	Por numeración de criterios de éxito	Según técnicas propias de la herramienta	Por principios de accesibilidad y por niveles de conformidad
Se indican las comprobaciones no realizadas	Sí	No	Parcialmente, se indica el número de errores suministrado por el servicio de validación de marcado de la W3C ²
Agrupación de resultados por principios	Sí	Sí	Sí
Consejos de desarrollo	No (suministra enlace a la página de la técnica en el sitio WCAG)	Sí	Sí
Se indican el grado de accesibilidad	No	Sí	Sí
Pre-visualización de la página evaluada	Sí	No	Sí
Se destaca el problema en el código fuente	Sí	Sí	Sí
Exportación e impresión de resultados en PDF	No	Sí	Sí
Registro histórico de verificaciones	No	Sí	Sí

² W3C Markup Validation Service, <https://validator.w3.org>

TABLA II. COMPARACIÓN DE LAS HERRAMIENTAS PARA EVALUACIÓN DE ACCESIBILIDAD TAW, HERA Y EAW (CONTINUACIÓN)

Características	TAW	HEVAC	EAW
Verificación de tecnología HTML	Sí	Sí	Sí
Verificación de tecnología CSS	Sí	No	Sí
Incorporación de comentarios del usuario	No	No	Sí
Revisión de las páginas hijas del URL	Sí	No	Sí

En el proceso de desarrollo se realizaron pruebas unitarias a las funcionalidades. Una vez desarrollada EAW, se verificó la usabilidad de la aplicación con la participación de diez (10) usuarios con distinto nivel de experiencia en uso de aplicaciones Web y con perfiles de desarrollador de aplicaciones, docente y público general.

Luego de utilizar EAW los usuarios respondieron un cuestionario con doce (12) preguntas, siete (7) obligatorias de selección simple y cinco (5) optionales de respuesta abierta. De esta prueba se obtuvieron sugerencias para mejorar los textos de ayuda que explican los criterios de accesibilidad verificados y funcionalidades, el despliegue de información del historial de páginas Web, la búsqueda de las páginas registradas, la explicación de los resultados detallados de los errores, entre otras. Ante la solicitud de evaluar la funcionalidad general de la aplicación en una escala de 1 (mínimo) a 10 (máximo), el 70% de los usuarios la calificaron con el máximo puntaje de 10, el 20% la calificó con 9 y el 10% con 8. En términos globales las opiniones fueron favorables, destacando que la herramienta es intuitiva en su uso, además de útil para las personas con discapacidad, para quienes los apoyan en el uso de aplicaciones Web y para los desarrolladores.

Finalmente, una instancia de EAW fue sometida a evaluación para identificar y corregir problemas de accesibilidad, lo cual se describe en la próxima sección.

IV. USO DE LA HERRAMIENTA EAW

Como se ha mencionado anteriormente, la herramienta EAW extendió el alcance de la verificación para incluir criterios relacionados con la accesibilidad de usuarios con discapacidad motriz, complementando los asociados a la discapacidad visual incluidos en investigaciones previas.

Estas dos categorías son de interés por ser comunes en la población, no sólo como condiciones de nacimiento o congénitas, sino como consecuencia de accidentes, enfermedades o por el avance en la edad. Incluyen deficiencias físicas, limitaciones de la actividad y restricciones de la participación que también afectan a usuarios de aplicaciones, servicios y productos Web. La OMS estimó que para el 2011 más de 1000 millones de personas vivían en todo el mundo con alguna forma de discapacidad; para 2014 había aproximadamente 285 millones de personas con discapacidad visual, de las cuales 39 millones eran ciegas y 246 millones presentaban baja

visión [2]. Tomando en cuenta que la población está envejeciendo, que el riesgo de discapacidad es superior entre los adultos mayores y también que aumentan las enfermedades crónicas (diabetes, cardiovasculares, cáncer y trastornos de la salud mental), la OMS, indica que la prevalencia de la discapacidad seguirá aumentando en años futuros [23].

Estos datos deben llamar a conciencia sobre la importancia de crear recursos que cumplan con los principios de accesibilidad para garantizar un diseño universal³ [24].

EAW se utilizó para evaluar sitios Web correspondientes a instituciones públicas y privadas en área que consideramos de importancia para el acceso a servicios, contenidos y el desarrollo de la ciudadanía digital. Ejemplos de los resultados de estas evaluaciones se presentan en la Tabla III, mostrando el porcentaje de accesibilidad, de recursos con problemas y el error de mayor incidencia en un conjunto de sitios Web relacionados con los sectores educación universitaria, instituciones gubernamentales y banca.

TABLA III. USO DE EAW PARA EVALUACIÓN DE SITIOS WEB DE INSTITUCIONES EN EL SECTOR EDUCATIVO, SALUD Y GOBIERNO (AÑO 2014)

Sitio Web	Característica Porcentaje de accesibilidad del sitio Web	Porcentaje de recursos con problemas	Error con mayor incidencia
Escuela de Computación UCV	93,27%	6,73%	a
Escuela Biología UCV	84,78%	15,22%	b
UCAB, Caracas	99,71%	0,29%	b
USB	91,48%	8,52%	a
Consejo Nacional Electoral	85,11%	14,89%	c
Gobernación de Miranda	85,11%	14,89%	c
Seniat	51,75%	48,25%	c
Banco de Venezuela	91,99%	8,01%	a
Banesco	83,05%	16,95%	a
Banco Mercantil	97,05%	2,95%	b

Los errores de mayor incidencia fueron:

- a. Uso del mismo valor para el atributo *name* en la etiqueta *<a>*, es decir, varios enlaces tienen el mismo nombre y esto ocasiona ambigüedad al momento de orientar al usuario en la navegación de la página Web.
- b. La etiqueta ** debe tener el atributo *alt*, lo cual se traduce en que no se incluyó texto alternativo para señalar la existencia de la imagen y describirla. Si la imagen no puede ser mostrada por el navegador o la página está siendo usada por una persona con discapacidad visual, el atributo *alt* permitiría indicar la presencia de la imagen, así como describírsela en la traducción de texto a voz del contenido de la página con el apoyo de alguna herramienta tiflotecnológica.

³ Diseño de productos, servicios o entornos para la mayor cantidad de usuarios posible, sin que tengan la necesidad de adaptarlos o rediseñarlos. Estos principios generales del diseño, son aplicables en la arquitectura, la ingeniería y también en la informática [24].

- c. En las tablas debe utilizarse las etiquetas `<tr>` y `<th>`, esto implica que las tablas deben presentar una estructura correcta, completa en sus etiquetas y atributos, y con delimitación de filas, celdas de título, celdas de datos y título que identifique a la tabla.

Otro error detectado con frecuencia fue la falta del atributo `tabindex` en los campos de los formularios, lo cual dificulta establecer el orden en que se recorren los campos utilizando el teclado, opción utilizada tanto por las personas con discapacidad motora, como con discapacidad visual.

Analizando los resultados de EAW para esta muestra de diez (10) sitios Web, sólo el 50% presenta un porcentaje de accesibilidad superior al 90%. Debe destacar que los sitios con el menor porcentaje de accesibilidad se relacionan con servicios bancarios y tributarios, en este caso, el sitio web de Banesco (83,05%) y del Seniat (51,75%).

El 20% (2 de 10) presenta menos del 5% de errores en los criterios de accesibilidad verificados, el 40% presenta entre 5% y 15% de criterios con problemas, y el 30% tiene más de 15% de criterios con errores de accesibilidad. En este indicador también destaca el sitio web del Seniat con 48,25% de recursos con problemas de accesibilidad.

En términos más generales, las dificultades que comúnmente afectan el acceso de las personas con limitación o discapacidad motora están relacionadas con el uso del ratón, el teclado y las pantallas táctiles como dispositivos para orientar la navegación y la selección de opciones en el sitio Web.

En el caso del uso del ratón puede que los usuarios no tengan la precisión o coordinación necesaria para ejecutar los movimientos; en el caso del teclado necesitan tener la fuerza y precisión para teclear, y en las pantallas táctiles la precisión, presión, control del tiempo y del movimiento para escribir o seleccionar las opciones. Algunas personas mayores y personas afectadas por la artritis u otras inflamaciones en las extremidades superiores, codos, manos, presentan dolores en las articulaciones que pueden causar fatiga y limitar el tiempo de empleo del ratón o del teclado.

En el caso de las personas con discapacidad visual, las limitantes más comunes se relacionan con no poder tener acceso a toda la información incluida en la página Web, no poder modificar el tamaño en que se presentan los contenidos o el uso de colores que no tienen buen contraste.

En las páginas Web se incorporan estructuras mal diseñadas, en especial tablas, capas, marcos, imágenes, animaciones, videos, botones e hipervínculos. También es común que no se incluyan identificadores únicos, metadatos, descripciones y textos alternativos. Estos errores son muy comunes a pesar de la existencia de pautas de accesibilidad Web que todo programador debería conocer y emplear.

Es importante realizar una correcta programación de estos elementos, en cuanto a etiquetas y atributos HTML o CSS que los conforman, pero también incluir información descriptiva, de manera que usuarios con discapacidad visual puedan valerse de otros canales de percepción para acceder a los contenidos. Estos usuarios pueden recurrir a técnicas

como el cambio de tamaño de las fuentes, combinaciones de colores con un alto contraste o apoyarse en el uso de software o dispositivos de hardware, por ejemplo, el uso de magnificadores de pantalla, ampliadores de imagen, sintetizadores de voz, grabadoras de sonido e incluso salidas en Braille, entre otras herramientas tiflotecnológicas.

Otra evaluación importante que se realizó en el marco de esta investigación se aplicó sobre la misma herramienta EAW. Debido a que también es una aplicación Web, era lógico y necesario verificar su grado de accesibilidad. Para ello se realizó una evaluación de pautas según los principios de accesibilidad (perceptible, comprensible, operable, robusto) sobre una versión espejo. Se detectaron algunos errores que fueron corregidos y se repitió la verificación hasta obtener como resultado un 100% de accesibilidad del recurso y un 0% de elementos con problemas, sobre 167 elementos evaluados que incluyeron etiquetas de títulos, imágenes, tablas, párrafos, hipervínculos, campos y nombres de campos. También se realizó la evaluación de EAW mediante el *W3C Markup Validation Service*, identificándose errores asociados a pautas que no están incluidas entre las verificaciones propias de la herramienta.

V. RESULTADOS

El resultado principal es el desarrollo del Evaluador de Accesibilidad Web, herramienta que permite la verificación del nivel de accesibilidad de una página Web e indica los errores encontrados en los recursos evaluados en ella. Esta herramienta amplía el alcance de investigaciones previas del CENEAC, específicamente de la aplicación HEVAC, al incluir criterios relacionados con la discapacidad motriz y la validación de código en estilos CSS, además de los criterios relacionados con la discapacidad visual y la verificación de HTML. EAW es una aplicación Web, disponible en línea, de libre acceso, que no requiere un proceso de instalación.

En un nivel más detallado, EAW identifica y destaca en su reporte de resultados las sentencias con errores, indica la naturaleza del problema según el criterio incumplido clasificando por etiqueta y por principio de accesibilidad, suministra recomendaciones para corregir cada error y permite socializar las evaluaciones mediante los comentarios que se pueden agregar a cada evaluación realizada.

Para cada URL suministrado, se obtiene una evaluación de la página Web a la cual enlaza y de las páginas que conforman el primer nivel de hipervínculos disponibles. Estas evaluaciones están a disposición del usuario en un historial de páginas revisadas, con la posibilidad de eliminar un resultado o actualizar una evaluación sin necesidad de volver a registrarla.

VII. CONCLUSIONES

En este estudio se utilizan los criterios para el desarrollo Web accesible y para la evaluación de accesibilidad desarrollados por la W3C a través de la *Web Accessibility Initiative* (WAI), en especial las pautas del estándar WCAG 2.0. Sin embargo, otros lineamientos de importancia para un correcto diseño, usabilidad y accesibilidad de recursos

informáticos se compilán en los estándares de la ISO, AENOR, y ANSI [26][27], entre otros, en la Sección 508, la Norma Requisitos de accesibilidad para contenidos en la Web (UNE 139803:2012) y en la *Accessible Rich Internet Applications* (WAI-ARIA).

El beneficio de herramientas como HEVAC y EAW tiene un alcance más amplio que el entorno de desarrolladores de aplicaciones y sitios Web. Pueden ser usadas por personas encargadas de seleccionar, recomendar o evaluar recursos con fines educativos, informativos, acceso a gobierno electrónico, gestión en línea, servicios públicos y a plataformas en línea, en donde debe garantizarse que todas las personas puedan realizar trámites, consultar información o utilizar productos y servicios.

Indiscutiblemente la información proporcionada por medio del Internet debe estar a disposición de la mayor cantidad de personas, más aún cuando vivimos en una sociedad que debe promover la inclusión. Es por ello que no se puede descuidar o ignorar las necesidades de los usuarios que presentan alguna limitación o discapacidad.

El proceso de verificación que se realizó con HEVAC y con EAW dejó entrever que un gran número de páginas Web de instituciones de educación superior, gobierno y banca se enfocan en entregar un diseño de calidad, en algunos casos con excelente presentación de contenido gráfico y textual, pero descuidan los aspectos de accesibilidad, dificultando su navegación y uso, especialmente por parte de personas con discapacidad visual o motora.

Es fundamental que se haga del diseño accesible una norma, garantizando el acceso a la información y servicios, no agravando la brecha digital o infoexclusión. En palabras de Tim Berners-Lee, creador de la World Wide Web, el poder de la Web está en su universalidad, siendo un aspecto esencial garantizar el acceso de todos, independientemente de su condición o discapacidad (Figura 7).

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect".

Sir Tim Berners-Lee.

Figura 7. El Poder de la Web Reside en su Universalidad

REFERENCIAS

- [1] WAI, *Web Accessibility Initiative*, Consorcio World Wide Web (W3C), <http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>
- [2] OMS, *Discapacidades*, Organización Mundial de la Salud, <http://www.who.int/topics/disabilities/es>
- [3] L. Valdez, *Discapacidad Visual*, Departamento de Educación Especial, Dirección Provincial de Educación del Guayas, Ecuador, 101 p. 2011.
- [4] M. Acosta, *Evaluador de Criterios de Accesibilidad Web para Pautas Relacionadas con Discapacidad Visual y Discapacidad Motora*. Trabajo Especial de Grado (Licenciatura en Computación). Tutora: Profa. Yusneyi Carballo Barrera. Caracas (Venezuela): Universidad Central de Venezuela. 137 p. 2014, http://saber.ucv.ve/bitstream/123456789/15048/1/Tesis_Ronald_Aguilera_Jun2012_conActa.pdf
- [5] WCAG 1.0, *Web Content Accessibility Guidelines 1.0*, W3C-WAI, <https://www.w3.org/WAI/intro/wcag10docs>
- [6] WCAG 2.0, *Web Content Accessibility Guidelines 2.0*, W3C-WAI, <https://www.w3.org/TR/WCAG20>
- [7] S. Luján Mora, Accesibilidad Web, Universidad de Alicante, <http://accesibilidadweb.dlsi.ua.es/?menu=pautas-accesibilidad-contenido-web>
- [8] WebAIM, *Constructing a POUR Website, Putting People at the Center of the Process*, <http://webaim.org/articles/pour>
- [9] *Evaluating Accessibility*, W3C-WAI, <https://www.w3.org/WAI/eval/Overview.html>
- [10] Y. Hassan y F. Martín, *Qué es la Accesibilidad Web*, No Solo Usabilidad, ISSN 1886-8592, 2013, <http://www.nosolousabilidad.com/articulos/accesibilidad.htm>
- [11] A. Olalere and J. Lazar, *Accessibility of U.S. Federal Government Home Pages: Section 508 Compliance and Site Accessibility Statements*. Government Information Quarterly 28(3), pp. 303-309, 2011.
- [12] E. Jackson-Sanborn, K. Odess-Harnish and N. Warren, *Website Accessibility: A Study of Six Genres*, Library Hi Tech, 20(3), pp. 308-317, 2002.
- [13] E. Loiacono, S. McCoy and W. Chin, *Federal Website Accessibility for People with Disabilities*, Information Technology Professional, 7(1), 27-31, 2005.
- [14] Government-wide Section 508 Accessibility Program (GSA), <https://www.section508.gov>
- [15] WebAIM, *Accessibility Evaluation Tools*, <http://webaim.org/articles/tools/>
- [16] WebAIM, *Web Accessibility Evaluation Tool*, <http://wave.webaim.org>
- [17] TAW, *Test de Accesibilidad Web*, Fundación CTIC, España, <http://www.tawdis.net>
- [18] HERA, *Hojas de Estilo para la Revisión de la Accesibilidad*, Fundación Sidar, España, <http://www.sidar.org/hera>
- [19] R. Aguilera, *Desarrollo de una Herramienta para Verificación de Criterios de Accesibilidad en Sitios Web*. Trabajo Especial de Grado (Licenciatura en Computación), Tutora: Profa. Yusneyi Carballo Barrera. Caracas (Venezuela): Universidad Central de Venezuela. 110 p. 2012, http://saber.ucv.ve/bitstream/123456789/15048/1/Tesis_Ronald_Aguilera_Jun2012_conActa.pdf
- [20] PHP.net, *Biblioteca URL Cliente cURL*, The PHP Group, <http://php.net/manual/es/book.curl.php>
- [21] W3Schools, *The HTML DOM Document Object*, https://www.w3schools.com/jsref/dom_obj_document.asp
- [22] PHP.net, *Clase DOMDocument*, The PHP Group, <http://php.net/manual/es/class.domdocument.php>
- [23] OMS, *Resumen Informe Mundial sobre la Discapacidad*, Organización Mundial de la Salud, 27 p. 2011.
- [24] NCSU, *Universal Design Resources*, Center for Universal Design, College of Design, North Carolina State University, https://www.ncsu.edu/ncsu/design/cud/about_ud/about_ud.htm
- [25] AGESIC, *Gobierno en Red*, Agencia de Gobierno Electrónico y Sociedad de la Información y del Conocimiento, Uruguay Digital, <https://www.agesic.gub.uy/innovaportal/v/168/1/agesic/principios-y-lineamientos-.html?idPadre=26>
- [26] C. Varela, A. Miñán, J. Hilera, F. Restrepo, H. Amado, M. Córdova y A. Villaverde, *Estándares y Legislación sobre Accesibilidad Web*, Actas del IV Congreso Internacional ATICA, Loja, Ecuador, pp. 46-53, 2012.
- [27] *Normas de Accesibilidad*, Portal de Administración Electrónica, Ministerio de Hacienda y Función Pública, Secretaría General de Administración Digital, Gobierno de España, http://administracionelectronica.gob.es/pae/Home/pae_Estrategias/pa_e_Accesibilidad/pae_normativa/pae_eInclusion_Normas_Accesibilida d.html#.WWYswIG1N

Propuesta para la Gestión de Proyectos Socio Tecnológicos del Programa Nacional de Formación en Informática

Iris Albaran, Liliana Silva, Colombia Amezquita, Marbella Castañeda

IUT "Dr. Federico Rivero Palacio"
Caracas, Venezuela

idealbarranp@iutfrp.edu.ve, iutrc.liliana@gmail.com, ceamezquitaz@iutfrp.edu.ve, mdrcastanedar@iutfrp.edu.ve

Resumen—A partir del año 2008, en el IUT “DR. Federico Rivero Palacio” comienza la gestión del Plan Nacional de Formación en Informática (PNFI). Este incorpora aspectos importantes como la formación por competencias y la definición del currículo por ejes transversales y longitudinales. El Proyecto Socio Tecnológico (PST), el cual forma parte del eje longitudinal, se plantea como una Unidad Curricular cuyas actividades en desarrollo propician la vinculación entre la Universidad y la Comunidad, mediante la satisfacción mutua de necesidades comunes. El objetivo de este trabajo es facilitar una propuesta que sustente la gestión y administración del PST y así garantizar que los productos de software generados por los estudiantes en la unidad curricular PST, sean de la satisfacción de las comunidades con las cuales se les vincula, adicional al logro académico esperado. Para ello se generó una propuesta de Gestión del Eje, aplicada a cuatro cohortes académicas.

Palabras Clave—Gestión-de-Proyectos, Proyecto-Socio Tecnológico, Comunidad.

I. INTRODUCCIÓN

En el año 2008, con el fin de regular los Programas Nacionales de Formación en Educación Superior, se crean los distintos Planes Nacionales de Formación (PNF). A partir de ese año se comienza a gestionar en el IUT “DR Federico Rivero Palacio” el Programa Nacional de Formación en Informática (PNFI), esto condujo a cambios en la gestión del currículo en cuanto a la estructuración y el abordaje de los conocimientos los cuales se enmarcan por ejes transversales y longitudinales. Esta estructuración y la forma de abordar el conocimiento fueron pensados con la finalidad de adecuar una currícula innovadora que orbitase alrededor de los retos de formación propios de la época y que son planteados por la UNESCO 2006.

El PNFI [1] plantea la formación total en cuatro (4) trayectos, generando en cada uno de los mismos una certificación o un título asociado a los conocimientos y competencias del perfil de egreso definido en el currículo. En cada uno de los trayectos se gestiona y administra un PST, el aprendiz a través de esta Unidad Curricular (UC) debe demostrar y aplicar los saberes asociados de las diferentes unidades curriculares generando un producto informático que dé solución a problemas reales de la comunidad, de manera que se refuerza la interacción Comunidad - Universidad

Es importante resaltar que desde el año 2008 hasta la fecha, se han desarrollado productos para las comunidades a través de los PSTs. Los resultados desde el punto de vista académico, en la mayoría de los casos, suelen ser satisfactorios, no obstante, no siempre se logra satisfacer las necesidades de las comunidades. Pareciera que esto se debe a que los tiempos de desarrollo (tiempo académico) del software superan el tiempo requerido por el usuario para utilizar el producto. Otro factor que ocurre con menos frecuencia pero que ha incidido en las fallas de las soluciones a la comunidad, está relacionada a la disgregación del equipo de trabajo, que se interesó en desarrollar el proyecto en su inicio, pero no logra dar respuesta al ente externo, por no contar con todos los recursos humanos y operativos. Así mismo, pueden ocurrir otros factores de riesgo en la aceptación del producto por parte de la Comunidad y esto se debe en algunos casos, a los cambios en las dependencias del ente, sobre todo en los niveles organizacionales decisores cuando el proyecto está avanzado y se originan nuevos requerimientos.

La problemática expresada anteriormente nos invita a reflexionar y a formularnos la siguiente interrogante: ¿De qué manera se puede lograr una solución en tecnología informática, real y adecuada, que satisfaga las necesidades de las comunidades en el tiempo requerido y que además, se garantice que los equipos de trabajo demuestren la aplicación rigurosa de los conocimientos y competencias que exige la academia?

Para dar respuesta a esta interrogante nos plantearnos como objetivo: proporcionar una guía de Gestión, para el desarrollo del Proyecto Socio Tecnológico, de manera que se puedan garantizar proyectos reales para las comunidades y que los estudiantes logren integrar los conocimientos y competencias aprendidas en las unidades curriculares del trayecto, especialmente la aplicación de las metodologías y modelos establecidos por la Ingeniería del Software y los perfiles de egreso planteados en el PNFI. Para construir la propuesta, se realizó el análisis del currículo basado en el eje proyecto, los diferentes perfiles de egreso y los ejes transversales que agregan entre otros los conocimientos propios de la profesión. Adicionalmente se realizó la propuesta de gestión de los proyectos Informáticos y por

último se estructuraron las estrategias de planificación y seguimientos por trayecto y por etapas.

II. MARCO REFERENCIAL

A. Programa Nacional de Formación en Informática (PNFI)

Por Resolución N° 2.963 del Ministerio del Poder Popular para la Educación Superior (MPPES), con fecha 14 de mayo de 2008, gaceta oficial N° 38.930, el Ejecutivo resuelve regular los Programas Nacionales de Formación en Educación Superior. Para ello define en el Artículo 2:

“... se entiende por: Programas Nacionales de Formación en Educación Superior: El conjunto de actividades académicas, conducentes a títulos, grados o certificaciones de estudios de educación superior, creados por iniciativa del Ejecutivo Nacional, a través del Ministerio del Poder Popular para la Educación Superior, diseñados con la cooperación de Instituciones de Educación Superior Nacionales, atendiendo a los lineamientos del Plan de Desarrollo Económico y Social de la Nación, para ser administrados en distintos espacios educativos del territorio Nacional.”

El Programa Nacional de Formación en Informática (PNFI) fue creado el 07 de octubre de 2008 en la resolución N° 3147. En la actualidad el PNFI es gestionado en 33 instituciones entre las que se encuentran: Institutos Universitarios de Tecnología (IUT), los Colegios Universitarios (CU) y las Universidades Nacionales Experimentales (UNE).

El PNFI presenta un currículo por competencias.

B. El Proyecto Socio Tecnológico (PST) en el Programa Nacional de Formación en Informática (PNFI)

La Unesco (2013) [2] plantea tres aspectos importantes relacionados con el qué y cómo educar, ellos son:

- 1) la necesidad de adoptar y desarrollar un enfoque integral al aprendizaje que considere no solamente los conocimientos académicos, el desarrollo cognitivo y las capacidades, sino también las dimensiones “no cognitivas” – actitudes, valores, emociones, cualidades personales
- 2) la exigencia de considerar la dimensión aplicada del conocimiento, puesto que no solamente cuenta lo que se sabe sino también lo que se puede hacer con este saber
- 3) la importancia de repensar enteramente la estructura disciplinar tradicional del currículo, la organización de las experiencias de aprendizaje, la manera de enseñar y los sistemas de evaluación si se quiere promover el desarrollo efectivo de competencias.

Se ha entendido que el planteamiento anterior refleja el espíritu concebido en la fundamentación del PNFI. Al respecto, en la construcción del conocimiento en el mencionado Plan, se estructura en cinco ejes temáticos, tres ejes transversales que incluyen los ejes temáticos: (1) Epistemológico-Heurístico, (2) Socio-Cultural-Económico-Histórico-ÉticoPolítico y (3) Profesional y Estético-Lúdico y

Ambiental; y dos ejes longitudinales conformados por (1) Proyecto Socio Tecnológico y (2) Formación Crítica.

En la fundamentación del PNFI se destacan tres características a resaltar de proyecto [1]:

- 1) “El Proyecto Socio Tecnológico (PST) etimológicamente se relaciona con la palabra socio proveniente del latín, *socius* lo cual significa grupo humano. Por su parte, tecnológico se asocia con tecnología, correspondiente a fabricar objetos, productos o servicios y modificar el medio ambiente, lo cual genera una combinación adecuada a la formación del participante del programa, a su inserción y contacto con la realidad. En ese contexto es señalado por el Diccionario de la Real Academia Española (2007).”
- 2) “Constituye por tanto, el PST, el núcleo central del Programa Nacional de Formación en Informática, referido como una unidad curricular en cada uno de los trayectos con una importante carga crediticia. De este modo, representa un eje longitudinal transversal que orienta y define el resto de las demás unidades curriculares, desarrollándose de forma incremental, aumentando su nivel de complejidad y profundidad en cada trayecto.”
- 3) “El PNFI propone el desarrollo de Proyectos Socio Tecnológicos como estrategia de aprendizaje que permite la construcción del conocimiento a partir del aprender haciendo, donde se propicia el reconocimiento en principio por el propio participante de sus conocimientos, habilidades y destrezas, que luego debe desarrollar a partir del Proyecto Socio Tecnológico convirtiéndose en crecimiento personal y confianza en el participante de su proceso formativo y del rol profesional a desempeñar”

El Proyecto Socio Tecnológico (PST) procura proporcionar un aprendizaje significativo que integre los saberes, valores, aptitudes, actitudes, habilidades y destrezas de un estudiante y donde el resultado se materialice en productos de software y que estos tengan un impacto dentro de las comunidades vinculadas proporcionando soluciones a problemas reales.

En el documento rector del PNFI se presentan las siguientes características del PST [1]:

- El planteamiento del proyecto se basa en un problema real local, regional o nacional que incorpore las áreas de los saberes de la informática.
- Provee oportunidades para que los participantes realicen investigaciones que les permitan aprender nuevos conceptos, aplicar la información y representar el conocimiento de diversas formas.
- Provee la posibilidad de trabajo en equipo y colaboración entre los participantes, profesores asesores y otras personas involucradas con el proyecto a fin de que el conocimiento sea compartido y distribuido.

- Posibilita el uso de herramientas cognitivas y ambientes de aprendizaje que motivan al participante a representar sus ideas y fomentar la construcción de sus conocimientos.
- Pueden abarcar más de un trayecto dependiendo de su objetivo y complejidad.
- Demandan la aplicación de conocimientos interdisciplinarios. Así, el participante puede apreciar la relación existente entre las diferentes disciplinas en el desarrollo de un proyecto en particular.
- Permiten la búsqueda de soluciones abiertas, dando así oportunidad al participante de generar nuevos conocimientos.

En general las propuestas de PST provienen de comunidades públicas o privadas las cuales plantean necesidades de automatización de acuerdo a la naturaleza de sus funciones. Se establece el vínculo Comunidad-Universidad a través de un PST asignado a uno o más grupos de estudiantes, dependiendo de la complejidad y el perfil de egreso del estudiante.

Si en este punto se triangula la propuesta de [1] y [2], encontramos que es un reto mejorar la Unidad Curricular Proyecto, la cual se considera además, un eje dentro del currículo y una estrategia de aprendizaje, de manera que garantice en dicha UC, el logro de las competencias de acuerdo al perfil y adicionalmente que el producto obtenido de ese logro sea de provecho directo a la comunidad con la cual se vincula.

C. Perfiles de Egreso del Programa Nacional de Formación en Informática (PNFI)

El perfil de egreso se refiere a las competencias logradas por un estudiante una vez alcanzado un nivel académico determinado, de acuerdo a lo planteado en el diseño curricular.

El PNFI presenta cuatro perfiles de egreso correspondiente a cada uno de los trayectos: dos constituyen títulos de nivel superior y dos son salidas intermedias o certificaciones. En la Tabla I, se muestran las salidas correspondientes a los perfiles según los trayectos académicos.

TABLA I. PERFILES DE EGRESO POR TRAYECTOS

Trayecto I	Trayecto II	Trayecto III	Trayecto IV
Soporte Técnico a usuarios y Equipos	TSU en Informática	Desarrollador(a) de Aplicaciones	Ingeniero(a) en Informática

Según lo presentado en [1] el egresado como TSU del PNFI: “es un profesional con formación integral, que se desempeña con idoneidad operativa y ética profesional en la construcción de productos tecnológicos informáticos en armonía con la preservación del ambiente y del progreso de su entorno, aplicando los saberes para:

1. Desarrollar y mantener componentes de software bajo estándares de calidad, priorizando el uso de software libre.
2. Caracterizar, seleccionar, ensamblar, configurar y mantener equipos informáticos.
3. Interpretar el modelo de datos e implementar y mantener, de forma operativa, las bases de datos.
4. Instalar, configurar y administrar operativamente redes de área local, bajo estándares de calidad, priorizando el uso de software libre.
5. Participar técnicamente en el proceso de evaluación, selección e instalación de software”

Así mismo define al Ingeniero o la Ingeniera en Informática como: “un profesional con formación integral que se desempeña con idoneidad y ética profesional, en la conceptualización y construcción de productos tecnológicos informáticos en armonía con la preservación del ambiente y del progreso de su entorno, aplicando los saberes para:

1. Participar en la administración de proyectos informáticos bajo estándares de calidad y pertinencia social.
2. Auditarse sistemas informáticos.
3. Desarrollar e implantar software bajo estándares de calidad y pertinencia social, priorizando el uso de plataformas libres.
4. Integrar y optimizar sistemas informáticos.
5. Diseñar, implementar y administrar bases de datos.
6. Diseñar, implementar y administrar redes informáticas bajo estándares de calidad, priorizando el uso de software libre”.

En el mismo orden de idea las salidas intermedias constituyen un reconocimiento a las habilidades y destrezas adquiridas en el desarrollo de los PSTs en los que participa, y para obtenerlas se comprueba que posee los saberes necesarios. En el caso del certificado en “Soporte Técnico a Usuarios y Equipos” debe demostrar que posee los saberes para aplicar los conocimientos en:

- “Ensamblar, configurar y realizar mantenimiento preventivo y correctivo de equipos de computación de acuerdo a los requerimientos del usuario.
- Realizar soporte técnico a usuarios y equipos.
- Participar técnicamente en el proceso de evaluación, selección e instalación de software.”

De manera similar, para obtener la certificación de “Desarrollador de Aplicaciones”, en el tercer trayecto, el estudiante comprueba mediante las actividades de PST que es capaz de aplicar los conocimientos para:

- “Desarrollar y mantener aplicaciones y componentes de software,
- Interpretar el modelo de datos e implementar y mantener, de forma operativa, las bases de datos.
- Instalar, configurar y administrar operativamente redes de área amplia.”

Se aprecia que algunos saberes se repiten en los perfiles de egreso, debido a que la formación es continua a lo largo de los trayectos y las competencias se van incrementando a lo largo de la ejecución del Plan.

D. Gestión de Proyectos Informáticos

Según [3] la gestión de un proyecto se define como: “Articular el método para alcanzar un objetivo único y no repetitivo en un plazo con principio y fin claros utilizando las técnicas que nos proporciona la gestión.”, indican los autores además, que las tareas primordiales son: planificar y establecer estrategias adecuadas, organizar a los miembros y equipos para lograr los objetivos que se quieren alcanzar, controlar y comprobar si se están alcanzando dichos objetivos. Según estos autores “la organización de un proyecto consiste en diseñar la estructura con la que vamos a establecer las dependencias entre individuos, departamentos, cosas... dentro del proyecto. Asimismo, debemos asignar las tareas más idóneas para esas capacidades y el tiempo estimado para cumplir las tareas o funciones”.

La gestión de los proyectos tiene dos aspectos que se resaltan en este trabajo, una son las fases y el otro es el seguimiento y control del proyecto.

1. Fases de un Proyecto

Según la metodología PMBOK [4] “Las fases del proyecto son divisiones dentro del mismo proyecto, donde es necesario ejercer un control adicional para gestionar eficazmente la conclusión de un entregable mayor”. Las fases de un proyecto se pueden relacionar de diferentes maneras una de forma secuencial donde el inicio de una depende de la culminación de la anterior, de una manera superpuesta, donde la fase siguiente comienza antes de que culmine la anterior y la tercera corresponde con una relación iterativa donde en un momento dado solo se planifica una fase y la siguiente depende del avance de proyecto y la generación de los entregables en la etapa actual. En la Figura 1, se muestra la estructura básica de un proyecto donde se agrupan diferentes componentes en cada fase.



Figura 1. Fases Principales de un Proyecto

Se observa que se propone como primera fase la planeación la cual incluye la definición del problema y la planificación propiamente dicha, en la segunda fase denominada ejecución se presenta la puesta en marcha, la fase productiva y la conclusión del proyecto, es decir contiene

todas las fases de desarrollo del software incluyendo la transición del producto a la comunidad y finalmente incorpora la fase de mantenimiento, recomendada para productos de software que ya se encuentran en producción.

Adicionalmente a lo planteados por [3], [4] incorpora algunos aspectos que deben considerarse en la fase de conclusión de proyectos, considerada en [4] como una fase y que incluye actividades como las siguientes:

- obtener la aceptación del cliente o del patrocinador,
- realizar una revisión tras el cierre del proyecto o la finalización de una fase,
- registrar los impactos de la adaptación a un proceso,
- documentar las lecciones aprendidas,
- aplicar actualizaciones apropiadas a los activos de los procesos de la organización,
- archivar todos los documentos relevantes del proyecto en el sistema de información
- para la dirección de proyectos para ser utilizados como datos históricos y
- cerrar las adquisiciones.

2. Seguimiento y Control del Proyecto

Las actividades de seguimiento y control, están constituidas por aquellas que reflejan todas las actividades requeridas para supervisar, analizar y regular el progreso y el desempeño del proyecto. El seguimiento de estas actividades arroja resultados que están relacionados con la dinámica propia del trabajo en cuestión, de manera que permite hacer correcciones, ajustes y algunos casos cambiar el rumbo inicialmente pautado. De manera que el seguimiento y control del proyecto es un proceso continuo y dinámico que requiere un esfuerzo constante de los involucrados. El grupo de seguimiento y control según [4] debe:

- Dar seguimiento y controlar el trabajo del proyecto
- Realizar control integrado de cambios
- Verificar el alcance
- Controlar el alcance
- Controlar el cronograma
- Controlar los costos
- Realizar control de calidad
- Informar el desempeño
- Dar seguimiento y control de riesgos
- Administrar las adquisiciones.

Esto permitirá a los involucrados: Controlar cambios, dar seguimiento a las actividades del proyecto y controlar los cambios de manera que se implementen cambios aprobados y no desvíe el alcance del proyecto.

E. Proceso Unificado de Desarrollo (RUP)

El proceso Unificado de Desarrollo fue creado por Booch, Jacobson, y Rumbaugh, en el año 2000 [5], con el fin de mitigar los riesgos en el proceso de desarrollo de software. Este proceso contiene todas las actividades necesarias para

convertir los requisitos de un usuario en un sistema software. Sigue un proceso iterativo e incremental.

Fases del Proceso Unificado de Desarrollo:

1. Fase de comienzo o inicio: el objetivo en esta etapa es concretar la visión del proyecto. Es la etapa donde se establecen los requerimientos y se delimita el alcance del proyecto. Finalmente se determinan los objetivos del ciclo de vida.
2. Fase de elaboración: en esta fase se planifican las actividades a ejecutar y los recursos necesarios, haciendo énfasis en la arquitectura del software a utilizar. Al término de esta fase se llega al punto de no retorno del proyecto, ya que luego de ésta se debe afrontar la fase de construcción que es la más costosa y arriesgada.
3. Fase de construcción: es la fase en la que se desarrolla el producto y se observa la evolución de la visión, la arquitectura y los planes hasta obtener la primera versión de lo que será manejado por el usuario final. En esta fase se hace especial énfasis en el control de las operaciones realizadas, administrando los recursos eficientemente de tal forma de minimizar costes, cumplir con el calendario y garantizar la calidad.
4. Fase de transición: en esta fase se efectúa la transición del producto a los usuarios, lo cual incluye: manejo del producto, envío, entrenamiento, documentación, soporte y mantenimiento del producto hasta que el cliente esté satisfecho.

Cada una de estas fases es desarrollada mediante el ciclo de iteraciones. Una iteración es un ciclo de desarrollo completo dando como resultado una entrega de producto ejecutable.

III. METODOLOGÍA

La necesidad de generar una propuesta de Gestión del eje longitudinal PST, que optimice la vinculación Universidad–Comunidad, se inicia con un proceso de entrevistas informales hechas a representantes de las comunidades que participaron como beneficiarios de los productos tecnológicos elaborados por estudiantes de las cohortes 2013-2014, 2014-2015, 2015-2016 y 2016-2017.

La propuesta se construye básicamente a partir del análisis documental de fuentes asociadas al problema abordado, el cual, aunado a la experiencia aportada por profesores del área, permitió extraer aspectos claves para la integración en una Guía, elementos que propicien: logro académico- satisfacción de la comunidad.

Elaborada la propuesta, y a fines de una primera valoración, se procedió a su aplicación en los trayectos correspondientes a las cohortes 2013-2014, 2014-2015, 2015-2016 y 2016-2017.

IV. RESULTADOS

Los resultados se presentan en tres etapas. La primera refleja el análisis del currículo basado en el eje proyecto, la

segunda presenta la propuesta de gestión de eje y la tercera etapa muestra la valoración de la propuesta.

A. Análisis del Currículo Basado en el Eje Proyecto

Se realizó el análisis de las Unidades Curriculares que contribuyen al desarrollo del eje longitudinal Proyecto por cada trayecto y que perfilan las competencias esperadas, con el fin de discernir la relación de estas con el perfil de egreso de cada trayecto.

En la Figura 2, denominada Visión Sistémica del Currículo, adecuada por [6], se muestra una estructuración simple del PNFI, especificándose en los laterales los ejes Longitudinales PST y Formación Crítica, y en el centro se hace la clasificación por áreas de las UCs que constituyen los ejes Transversales. Se agruparon bajo el eje de formación básica algunas UCs no incluidas en las áreas presentadas en el documento rector pero que han sido incluidas como cursos en el mismo.

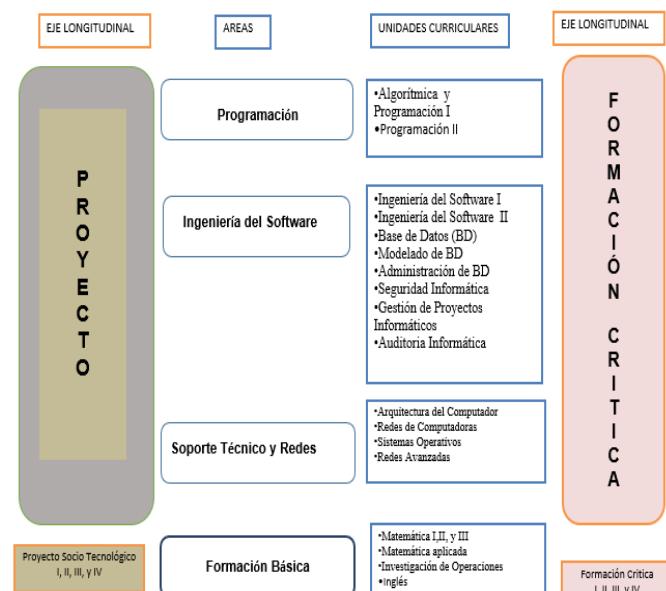


Figura 2. Visión Sistémica del Currículo

A continuación se presentan por cada trayecto una tabla (Tabla II, Tabla III, Tabla IV y Tabla V) donde se resaltan el conjunto de competencias requeridas por el aprendiz y el perfil de egreso pertinente al trayecto respectivo y que se relacionan con el PST. Ésta estructuración es tomada de [6] y es producto de la adecuación del análisis del documento rector. Las tablas se construyen tomando en cuenta todos los ejes de formación, de tal manera que se complemente el perfil técnico plasmado en el documento rector.

En la Tabla II, Atributos del Trayecto I, la contribución directa y fundamentalmente de las competencias técnicas para la construcción del PST en el Trayecto I, corresponde a la Arquitectura del Computador. La integración con el eje socio crítico permite al estudiante potenciar e integrar las características éticas sociales y de participación ciudadana, que son requeridas para el trabajo en equipo y respeto a la

propiedad intelectual, entre otras. La UC Algorítmica y Programación provee las competencias básicas y técnicas para el desarrollo de componentes de software que son requeridas en el Trayecto II, en conjunto con otras UC.

Por otra parte, en la Tabla III Atributos del Trayecto II, a través de estos componentes curriculares mostrados, el estudiante aprende herramientas, metodologías y lenguajes de programación que le permiten abordar adecuadamente el desarrollo del PST. Es importante resaltar que el alcance de la aplicación informática que debe construir el equipo de trabajo en un PST, se limita al desarrollo de componentes. En este sentido, la definición de componentes además de otros conceptos son asumidos por la UC Ingeniería del Software. Los conocimientos y competencias adquiridos por los aprendices en el Trayecto I son acumulativos y se refuerzan y potencian a través del eje Longitudinal. Hasta el Trayecto II el estudiante éticamente puede realizar actividades de soporte y además desarrollar componentes de software.

TABLA II. ATRIBUTOS DEL TRAYECTO I

COMPETENCIAS PROFESIONALES	UNIDADES CURRICULARES	PROYECTO I
Aplicar el análisis lógico y el razonamiento deductivo y deductivo en la solución de problemas, a partir del desarrollo alcanzado en el pensamiento abstracto.	PROGRAMACIÓN Algorítmica y Programación I	Certificación en Soporte Técnico a Usuarios y Equipos
Operar una computadora personal conociendo sus características y funcionalidades.		Plantear alternativas de soluciones ante situaciones y problemas reales, relacionados con soporte técnico a usuarios y equipos.
Caracterizar y seleccionar equipos de computación.		
Ensamblar y configurar equipos de computación de acuerdo a los requerimientos del usuario.	SOPORTE TÉCNICO Y REDES	
Realizar el mantenimiento preventivo y correctivo de equipos de computación.	Arquitectura del Computador	
Realizar soporte a usuarios y equipos de computación.	Cursos Matemáticas I Formación Crítica I Inglés	
Desarrollar algoritmos de mediana complejidad, implementarlos y ponerlos a punto usando los estándares adecuados.		
Utilizar el idioma inglés para manejar bibliografía técnica relacionada con la informática.		
Fomenta la cultura de la innovación para contribuir con la soberanía y seguridad tecnológica.		

TABLA III. ATRIBUTOS DEL TRAYECTO II

COMPETENCIAS PROFESIONALES	UNIDADES CURRICULARES	PROYECTO II
Aplicar el pensamiento y la reflexión lógica en la organización y formalización de conocimientos relacionados con el cálculo integral y conceptos básicos matemáticos en otras áreas de saberes.	PROGRAMACIÓN • Programación II	Técnico Superior Universitario o Técnico Superior Universitaria en Informática*
Elaborar algoritmos de alta complejidad utilizando estructuras de datos en memoria interna y externa, así como, programarlos en un lenguaje de alto nivel.	INGENIERÍA DEL SOFTWARE • Ingeniería del Software I Base de Datos (BD)	Plantear alternativas de soluciones ante situaciones y problemas reales, relacionados con soluciones informáticas.
Desarrollar y mantener componentes de software, bajo estándares de calidad, priorizando el uso de software libre.	SOPORTE TÉCNICO Y REDES	Diseño y desarrollo de la aplicación informática, acorde al alcance del proyecto
Instalar, configurar y administrar operativamente redes de área local, bajo estándares de calidad, priorizando el uso de software libre.	Cursos • Redes de Computadoras	Mostrar el módulo funcional realizado
Interpretar modelos de datos		
Implementar y mantener bases de datos de pequeña y mediana complejidad.		
Desarrollar algoritmos para manipular bases de datos de pequeña a mediana complejidad.		
Fomenta la cultura de la innovación para contribuir con la soberanía y seguridad tecnológica.		

En el Trayecto III se incorporan elementos técnicos de mayor profundidad, así mismo el proceso de abstracción del estudiante se ha desarrollado complementado con las UCs del componente de formación general y el eje socio crítico. El PST en este trayecto le permite al estudiante concebir un proyecto de una complejidad más alta. El equipo de trabajo está en capacidad de desarrollar un sistema completo partiendo desde el análisis de necesidades y requerimientos, construyendo el diseño e implementación de soluciones automatizadas completas y finalmente realizando un conjunto de pruebas que pondría a tono el sistema. Desde el punto de vista académico y de la gestión de PST como UC, el alcance del proyecto es mayor, por lo que se requiere por parte del equipo docente especialistas del área técnica, evaluar proyectos de alta complejidad para ser distribuidos a los equipos de estudiantes interesados en brindar solución a través de un producto software a una comunidad y así mismo, demostrar sus sólidos conocimientos en la aplicación de metodologías, herramientas, métodos y análisis de problemas complejos. También, en este momento, el estudiante estará en capacidad de desarrollar productos aplicando estándares de calidad y hacer propuestas innovadoras desde su PST.

En el Trayecto IV, se da por sentado que los aprendices cuentan con todos los saberes acumulados desde el inicio de la carrera al momento de desarrollar un PST. Ahora corresponde incorporar un nuevo elemento en el desarrollo del software, el cual se refiere a la seguridad informática. En resumen: el estudiante desde el punto de vista técnico e integral está en capacidad de cumplir con todas las etapas planteadas en el desarrollo de sistemas, aplicando además criterios de calidad y seguridad informática. Tiene la capacidad de analizar y desarrollar proyectos complejos, pudiendo continuar con proyectos realizados en el Trayecto III o sobre cualquier otro proyecto que haya sido desarrollado en la comunidad y que el estudiante pueda añadir valor agregado desde su conocimiento y competencia.

TABLA IV. ATRIBUTOS DEL TRAYECTO III

COMPETENCIAS PROFESIONALES	UNIDADES CURRICULARES	PROYECTO III
Desarrollar aplicaciones informáticas basadas en los principios de la ingeniería de software.	INGENIERÍA DEL SOFTWARE • Ingeniería del Software II • Modelado de BD	Desarrollador de Aplicaciones Diseño y desarrollo de la aplicación informática Seleccionar y justificar la Metodología de desarrollo de software a utilizar en el proyecto socio tecnológico.
Aplicar estándares de calidad, usabilidad y accesibilidad en el desarrollo de aplicaciones informáticas.	SOPORTE TÉCNICO Y REDES Cursos • Sistemas Operativos	Discutir acerca de Metodologías de Desarrollo de Software Metodologías Tradicionales Vs Ágiles.
Elaborar la documentación técnica de una aplicación informática.		Visión general de las distintas metodologías de desarrollo de software. (RUP, WATCH, MERINDE, XP, SCRUM, otras).
Instalar, configurar y manejar sistemas operativos en equipos de computación.		Productos entregables de un proyecto de desarrollo de software
Diseñar bases de datos.		
Aplicar técnicas estadísticas y de la programación matemática para apoyar la toma de decisiones.		
Fomenta la cultura de la innovación para contribuir con la soberanía y seguridad tecnológica.		

TABLA V. ATRIBUTOS DEL TRAYECTO IV

COMPETENCIAS PROFESIONALES	UNIDADES CURRICULARES	PROYECTO IV
Gestionar proyectos informáticos, aplicando estándares reconocidos de calidad y pertinencia social.		"Ingeniero o Ingeniera en Informática"
Aplicar metodologías para realizar auditorías a sistemas informáticos.	INGENIERÍA DEL SOFTWARE <ul style="list-style-type: none"> • Administración de BD • Seguridad Informática • Gestión de Proyectos Informáticos • Auditoría Informática 	Diagnóstico, diseño, ejecución, gestión, implantación y evaluación de proyecto informático bajo estándares de calidad y pertinencia social
Administrar bases de datos.	SOPORTE TÉCNICO Y REDES	
Diseñar, implementar y administrar redes informáticas, bajo estándares de calidad, priorizando el uso de software libre.	Cursos <ul style="list-style-type: none"> • Redes Avanzadas • Investigación de Operaciones • Formación Crítica IV • Inglés 	
Aplicar los principios básicos de la seguridad informática.		
Utilizar el idioma inglés para comunicar los resultados de los proyectos desarrollados mediante el uso de la terminología técnica del idioma inglés.		
Fomenta la cultura de la innovación para contribuir con la soberanía y seguridad tecnológica.		

B. Propuesta de Gestión de los PSTs

A continuación se presenta la propuesta para la gestión de los PSTs. En la misma se presenta el objetivo general, los objetivos específicos, la estrategia para la ejecución del proyecto, las etapas del proyecto y la guía para el desarrollo de los proyectos de software.

La propuesta de gestión de PSTs, se basa en aspectos observados durante la ejecución del PNFI a lo largo de los años de su ejecución. Uno de los aspectos a resaltar es que el 85% de los proyectos que se han implementado en el instituto utilizan metodologías de desarrollo hasta la fase de construcción, lo que implica que cuando requiere pasar a producción se generan errores y la mayoría de las veces termina como software inutilizado. Otra situación observada es que la selección de las comunidades es mayoritariamente realizada por los estudiantes. En este sentido, se dificulta el proceso de transición una vez que los estudiantes egresan de la institución, sobre todo en los Trayectos II y IV que proporcionan un título universitario que le permite insertarse directamente en el mercado laboral. La propuesta hace énfasis en que los proyectos deben llegar a la fase de Transición de acuerdo a lo establecido en la Ingeniería del Software.

Se plantea en este aspecto la aplicación de un enfoque sistemático en el cual el equipo de docentes de Proyecto del Departamento de Informática sea el ente que canalice y asigne prioridades para la aceptación y desarrollo de los proyectos. De esta forma, existiría un mayor compromiso de parte de los docentes, los estudiantes y también las comunidades que se atienden y que deben colaborar en todas las fases del proyecto.

A continuación se presenta esquemáticamente la propuesta de gestión:

1) Objetivo General: Gestionar los proyectos Socio-tecnológicos desarrollados por los estudiantes del PNFI del IUT "Dr. Federico Rivero Palacio".

2) Objetivos Específicos:

- Seleccionar los proyectos a gestionar.
- Seleccionar las metodologías en el ámbito tecnológico a aplicar.
- Seguimiento a la Aplicación de la metodología seleccionada para los proyectos.

3) Estrategias para la Ejecución del PST: el Programa Nacional de Formación en Informática (PNFI) establece un número de 6h semanales presenciales para Proyecto Socio-Tecnológico. En este tiempo se debe orientar a los equipos de trabajo de proyecto en el trayecto correspondiente, tomando en cuenta el perfil y las competencias a aplicar, tal como se mostró en el análisis del currículo que se realizó en la fase anterior. Ahora bien, para garantizar la calidad del PST, se ha trabajado con una división interna del proyecto en dos bloques. Un bloque que se ha denominado Proyecto Metodológico, en el cual se realizan las actividades propias de una investigación, tales como: plantear el problema a resolver, los objetivos, las bases teóricas y metodológicas y normas APA, entre otras. El segundo bloque se denominó Proyecto Técnico: en donde se realizan actividades complementarias que refuerzan los conocimientos y competencias técnicas para la ejecución exitosa del PST. Adicionalmente, existe la figura de un especialista técnico o tutor que trabaja en conjunto con el equipo de estudiantes y además se vinculan con la comunidad que requiere el software.

La estrategia que se plantea para lograr el objetivo, es emplear los principios establecidos por la Ingeniería del Software y el enfoque de Sistemas y Gestión de Proyectos Informáticos, para realizar el seguimiento del proyecto y asegurar la calidad del producto a entregar a la comunidad que ha solicitado el proyecto desde los inicios del mismo. Esto serviría para potenciar el área de Ingeniería del Software en la formación de los estudiantes del instituto y la gestión propiamente dicha, fomentando la aprehensión de la competencia profesional desde el inicio mismo del proyecto de manera práctica. Así como estrechar los vínculos Comunidad-Universidad necesarios para garantizar un desempeño efectivo del eje de Proyecto.

4) Etapas de la Ejecución del PST: Se propone un esquema para la gestión de PSTs adecuando RUP [5]. En este sentido, se plantean cuatro etapas, que consideran las distintas fases de RUP, dependiendo del trayecto del proyecto y se incorpora una etapa de selección de proyectos, la cual es administrada por el grupo docente encargado de proyectos.

5) Guía para el Desarrollo de Proyectos de Software: Para el desarrollo del software se propone como base la aplicación de una guía de gestión de proyectos de Software. Se hace énfasis en los proyectos de desarrollo de software que constituyen tres cuartos del eje longitudinal PST, la naturaleza de los proyectos de soporte técnico a usuarios y equipos plantean otros tipos de metodologías no presentadas en este documento. Lo cual no implica que las fases de esta guía puedan ser adaptadas para el PST del Trayecto I.

C. Etapas de la Ejecución de los PSTs

Para la etapa de ejecución, se diseñó un cronograma que permite realizar un seguimiento y control dentro de cada etapa.

De esta manera se puede observar en la Tabla VI una estructuración donde para cada uno de los trayectos, se especifica una fecha para realizar el análisis y selección del PST del conjunto de los PSTs, esto con la finalidad de que pueda convertirse en un posible elegible, la determinación del inicio, diagnóstico o construcción (etapa 2), inventario técnico, elaboración o transición (etapa 3), la ejecución, construcción o transición (etapa 4) y el desarrollo y entrega del producto e informe (etapa 5). En definitiva, la Tabla VI representa un cronograma que permite asignarle tiempo (en 9 meses o 36 semanas) para la realización de cada una de las fases del PST del trayecto correspondiente. Esta tabla, sirve de guía para presentar las diferentes fases en el eje longitudinal. La primera columna contiene las fases previstas, y las columnas sucesivas contienen el tiempo expresado en meses. Una intercepción de una fila con una columna indica el mes en que debería realizarse esa fase para el proyecto especificado. De manera que se puede ubicar, dada una fase de un proyecto de un trayecto dado, el mes en que debería realizarla o dado un mes cuál debería ser el avance en cuanto a las etapas del proyecto de un trayecto dado.

La explicación de cada una de las etapas se expresa posteriormente a la presentación de la Tabla VI.

TABLA VI. CRONOGRAMA DE ETAPAS VS TIEMPO

Etapas/Tiempo (en meses)	1	2	3	4	5	6	7	8	9
Ejecución PST Etapa 1	X	X							
Ejecución PST Etapa 2			X						
Ejecución PST Etapa 3				X	X				
Ejecución PST Etapa 4						X	X	X	
Ejecución PST Etapa 5									X

1) Ejecución PST etapa 1: en esta etapa el objetivo es analizar y seleccionar las comunidades y los proyectos asociados a ellas. Las comunidades provienen del entorno público, departamentos internos, empresas privadas, convenios interinstitucionales y podrán ser propuestas por las autoridades, auto postulándose o por los estudiantes. Se propone establecer al menos dos reuniones con las comunidades.

- 1era reunión: Diagnóstico general: se determinan las necesidades de automatización y/o soporte técnico de la comunidad participante.
- 2da reunión: Presentación de la comunidad. Una vez determinadas las propuestas de proyectos factibles a realizar en ese trayecto académico. Los PSTs ya estarán clasificados por parte de los expertos (profesores especialistas de PST) de acuerdo a las prioridades establecidas ya sea por compromisos de convenios interinstitucionales, necesidades internas a la institución, proyectos de investigación o

continuidad de proyectos anteriores, entre otros. Los estudiantes democráticamente escogen el proyecto con el cual sienten afinidad para trabajar.

- 2) Ejecución PST etapa 2: esta etapa depende del trayecto en el cual se está realizando el proyecto. De forma tal que, en el Trayecto I corresponderá con el Diagnóstico técnico, en el Trayecto II y III corresponderá a la fase de inicio o equivalente y en Trayecto IV a la fase de construcción o equivalente.
- 3) Ejecución PST etapa 3: en el Trayecto I corresponde al inventario técnico, en el Trayecto II y Trayecto III a la fase de elaboración o equivalente y en el Trayecto IV a la Auditoría de Sistemas.
- 4) Ejecución PST etapa 4: en el Trayecto I corresponde a la acción de corregir las fallas del equipo de hardware o proponer la inclusión de componentes nuevos que forman parte de un hardware completo, en el Trayecto II y III corresponde a la fase de construcción o equivalente y en el Trayecto IV corresponde a la fase de transición.
- 5) Ejecución PST etapa 5: esta etapa corresponde a las actividades de cierre académico, tales como entrega de informes, presentación oral, entre otras.

D. Guía para el Desarrollo del PST

Para el desarrollo del software se propone como base la aplicación de una guía de gestión de proyectos de Software. Se hace énfasis en los proyectos de desarrollo de software que constituyen tres cuartos del eje longitudinal PST, la naturaleza de los proyectos de soporte técnico a usuarios y equipos plantean otros tipos de metodologías no presentadas en este documento.

En las fases de desarrollo de un sistema informático, presentadas en [3] se indica que la primera es la Planeación, no obstante al trabajar con proyectos académicos, con grupos integrados por dos o más estudiantes, es importante establecer relaciones de trabajo y organizacionales entre ellos. Según [4], la cultura organizacional tiene una influencia en cómo los proyectos son ejecutados, así mismo en [7] y [8], se plantea la importancia de organizar, definir y conocer el equipo que trabajará en un proyecto determinado. Es por ello que para los proyectos académicos en particular se añade una fase inicial relacionada con el manejo del grupo. Así mismo es importante crear un marco regulatorio de aspectos que deben ser alcanzados, ese marco debe ser flexible y no limitativo, de manera que todos los proyectos tengan un mínimo académico requerido de acuerdo a los perfiles deseados y donde se respete la naturaleza del PST. Este conjunto de requisitos mencionados anteriormente constituyen los procesos de dirección del proyecto. Así mismo en un proyecto académico-comunitario es necesario incorporar una fase de cierre del proyecto donde se incluyen actividades propias de la finalización de un proyecto académico y directamente

relacionado con las directrices de evaluación, las cuales no son consideradas en este proceso directamente.

La guía de gestión plantea fundamentalmente las etapas siguientes:

- Conformación de grupo
- Análisis preliminar
- Desarrollo de la propuesta de solución
- Presentación de propuesta
- Aplicación de la Metodología de desarrollo de software en las etapas de: Análisis, diseño implementación, pruebas y transición
- Cierre de proyecto.

Estas etapas se complementan con el desarrollo de los siguientes componentes:

- Proceso de Planificación, fundamental para el control seguimiento y evaluación de los estudiantes participantes del proyecto
- Proceso de control de versiones que permita mantener los desarrollos de software,
- Proceso de organización documental, que permita mantener los distintos productos generados en el trabajo en la aplicación de la metodología, separándolos dependiendo de los actores, por ejemplo la comunidad, el documento de PST, proyecto técnico, entre otros.

A continuación se desarrollan las etapas de la guía de gestión propuesta.

1) La conformación de grupo, se refiere a las actividades conducentes a integrar bajo una misión al grupo, cohesionarlos y que entiendan sus ventajas y diferencias, de manera que tengan herramientas para el manejo de conflictos, minimizando el riesgo de ruptura del equipo en algún momento de la ejecución del mismo. En esta etapa, los estudiantes interactúan como equipo y escogen líderes en aspectos técnicos, organizacionales, comunicacionales, entre otros. Además definen su misión, visión, metas y estrategias; como grupo. Esto es importante porque todos los estudiantes deberán rotar y ejercer los diferentes roles presentes en el perfil, a lo largo de la ejecución del proyecto y caracteriza cultura de los miembros del grupo y permite establecer la organización del mismo. En esta etapa los estudiantes definen su primera planificación de actividades generales a desarrollar en el proyecto, a través de un diagrama de Gantt.

2) Análisis preliminar: para el análisis se prevén las actividades de recolección de información. Análisis del sistema actual, entrevistas con la comunidad, realización de encuestas, entrevistas a los usuarios, determinación de tendencias, indagación de estándares de desarrollo del tipo de aplicaciones que se pretende desarrollar, entre otras que puedan surgir por la naturaleza del proyecto. Una vez culminada esta actividad de recolección se genera la lista de requerimientos. La lista de

requerimientos permite formular el modelo del negocio y los casos de uso generales. Adicionalmente el estudiante debe incorporar los requerimientos no funcionales de acuerdo a las normas de calidad y estándares de desarrollo. Esto es fundamental, pues con ellos se generará la propuesta de solución.

- 3) Desarrollo de la propuesta de solución: consiste en determinar en qué consistirá el proyecto, evaluar su factibilidad estimando costos, duración y esfuerzo. Para ello es necesario guiar a los estudiantes a utilizar alguna herramienta de estimación de costo y esfuerzo; se sugiere la utilización de alguna basada en los casos de uso. Aunque generalmente de la estimación pueden despreciarse los costos por ser un PST, las estimaciones del esfuerzo, horas hombre y duración del proyecto resultan importantes para la toma de decisiones en el proyecto y para la planificación. Adicionalmente los estudiantes deben indagar en la comunidad sobre cuál es la tecnología con la que cuentan para el proyecto y con cuál tecnología cuentan los estudiantes para el desarrollo del mismo ya sea propia, de la comunidad o del Instituto o universidad.
- 4) Presentación de la propuesta o anteproyecto ante la comunidad: en esta actividad los estudiantes presentan a la comunidad el resumen de los casos de uso que determinaron en el análisis preliminar, presentan un prototipo de cómo funcionaría el sistema y la factibilidad, la estimación y los riesgos del desarrollo del mismo. Así mismo, se presenta la planificación del proyecto general y un diagrama PER CPM para mostrar las actividades críticas del mismo. La comunidad puede visualizar la posible solución, la duración, los riesgos, las actividades críticas que puedan poner en riesgo la planificación, entre otras y hacer un nuevo intercambio para fijar el alcance real del proyecto durante el trayecto, que puede ser mayor o menor, de acuerdo a los resultados obtenidos en la estimación. Cuando se habla de menor no significa que no se desarrollará la aplicación completa a la comunidad sino que pueden tomarse otras alternativas, como que el mismo grupo desarrolle un porcentaje de los requerimientos y quede para el próximo trayecto el resto de las funcionalidades, esto en el caso de Trayecto III y IV. En el caso de Trayecto II, se acorta el número de componentes que puede desarrollar. En todo caso los involucrados documentan esta situación mediante un acta de requerimientos.
- 5) Aplicación de la Metodología de desarrollo de Software: aun cuando los estudiantes realizan un análisis preliminar a fin de presentar la propuesta, en este punto se comienza a desarrollar como tal el proyecto de software de manera que una vez acordados los requisitos del sistema, el estudiante proceda con las demás actividades planteadas en la metodología realizando el análisis detallado de los requerimientos. La actividad subsiguiente sería proceder a realizar el diseño, generando para ello los artefactos necesarios de acuerdo

a la metodología que esté utilizando, diagramas de clase, de interacción, de secuencia, de estado, etc. Todos aquellos que permitan explicar el funcionamiento del sistema.

Así mismo se debe diseñar el modelo de datos correspondiente, valiéndose de herramientas como el diagrama de clases, análisis del discurso, entre otros. Esta etapa debe ir acompañada de su respectiva planificación de la fase de diseño del proyecto, la cual incluye las posibles iteraciones según la metodología. Se recomienda que la planificación se haga por roles, es decir cada uno de los estudiantes debe identificar las tareas a desarrollar en cada fase y preferiblemente todos los integrante del equipo deben participar proporcionalmente. También debe incluir la entrega de productos, las reuniones de grupo con el asesor y de ser requerido, con la comunidad. El seguimiento de la planificación en esta fase, mínimo debe hacerse semanalmente y mostrar los avances porcentuales de las actividades.

Implementación: basado en que el diseño ha sido óptimo el desarrollo debe concentrarse en codificar los requerimientos e implantar el modelo físico de la base de datos. La codificación lleva también la revisión con pruebas de caja blanca y caja negra.

La verificación del avance puede hacerse de manera individual con el asesor técnico o de forma grupal, donde cada quien muestra su avance, de manera semanal como mínimo. Durante el desarrollo es posible utilizar metodologías ágiles, pues en ese punto se supone el conocimiento pleno de negocio y sus interacciones. Las iteraciones se concentran en resolver los problemas de codificación y avance del proyecto en el menor tiempo posible.

Pruebas: Para realizar esta actividad se recomienda que los estudiantes conozcan diferentes softwares de pruebas para requerimientos Funcionales y No funcionales de manera que puedan aplicar las verificaciones según el tipo de sistema. Este proceso genera documentos que se utilizan para planificar las correcciones pertinentes. En esta etapa lo relevante en la planificación es determinar cuáles serán las pruebas a ejecutar y cuales insumos se necesita para ello. Una vez realizadas las pruebas deben generarse los informes técnicos con los hallazgos, de manera que se solventen en tiempos cortos.

Transición: en algunos casos los equipos de trabajo realizan sus actividades en ambientes colaborativos y el proyecto se va integrando a medida que se desarrolla, en otros casos, trabajan de manera separada y es necesario integrar los componentes desarrollados. Una vez culminado el proyecto, es necesario planificar cómo se implantará en la comunidad destino. Es necesario considerar la disposición de la comunidad para permitir a los estudiantes manejar sus activos o involucrar a los mismos en los equipos de trabajo de la organización. De igual forma se prevé la realización de las pruebas de instalación e integración, la migración de datos y el

entrenamiento a los usuarios, entre otros. Finalmente se requiere que la comunidad realice las pruebas de aceptación para cada uno de los requerimientos convenidos. En este punto se considera culminado el proyecto y pasa a la fase de producción en la comunidad.

6) **Cierre del Proyecto:** en este punto pareciera culminado el proyecto, sin embargo, deben realizarse algunas actividades que permitan cerrar el proceso. La relación Comunidad-universidad se verá fortalecido y se preparará para la apertura de nuevos ciclos de proyectos. Dentro de las actividades pautadas. Se presentan entonces los informes técnicos referentes al proceso y los manuales de uso para los usuarios. Se planifica la presentación final, la entrega del trabajo final de proyecto. Una vez culminados todos los proyectos es importante considerar la evaluación del grupo docente de proyectos, generación del informe final y presentación ante las autoridades. También se generan candidatos a presentaciones en congresos, eventos científicos y otras actividades académicas, sociales, científicas; relacionadas con los proyectos.

E. Valoración de la Propuesta de Gestión

Se aplicó la propuesta de Gestión de PST a cuatro cohortes comprendidas entre 2013 y 2017, en la gestión diurna del PNFI.

Las peticiones de proyectos se clasificaron en cuatro grupos:

- solicitudes internas a la institución, procedentes de departamentos académicos o dependencias administrativas
- solicitudes procedentes de instituciones públicas del Estado Venezolano, tales como instituciones educativas, Consejos Comunales, Ministerios Públicos, entre otros
- peticiones procedentes de Organizaciones no Gubernamentales (ONG) en ese grupo están generalmente fundaciones sin fines de lucro
- solicitudes del sector privado, procedente de empresas generalmente desarrolladoras de software o fundaciones privadas.

En total se analizaron un total de 79 proyectos, repartidos en las cuatro cohortes estudiadas.

En la Tabla VII se muestra la cantidad de proyectos asignados a cada tipo de comunidad. Como se observa la distribución de los proyectos es variable, el hecho de no asignar proyectos a un sector no implica que no hubo peticiones del mismo.

En la Tabla VIII, se muestra la distribución según el tipo de comunidad y los trayectos. El mayor número de proyectos se reparte en los trayectos iniciales, dado que en ellos se concentra la mayor cantidad de estudiantes. La disminución que se observa en el número de PST entre el Trayecto II y el Trayecto III, se debe a que algunos estudiantes al obtener el título de TSU, deciden no continuar con la carrera hacia la ingeniería. Por otra parte se observa que en el Trayecto IV,

aparece un número de proyectos para el sector privado que no proviene de trayectos anteriores, correspondiente al sector privado, esto surgió por la migración de estudiantes del turno nocturno a la gestión diurna, como era de esperarse continuaron con sus comunidades y proyectos.

TABLA VII. ASIGNACIÓN DE PROYECTOS POR COMUNIDAD Y COHORTE

Cohortes \ Comunidad	2013-2014	2014-2015	2015-2016	2016-2017
Proyectos Internos	9	6	9	10
Proyectos Sector público	8	6	4	9
Fundaciones y ONGs	0	1	1	1
Proyectos sector Privado	1	0	9	5
Total	18	13	23	25

En total se asignaron 16 proyectos de soporte técnico al Trayecto I, 29 proyectos de desarrollo de componentes al Trayecto II, que representaban gestiones administrativas sencillas los cuales incluían hasta tres procesos de: creación, reportes, actualización y eliminación. Así mismo se asignaron 17 proyectos de mayor complejidad a equipos de trabajo del Trayecto II y se atendieron proyectos rezagados en Trayecto IV para un total de 17 PSTs.

TABLA VIII. ASIGNACIÓN SEGÚN EL TIPO DE COMUNIDAD Y LOS TRAYECTOS

Trayectos \ Comunidad	I	II	III	IV
Proyectos Internos	12	9	7	6
Proyectos Sector Público	0	10	9	8
Fundaciones y ONGs	0	2	1	0
Proyectos sector Privado	4	8	0	3
Total	16	29	17	17

En la Tabla IX, aparece el total de proyectos culminados satisfactoriamente, cumpliendo con los requisitos académicos y los requerimientos de la comunidad.

Se observa que en el Trayecto I, solo el 37% de los proyectos se culminó satisfactoriamente. De acuerdo al estudio realizado, esto se debe que los estudiantes abandonan la UC proyecto I. Otro factor es que las instituciones declinan ante la posibilidad de ejecución del PST, sobre todo cuando se trata de soporte técnico a equipos hardware. Es importante resaltar que el total de proyectos culminados en general tienen el 100% de satisfacción en la comunidad y en su mayoría corresponde con actualizaciones de software, instalación de sistema operativo Linux y entrenamiento a usuarios.

Para el Trayecto II, se evidencia que los proyectos internos presentan problemas para la implantación del mismo y por lo tanto queda archivados, aun cuando cumple las expectativas de funcionamiento. Presumiblemente los estudiantes entregan el producto que no ha sido probado exhaustivamente y falla al ponerse en funcionamiento. En algunos casos no se realiza la migración de datos y la comunidad decide seguir usando su sistema ante la imposibilidad de tener el historial requerido. Otro aspecto tiene que ver con los recursos técnicos humanos, disponibles en la organización, de manera que si no tienen expertos de software, la aplicación queda sin soporte de mantenimiento y por ende cae en desuso lo que finalmente genera insatisfacción. No así para las instituciones privadas que tienen equipos humanos para gestionar los componentes de software que piden realizar, en general éstas tienen un 100% de satisfacción en los componentes recibidos al culminar el proyecto. En general se observó que las comunidades satisfechas regresaron a la institución solicitando mejoras y nuevos requerimientos para el software.

En el Trayecto III, los estudiantes desarrollaron un porcentaje del 70% de la aplicación requerida. Se estableció la entrega de documentos hasta el desarrollo alcanzado y se realizó la presentación de los logros a la comunidad. En general las comunidades quedan satisfechas con los logros alcanzados, al observar los avances del proyecto, aun cuando el software no esté concluido la puesta en marcha del mismo en la comunidad es posible, y esa instalación es tratada como una versión de prueba. Un aspecto negativo es que en caso de que algunos proyectos queden reprobados por incumplimiento por parte de los estudiantes con los requerimientos académicos, la comunidad queda desatendida, hasta una nueva apertura de trayecto.

TABLA IX. PROYECTOS CULMINADOS SATISFACTORIAMENTE POR TRAYECTOS

Trayectos \ Comunidad	I	II	III	IV
Proyectos Internos	6	1	7	5
Proyectos Sector Público	0	3	9	5
Fundaciones y ONGs	0	1	1	0
Proyectos sector Privado	0	8	0	3
Porcentaje satisfacción	37,5%	44,8%	17	76,4%

En el Trayecto IV, se continúan con los proyectos del Trayecto III, en las cohortes de estudio, tres grupos de estudiantes del turno nocturno realizaron el cambio al turno diurno, por ello las cifras coinciden, pero en realidad hubo proyectos que reprobaron o desertaron. No obstante se culminaron el 76% de los proyectos, los cuales cumplieron con todas las fases de la ingeniería del software y la aceptación de la comunidad. En este punto es importante aclarar que un grupo puede culminar el 100% de los

requerimientos en el Trayecto III, no obstante debe continuar con el mismo para aplicarle los conceptos referidos a la seguridad, administración de base de datos, auditoria de sistemas, entre otras competencias que adquiere en el último trayecto del Plan de Formación.

A través de la valoración de la propuesta y los resultados obtenidos, se evidencia que la aplicación de la misma es una adecuada alternativa de solución que permite una vinculación y relación enriquecedora de la Comunidad-Universidad, ya que se logra cumplir con el fin académico y con la construcción de un producto final que queda operativo en la organización. Igualmente se muestra que hay aspectos que pueden mejorarse, como es el manejo de grupo y la gestión de conflictos, entre otros.

La propuesta deja en evidencia que los estudiantes de Trayecto IV, tienen los conocimientos y competencias para lograr construir un producto que quede totalmente operativo y cumplir con los aspectos académicos requeridos. Los proyectos del Trayecto II, solo alcanzan algunas fases de la ingeniería del software y es posible que sea la razón de falla al ser instalado en la comunidad, no obstante funcionan muy bien, cuando la comunidad tiene un equipo de especialistas que tome el componente desarrollado para integrarlos a un sistema. Igualmente los proyectos del Trayecto III, resultan inconclusos con algunos aspectos de calidad del software, pero se instalan como versiones preliminares, hasta completar la versión final en el Trayecto IV. Los proyectos de Trayecto I, finalizados muestran excelentes resultados en la satisfacción de la comunidad, esto se debe a que los estudiantes logran dejar operativos los equipos o diagnosticar fallas que luego puedan ser atendidas por los especialistas.

V. CONCLUSIONES

En este trabajo se presenta una propuesta de Gestión de Proyectos para el eje longitudinal Proyecto Socio Tecnológico. Esta propuesta es el resultado de la adecuación y combinación de la metodología de desarrollo de sistemas RUP y la de gestión de proyectos PMBOK.

La propuesta fue aplicada en las cohortes 2013-2014, 2014-2015, 2015-2016 y 2016-2017 y la valoración de los resultados obtenidos demostró resultados favorables en la culminación de proyectos y el grado de satisfacción del producto en la relación Universidad-Comunidad.

La gestión del PST con miras a satisfacer tanto las necesidades de formación del estudiantado conforme al perfil de competencias expresadas en el PNFI, como las de los beneficiarios del proyecto, implica una negociación previa entre las partes (Universidad- Comunidad), donde se establezcan los alcances de cada PST, de tal forma que el estudiantado en cuestión, solo hará aportes significativos a la

comunidad, en función de las competencias exigidas en el pensum de estudio.

Para garantizar que todos los proyectos lleguen a la fase de Transición de acuerdo a lo establecido en la Ingeniería del Software, se estima conveniente que sea el personal docente involucrado en la gestión de los Proyectos quienes canalicen y asignen prioridades en la aceptación y desarrollo de proyectos.

La calidad del PST ejecutado por los estudiantes, implica la puesta en práctica de conocimientos no solo en el ámbito de la Informática, sino además como investigadores. Esto es posible, distribuyendo el tiempo asignado a PST, en dos bloques: Proyecto Metodológico y Proyecto Técnico.

En el análisis curricular se demuestra que los proyectos de software deben cumplir todas las fases de la ingeniería de software y éstas solo pueden ser completadas por estudiantes del Trayecto IV.

Para el Trayecto I, la organización del proyecto condujo a resultados exitosos, ya que todos los proyectos ejecutados satisficieron las necesidades de soporte técnico a usuarios y equipos de la comunidad.

Los proyectos de Trayecto II, solo alcanzan algunas fases de la ingeniería del software y es posible que sea la razón de falla al ser instalado en la comunidad, funcionan muy bien, cuando la comunidad tiene un equipo de especialistas que tome el componente desarrollado para integrarlos a un sistema.

Un aspecto que aún queda abierto es por cuánto tiempo se mantienen los proyectos en funcionamiento en las comunidades, se propone el tema como trabajo futuro para mejorar la propuesta actual.

REFERENCIAS

- [1] MPPEU. Ministerio de Poder Popular Para la Educación Universitaria “Plan Nacional de Formación en Informática “. pp. 11, pp. 14, pp. 22, pp. 28, pp. 29. 2012.
- [2] M. Amadio, R. Operetti, J. Tedeski. “Ejes de Formación y Enfoques Curriculares”. UNESCO. Boletín No. 13. Suiza, Ginebra. 2013.
- [3] G. Maigua, E. Lopez. “Buenas Prácticas en la Dirección y Gestión de Proyectos Informáticos”. Edutecne. Argentina. 2012.
- [4] Project Management Institute. “Guia de los Fundamentos para la Dirección de Proyectos (PMBOK Guide)”. 2 Edition. Estados Unidos. 2008.
- [5] I. Jacobson, G. Booch, J. Rumbaugh,. “El Proceso Unificado de Desarrollo de Software”, Addison Wesley. 2000.
- [6] C. Amezquita, M. Castañeda. “Plan de Formación y Capacitación del Perfil Profesional Docente del Departamento de Informática, Departamento de Informática, IUTFRP”. 2015.
- [7] R. Pressman. “Ingeniería del Software un Enfoque Práctico”. 7 edición. Estados Unidos. McGrawHill. 2012.
- [8] A. Weitzenfeld, “Ingeniería de Software Orientada a Objetos con UML. Java e Internet”, México: Edamsa impresiones, S.A. de C.V, pp. 68–73. 2008.

Propiedades Algebraicas y Decidibilidad del Transformador de Predicados *wp* sobre la Teoría de Conjuntos

Federico Flaviani
 Universidad Simón Bolívar
 Caracas, Venezuela
 fflaviani@usb.ve

Resumen—En este trabajo se presentan nuevas propiedades algebraicas del transformador de predicado *wp* sobre los operadores \wedge , \vee , \neg , \Rightarrow , \forall , \exists , \min y \max , demostrados independientemente del lenguaje de programación, usando propiedades generales de la semántica denotacional de los lenguajes.

Adicionalmente se muestra un resultado que habla sobre la decidibilidad y cerradura del transformador de predicados *wp* sobre el lenguaje de programación GCL y usando aseraciones escritas en el lenguaje de la teoría de conjuntos de Zermelo-Frankel-Skolem. En este trabajo se muestra que calcular *wp* de una instrucción de GCL y una asercción escrita en el lenguaje ZFS, es decidable y es otra asercción escrita en ZFS. No necesariamente se puede decidir el valor de verdad de dicha asercción resultante de calcular *wp*, aun teniendo todos los valores de sus variables libres, por lo que el resultado no contradice la indecidibilidad del problema de la parada.

Palabras Clave—Precondición más débil; Semántica denotacional; Decidibilidad; GCL.

I. INTRODUCCIÓN

La lógica de Dijkstra [1] para la corrección de programas se basa en el transformador de predicados *wp* (weakest precondition), que es básicamente una función sintáctica de dos variables que devuelve de forma simbólica la precondición más débil de una instrucción *inst* dado una postcondición *Post* (usando la notación clásica de funciones de dos variables, la notación *wp(inst, Post)* se refiere al resultado de aplicarle a la función *wp*, los argumentos *inst* y *Post*, este resultado es la precondición más débil, simbólicamente hablando, de la instrucción *inst* con la postcondición *Post*). El uso sucesivo de *wp* permite ir calculando precondiciones más débiles entre instrucción e instrucción, desde el final del programa hasta el inicio.

Dijkstra en [1] estableció las reglas que definen la función de transformación sintáctica *wp* según el párrafo siguiente:

Si B, B_0, \dots, B_n y S, S_0, \dots, S_n son expresiones booleanas e instrucciones del lenguaje *GCL* respectivamente, si se abre-
 via *IF* y *Do* como las instrucciones *if* $B_0 \rightarrow S_0 \parallel \dots \parallel B_n \rightarrow S_n$ *fi* y *do* $B \rightarrow S$ *od* respectivamente y si se denota $\text{domain}(B_0, \dots, B_n)$ como un predicado que de satisfacerse en un estado, ninguna de las expresiones B_i , al evaluarse en ese estado, incurren en una operación ilegal (como dividir entre 0), entonces:

- $wp(SKIP, Post) := Post$
- $wp(y_{i_1}, \dots, y_{i_k} := Exp_1, \dots, Exp_k, Post) := \text{domain}(Exp_1, \dots, Exp_k) \wedge Post[y_{i_1}, \dots, y_{i_k} := Exp_1, \dots, Exp_k]$
- $wp(S_0; S_1, Post) := wp(S_0, wp(S_1, Post))$
- $wp(IF, Post) := \text{domain}(B_0, \dots, B_n) \wedge (B_0 \vee \dots \vee B_n) \wedge (B_0 \Rightarrow wp(S_0, Post)) \wedge \dots \wedge (B_n \Rightarrow wp(S_n, Post))$
- $wp(Do, Post) := (\exists k | k \geq 0 : H_k(Post))$

en donde $H_k(Post)$ es un predicado que satisface las ecuaciones:

$$H_0(Post) \equiv \text{domain}(B) \wedge \neg B \wedge Post$$

$$H_k(Post) \equiv$$

$$H_0(Post) \vee (\text{domain}(B) \wedge B \wedge wp(S, H_{k-1}(Post)))$$

para $k \geq 1$

La definición recursiva anterior de $H_k(Post)$, tiene la desventaja de que la regla que define a *wp* para la instrucción *Do* esta escrita en lógica de segundo orden, por lo que no es aplicable directamente. Es necesario resolver la recurrencia de fórmulas $H_k(Post)$ primero, antes de aplicar la regla de *wp* para el *Do*, y también es necesario demostrar que la solución a esa recurrencia se puede escribir en el mismo lenguaje en que esta escrito *Post*, esto con el objetivo de demostrar que la función sintáctica $wp(S, .)$ es cerrada con respecto al lenguaje en que se escribe las aseraciones (Es decir que si una postcondición *Post* esta escrito en cierto lenguaje, entonces *wp(S, Post)* es equivalente a una fórmula escrita en el mismo lenguaje).

Por otro lado existe un álgebra del transformador de predicado *wp* sobre los operadores \wedge , \vee y \Rightarrow , determinada por las siguientes reglas

- $wp(S, \text{false}) \equiv \text{false}$
- $wp(S, P \wedge Q) \equiv wp(S, P) \wedge wp(S, Q)$
- $wp(S, P \vee Q) \equiv wp(S, P) \vee wp(S, Q)$ si S es una instrucción determinística
- Si $P \Rightarrow Q$, entonces $wp(S, P) \Rightarrow wp(S, Q)$

En este trabajo se demuestran propiedades del álgebra de *wp* distintas a las anteriores.

A. Contribución

Asumiendo que las aseraciones de este trabajo se escriben en el lenguaje de la teoría de conjuntos de Zermelo-Frankel-Skolem, se demuestra la cerradura de $wp(S,.)$ sobre este lenguaje para cualquier instrucción S de GCL. Adicionalmente se muestra que, usando el teorema de las definiciones recursivas de la teoría de conjuntos, si se tiene un predicado $Post$, se muestra de forma constructiva una fórmula en ZFS que es equivalente a $wp(S, Post)$, demostrando que el problema del cálculo de wp es decidible sobre el lenguaje ZFS.

El problema de la parada para la instrucción S , es equivalente al problema de conocer el valor de verdad de la asercción $wp(S, true)$ conociendo todos los valores de las variables libres de $wp(S, true)$ (que corresponden a los valores del estado inicial del programa antes de la ejecución). La decidibilidad del cálculo de $wp(S, Post)$ sobre el lenguaje ZFS no es una contradicción a la indecidibilidad del problema de la parada, ya que el problema de conocer el valor de verdad de la asercción resultante de escribir $wp(S, Post)$ en el lenguaje de la teoría de conjuntos de ZFS, teniendo todos los valores de las variables libres de la asercción, no es decidible, ya que la fórmula puede contener combinaciones de cuantificadores \forall y \exists sobre conjuntos infinitos, obligando a que la única forma de conocer el valor de verdad de la asercción sea haciendo una demostración matemática elaborada, y el problema de decidir si una fórmula es un teorema de ZFS, no es decidible.

Por otro lado en este trabajo usando semántica denotacional se demuestran propiedades algebraicas del transformador de predicados wp sobre los operadores \wedge y \vee , cuando uno de los predicados en conjunción o disyunción no es modificado o es modificado determinísticamente por $wp(S,.)$. También se demuestran propiedades algebraicas de wp sobre los operadores $\neg, \Rightarrow, \forall, \exists, min$ y max .

B. Trabajos Relacionados

Originalmente en [1] la definición recursiva de wp que se expuso al inicio no incluía la función sintáctica *domain* en sus reglas, esto fue corregido en [2], donde lo incorpora a la regla de wp de la asignación, pero no en las demás reglas como se definió al inicio de la introducción. Una justificación de la incorporación de *domain* en las reglas de wp del *IF* y *Do*, se encuentra en [3], donde se hace una revisión de la semántica denotacional de *GCL* incluyendo el estado *abort*. La función sintáctica *domain*, aplica sobre expresiones, pero su incorporación en las reglas de construcción de wp del *IF* y *Do*, traen dificultades adicionales que no se tenían en [2]. Para manejar estas dificultades, en [4] se definió la función sintáctica *support*, que viene siendo el análogo a *domain*, pero aplica sobre instrucciones en lugar de expresiones.

Gracias a la incorporación de la función sintáctica *support* en la teoría de wp , es posible demostrar algunas propiedades algebraicas nuevas del transformador de predicados wp , dichas demostraciones se encuentran en este trabajo usando propiedades generales de la semántica denotacional, de esta forma los resultados de estas propiedades algebraicas no dependen del lenguaje de programación usado. Algunas de las propiedades algebraicas de wp que se enuncian aquí, ya

fueron enunciadas en [5], en donde se afirma que las demostraciones se pueden realizar usando inducción estructural sobre el tamaño de la instrucción, pero dichas demostraciones no se encuentran en [5].

Por otro lado en [6] se demuestra un teorema de cerradura del transformador de predicados $wp(S, Post)$ sobre el lenguaje de la aritmética de Peano. La demostración muestra una forma constructiva de resolver la recurrencia $H_k(Post)$ usando las funciones β de Gödel, lo cual demuestra que el cálculo de $wp(S, Post)$ es decidible sobre el lenguaje de la aritmética. En [4] se tomó la misma idea de [6] pero resolviendo la recurrencia $H_k(Post)$ usando la versión numerable del teorema de recursión transfinita que es conocido, como el teorema de las definiciones recursivas, sin embargo en [4] no se toma en cuenta el aspecto constructivo del teorema de las definiciones recursivas y por lo tanto no se mostró un resultado de decidibilidad del cálculo de wp . En este trabajo se retoma la demostración de [4] considerando los aspectos constructivos de la demostración con el objetivo de demostrar la decidibilidad del cálculo de wp sobre el lenguaje de la teoría de conjuntos de ZFS.

En el área de derivación automática de invariantes ha habido un interés reciente en los últimos años [7][8][9][10][11][12][13][14][15], adicionalmente existen aplicaciones como [16][17] que pueden calcular invariantes para ciclos donde las expresiones de las asignaciones del cuerpo del ciclo son todas lineales o traducibles a sistemas de transición lineales, de igual forma en [18] se encuentra otra técnica que es aplicable sólo a ciclos donde el cuerpo es traducible a una transformación afín de espacios vectoriales. Aplicaciones basadas en lógica de Hoare y separación tenemos a [19][20][21] y basadas en wp se encuentra [22], sólo que funciona para programas no estructurados.

El resultado sobre la decidibilidad del cálculo de wp abre posibilidades en el campo de la derivación automática de invariantes para algunos casos, ya que una precondición más débil de un ciclo es un invariante. Sin embargo, la precondición más débil obtenida del teorema de decidibilidad, no es en general una asercción cuyo valor de verdad es decidible cuando se tienen todos los valores las variables libres de la asercción, de esta forma no siempre es práctico usar este teorema de decidibilidad para derivar automáticamente un invariante, y más aún si se cuenta con alguna otra técnica, que en el caso en cuestión, pueda calcular un invariante equivalente y decidible (que el valor de verdad de la fórmula sea decidible cuando se tienen todos los valores de las variables libres de la asercción).

C. Estructura del Artículo

A continuación se presentan tres secciones de las cuales, en la primera de ellas se exponen todas las definiciones de semántica denotacional de [3] necesarias para demostrar los teoremas de las siguientes secciones. En la sección siguiente, se demuestran nuevas propiedades algebraicas del transformador de predicados wp . En la última sección se demuestra el teorema que afirma que el cálculo de wp sobre GCL y la teoría de conjuntos es decidible sobre el lenguaje de la teoría de conjuntos de ZFS.

II. SEMÁNTICA DENOTACIONAL DE UN LENGUAJE DE PROGRAMACIÓN

Algunas de las propiedades del transformador de predicados wp que se encuentran en la siguiente sección, se enunciaron por primera vez en [5], donde se afirma que su demostración se hace por inducción estructural sobre el tamaño de la instrucción. Demostrar propiedades por inducción estructural tiene la desventaja de que la demostración depende de las instrucciones que tenga el lenguaje GCL, es decir, si en un futuro se extiende el lenguaje GCL con nuevas instrucciones, entonces sería necesario revisar todas las demostraciones por inducción estructural que se han hecho en la teoría, para incluir los casos correspondientes a las nuevas instrucciones.

Demostraciones independiente al lenguaje de programación son posibles usando semántica denotacional, en donde se consideran sólo las hipótesis generales que tiene una interpretación de una instrucción. A continuación se presenta un resumen de la semántica denotacional para lenguajes de programación propuesta en [3].

Definición (Espacio de Estados del Algoritmo). *Se considera el siguiente algoritmo*

```
[Const  $\bar{x} : \bar{T}$ ;
Var  $\bar{y} : \bar{T}'$ ;
S
]
```

Donde S es el código del algoritmo, \bar{x} es la lista de constantes del algoritmo y T es la lista de tipos de cada \bar{x} , \bar{y} es la lista de variables del algoritmo y \bar{T}' es la lista de tipos de cada \bar{y} .

Si $\bar{T} = T_1, T_2, \dots, T_n$ y $\bar{T}' = T_{n+1}, T_{n+2}, \dots, T_{n'}$, entonces se define el espacio de estados del algoritmo anterior como

$$\prod_{i=1}^{n'} T_i$$

Ejemplo. Se considera el siguiente algoritmo:

```
[Const  $n : \text{Entero}$ ;
Var  $x : \text{Real}$ ;
 $z : \text{Real}$ ;
S
]
```

Como las constantes y variables del algoritmo son n , x y z de tipos Entero, Real y Real respectivamente, entonces el espacio de estados del algoritmo anterior es $\mathbb{Z} \times \mathbb{R} \times \mathbb{R}$.

Ejemplo. Se considera el siguiente algoritmo:

```
[Const  $n : \text{Entero}$ ;
Var  $A : \text{arreglo } [3..7] \text{ de Reales}$ ;
 $z : \text{Real}$ ;
S
]
```

Como un arreglo de tipo T , es una función de una parte de los enteros a T , entonces el espacio de estados del algoritmo del ejemplo es $\mathbb{Z} \times \mathbb{R}^{[3..7]} \times \mathbb{R}$

Notación. En un algoritmo con espacio de estados Esp se denotará como \vec{x} a la lista de constantes y variables que se encuentra en el orden en que fueron declaradas, es decir, $\vec{x} = \bar{x} \parallel \bar{y}$, donde \bar{x} y \bar{y} son las listas de constantes y variables de la definición de espacio de estados y \parallel es el operador de concatenación de listas.

Notación. Para simplificar la notación, el vector $(\vec{x}) = (\bar{x}, \bar{y})$ se denota simplemente como \vec{x} , por lo que la notación \vec{x} puede entenderse según el contexto como una lista de variables de tipo sintáctica $\bar{x} \parallel \bar{y}$, ó como una tupla (\bar{x}, \bar{y}) .

Definición. Sea un algoritmo con espacio de estados Esp y se toma un elemento $\text{abort} \notin Esp$, entonces se define el espacio de estados extendido al abort como $Esp' := Esp \cup \{\text{abort}\}$

Un algoritmo además de la descripción del espacio de estados consta de frases del lenguaje que se llaman instrucciones y dentro de las instrucciones se encuentran otras frases llamadas expresiones de tipo T (donde T es un conjunto). Dichas frases se interpretan denotacionalmente con la función de interpretación \mathcal{E} . Dicha función de interpretación toma una expresión Exp sintácticamente hablando y devuelve una función con rango en T , que se denota $\mathcal{E}[Exp]$.

Definición. En un algoritmo con espacio de estados Esp , una expresión Exp de tipo T , es una frase del lenguaje que se interpreta como una función

$$\mathcal{E}[Exp] : Dom(\mathcal{E}[Exp]) \subseteq Esp \rightarrow T.$$

Nota. Se deja abierta a cualquier posibilidad la sintaxis de las expresiones y el valor de su función de interpretación, ya que la teoría que aquí se desarrolla es válida para cualquier tipo de expresión y función de interpretación.

Notación. De ahora en adelante para abbreviar se denota a la interpretación de la expresión Exp evaluada en \vec{x} por $Exp(\vec{x})$, en lugar de $\mathcal{E}[Exp](\vec{x})$ y se denota el dominio de la expresión por $Dom(Exp)$ en lugar de $Dom(\mathcal{E}[Exp])$.

Notación. Dada una expresión Exp dentro de un algoritmo con espacio de estados Esp , se denota como $domain(Exp)$ a una fórmula con variables libres \vec{x} , tal que

$$Dom(Exp) = \{\vec{x} \in Esp | domain(Exp)\}$$

y si Exp_1, \dots, Exp_n son expresiones del mismo algoritmo definiremos

$$domain(Exp_1, \dots, Exp_n) :=$$

$$domain(Exp_1) \wedge \dots \wedge domain(Exp_n)$$

A continuación se define el concepto de instrucción e interpretación de la misma. Para dar semántica denotacional a las instrucciones, se usará una función de interpretación que se denota C , dicha función recibe una instrucción S sintácticamente hablando y devuelve una interpretación, que

se denota como $\mathcal{C}[S]$, que no es más que una relación de la teoría de conjuntos.

Definición. En un algoritmo con espacio de estados Esp una instrucción S es una frase del lenguaje que se interpreta como una relación

$$\mathcal{C}[S] : Esp' \rightarrow Esp'$$

tal que el dominio de $\mathcal{C}[S]$ es todo Esp' ,

$$\mathcal{C}[S](\{abort\}) = \{abort\}$$

y

$$\bar{x}' = \bar{x} \text{ si } (\bar{x}', \bar{y}') \in \mathcal{C}[S](\{\bar{x}, \bar{y}\}) \text{ y } \bar{x}' \in \bar{T}.$$

Ejecutar esta instrucción para unos valores iniciales \vec{x}_0 se entiende como evaluar la relación anterior en los valores \vec{x}_0 donde la salida de la ejecución pudiera ser cualquier imagen del punto \vec{x}_0 .

Adicionalmente, se define el soporte $supp(\mathcal{C}[S])$ como el conjunto de los estados que no resultan en un abort al ejecutar la instrucción, es decir;

$$supp(\mathcal{C}[S]) := \{\vec{x} \in Esp | abort \notin \mathcal{C}[S](\{\vec{x}\})\}$$

Nota. Note que como el dominio de $\mathcal{C}[S]$ es todo Esp' entonces, $\mathcal{C}[S](\{x\}) \neq \emptyset$ para todo $x \in Esp'$

Definición. Si \bar{x} y \bar{y} son la lista de constantes y variables de un algoritmo A en el orden en que fueron declaradas, \bar{Y} una lista de variables distintas a \bar{x} y \bar{y} , \bar{Y}_0 una lista de valores de tipos iguales a las variables \bar{Y} , S una instrucción, $Pre(\bar{x}, \bar{y}, \bar{Y})$ y $Post(\bar{x}, \bar{y}, \bar{Y})$ predicados que solo tienen como variables libres a \bar{x} , \bar{y} y \bar{Y} y las familias de conjuntos

$$Dom_{\bar{Y}} := \{(\bar{x}, \bar{y}) \in Esp | Pre(\bar{x}, \bar{y}, \bar{Y})\}$$

y

$$Rgo_{\bar{Y}} := \{(\bar{x}, \bar{y}) \in Esp | Post(\bar{x}, \bar{y}, \bar{Y})\},$$

entonces una tripleta de Hoare dentro del algoritmo A es un predicado que tiene la forma

$$\{Pre(\bar{x}, \bar{y}, \bar{Y})\} S \{Post(\bar{x}, \bar{y}, \bar{Y})\}$$

y es verdadera si y sólo si

$$\mathcal{C}[S](Dom_{\bar{Y}_0}) \subseteq Rgo_{\bar{Y}_0}$$

Notación. En caso en que Pre y $Post$ no tengan valores fijos instanciando las variables libres \bar{Y} , entonces la tripleta

$$\{Pre(\vec{x}, \bar{Y})\} S \{Post(\vec{x}, \bar{Y})\}$$

es una abreviación de

$$(\forall \bar{Y} : \{Pre(\vec{x}, \bar{Y})\} S \{Post(\vec{x}, \bar{Y})\}),$$

es decir, que para cada instancia \bar{Y}_0 de valores fijos para las variables \bar{Y} , se tiene que

$$\{Pre(\vec{x}, \bar{Y}_0)\} S \{Post(\vec{x}, \bar{Y}_0)\}$$

es verdadera.

Nota. Como \bar{Y} es una lista de variables ligadas en el predicado $\{Pre(\vec{x}, \bar{Y})\} S \{Post(\vec{x}, \bar{Y})\}$ según la notación anterior,

entonces el nombre de éstas no importa y pueden cambiarse según convenga.

Nota. Como $Dom_{\bar{Y}} := \{\vec{x} \in Esp | Pre(\vec{x}, \bar{Y})\}$ y $Rgo_{\bar{Y}} := \{\vec{x} \in Esp | Post(\vec{x}, \bar{Y})\}$, entonces para evitar redundancia se convendrá que en $Pre(\vec{x}, \bar{Y})$ y $Post(\vec{x}, \bar{Y})$ no aparece el predicado $\vec{x} \in Esp$, y dicho predicado siempre se tomará como una hipótesis sobrentendida.

Definición. Sea R una relación de $A \times B$ y un subconjunto $Rgo \subseteq B$, entonces

$$M := \{x \in A | R(\{x\}) \neq \emptyset \wedge R(\{x\}) \subseteq Rgo\}$$

se denomina como “dominio máximo de R con rango Rgo ”.

Lema 1. Dado un algoritmo con espacio de estados Esp , una instrucción S y un predicado $Post(\vec{x}, \bar{Y})$ con $\vec{x} \in Esp$ y \bar{Y}_0 lista de valores del mismo tipo que \bar{Y} , entonces el dominio máximo de $\mathcal{C}[S]$ y $Rgo_{\bar{Y}_0}$ es igual a

$$\{\vec{x} \in Esp | \vec{x} \in supp(\mathcal{C}[S]) \wedge (\forall y | : y \in Rg_{\mathcal{C}[S]} \upharpoonright_{supp(\mathcal{C}[S])} (\{\vec{x}\}) \Rightarrow Post(y, \bar{Y}_0))\}$$

Donde $Post(y, \bar{Y}_0)$ es una notación simplificada que significa

$$(\exists \bar{y}' | y = (\bar{x}, \bar{y}') : Post(\bar{x}, \bar{y}', \bar{Y}_0))$$

Demostración: Ver [3] ■

Definición. Dado un algoritmo cuyo espacio de estados es Esp con lista de constantes y variables igual a \vec{x} y la tripleta $\{Pre\} S \{Post\}$ tiene como variables libres distintas a las del espacio de estado a \bar{Y} , entonces se dice que Pre es una precondición más débil de S y $Post$ si y sólo si $\{\vec{x} \in Esp | Pre(\vec{x}, \bar{Y}_0)\}$ es el dominio máximo de la relación $\mathcal{C}[S]$ con rango $Rgo_{\bar{Y}_0}$ para cualquier valor \bar{Y}_0 de las variables libres \bar{Y} .

Nota. Según la definición anterior la precondición más débil de la instrucción S y $Post$ es equivalente a

$$\{\vec{x} \in supp(\mathcal{C}[S]) \wedge (\forall y | : y \in Rg_{\mathcal{C}[S]} \upharpoonright_{supp(\mathcal{C}[S])} (\{\vec{x}\}) \Rightarrow Post(y, \bar{Y}))\}$$

De modo que en el lenguaje de la teoría de conjuntos, el predicado $wp(S, Post(\vec{x}, \bar{Y}))$ es equivalente al predicado anterior.

III. NO DETERMINISMO Y PROPIEDADES DE wp Y $support$

En esta sección se introduce la función de tipo sintáctica $support$, que es análoga a la función $domain$ salvo, que $support$ aplica a instrucciones mientras que $domain$ aplica a expresiones. Incluir en la teoría del transformador de predicados wp a la función sintáctica $support$, tiene la ventaja de que con su ayuda, pueden enunciarse propiedades de tipo algebraicas de wp sobre operadores como \neg , $\Rightarrow min$ y max .

Los siguientes teoremas son propiedades algebraicas del transformador de predicados wp , los lemas principales serán demostrados usando aserciones escritas en el lenguaje de la

teoría de conjuntos de ZFS y usando la fórmula de precondición más débil que se deduce de la semántica denotacional al final de la sección anterior. Dicha fórmula tiene la ventaja de ser independiente del lenguaje de programación, ya que sólo usa el concepto general de interpretación de una instrucción $\mathcal{C}[S]$, sin usar hipótesis de como es el comportamiento específico de la instrucción S al ser interpretada.

Lema 2. $wp(S, P \wedge Q) \equiv wp(S, P) \wedge wp(S, Q)$

La interpretación de la instrucción S es una relación $\mathcal{C}[S]$, si esta relación es determinística con respecto a las coordenadas i_1, \dots, i_k , entonces la relación se comporta como una función sobre esas coordenadas. Dichas funciones se les pondrá el nombre de “funciones componentes” y se denotará como $\mathcal{C}[S]^j$ a la función componente de la coordenada j .

Ejemplo.

```
[Const x : Real;
Var y : Entero;
z : Real;
if y ≥ 0 →
y := -y;
z := z/x
[] y ≥ 3 →
y := 3;
z := z/x
fi]
```

Denotemos el cuerpo del algoritmo anterior por S . El espacio de estados Esp del algoritmo es $\mathbb{R} \times \mathbb{Z} \times \mathbb{R}$, por lo tanto si $\vec{x} \in Esp$, entonces \vec{x} es de la forma (x, y, z) en donde la tercera coordenada es modificada por $\mathcal{C}[S]$ de forma determinística. De esta forma, la tercera coordenada es gobernada por la función componente

$$\mathcal{C}[S]^3 : supp(\mathcal{C}[S]) \subseteq \mathbb{R} \times \mathbb{Z} \times \mathbb{R} \rightarrow \mathbb{R}$$

$$\mathcal{C}[S]^3(x, y, z) := \frac{z}{x}$$

y todo elemento perteneciente a $\mathcal{C}[S](\{(x, y, z)\})$ con $(x, y, z) \in supp(\mathcal{C}[S])$ es de la forma

$$(x, y', \mathcal{C}[S]^3(x, y, z))$$

para algún $y' \in \mathbb{Z}$

Lema 3. Sean \vec{x} y \vec{y} la lista de constantes y variables declaradas en un algoritmo respectivamente y \vec{Y} una lista de variables de especificación. Sea S una instrucción que se comporta determinísticamente sobre los valores de las variables y_{i_1}, \dots, y_{i_k} de la lista \vec{y} . Sea Q un predicado y $P(\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y})$ un predicado que sólo depende de $\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y}$, entonces

$$wp(S, P(\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y})) \vee Q(\vec{x}, \vec{y}, \vec{Y}) \equiv$$

$$\vec{x} \in supp(\mathcal{C}[S]) \wedge$$

$$(P(\vec{x}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, \mathcal{C}[S]^{i_k}(\vec{x}), \vec{Y}) \vee wp(S, Q(\vec{x}, \vec{y}, \vec{Y})))$$

Donde $\mathcal{C}[S]^j(\vec{x})$ es la función componente de $\mathcal{C}[S]$ en la coordenada j

Demostración: Según la última fórmula de la sección anterior se tiene que la precondición más débil de una instrucción S y la postcondición $Post$ es equivalente a

$$\begin{aligned} & (\vec{x}, \vec{y}) \in supp(\mathcal{C}[S]) \wedge \\ & (\forall y| : y \in \mathcal{C}[S] \upharpoonright_{supp(\mathcal{C}[S])} (\{(\vec{x}, \vec{y})\}) \Rightarrow \\ & \quad (\exists \vec{y}'| y = (\vec{x}, \vec{y}') : Post(\vec{x}, \vec{y}', \vec{Y}))). \end{aligned}$$

Como la interpretación $\mathcal{C}[S]$ de la instrucción S modifica las coordenadas i_1, \dots, i_k de forma determinística con las funciones componentes $\mathcal{C}[S]^{i_j}(\vec{x})$, entonces el predicado anterior es equivalente a:

$$\begin{aligned} & (\vec{x}, \vec{y}) \in supp(\mathcal{C}[S]) \wedge \\ & (\forall y| : y \in \mathcal{C}[S] \upharpoonright_{supp(\mathcal{C}[S])} (\{(\vec{x}, \vec{y})\}) \Rightarrow \\ & \quad (\exists y'_1, \dots, y'_{i_1-1}, y'_{i_1+1}, \dots, y'_{i_k-1}, y'_{i_k+1}, \dots, y'_m| \\ & \quad y = (\vec{x}, y'_1, \dots, y'_{i_1-1}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, y'_{i_k-1}, \mathcal{C}[S]^{i_k}(\vec{x}), \dots, y'_m) : \\ & \quad Post(\vec{x}, y'_1, \dots, y'_{i_1-1} \mathcal{C}[S]^{i_1}(\vec{x}), \dots, y'_{i_k-1}, \mathcal{C}[S]^{i_k}(\vec{x}), \dots, \vec{Y}))) \end{aligned}$$

Si la postcondición $Post$ es $P(\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y}) \vee Q(\vec{x}, \vec{y}, \vec{Y})$, entonces la precondición más débil de la instrucción S con esta postcondición es equivalente a:

$$\begin{aligned} & (\vec{x}, \vec{y}) \in supp(\mathcal{C}[S]) \wedge \\ & (\forall y| : y \in \mathcal{C}[S] \upharpoonright_{supp(\mathcal{C}[S])} (\{(\vec{x}, \vec{y})\}) \Rightarrow \\ & \quad (\exists y'_1, \dots, y'_{i_1-1}, y'_{i_1+1}, \dots, y'_{i_k-1}, y'_{i_k+1}, \dots, y'_m| \\ & \quad y = (\vec{x}, y'_1, \dots, y'_{i_1-1}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, y'_{i_k-1}, \mathcal{C}[S]^{i_k}(\vec{x}), \dots, y'_m) : \\ & \quad P(\vec{x}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, \mathcal{C}[S]^{i_k}(\vec{x}), \vec{Y}) \vee \\ & \quad Q(\vec{x}, y'_1, \dots, \mathcal{C}[S]^{i_1}(\vec{x}), y'_{i_1+1}, \dots, \mathcal{C}[S]^{i_k}(\vec{x}), y'_{i_k+1}, \dots, y'_m, \vec{Y}))) \\ & \equiv \\ & (\vec{x}, \vec{y}) \in supp(\mathcal{C}[S]) \wedge \\ & (\forall y| : y \in \mathcal{C}[S] \upharpoonright_{supp(\mathcal{C}[S])} (\{(\vec{x}, \vec{y})\}) \Rightarrow \\ & \quad P(\vec{x}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, \mathcal{C}[S]^{i_k}(\vec{x}), \vec{Y}) \vee \\ & \quad (\exists y'_1, \dots, y'_{i_1-1}, y'_{i_1+1}, \dots, y'_{i_k-1}, y'_{i_k+1}, \dots, y'_m| \\ & \quad y = (\vec{x}, y'_1, \dots, y'_{i_1-1}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, y'_{i_k-1}, \mathcal{C}[S]^{i_k}(\vec{x}), \dots, y'_m) : \\ & \quad Q(\vec{x}, y'_1, \dots, y'_{i_1-1}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, y'_{i_k-1}, \mathcal{C}[S]^{i_k}(\vec{x}), \dots, y'_m, \vec{Y}))) \\ & \equiv \\ & (\vec{x}, \vec{y}) \in supp(\mathcal{C}[S]) \wedge \\ & (\forall y| : y \in \mathcal{C}[S] \upharpoonright_{supp(\mathcal{C}[S])} (\{(\vec{x}, \vec{y})\}) \Rightarrow \\ & \quad P(\vec{x}, \mathcal{C}[S]^{i_1}(\vec{x}), \dots, \mathcal{C}[S]^{i_k}(\vec{x}), \vec{Y}) \vee (\exists \vec{y}'| y = (\vec{x}, \vec{y}') : Q(\vec{x}, \vec{y}', \vec{Y}))) \end{aligned}$$

$(\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge$ $(\forall y : y \notin \mathcal{C}\llbracket S \rrbracket \upharpoonright_{supp(\mathcal{C}\llbracket S \rrbracket)} (\{(\bar{x}, \bar{y})\}) \vee$ $P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee (\exists \bar{y}' y = (\bar{x}, \bar{y}') : Q(\bar{x}, \bar{y}', \bar{Y})) \quad wp(S, P(\bar{x}, y_{i_1}, \dots, y_{i_k}, \bar{Y}) \vee Q(\bar{x}, y_{j_1}, \dots, y_{j_{k'}}, \bar{Y}))$ \equiv $(\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge$ $(P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee$ $(\forall y : y \notin \mathcal{C}\llbracket S \rrbracket \upharpoonright_{supp(\mathcal{C}\llbracket S \rrbracket)} (\{(\bar{x}, \bar{y})\}) \vee$ $(\exists \bar{y}' y = (\bar{x}, \bar{y}') : Q(\bar{x}, \bar{y}', \bar{Y}))) \quad wp(S, P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee Q(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{j_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{j_{k'}}(\vec{x}), \bar{Y}))$ \equiv $(\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge$ $(P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee$ $(\forall y : y \in \mathcal{C}\llbracket S \rrbracket \upharpoonright_{supp(\mathcal{C}\llbracket S \rrbracket)} (\{(\bar{x}, \bar{y})\}) \Rightarrow$ $(\exists \bar{y}' y = (\bar{x}, \bar{y}') : Q(\bar{x}, \bar{y}', \bar{Y}))) \quad wp(S, P \vee wp(S, Q))$ $\equiv p \wedge (q \vee r) \equiv p \wedge (q \vee (p \wedge r)) >$ $(\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee$ $((\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (\forall y : y \in \mathcal{C}\llbracket S \rrbracket \upharpoonright_{supp(\mathcal{C}\llbracket S \rrbracket)} (\{(\bar{x}, \bar{y})\}) \Rightarrow$ $(\exists \bar{y}' y = (\bar{x}, \bar{y}') : Q(\bar{x}, \bar{y}', \bar{Y})))) \quad wp(S, P \vee wp(S, Q))$ \equiv	<p><i>Demostración:</i> Si P depende de \bar{x}, \bar{Y} y de las variables y_{i_1}, \dots, y_{i_k} y Q depende de \bar{x}, \bar{Y} y de las variables $y_{j_1}, \dots, y_{j_{k'}}$, entonces:</p> $wp(S, P(\bar{x}, y_{i_1}, \dots, y_{i_k}, \bar{Y}) \vee Q(\bar{x}, y_{j_1}, \dots, y_{j_{k'}}, \bar{Y})) \equiv <\text{Lema 4}>$ $\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee Q(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{j_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{j_{k'}}(\vec{x}), \bar{Y}))$ \equiv $(\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y})) \vee (\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge Q(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{j_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{j_{k'}}(\vec{x}), \bar{Y}))$ $\equiv <\text{Lema 4}>$ $wp(S, P) \vee wp(S, Q) \quad \blacksquare$ <p>Lema 6. Sea P un predicado y S una instrucción que no modifica los valores de las variables de P, entonces</p> $wp(S, P) \equiv support(S) \wedge P$ <p>donde $support(S)$ es un predicado que depende de las constantes y variables declaradas en el programa, tal que un estado lo satisface si y sólo si la instrucción S no aborta al ser ejecutado en dicho estado.</p> <p>Por ejemplo <i>true</i> es un predicado que para cualquier S, se tiene que S no modifica sus variables, por lo que una forma de calcular $support(S)$ es calculando $wp(S, \text{true}) \equiv support(S) \wedge \text{true} \equiv support(S)$. Por ejemplo si S es la instrucción</p> $\begin{array}{l} \text{if } a > -3 \rightarrow \\ \quad b := b/a \\ \text{else if } a \leq -3 \rightarrow \\ \quad b := 2 \\ \quad \text{fi,} \end{array}$ <p>entonces</p> $\begin{array}{l} wp(S, \text{true}) \\ \equiv \\ (a > -3 \Rightarrow domain(b/a) \wedge true[b := b/a]) \wedge \\ (a \leq -3 \Rightarrow true[b := 2]) \\ \equiv \\ (a > -3 \Rightarrow a \neq 0) \wedge true \end{array}$ <p>Con lo que $support(S) \equiv a > -3 \Rightarrow a \neq 0$.</p> <p>A continuación se demostrará el Lema 6.</p> <p><i>Demostración:</i> Si P depende de \bar{x}, \bar{Y} y de las variables y_{i_1}, \dots, y_{i_k}, entonces</p> $wp(S, P(\bar{x}, y_{i_1}, \dots, y_{i_k}, \bar{Y})) \equiv <\text{Lema 4}>$
--	--

$$wp(S, P \vee Q) \equiv wp(S, P) \vee wp(S, Q)$$

$$\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y})$$

support(S) \wedge

$$(wp(S, P(\bar{x}, \bar{y}, \bar{Y})) \vee wp(S, Q(\bar{x}, \bar{y}, \bar{Y})))$$

■

Como la instrucción S no modifica los valores de las variables y_{i_1}, \dots, y_{i_k} , entonces las funciones componentes $\mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x})$ son funciones identidad y por lo tanto la expresión anterior es equivalente a:

$$\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge P(\bar{x}, y_{i_1}, \dots, y_{i_k}, \bar{Y})$$

\equiv <Notación>

$$support(S) \wedge P$$

■

Lema 7. $wp(S, P) \Rightarrow support(S)$

Demostración:

$$wp(S, P)$$

\equiv <Lema 1>

$$\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (\forall y : y \in R \upharpoonright_{supp(\mathcal{C}\llbracket S \rrbracket)} (\{\vec{x}\}) \Rightarrow P(y, \bar{Y}))$$

\Rightarrow <Debilitamiento>

$$\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket)$$

\equiv <Notación>

$$support(S)$$

■

Lema 8. Sean P y Q predicados y S una instrucción que se comporta determinísticamente sobre los valores de las variables de P , entonces

$$wp(S, P \vee Q) \equiv support(S) \wedge (wp(S, P) \vee wp(S, Q))$$

Demostración: Si P depende de \bar{x}, \bar{Y} y de las variables y_{i_1}, \dots, y_{i_k} , entonces

$$wp(S, P(\bar{x}, y_{i_1}, \dots, y_{i_k}, \bar{Y})) \vee wp(S, Q(\bar{x}, \bar{y}, \bar{Y}))$$

\equiv <Lema 3>

$$\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y}) \vee wp(S, Q(\bar{x}, \bar{y}, \bar{Y})))$$

\equiv

$$((\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge P(\bar{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \bar{Y})) \vee wp(S, Q(\bar{x}, \bar{y}, \bar{Y}))$$

\equiv <Lema 4>

$$(\bar{x}, \bar{y}) \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (wp(S, P(\bar{x}, \bar{y}, \bar{Y})) \vee wp(S, Q(\bar{x}, \bar{y}, \bar{Y})))$$

\equiv <Notación>

Lema 9. Sean P y Q predicados y S una instrucción que no modifica los valores de las variables de P , entonces

$$wp(S, P \wedge Q) \equiv P \wedge wp(S, Q)$$

■

$$wp(S, P \vee Q) \equiv support(S) \wedge (P \vee wp(S, Q))$$

Demostración:

$$wp(S, P \wedge Q)$$

\equiv <Lema 2>

$$wp(S, P) \wedge wp(S, Q)$$

\equiv <Lema 6>

$$support(S) \wedge P \wedge wp(S, Q)$$

\equiv <Lema 7>

$$P \wedge wp(S, Q)$$

Por otro lado si S no modifica los valores de las variables de P , entonces S se comporta determinísticamente sobre los valores de las variables de P , por lo tanto se puede aplicar el Lema 8 de la siguiente manera:

$$wp(S, P \vee Q)$$

\equiv <Lema 8>

$$support(S) \wedge (wp(S, P) \vee wp(S, Q))$$

\equiv <Lema 6>

$$support(S) \wedge ((support(S) \wedge P) \vee wp(S, Q))$$

$$\equiv support(S) \wedge (P \vee wp(S, Q))$$

■

Lema 10. Sea P un predicado y S una instrucción que se comporta determinísticamente sobre los valores de las variables de P , entonces

$$wp(S, \neg P) \equiv support(S) \wedge \neg wp(S, P)$$

Demostración: Si P depende de \bar{x}, \bar{Y} y de las variables y_{i_1}, \dots, y_{i_k} , entonces:

$$wp(S, \neg P(\bar{x}, y_{i_1}, \dots, y_{i_k}, \bar{Y}))$$

\equiv <Lema 4>

$$\begin{aligned}
 & \vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge \neg P(\vec{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \vec{Y}) \\
 \equiv & \text{<Absorción>} \\
 & \vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge (\vec{x} \notin supp(\mathcal{C}\llbracket S \rrbracket) \vee \\
 & \quad \neg P(\vec{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \vec{Y})) \\
 \equiv & \vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge \neg(\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket)) \wedge \\
 & \quad P(\vec{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \vec{Y}) \\
 \equiv & \text{<Lema 4>}
 \end{aligned}$$

$$\vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge \neg wp(S, P(\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y}))$$

\equiv <Notación>

$$support(S) \wedge \neg wp(S, P)$$

\equiv <Lema 6>

$$\begin{aligned}
 & support(S) \wedge (support(S) \wedge P \Rightarrow wp(S, Q)) \\
 \equiv & support(S) \wedge (P \Rightarrow wp(S, Q))
 \end{aligned}$$

Lema 11. Sean P y Q predicados y S una instrucción que se comporta determinísticamente sobre los valores de las variables de P , entonces

$$wp(S, P \Rightarrow Q) \equiv support(S) \wedge (wp(S, P) \Rightarrow wp(S, Q))$$

Demostración: Como S es una instrucción que se comporta determinísticamente sobre los valores de las variables de P , entonces S es una instrucción que se comporta determinísticamente sobre los valores de las variables de $\neg P$, entonces:

$$\begin{aligned}
 & wp(S, P \Rightarrow Q) \\
 \equiv & wp(S, \neg P \vee Q)
 \end{aligned}$$

\equiv <Lema 8>

$$support(S) \wedge (wp(S, \neg P) \vee wp(S, Q))$$

\equiv <Lema 10>

$$\begin{aligned}
 & support(S) \wedge (\neg wp(S, P) \vee wp(S, Q)) \\
 \equiv & support(S) \wedge (wp(S, P) \Rightarrow wp(S, Q))
 \end{aligned}$$

Lema 12. Sean P y Q predicados y S una instrucción que no modifica los valores de las variables de P , entonces

$$wp(S, P \Rightarrow Q) \equiv support(S) \wedge (P \Rightarrow wp(S, Q))$$

Demostración: Como S es una instrucción que no modifica los valores de las variables de P , entonces S es una instrucción que se comporta determinísticamente sobre las variables de P , entonces:

$$wp(S, P \Rightarrow Q)$$

\equiv <Lema 11>

$$support(S) \wedge (wp(S, P) \Rightarrow wp(S, Q))$$

$$\begin{aligned}
 & support(S) \wedge (support(S) \wedge P \Rightarrow wp(S, Q)) \\
 \equiv & support(S) \wedge (P \Rightarrow wp(S, Q))
 \end{aligned}$$

Lema 13. Sea P un predicado, S una instrucción que se comporta determinísticamente sobre los valores de las variables de P , y ϵ una variable no declarada en el programa, entonces

$$wp(S, (\exists \epsilon : P)) \equiv (\exists \epsilon : wp(S, P))$$

Demostración: Si P depende de \vec{x} , \vec{Y} y de las variables y_{i_1}, \dots, y_{i_k} , entonces:

$$wp(S, (\exists \epsilon : P(\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y})))$$

\equiv <Lema 4>

$$\begin{aligned}
 & \vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket) \wedge \\
 & \quad (\exists \epsilon : P(\vec{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \vec{Y}))
 \end{aligned}$$

\equiv < ϵ no ocurre en el vector de variables declaradas \vec{x} >

$$\begin{aligned}
 & (\exists \epsilon : \vec{x} \in supp(\mathcal{C}\llbracket S \rrbracket)) \wedge \\
 & \quad P(\vec{x}, \mathcal{C}\llbracket S \rrbracket^{i_1}(\vec{x}), \dots, \mathcal{C}\llbracket S \rrbracket^{i_k}(\vec{x}), \vec{Y}))
 \end{aligned}$$

\equiv <Lema 4>

$$(\exists \epsilon : wp(S, P(\vec{x}, y_{i_1}, \dots, y_{i_k}, \vec{Y})))$$

Lema 14. Sean P y R predicados, S una instrucción que se comporta determinísticamente sobre los valores de las variables de P y no modifica los valores de las variables de R , y ϵ una variable no declarada en el programa, entonces

$$wp(S, (\exists \epsilon | R : P)) \equiv (\exists \epsilon | R : wp(S, P))$$

Demostración: Como S no modifica los valores de las variables de R , entonces S se comporta determinísticamente sobre los valores de las variables de R y como adicionalmente S se comporta determinísticamente sobre los valores de las variables de P , entonces S se comporta determinísticamente sobre los valores de las variables de $R \wedge P$.

$$wp(S, (\exists \epsilon | R : P))$$

$$\equiv wp(S, (\exists \epsilon : R \wedge P))$$

\equiv <Lema 13>

$$(\exists \epsilon : wp(S, R \wedge P))$$

\equiv <Lema 9>

$$\begin{aligned}
 & (\exists \epsilon : R \wedge wp(S, P)) \\
 \equiv & (\exists \epsilon | R : wp(S, P))
 \end{aligned}$$

Lema 15. Sea S una instrucción, P un predicado y ϵ una variable no declarada en el programa, entonces

$$wp(S, (\forall \epsilon : P)) \equiv (\forall \epsilon : wp(S, P))$$

Demostración: Dado que $\mathcal{C}[S] \upharpoonright_{\text{supp}(\mathcal{C}[S])}$ es una relación, es fácil demostrar que la fórmula de $wp(S, P)$ del final de la sección anterior es equivalente a

$$\begin{aligned} & (\bar{x}, \bar{y}) \in \text{supp}(\mathcal{C}[S]) \wedge \\ & (\forall \bar{y}' | : (\bar{x}, \bar{y}') \in \mathcal{C}[S] \upharpoonright_{\text{supp}(\mathcal{C}[S])} (\{(\bar{x}, \bar{y})\}) \Rightarrow \\ & \quad P(\bar{x}, \bar{y}', \bar{Y})) \end{aligned}$$

Como ϵ no ocurre en la lista de variables libres \bar{x}, \bar{y} , entonces de la fórmula $(\forall \epsilon | : wp(S, P))$ se puede sacar del $\forall \epsilon$ el lado izquierdo del \wedge , el $\forall \bar{y}'$ y el antecedente de \Rightarrow quedando

$$\begin{aligned} & (\bar{x}, \bar{y}) \in \text{supp}(\mathcal{C}[S]) \wedge \\ & (\forall \bar{y}' | : (\bar{x}, \bar{y}') \in \mathcal{C}[S] \upharpoonright_{\text{supp}(\mathcal{C}[S])} (\{(\bar{x}, \bar{y})\}) \Rightarrow \\ & \quad (\forall \epsilon | : P(\bar{x}, \bar{y}', \bar{Y}))) \\ & \equiv \\ & wp(S, (\forall \epsilon | : P)) \quad \blacksquare \end{aligned}$$

Lema 16. Sean P y R predicados, S una instrucción que no modifica los valores de las variables de R , y ϵ una variable no declarada en el programa. Si $(\exists \epsilon | : R) \equiv \text{true}$, entonces

$$wp(S, (\forall \epsilon | R : P)) \equiv (\forall \epsilon | R : wp(S, P))$$

Demostración:

$$\begin{aligned} & wp(S, (\forall \epsilon | R : P)) \\ & \equiv \\ & wp(S, (\forall \epsilon | : \neg R \vee P)) \end{aligned}$$

$\equiv \text{Lema 15} >$

$$(\forall \epsilon | : wp(S, \neg R \vee P))$$

$\equiv \text{Lema 9} >$

$$(\forall \epsilon | : \text{support}(S) \wedge (\neg R \vee wp(S, P)))$$

$$\equiv$$

$$support(S) \wedge (\forall \epsilon | : \neg R \vee wp(S, P))$$

$$\equiv$$

$$support(S) \wedge (\forall \epsilon | R : wp(S, P))$$

$\equiv < (\exists \epsilon | : R) \equiv \text{true} >$

$$(\forall \epsilon | R : support(S) \wedge wp(S, P))$$

$\equiv \text{Lema 7} >$

$$(\forall \epsilon | R : wp(S, P)) \quad \blacksquare$$

Lema 17. Sea R un predicado y S una instrucción que se comporta determinísticamente sobre los valores de las

variables de R . Si i_f y ϵ son variables no declarada en el programa, entonces

$$wp(S, \epsilon \neq (\min i_f | R : i_f)) \equiv$$

$$support(S) \wedge \epsilon \neq (\min i_f | wp(S, R) : i_f)$$

y

$$wp(S, \epsilon = (\min i_f | R : i_f)) \equiv$$

$$support(S) \wedge \epsilon = (\min i_f | wp(S, R) : i_f)$$

Demostración: $\epsilon \neq (\min i_f | R : i_f)$ es equivalente a

$$(\neg(\exists i_f | : R) \Rightarrow \epsilon \neq \infty) \wedge$$

$$((\exists i_f | : R) \Rightarrow \neg(R[i_f := \epsilon]) \vee (\exists i_f | : R \wedge i_f < \epsilon)).$$

De modo que:

$$wp(S, \epsilon \neq (\min i_f | R : i_f))$$

$\equiv < wp(S, P) \Rightarrow support(S) \text{ para cualquier } P >$

$$support(S) \wedge wp(S, \epsilon \neq (\min i_f | R : i_f))$$

\equiv

$$support(S) \wedge wp(S, (\neg(\exists i_f | : R) \Rightarrow \epsilon \neq \infty) \wedge ((\exists i_f | : R) \Rightarrow \neg(R[i_f := \epsilon]) \vee (\exists i_f | : R \wedge i_f < \epsilon)))$$

$\equiv \text{Lema 2} >$

$$support(S) \wedge wp(S, \neg(\exists i_f | : R) \Rightarrow \epsilon \neq \infty) \wedge$$

$$wp(S, (\exists i_f | : R) \Rightarrow \neg(R[i_f := \epsilon]) \vee (\exists i_f | : R \wedge i_f < \epsilon))$$

$\equiv \text{Lema 11} >$

$$support(S) \wedge (wp(S, \neg(\exists i_f | : R)) \Rightarrow wp(S, \epsilon \neq \infty)) \wedge$$

$$(wp(S, (\exists i_f | : R)) \Rightarrow wp(S, \neg(R[i_f := \epsilon])) \vee$$

$$(\exists i_f | : R \wedge i_f < \epsilon)))$$

$\equiv \text{Lema 8} >$

$$support(S) \wedge (wp(S, \neg(\exists i_f | : R)) \Rightarrow wp(S, \epsilon \neq \infty)) \wedge$$

$$(wp(S, (\exists i_f | : R)) \Rightarrow wp(S, \neg(R[i_f := \epsilon])) \vee$$

$$wp(S, (\exists i_f | : R \wedge i_f < \epsilon)))$$

$\equiv \text{Lema 10} >$

$$support(S) \wedge (\neg wp(S, (\exists i_f | : R)) \Rightarrow wp(S, \epsilon \neq \infty)) \wedge$$

$$(wp(S, (\exists i_f | : R)) \Rightarrow \neg wp(S, R[i_f := \epsilon])) \vee$$

$$wp(S, (\exists i_f | : R \wedge i_f < \epsilon)))$$

$\equiv \text{Lemas 13} >$

$$support(S) \wedge (\neg(\exists i_f | : wp(S, R)) \Rightarrow wp(S, \epsilon \neq \infty)) \wedge$$

$$((\exists i_f | : wp(S, R)) \Rightarrow \neg wp(S, R[i_f := \epsilon])) \vee$$

$$(\exists i_f | : wp(S, R \wedge i_f < \epsilon)))$$

$\equiv \text{Lema 2} >$

$$support(S) \wedge (\neg(\exists i_f | : wp(S, R)) \Rightarrow wp(S, \epsilon \neq \infty)) \wedge$$

$$((\exists i_f : wp(S, R)) \Rightarrow \neg wp(S, R[i_f := \epsilon]) \vee (\exists i_f : wp(S, R) \wedge wp(S, i_f < \epsilon)))$$

\equiv <Lema 6>

$$\begin{aligned} support(S) \wedge (\neg(\exists i_f : wp(S, R)) \Rightarrow \epsilon \neq \infty) \wedge \\ ((\exists i_f : wp(S, R)) \Rightarrow \neg wp(S, R[i_f := \epsilon]) \vee (\exists i_f : wp(S, R) \wedge i_f < \epsilon)) \end{aligned}$$

\equiv < S no modifica las variables i_f, ϵ >

$$\begin{aligned} support(S) \wedge (\neg(\exists i_f : wp(S, R)) \Rightarrow \epsilon \neq \infty) \wedge \\ ((\exists i_f : wp(S, R)) \Rightarrow \neg (wp(S, R)[i_f := \epsilon]) \vee (\exists i_f : wp(S, R) \wedge i_f < \epsilon)) \\ \equiv \\ support(S) \wedge \epsilon \neq (min i_f | wp(S, R) : i_f) \end{aligned}$$

Por otro lado

$$\begin{aligned} wp(S, \epsilon = (min i_f | R : i_f)) \\ \equiv \\ wp(S, \neg(\epsilon \neq (min i_f | R : i_f))) \end{aligned}$$

\equiv <Lema 10>

$$support(S) \wedge \neg wp(S, \epsilon \neq (min i_f | R : i_f))$$

\equiv <Resultado anterior>

$$\begin{aligned} support(S) \wedge \neg(support(S) \wedge \epsilon \neq (min i_f | wp(S, R) : i_f)) \\ \equiv \\ support(S) \wedge (\neg support(S) \vee \epsilon = (min i_f | wp(S, R) : i_f)) \end{aligned}$$

\equiv <Absorción>

$$support(S) \wedge \epsilon = (min i_f | wp(S, R) : i_f)$$

■

Lema 18. Sea R un predicado y S una instrucción que se comporta determinísticamente sobre los valores de las variables de R . Si i_f y ϵ son variables no declarada en el programa, entonces

$$\begin{aligned} wp(S, \epsilon \neq (max i_f | R : i_f)) \equiv \\ support(S) \wedge \epsilon \neq (max i_f | wp(S, R) : i_f) \end{aligned}$$

y

$$\begin{aligned} wp(S, \epsilon = (max i_f | R : i_f)) \equiv \\ support(S) \wedge \epsilon = (max i_f | wp(S, R) : i_f) \end{aligned}$$

Demostración: La demostración es análoga a la del Lema 17 pero sustituyendo la expresión $i_f < \epsilon$ por $i_f > \epsilon$.

■

Nota. las demostraciones de estas propiedades con aserciones escritas en el lenguaje de la teoría de conjuntos de ZFS, implican la veracidad de las mismas en un fragmento de un lenguaje que incluya la teoría de ZFS y que sea cerrado sobre los operadores $\wedge, \vee, \neg, \Rightarrow, \forall, \exists, min, max$ y $wp(S, .)$.

Por ejemplo el lenguaje de las aserciones decidibles de [2] (aserciones donde conocer el valor de verdad, si se tienen

todos los valores de las variables libres de la aserción, es decible) es tal, que se puede extender para poder escribir todos los axiomas de ZFS consistentemente, a modo que el lenguaje de [2] pudiera entenderse, como un fragmento de ZFS. En este lenguaje de [2] las propiedades de esta sección son ciertas para toda instrucción S tal que $wp(S, .)$ sea cerrado, ya que el fragmento es cerrado sobre $\wedge, \vee, \neg, \Rightarrow, \forall, \exists, min$ y max .

Para mostrar la afirmación de la nota anterior para el Lema 5 se supone que S es una instrucción que se comporta determinísticamente sobre las variables de P y Q , que son predicados escritos en el lenguaje de las aserciones decibiles de [2] y $wp(S, .)$ es cerrado en ese lenguaje. Como puedo escribir consistentemente los axiomas de ZFS con la sintaxis del lenguaje de [2], entonces este lenguaje puede entenderse como un fragmento de ZFS y por lo tanto el lema 5 es cierto obteniendo

$$wp(S, P \vee Q)$$

\equiv <Lema 5>

$$wp(S, P) \vee wp(S, Q)$$

Como $wp(S, .)$ es cerrado sobre el lenguaje de [2], entonces $wp(S, P)$ y $wp(S, Q)$ son fórmulas del lenguaje de [2], y como dicho lenguaje es cerrado sobre el operador \vee , entonces $wp(S, P) \vee wp(S, Q)$ es una fórmula en el lenguaje de [2]. Esto demuestra que el lema 5 es cierto si se restringen las aserciones a este lenguaje y $w(S, .)$ es cerrado en el fragmento.

IV. CERRADURA Y DECIDIBILIDAD DEL CÁLCULO DE wp

Del lenguaje y axiomatización de la teoría de conjuntos de ZFS no es directo que todo conjunto definido recursivamente exista, sin embargo todo conjunto que quiera definirse de forma recursiva, puede definirse con una fórmula en el lenguaje de ZFS que es equivalente a la recursión inicial. El proceso de conseguir la fórmula del lenguaje de primer orden de ZFS, que define equivalentemente el conjunto que inicialmente se encontraba definido recursivamente, se conoce como “metateorema de recursión transfinita”. Dado una definición de un conjunto hecho de forma recursiva, dicho metateorema muestra de forma constructiva, cuál es la fórmula dentro del lenguaje de ZFS, que define al mismo conjunto.

Una demostración detallada del metateorema de recursión transfinita se encuentra en [23], sin embargo es más general de lo que se necesita para esta sección, ya que es válido para hacer recursión sobre cualquier ordinal. En esta sección se usará una versión de dicho teorema restringido a ω , cuyo enunciado es:

Teorema 1. Si se tiene un predicado φ tal que satisface $(\forall k, F| : (\exists!y| : \varphi(k, F, y)))$. Definiendo $G(k, F)$ como el único y tal que $\varphi(k, F, y)$. Entonces se puede escribir una fórmula ψ donde lo siguiente es demostrable:

- 1) $(\forall k| : (\exists!y| : \psi(k, y)))$, es decir ψ define una función F tal que $\psi(k, F(k))$
- 2) $(\forall k|k \in \omega| : F(k) = G(k, F \upharpoonright_{k-1}))$

Una explicación verboza del teorema anterior sería que la expresión $F(k) = G(k, F \upharpoonright_{k-1})$ es una definición recursiva del conjunto $F(k)$ y la fórmula $\psi(k, y)$ es una versión en el lenguaje de ZFS, que define por comprensión un conjunto y que viene siendo igual $F(k)$. La idea de la demostración es la siguiente:

Demostración: Se define $\psi(k, y)$ como

$$(k \notin \omega \wedge y = \emptyset) \vee (k \in \omega \wedge (\exists d, h)(App(d, h) \wedge k \in d \wedge h(k) = y))$$

En donde $App(d, h)$ es un predicado definido como

$$esFuncion(h) \wedge$$

$$d = Dom(h) \subseteq \omega \wedge (\forall m)(m \in d \Rightarrow m - 1 \subseteq d) \wedge$$

$$(\forall m)(m \in d \Rightarrow \varphi(m, h \upharpoonright_{m-1}, h(m)))$$

El resto de la demostración consiste en demostrar $(\exists!y : \psi(k, y))$ por inducción fuerte y luego que la función $F(k)$ existe usando el axioma de reemplazo. ■

Nota. Note que la demostración del teorema anterior dice que si se tiene la fórmula del predicado φ , entonces se tiene la fórmula para $\psi(k, y)$, que es una fórmula escrita en el lenguaje de primer orden de la teoría de conjuntos de ZFS.

Teorema 2. Si el lenguaje que se usa para escribir los predicados de las aserciones en GCL, es el lenguaje de primer orden de la teoría de conjuntos de Zermelo-Frankel-Skolem, entonces por cada caso particular de predicado Post, de expresión B_0 e instrucción S_0 , existe una fórmula escrita en el lenguaje de primer orden de la teoría de conjuntos de Zermelo-Frankel-Skolem, que es equivalente a $wp(do B_0 \rightarrow S_0 od, Post)$

Demostración:

Se definen recursivamente las siguientes instrucciones

$$If := if B_0 \rightarrow S_0 [] \neg B_0 \rightarrow SKIP fi$$

$$Do_0 := if \neg B_0 \rightarrow SKIP fi$$

$$Do_{k+1} := If; Do_k.$$

Como la interpretación de una secuenciación de instrucciones es la composición de las interpretaciones, entonces la interpretación de la instrucción Do_k satisface la siguiente recurrencia:

$$\mathcal{C}[Do_0] := \mathcal{C}[if \neg B_0 \rightarrow SKIP fi]$$

$$\mathcal{C}[Do_{k+1}] := \mathcal{C}[Do_k] \circ \mathcal{C}[If]$$

Por otro lado la fórmula definida por

$$\varphi(k, F, y) :=$$

$$(k = 0 \wedge y = \mathcal{C}[Do_0]) \vee$$

$$(k \neq 0 \wedge k \in \omega \wedge esFuncion(F) \wedge y = F(k - 1) \circ \mathcal{C}[If]) \vee$$

$$(\neg(k = 0 \vee (k \neq 0 \wedge k \in \omega \wedge esFuncion(F))) \wedge y = F)$$

satisface que:

$$(\forall k, F | : (\exists!y | : \varphi(k, F, y))).$$

Si se define a $G(k, F)$ como el único y que satisface $\varphi(k, F, y)$, entonces por el teorema 1, se tiene que existe una fórmula ψ que satisface que:

- 1) $(\forall k | : (\exists!y | : \psi(k, y)))$, es decir ψ define una función F tal que $\psi(k, F(k))$
- 2) $(\forall k | k \in \omega | : F(k) = G(k, F \upharpoonright_{k-1}))$

Como $G(k, F)$ en notación de llaves es la función a trozos

$$G(k, F) = \begin{cases} \mathcal{C}[Do_0] & \text{si } k = 0 \\ F(k - 1) \circ \mathcal{C}[If] & \text{si } k \in \omega \text{ y} \\ & \text{esFuncion}(F) \\ F & \text{sino} \end{cases}$$

entonces la fórmula

$$(\forall k | k \in \omega | : F(k) = G(k, F \upharpoonright_{k-1}))$$

es equivalente a la recurrencia que define a $\mathcal{C}[Do_k]$ arriba y por lo tanto, $F(k)$ debe ser igual a $\mathcal{C}[Do_k]$. Por esta razón, como se tiene que F es una función tal que $F(k)$ es el único valor en el que $\psi(k, F(k))$ es verdad, entonces cuando el predicado

$$\psi(k, R)$$

sea cierto, debe ocurrir que $R = \mathcal{C}[Do_k]$.

Por otro lado usando las notaciones de [3], las cuales son: \vec{x} para indicar el vector de constantes y variables declaradas en un algoritmo, Esp para indicar el espacio de estados del algoritmo y $Rgo_{\bar{Y}}$ para referirse al conjunto $\{\vec{x} \in Esp | Post(\vec{x}, \bar{Y})\}$, se demostró que

$$(\exists k | k \geq 0 : \mathcal{C}[Do_k](\{\vec{x}\}) \subseteq Rgo_{\bar{Y}})$$

es un predicado equivalente a $wp(do B_0 \rightarrow S_0 od, Post)$. Sin embargo, usando el predicado ψ se puede reescribir la fórmula anterior como

$$(\exists k | k \geq 0 : \psi(k, R) \wedge R(\{\vec{x}\}) \subseteq Rgo_{\bar{Y}})$$

la cual está escrita en el lenguaje de primer orden de la teoría de conjuntos. ■

Como puede observarse la fórmula de la precondición más débil de la demostración del teorema anterior, se consigue de forma constructiva, ya que el predicado $\psi(k, R)$ se extrae del metateorema de recursión transfinita, y como se tiene la fórmula explícita para φ , se puede construir la fórmula de $\psi(k, R)$ y por ende la fórmula de la precondición más débil dentro del lenguaje de primer orden de la teoría de conjuntos.

Por lo dicho en el párrafo anterior, este último teorema muestra que el cálculo de $wp(Do, .)$ para cualquier instrucción Do es decidible dentro del lenguaje de la teoría de conjuntos y por ende, como las reglas de cálculo de wp (que se enunciaron en la introducción) para las instrucciones distintas de Do , son aplicables directamente sobre lenguajes de primer orden, entonces el cálculo de $wp(S, .)$ para cualquier instrucción S es decidible sobre ZFS.

De la demostración se observa que la aserción $wp(Do, Post)$ que se extrae del teorema 2 es muy complicada, incluso la verificación del valor de verdad de dicha aserción teniendo los valores de las variables libres puede ser no decidible. Esto es debido a que el problema de la parada

se puede escribir equivalentemente, como el problema de verificar el valor de verdad de la precondición $wp(S, \text{true})$ para valores iniciales de las variables y constantes del programa S , es decir, para no contradecir la indecidibilidad del problema de la parada, debe ocurrir que verificar el valor de verdad de la aserción $wp(S, \text{true})$, no es decidible en general.

V. CONCLUSIONES

El teorema de decidibilidad del cálculo de wp provee una alternativa para calcular precondiciones más débiles de forma automática, sin embargo estas precondiciones, en muchos casos, no serían una aserción decidible. Por esta razón el teorema sugiere una aplicación de software para el cálculo de wp que maneje aserciones no decidibles de forma simbólica.

La idea práctica es usar todas las técnicas conocidas para calcular precondiciones o invariantes, y en caso de que estas técnicas fallen en el cálculo de una aserción decidible, la aplicación arroja como última opción, la aserción resultante del teorema de decidibilidad aquí mostrado.

Una aplicación de este estilo trabajaría en muchas ocasiones con aserciones no decidibles, por lo que no tiene sentido manejar estas aserciones dentro del lenguaje de programación en cuestión, porque en estos casos las aserciones no serían programables. Por esta razón sugiero que una aplicación que haga uso del teorema de decidibilidad aquí mostrado, maneje las aserciones como comentarios al código. Por ejemplo se pudiera desarrollar un IDE que inserte a modo de comentario, de forma automática la precondición más débil de cada instrucción. Un IDE de este tipo siempre puede computar una aserción entre todas las instrucciones del programa, aunque algunas de ellas sean no decidibles, pudiera ser provechoso, porque es posible que en el cálculo sucesivo de wp , las aserciones se simplifiquen y se obtenga, al final del proceso, una precondición más débil de todo el programa, que sea una aserción decidible.

Por otro lado desde el punto de vista teórico, los resultados aquí presentados muestran que la teoría de wp de Dijkstra es aplicable sobre la teoría de conjuntos, y por ende también el teorema de la invariancia y todas las reglas de Hoare derivadas de wp . De esta forma se pueden usar las técnicas de corrección formal sobre algoritmos con tipos de datos pertenecientes a la teoría de conjuntos. Por ejemplo, se puede usando wp o reglas de Hoare, corregir algoritmos donde los tipos de las variables son objetos como ordinales, cardinales, filtros, ultrafiltros, espacios topológicos, etc.

REFERENCIAS

- [1] E. W. Dijkstra. *Guarded Commands, Nondeterminacy and Formal Derivation of Programs*. Communications of the ACM, 18(8):453-457, 1975.
- [2] D. Gries. *The Science of Programming*. New York, New York: Springer, 1981.
- [3] F. Flaviani. *Modelo Relacional de la Teoría Axiomática del Lenguaje GCL de Dijkstra*. CoNCISa 2015, Valencia, Venezuela, Noviembre 2015, pp. 153-164.
- [4] F. Flaviani. *Cálculo de Precondiciones Más Débiles*. ReVeCom, Diciembre 2016, Vol. 3, No. 2, pp. 68-80.
- [5] F. Flaviani. *Calculation of Invariants Assertions*. CLEI, Septiembre 2017, <http://www.clei2017-46jaio.sadio.org.ar/sites/default/files/Mem/SLTC/sltc-04.pdf>
- [6] G. Winskel. *The Formal Semantics of Programming Languages: An Introduction*. MIT Press, 1993.
- [7] J. Berdine, A. Chawdhary, B. Cook, D. Distefano, and P. O'Hearn. *Variance Analyses from Invariance Analyses*. Proceedings of the 34th Annual Symposium on Principles of Programming Languages, Nice, France, 2007.
- [8] E. Rodriguez Carbonell and D. Kapur. *Program Verification using Automatic Generation of Invariants*. International Conference on Theoretical Aspects of Computing, 2004, Vol. 3407, pp. 325-340.
- [9] J. Carette and R. Janicki. *Computing Properties of Numeric Iterative Programs by Symbolic Computation*. Fundamentae Informatica, 80(1-3):125-146, March 2007.
- [10] M. A. Colon, S. Sankaranarayana, and H. B. Sipma. *Linear Invariant Generation using non Linear Constraint Solving*. Computer Aided Verification, 2003, Vol. 2725, pp. 420-432.
- [11] M. D. Ernst, J. H. Perkins, P. J. Guo, S. McCamant, C. Pacheco, M. S. Tschantz, and C. Xiao. *The Daikon System for Dynamic Detection of Likely Invariants*. Science of Computer Programming, 2006.
- [12] J.C. Fu, F. B. Bastani, and I-L. Yen. *Automated Discovery of Loop Invariants for High Assurance Programs Synthesized using ai Planning Techniques*. HASE 2008: 11th High Assurance Systems Engineering Symposium, 2008, pp. 333-342, Nanjing, China.
- [13] L. Kovacs and T. Jebelean. *Automated Generation of Loop Invariants by Recurrence Solving in Theorema*. In D. Petcu, V. Negru, D. Zaharie, and T. Jebelean, editors, Proceedings of the 6th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC04), pages 451-464, Timisoara, Romania, 2004. Mirton Publisher.
- [14] L. Kovacs and T. Jebelean. *An Algorithm for Automated Generation of Invariants for Loops with Conditionals*. D. Petcu, editor, Proceedings of the Computer-Aided Verification on Information Systems Workshop (CAVIS 2005), 7th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2005), pages 161-169, Department of Computer Science, West University of Timisoara, Romania, 2005.
- [15] S. Sankaranarayana, H. B. Sipma, and Z. Manna. *Non Linear Loop Invariant Generation Using Groebner Bases*. In Proceedings, ACM SIGPLAN Principles of Programming Languages, POPL 2004, pages 381-392, 2004.
- [16] A. Gupta, A. Rybalchenko. *InvGen: An Efficient Invariant Generator*. International Conference on Computer Aided Verification, 2009, pp. 634-640.
- [17] Stanford Invariant Generator, 2006, <http://theory.stanford.edu/~sriram/Software/sting.html>
- [18] E. Rodriguez-Carbonell and D. Kapur. *Generating all Polynomial Invariants in Simple Loops*. J. Symbolic Comput. 42 (2007), no. 4, 443-476.
- [19] S. Magill, A. Nanevski, E. Clarke, and P. Lee. *Inferring Invariants in Separation Logic for Imperative List-processing Programs*. SPACE, 1(1):57, 2006.
- [20] J. Berdine, B. Cook, and S. Ishtiaq. *SLAyer: Memory Safety for Systems-Level Code*. Gopalakrishnan, Springer, Heidelberg 6806:178-183, 2011.
- [21] C. Varming, L. Birkedal. *Higher-order Separation Logic in Isabelle/holcf*. Electronic Notes in Theoretical Computer Science, 218:371-389, 2008.
- [22] M. Barnett, K. Rustan, and M. Leino, Microsoft Research. *Weakest-Precondition of Unstructured Programs*. Proceedings of the 6th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, 31(2006):1, pp. 82-87.
- [23] K. Kunen. *Set Theory*. College Publications, London, UK, 2013.



Sesión de Investigación

Artículos Cortos

Una Propuesta para el Ordenamiento de Datos Bimodales

José T. Cadenas, Soraya Carrasquel, David Coronado, Rosseline Rodríguez, Leonid Tineo

Departamento de Computación y Tecnología de la Información

Universidad Simón Bolívar

Caracas, Venezuela

jtcadenas@usb.ve, scarrasquel@usb.ve, dcoronado@usb.ve, crodrig@usb.ve, leonid@usb.ve

Resumen—Lofti Zadeh, al establecer en 2015 el Principio de la Información, afirma que toda información es posibilística, probabilística o bimodal, la cual es tanto posibilística como probabilística. Ante tal principio, sería deseable poder manipular información bimodal en bases de datos, esto requeriría, entre otras cosas, el poder ordenar resultados de consultas. Aquí se presenta una propuesta de ordenamiento de datos bimodales, la cual usa un ordenamiento, establecido previamente, de números difusos. El orden propuesto puede ser usado en bases de datos con atributos bimodales en consultas que involucren ordenamientos y agrupamientos.

Palabras Clave—Principio de la Información, información bimodal, datos bimodales, ordenamiento.

I. INTRODUCCIÓN

Zadeh [1] estableció el Principio de la Información el cual consta de tres partes: (1) la información es *restricción*, (2) existen tres tipos básicos de información: posibilística, probabilística y bimodal (tanto posibilística como probabilística) y (3) la información posibilística y la probabilística son ortogonales (no derivables la una de la otra). En la Figura 1, tomada de [1], se muestra la esencia de este principio.

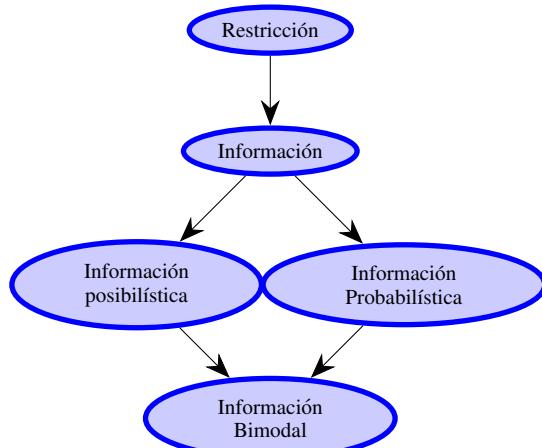


Figura 1: Principio de la Información

Zadeh [1] afirma que una observación empírica importante es que las proposiciones en lenguaje natural contienen, principalmente información difusa posibilística e información difusa bimodal.

La información difusa posibilística ha sido ampliamente estudiada por investigadores como Fukami et al. [2], Buckles y Petri [3], para la incorporación de conceptos difusos en bases de datos. Medina y Pons [4] clasificaron los datos difusos en 11 tipos, en el modelo extendido generalizado para bases de datos relacionales, conocido como GEFRED. Posteriormente, Galindo, Urrutia y Piattini [5] los reclasifican en cuatro tipos, en el modelo FuzzyEER. Estos datos, cuando son considerados como atributos difusos de una base de datos, pueden ser tratados en diversas operaciones como por ejemplo, en ordenamientos [6] [7]. El objetivo principal de este trabajo es presentar un ordenamiento de datos difusos bimodales. El mismo está basado en un ordenamiento de números difusos que utiliza el Principio de Extensión de Zadeh [8].

El trabajo está organizado como sigue, en la segunda sección se presenta el Principio de la Información de Zadeh, conceptos relacionados y algunos ejemplos. La tercera sección describe los conceptos relacionados a la información imperfecta, datos posibilísticos, probabilísticos y bimodales, así como un tipo de ordenamiento de datos posibilísticos. En la cuarta sección, se muestra la propuesta de ordenamiento de datos bimodales. Finalmente, se plantean conclusiones y trabajos futuros.

II. EL PRINCIPIO DE LA INFORMACIÓN

Para poder comprender el Principio de la Información, se debe conocer primero el concepto de restricción.

A. Restricciones sobre Variables

Una *restricción*, $R(X)$, sobre una variable X , es una limitación de los valores que puede tomar X . En ocasiones, una restricción es descrita en lenguaje natural [1]. Por ejemplo, la edad de los aspirantes debe estar entre 18 y 22 años.

Definición 2.1: Tipos de restricciones

1. Una *restricción posibilística* esta expresada como

$$R(X) : X \text{ is } A,$$

donde A es un conjunto difuso en el universo U con función de membresía μ_A , is significa que X tiene distribución de posibilidad A .

2. Una *restricción probabilística* esta expresada como

$$R(X) : X \text{ isp } p,$$

donde p es la función de densidad de probabilidad de X :

$$\text{Prob}(u \leq X \leq du) = p(u)du,$$

isp significa X tiene distribución de probabilidad p .

3. Una *restricción bimodal* es una combinación de una restricción posibilística con una probabilística.

A manera de ejemplo, Si decimos X es “pequeño” tenemos a la variable X con una restricción posibilística, ya que *pequeño* se representa mediante un conjunto difuso. Si decimos X tiene probabilidad p de ser 1,60 cm es porque existe una distribución de probabilidad, por ejemplo normal, que determina la frecuencia con la cual el valor de la variable X es 1,60 cm, es decir, la variable X tiene una restricción probabilística. Pero al afirmar *es p probable que el valor de X sea “pequeño”* estamos afirmando que existe una distribución de probabilidad que mide la frecuencia con la cual, el valor de variable X pertenece al conjunto difuso “pequeño”, es decir, X tiene restricción bimodal.

B. Tipos de Información

Dado que el Principio de la Información establece la relación entre restricción e información, se pueden clasificar los tipos de información a partir de los tipos de restricciones. De acuerdo al Principio de la información, ésta se puede clasificar como:

1. *Información posibilística*: cuando ocurre alguno de los siguientes casos:
 - a) Información posibilística sobre X : si existe una restricción posibilística sobre X .
 - b) Información posibilística sobre X es precisa si la restricción sobre X lo es.
 - c) Información posibilística sobre X es difusa si la restricción sobre X lo es.
2. *Información probabilística* sobre X : cuando existe una restricción probabilística sobre X .
3. *Información bimodal* sobre X : existe una restricción bimodal sobre X .

Tanto la información posibilística como la probabilística pueden ser a su vez, precisa o difusa. En la información bimodal precisa, tanto las posibilidades como las probabilidades son precisas. En la información bimodal difusa, la posibilidad, la probabilidad o ambas, son difusas.

La diferencia principal entre la información posibilística y la probabilística reside en que la primera define lo que *puede* ocurrir, con un número en $[0, 1]$ el cual establece *que tan posible es*, siendo 1 el valor para totalmente posible y 0 para no posible, mientras que la segunda dice *con que frecuencia* se presentan estos valores posibles, con números en $[0, 1]$, con la condición adicional de que la suma de las probabilidades de todos los valores posibles debe ser uno (1). Para ser *probable*, primero debe ser *possible*.

Así, la información probabilística hace referencia a la repetición de un evento, mientras que la información posibilística hace referencia a los posibles resultados de un sólo evento.

La información difusa comprende los casos de incertidumbre, en los cuales la naturaleza de los datos es imprecisa,

vaga, incierta o nula. La incertidumbre se encuentra cuando se afirma, por ejemplo, *la edad de Ana es 19, 20, 21 o 22 años, pero es una de esas*, pues no se dà un valor preciso sobre la edad de Ana. Pero la frase *la edad de Ana está entre 2 y 30 años* es muy seguro que sea cierta pero se puede considerar en la práctica como imprecisa, por ejemplo, al querer asignarle el valor a un boleto de avión. También aparece la incertidumbre cuando hay varias fuentes de información no concordantes, por ejemplo *según Luis, la edad de Ana es 19 o 20 años, pero según Juan, Ana es mayor de 30 años*. La información nula ocurre cuando, por ejemplo, no se tiene ningún valor posible para la edad de Ana.

C. Notación

Para denotar probabilidades y posibilidades, comúnmente se usan funciones llamadas distribución de probabilidad y distribución de posibilidad.

Si un dato D puede tomar los valores $\{x_1, x_2, \dots, x_n\}$ en un universo X , con $p_i = \text{Prob}(D = x_i)$, la distribución de probabilidad de D es

$$\text{Prob}(D) = \{p_1/x_1, \dots, p_n/x_n\}.$$

Donde $\text{Prob}(D)$ es normalizada, lo cual se entiende como que la suma de las probabilidades es 1.

Análogamente, con $\mu_i = \text{Pos}(D = x_i)$, la distribución de posibilidad de D es

$$\text{Pos}(D) = \{\mu_1/x_1, \dots, \mu_n/x_n\}.$$

Donde $\text{Pos}(D)$ es normalizada, lo cual se entiende como que al menos una de las posibilidades es 1.

Adicionalmente, si X es un universo numérico, y $\text{Pos}(D)$ es convexa en X , (en el sentido analítico: $\text{Pos}(\lambda x + (1 - \lambda)y) \leq \lambda \text{Pos}(x) + (1 - \lambda) \text{Pos}(y)$, o en el sentido geométrico: todos los puntos de cualquier segmento de recta con extremos x, y en D , está contenido en D) se dice que $\text{Pos}(D)$ es un número difuso.

Para la distribución de posibilidad de un dato posibilístico D , suele usarse la notación

$$D = \{\mu_D(x_1)/x_1, \dots, \mu_D(x_n)/x_n\}.$$

III. INFORMACIÓN IMPERFECTA

Con el Principio de la Información, los datos imperfectos, no sólo pueden ser considerados como información posibilística, como se han tratado, [2] [3] [4] [5] [7], permitiendo la utilización de atributos con valores difusos dentro de bases de datos, sino como información bimodal, cuya utilización en base de datos está en estudio.

A. Datos Posibilísticos

Los tipos de datos difusos han sido ampliamente estudiados. En [5] se establece el modelo *FuzzyEER*, el cual clasifica los datos difusos en cuatro tipos:

Tipo 1 Datos precisos consultados mediante etiquetas lingüísticas (conjuntos difusos).

Tipo 2 Distribuciones de posibilidad sobre conjuntos ordenados. Son números difusos.

Tipo 3 Etiquetas lingüísticas provistas de relación de similitud. Además, distribuciones de posibilidad sobre estas.

Tipo 4 Como los Tipo 3, pero sin la relación de similitud.

En [9] [10], se evidencia la existencia de varias propuestas de ordenamiento para números difusos. Una de las propuestas de ordenamiento de números difusos se deriva de la aplicación del Principio de Extensión de Zadeh [7]. Este ordenamiento será la base para resolver el problema planteado.

B. Datos Bimodales

Un dato difuso bimodal, X , está caracterizado por dos distribuciones, una posibilística, $\text{Pos}(X)$ y otra probabilística, $\text{Prob}(X)$. Es decir, un número difuso dado por extensión junto con una distribución de probabilidad asociada. A continuación, algunos ejemplos.

Ejemplo 3.1: Es información posibilística: *la edad de Ana está en el conjunto difuso Edad(Ana) = {.6/18, 1/19, 1/20, .6/21}*, el cual es representado gráficamente como en la Figura 2 y se interpreta como: *es totalmente posible que la edad de Ana sea 19 o 20 años, menos posible sea 18 o 21 y es totalmente imposible que sea menor que 18 o mayor que 21*.

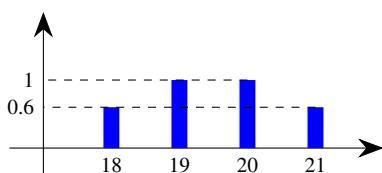


Figura 2: Edad de Ana Representada como Información Posibilística

Ejemplo 3.2: Es información probabilística: *la edad de los aspirantes es 18, 19 o 20 años, con distribución de probabilidad Edad(aspirantes) = {.2/18, .5/19, .3/20}* la cual se representa, por ejemplo, con el gráfico de torta mostrado en la Figura 3.

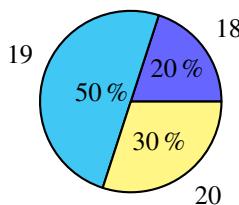


Figura 3: Edad de los Aspirantes Representada como Información Probabilística

Ejemplo 3.3: Se entrevistan aspirantes a un cargo cuya edad debe estar entre 18 y 20 años. Es decir, el conjunto de posibles valores es $\{18, 19, 20\}$ con alguna distribución de posibilidad asociada. Para un aspirante se puede dar el caso que no se conoce el valor exacto de su edad. Así, se puede tener una restricción de posibles valores de la edad, una distribución de posibilidad. Adicionalmente, puede que se conozca la frecuencia de distribuciones de edades en el grupo, lo cual también aporta información sobre la edad de

este aspirante, una distribución de probabilidad. De manera que el dato sería un dato difuso bimodal. Por ejemplo, se podrían tener las siguientes distribuciones:

$$\text{Pos(Edad)} = \{.8/18, 1/19, 1/20\}$$

$$\text{Prob(Edad)} = \{.2/18, .3/19, .5/20\}$$

Y se representan gráficamente como en la Figura 4: a la izquierda su distribución de posibilidad y a la derecha su distribución de probabilidad.

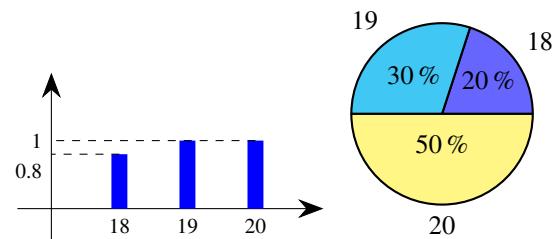


Figura 4: Edad de los Aspirantes Representada como Información Bimodal

C. Ordenamiento de Números Difusos Basado en el Principio de Extensión de Zadeh

Un número difuso A definido por Extensión se representa con una distribución de posibilidad

$$A = \{\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \dots, \mu_A(x_n)/x_n\}.$$

Definición 3.1: Dados dos números difusos, A y B , aplicando el Principio de Extensión de Zadeh [8], en [7] se define

$$\Pi_{A < B} = \left\{ \begin{array}{l} \bigvee_{\substack{x_1 < x_2 \\ x_1, x_2 \in X}} (\mu_A(x_1) \wedge \mu_B(x_2)) / \text{true}, \\ \bigvee_{\substack{x_1 \geq x_2 \\ x_1, x_2 \in X}} (\mu_A(x_1) \wedge \mu_B(x_2)) / \text{false} \end{array} \right\}$$

Donde \wedge es el mínimo y \vee el máximo. Con $\Pi_{A < B}$ se definen las relaciones entre los números difusos A y B , *posiblemente menor*, $<_p$ y *posiblemente igual*, $=_p$, como:

$$\begin{array}{ll} A <_p B \leftrightarrow & \Pi_{A < B}(\text{true}) > \Pi_{B < A}(\text{true}) \\ A =_p B \leftrightarrow & \Pi_{A < B}(\text{true}) = \Pi_{B < A}(\text{true}) \end{array}$$

Y así se puede definir el operador posible menor o igual como

$$A \leq_p B \leftrightarrow A <_p B \vee A =_p B.$$

De forma equivalente:

$$A \leq_p B \leftrightarrow \Pi_{A < B}(\text{true}) \geq \Pi_{B < A}(\text{true})$$

Ejemplo 3.4: Sean A y B los números difusos en X que representan la edad de Ana y de José, respectivamente, con:

$$A = \{1/19, .8/20, .5/21, .2/22\}$$

$$B = \{.2/18, .5/19, .8/20, 1/21\}$$

cuyas representaciones son, A en azul y B en rojo en la Figura 5, se quiere establecer la distribución de posibilidad de $A \leq B$. Para ello, como $19 < 21$, $\mu_A(19) = 1$ y $\mu_B(21) = 1$, se tiene $\mu_A(19) \wedge \mu_B(21) = 1$ y como 1 es el máximo valor posible, se establece

$$\bigvee_{\substack{x_1 < x_2 \\ x_1, x_2 \in X}} (\mu_A(x_1) \wedge \mu_B(x_2)) = 1.$$

Para calcular $\Pi_{A < B}(\text{false})$ debemos encontrar todos los valores para x_1 y x_2 tales que $x_1 \geq x_2$ y para cada par, calcular $\mu_A(x_1) \wedge \mu_B(x_2)$, luego tomar el máximo entre estos valores. La Tabla I muestra estos valores, obteniendo el máximo .8.

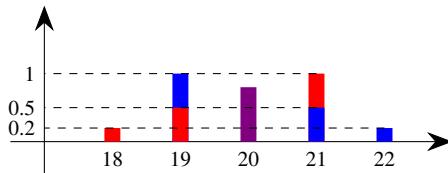


Figura 5: Edades de Ana y José Representadas como Números Difusos

TABLA I. POSIBLES VALORES DE x_1, x_2 TALES QUE $x_1 \geq x_2$

x_1	x_2	$\mu_A(x_1)$	$\mu_B(x_2)$	$\mu_A(x_1) \wedge \mu_B(x_2)$
19	18	1	.2	.2
19	19	1	.5	.5
20	18	.8	.2	.2
20	19	.8	.5	.5
20	20	.8	.8	.8
21	18	.5	.2	.2
21	19	.5	.5	.5
21	20	.5	.8	.5
21	21	.5	1	.5
22	18	.2	.2	.2
22	19	.2	.5	.2
22	20	.2	.8	.2
22	21	.2	1	.2
22	22	.2	0	0

Así tenemos que

$$\Pi_{A < B} = \{1/\text{true}, .8/\text{false}\}.$$

De manera análoga, se establece la distribución de posibilidad $\Pi_{B < A}$ usando la Tabla II y la inspección simple se concluye que el máximo de los valores es 0.5.

TABLA II. POSIBLES VALORES DE x_1, x_2 TALES QUE $x_1 < x_2$

x_1	x_2	$\mu_B(x_1)$	$\mu_A(x_2)$	$\mu_B(x_1) \wedge \mu_A(x_2)$
18	19	.2	1	.2
18	20	.2	.8	.2
18	21	.2	.5	.2
18	22	.2	.2	.2
19	20	.5	.8	.5
19	21	.5	.5	.5
19	22	.5	.2	.2
20	21	.8	.5	.5
20	22	.8	.2	.2
21	22	1	.2	.2

Por simple inspección, como $\mu_A(19) = 1$ y $\mu_B(21) = 1$, entonces $\mu_B(21) \wedge \mu_A(19) = 1$. Así se tiene

$$\Pi_{B < A} = \{.5/\text{true}, 1/\text{false}\}.$$

Por la Definición 3.1, se concluye que $[A \leq_P B]$. Se pueden tomar los valores comparados para establecer un orden difuso entre A y B :

$$\begin{aligned} \Pi_{A \leq B} &= \{\Pi_{A < B}(\text{true})/\text{true}, \Pi_{B < A}(\text{true})/\text{true}\} \\ &= \{1/\text{true}, .5/\text{false}\}. \end{aligned}$$

IV. ORDEN DE DATOS BIMODALES

Yuan [11] propone que al ordenar un conjunto difuso, no se debe usar un ordenamiento preciso, sino uno difuso, es decir, dados dos números difusos a y b , al aplicarle un ordenamiento $<_d$ se obtiene una medida de posibilidad para para $a <_d b$ y otra para $b <_d a$. Esta es una de las razones por las que existen diferentes ordenamientos para números difusos. El objetivo de este trabajo es establecer un orden $<_b$ entre un par de datos bimodales, adaptando el criterio de Yuan, el ordenamiento $<_b$ producido será bimodal. Esto es, dados dos datos bimodales A y B con distribuciones $\text{Pos}(A)$, $\text{Prob}(A)$, $\text{Pos}(B)$ y $\text{Prob}(B)$, se propone presentar la medida de posibilidad y la medida de probabilidad asociada a cada orden, $A <_b B$ y $B <_b A$. En la sección anterior, ya se presentó el cálculo de las posibilidades en un ordenamiento difuso, este cálculo se usaría para la posibilidad asociada al orden bimodal. De manera que sólo resta presentar el cálculo de la distribución de probabilidad del orden bimodal.

Por ejemplo, si se toman los números difusos A y B del Ejemplo 3.4, y se complementan con distribuciones de probabilidad, se tendrían

$$\begin{aligned} \text{Pos}(A) &= \{1/19, .8/20, .5/21, .2/22\} \\ \text{Prob}(A) &= \{.4/19, .32/20, .2/21, .08/22\} \\ \text{Pos}(B) &= \{.2/18, .5/19, .8/20, 1/21\} \\ \text{Prob}(B) &= \{.08/18, .2/19, .32/20, .4/21\} \end{aligned}$$

Que ahora se representan gráficamente como en las Figuras 6 y 7, respectivamente.

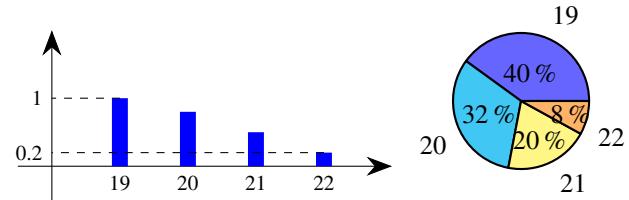


Figura 6: Representación de A como Dato Bimodal

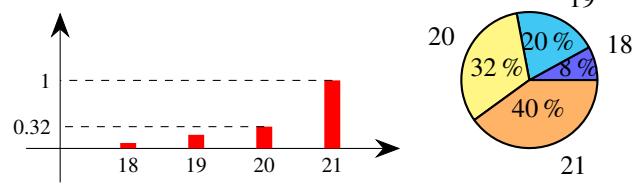


Figura 7: Representación de B como Dato Bimodal

Se quiere establecer un ordenamiento bimodal entre A y B , es decir, dar una distribución de posibilidad y una

distribución de probabilidad de la relación \leq . En el Ejemplo 3.4, se estableció la distribución de posibilidad $\text{Pos}(A \leq B) = \{1/\text{true}, .5/\text{false}\}$. Para establecer la distribución de probabilidad, se propone calcular la probabilidad tomando la suma de las probabilidades conjuntas, como sigue. Dadas $S_1 = \sum_{\substack{x_1 \leq x_2 \\ x_1, x_2 \in X}} (p_A(x_1) \cdot p_B(x_2))$ y $S_2 = \sum_{\substack{x_2 > x_1 \\ x_1, x_2 \in X}} (p_A(x_1) \cdot p_B(x_2))$. Entonces $S_1 + S_2 = 1$. En efecto, desarrollando las sumas

$$S_1 + S_2 = \sum_{\substack{x_1 \leq x_2 \\ x_1, x_2 \in X}} (p_A(x_1) \cdot p_B(x_2)) + \sum_{\substack{x_2 > x_1 \\ x_1, x_2 \in X}} (p_A(x_1) \cdot p_B(x_2)).$$

Como el sumando es común a ambas sumas, y la disjunción de las condiciones es todo el universo se tiene que:

$$S_1 + S_2 = \sum_{x_1, x_2 \in X} p_A(x_1) \cdot p_B(x_2) = \sum_{x_2 \in X} p_B(x_2) \cdot \sum_{x_1 \in X} p_A(x_1) = 1.$$

Definición 4.1: Dados dos datos bimodales A y B con distribuciones de probabilidad p_A y p_B respectivamente, entonces

$$\text{Prob}(A \leq B) = \{S_1/\text{true}, S_2/\text{false}\}.$$

Aplicado al ejemplo, se calculan las probabilidades en las Tablas III y IV. Así $S_1 = 0.6784$ y $S_2 = 0.3216$ resultando:

$$\text{Prob}(A \leq B) = \{.6784/\text{true}, .3216/\text{false}\}.$$

El cual tiene representación gráfica como en la Figura 8. Este resultado se puede interpretar como, es totalmente posible que $A \leq B$ y ocurre el 68 % de la veces, mientras que es poco posible que $B > A$, y ocurre el 32 % de las veces.

TABLA III. PROBABILIDADES x_1, x_2 TALES QUE $x_1 \leq x_2$

x_1	x_2	$p_A(x_1)$	$p_B(x_2)$	$p_A(x_1) \cdot p_B(x_2)$
19	19	.4	.2	.08
19	20	.4	.32	.128
19	21	.4	.4	.16
20	20	.32	.32	.1024
20	21	.32	.4	.128
21	21	.2	.4	.08
Suma =				.6784

TABLA IV. PROBABILIDADES DE x_1, x_2 TALES QUE $x_1 > x_2$

x_1	x_2	$p_A(x_1)$	$p_B(x_2)$	$p_A(x_1) \cdot p_B(x_2)$
19	18	.4	.08	.032
20	18	.32	.08	.0256
20	19	.32	.2	.064
21	18	.2	.08	.016
21	19	.2	.2	.04
21	20	.2	.32	.064
22	18	.08	.08	.0064
22	19	.08	.2	.016
22	20	.08	.32	.0256
22	21	.08	.4	.032
Suma =				.3216

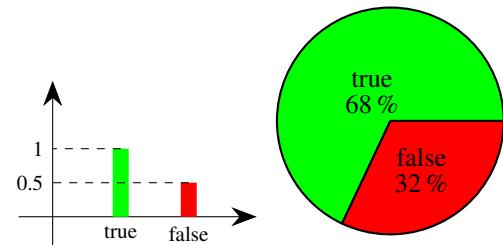


Figura 8: Representación de $A \leq B$

V. CONCLUSIONES Y TRABAJOS FUTUROS

El trabajo presentado provee una primera propuesta de ordenamiento de datos bimodales. Éste puede ser usado en consultas SQL sobre bases de datos difusas, extendidas con datos bimodales, cuando se usa el operador ORDER BY. En este caso los atributos serían del Tipo 2, según el modelo *Fuzzy-EER*, extendidos a bimodales, agregándole la distribución de probabilidad asociada.

La propuesta usa un orden total sobre números difusos que aplica el Principio de Extensión Generalizado de Zadeh, calculando un orden similar para las distribuciones de probabilidad asociadas a los datos bimodales a comparar.

Para incorporar esto en SQL, será necesario proveer una extensión a este lenguaje que permita definir y manipular dominios de atributos Tipo 2 extendidos con distribuciones de probabilidad, de manera que se puedan utilizar para la creación y operaciones de una base de datos.

También se requiere dar semántica al operador ORDER BY de SQL cuando se involucran atributos sobre estos dominios. La propuesta que hacemos aquí es meramente teórica, habría que explorar sus implicaciones prácticas al incorporarla a SQL y hacer implementaciones reales de la surja.

Se propone a futuro aplicar esta propuesta a algunos casos de estudio y analizar su comportamiento a fin de evaluar la utilidad de la propuesta y su utilidad práctica al usar datos bimodales en la representación de información en bases datos. Como existen otros operadores de consulta en SQL que se basan en el ordenamiento de datos como el GROUP BY y el WINDOW, para ellos es también necesario establecer un orden.

AGRADECIMIENTOS

En un mundo bimodal, donde muchas cosas son imprecisas e inciertas, agradecemos a quien nos da plena certeza para seguir adelante en nuestras vidas: “Jesús le dijo: *Si puedes creer, al que cree todo le es posible*”. (Marcos 9:23)

REFERENCIAS

- [1] L. A. Zadeh, “The Information Principle,” *Information Sciences*, vol. 294, pp. 540–549, February 2015.
- [2] S. Fukami, M. Umano, M. Muzimoto, and H. Tanaka, “Fuzzy Database Retrieval and Manipulation Language,” in *IEICE Technical Reports*, vol. 78, no. 233, pp. 65–72, 1979.
- [3] B. P. Buckles and F. E. Petry, “A Fuzzy Representation of Data for Relational Databases,” *Fuzzy sets and systems*, vol. 7, no. 3, pp. 213–226, May 1982.

- [4] Medina, O., Pons, and A. Vila, "GEFRED: A Generalized Model of Fuzzy Relational Databases," *Information Sciences*, vol. 77, no. 6, pp. 87–109, December 1994.
- [5] J. Galindo, A. Urrutia, and M. Piattini, *Fuzzy Databases: Modeling, Design and Implementation*. IGI Global, 2005.
- [6] S. Carrasquel, R. Rodríguez, and L. Tineo, "Consultas con Ordenamiento basado en Similitud," *Telematique*, vol. 12, no. 1, pp. 24–45, January-June 2013.
- [7] D. Coronado, S. Carrasquel, R. Rodriguez, and L. Tineo, "Ordenamiento de Números Difusos," in *Memorias de la Segunda Conferencia Nacional de Computación, Informática y Sistemas*. Caracas, Venezuela, pp. 205–208, October 2014.
- [8] L. A. Zadeh, "Fuzzy Sets," *Information and control*, vol. 8, no. 3, pp. 338–353, June 1965.
- [9] S. Abbasbandy and B. Asady, "Ranking of Fuzzy Numbers by Sign Distance," *Information Sciences*, vol. 176, no. 16, pp. 2405 – 2416, August 2006.
- [10] P. Singh, "A Novel Method for Ranking Generalized Fuzzy Numbers," *Journal of Information Science and Engineering*, vol. 31, May 2015.
- [11] Y. Yuan, "Criteria for Evaluating Fuzzy Ranking Methods," *Fuzzy Sets and Systems*, vol. 43, no. 2, pp. 139—157, September 1991.



Sesión de Tecnologías y Herramientas Computacionales

Artículos

Metodología para el Desarrollo de Aplicaciones Web Utilizando Datos Enlazados

Taniana Rodríguez¹, Ricardo Dos Santos¹, Jose Aguilar^{1,*}

¹Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID)
Universidad de Los Andes, Mérida, Venezuela
{taniana, aguilar}@ula.ve, ricardojdsg@gmail.com

*Prometeo researcher, Escuela Politécnica Nacional, Quito, Ecuador
Universidad Técnica Particular de Loja, Ecuador

Resumen—El objetivo de este trabajo es proponer una Metodología para el desarrollo de Aplicaciones Web utilizando datos enlazados (Linked Data), con el fin de publicar los datos de la organización de una forma clara, precisa y más amigable. La metodología se basa en el uso de las nuevas tendencias tecnológicas de Datos Abiertos Enlazados (Linked Open Data). Para probar la Metodología, se presenta el caso de estudio de la producción intelectual de estudios de cuarto nivel de la Universidad de Los Andes (ULA), en específico, el programa del Postgrado en Computación y el Doctorado de Ciencias Aplicadas de la Facultad de Ingeniería de la Universidad de Los Andes de Mérida-Venezuela.

Palabras Claves—Producción Intelectual; Linked Open Data; Ontologías; Web Semántica;

I. INTRODUCCIÓN

La Web ha venido evolucionando drásticamente estos últimos años, comenzando con una Web de Hipertexto que hizo posible enlazar y conectar los documentos (sitios web) compuestos por texto, imágenes, videos, etc. Inicialmente, la Web era interpretada por los humanos y no por las máquinas, pero con el surgimiento de la Web 2.0 se logró compartir los documentos y permitir la colaboración, aunque igualmente la interpretación seguía siendo realizada por los humanos. En la Web 3 o Web Semántica [3], se busca enlazar los metadatos semánticos y ontológicos que describen el contenido de los sitios web, a través del paradigma de “Datos Enlazados”, que permite identificar, describir, conectar y relacionar los elementos de datos de la Web.

La ventaja de utilizar el paradigma de Datos Abiertos Enlazados en la Web Semántica es poder conectar los datos de la web, independientemente de dónde residan. La vinculación se hace utilizando un identificador global llamado Identificador Uniforme del Recurso (URI, por sus siglas en inglés *Uniform Resource Identifier*). Ahora bien, en la actualidad no se cuenta con una metodología que permita desarrollar aplicaciones Web basada en datos enlazados.

Este trabajo tiene como objetivo el desarrollo de una Metodología para el desarrollo de Aplicaciones Web utilizando Datos Enlazados. Esto permite que los datos de la organización se puedan descubrir, reutilizar, por parte de otras aplicaciones. Además, la metodología se prueba en el caso de estudio de la

producción intelectual de cuarto nivel de la Universidad de Los Andes (ULA).

En particular, nosotros proponemos un conjunto de etapas a seguir para especificar una aplicación Web basada en Datos Enlazados, y proponemos una Metodología, llamado Metodología para el Desarrollo de Aplicaciones Web utilizando Datos Enlazados (MEDAWEDE), que facilita el desarrollo de dichas aplicaciones.

II. MARCO TEÓRICO

A. Datos Enlazados

Los datos enlazados o vinculados (en inglés *Linked Data*) describen una forma de publicar los datos estructurados para que se puedan interconectar entre ellos. Para ello, se basa en tecnologías Web, tales como RDF¹ (Resource Description Framework) y los URIs, para compartir información que pueda ser leída automáticamente por los computadores, para conectar y consultar los datos desde diferentes fuentes.

Los Datos Enlazados se han aplicado con éxito en proyectos como Dbpedia², que es un proyecto de la Universidad de Leipzig y la Universidad Libre de Berlín, con la finalidad de transformar de forma automática, los contenidos de Wikipedia en datos estructurados RDF; Europeana [8], que es una biblioteca digital semántica que constituye una agregación de los fondos editoriales de numerosas instituciones culturales de todos los países miembro de la Unión Europea, a través de una única interfaz, para tener acceso a millones de objetos digitales bibliográficos, archivísticos y de arte, que conforma el acervo cultural europeo; EDM, que es el modelo de datos de la Unión Europea, que incluye ontologías como SKOS³ (Simple Knowledge Organization System), DC⁴ (Dublin Core) y FOAF⁵ (Friend Of A Friend); y Datos.bne.es⁶ [9], que es el resultado del proyecto de datos enlazados de la Biblioteca Nacional de España.

¹ <https://www.w3.org/RDF>

² <http://wiki.dbpedia.org>

³ <https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html>

⁴ <http://dublincore.org/documents/dc-rdf>

⁵ <http://xmlns.com/foaf/spec>

⁶ <http://datos.bne.es/ocio.html>

B. Principios de los Datos Enlazados

Linked Data describe un conjunto de prácticas para publicar, compartir y conectar piezas de datos, información y conocimiento en la Web Semántica, usando identificadores URIs y RDF para describir los recursos [4]. Berners-Lee propuso las siguientes reglas para la publicación de Datos Enlazados [5, 11,12]⁷:

- *Identidad:* Utilización de URIs para identificar las cosas
- *Accesibilidad:* Usar URIs HTTP (Hypertext Transfer Protocol) para que las personas puedan buscar recursos
- *Estructura:* Utilizar estándares RDF para describir recursos y SPARQL⁸ (Query Language for RDF) para realizar consultas.
- *Navegación:* Incluir enlaces a otras URIs para descubrir más cosas.

Por lo tanto, los Datos Enlazados se basan en la creación de recursos con información expresada directamente en RDF, relacionados entre sí para realizar consultas. Cada objeto dentro del recurso de los datos enlazados cuenta con un nombre único, su URI, que permite referenciarlo de forma unívoca.

La gran ventaja de los datos enlazados es que permite utilizar y combinar datos procedentes de diferentes fuentes (otros recursos de datos enlazados), y a partir de su integración, extrae nuevo conocimiento.

III. METODOLOGÍA DE DESARROLLO DE APLICACIÓN WEB UTILIZANDO DATOS ENLAZADOS (MEDAWEDE)

A. Fases de la Metodología

Los datos enlazados implican un proceso complejo con diferentes etapas no excluyentes, por lo tanto, el ciclo de vida de los datos bajo tecnologías dato enlazado no es un proceso estático sino dinámico [11]. Así, nosotros proponemos las siguientes etapas en nuestra metodología:

- *Especificación:* se centra en el análisis de fuentes de datos. En esta fase se selecciona el conjunto de datos, algunos datos es probable que se extraigan utilizando técnicas de Big Data. Además, se deben diseñar los URIs a utilizar en los dataset o metadatos, así como la definición de las licencias de los diferentes dataset o metadatos a usar.
- *Modelado:* se centra en la creación del modelo que describe el conocimiento del área de estudio. Para ello se reutilizan ontologías, se utilizan vocabularios estándares, e incluso, se diseñan ontologías propias al área de estudio, utilizando algunas de las metodologías de desarrollo de ontologías [1,2,14]. Para la implementación del modelo se utiliza un editor de ontología, por ejemplo protégé⁹
- *Generación:* se centra en la transformación de los datos al lenguaje de RDF. En la actualidad, existen muchas herramientas para la transformación de datos a RDF. Por

ejemplo, openRefine¹⁰ para datos en formatos CVS o Excel, y el D2RQ¹¹ para base de datos relacionales.

- *Vinculación:* en esta etapa se vinculan los datos con otros conjuntos de datos o datasets para aumentar su valor, visibilidad y calidad. Con estos datos vinculados, se puede mostrar nuevo conocimiento que se encuentra oculto en los datasets. Una de las posibles herramientas utilizadas para la vinculación es Silk¹², también se puede vincular de forma manual, a través de los siguientes vocabularios: owl:sameAs, rdfs:seeAlso, rdfs:subClassOf, foaf:knows, rdfs:subPropertyOf.
- *Publicación:* del conjunto de datos en un triplestore o almacenamiento RDF, también llamado almacen de tripletas. Para ello, se usan herramientas tales como Virtuoso¹³, Pubby¹⁴, etc.
- *Explotación de la información:* Esta etapa permite el manejo e integración de distintas interfaces para consumir los recursos publicados de manera agradable y sencilla. Para esta fase se propone el uso de una arquitectura orientada a servicios [15].

B. Tareas Principales de la Metodología para Desarrollar Aplicaciones Web

Las etapas de nuestra metodología llevan a la realización de dos grandes tareas, que facilitan el desarrollo de las aplicaciones Web. En la Figura 1 se muestran esas dos tareas en la Metodología propuesta:

- La primera es modelar los datos de la organización, utilizando los principios de Datos Enlazados
- La segunda es generar la aplicación Web en base a los datos generados en la primera tarea.

Las siguientes subsecciones describen las dos tareas.

C. Modelar los Datos de la Organización

El modelado de los datos de la organización se realiza en dos fases, en la primera se modela la base de conocimiento de la organización, es decir, se realiza el modelo ontológico de la organización, y luego, en la segunda fase se transforman los datos de la organización con los principios de datos enlazados.

Las fases detalladas que se siguen en nuestra metodología en esta tarea se indican a continuación:

- *Modelo ontológico de la organización:* en el modelado se analiza la organización en base a la información que la organización posee, por ejemplo: sus páginas Web, sus sistemas computacionales, los documentos que describen a la organización, entre otros. Una vez obtenida toda la información, se genera el esquema conceptual ontológico de la organización de manera semiautomática, según donde ella se encuentre:

⁷ <https://www.w3.org/DesignIssues/LinkedData.html>

⁸ <https://www.w3.org/TR/rdf-sparql-query>

⁹ <http://protege.stanford.edu>

¹⁰ <http://openrefine.org>

¹¹ <http://d2rq.org>

¹² <http://silkframework.org>

¹³ <https://virtuoso.openlinksw.com>

¹⁴ <http://wifo5-03.informatik.uni-mannheim.de/pubby>

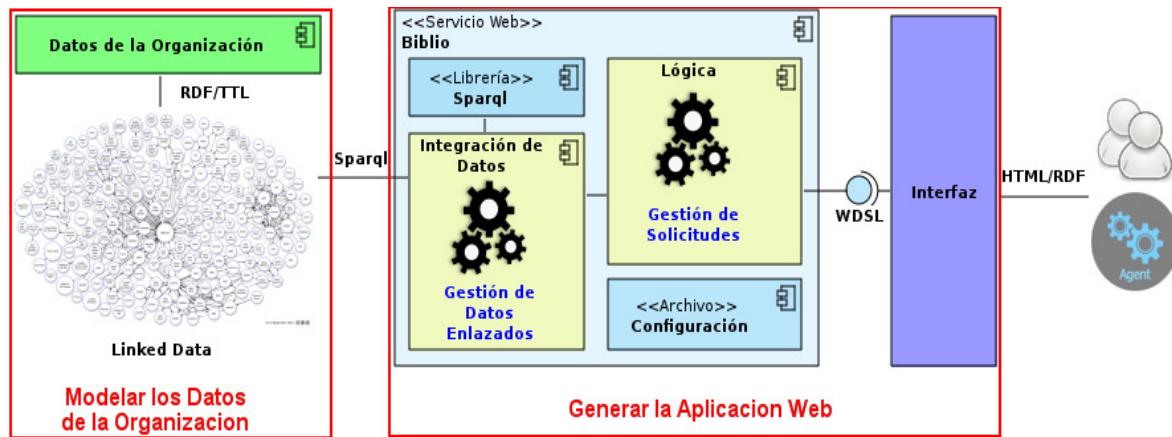


Figura 1. Metodología para el Desarrollo de Aplicación Web utilizando Datos Enlazados (MEDAWEDE)

- *Caso 1. La información está en páginas web o documentos:* en este caso se usa un sistema de extracción de entidades y relaciones [10], o técnicas de minería semántica [13, 14], para la generación del esquema conceptual.
- *Caso 2. La información se encuentra en bases de datos:* el esquema conceptual se extrae del esquema de la bases de datos.
- *Caso 3. La información se encuentra en páginas web, documentos y bases de datos:* en este caso, si se realiza el caso 1 y 2 por separado, luego se analizan los esquemas generados y se fusionan en el esquema conceptual

Una vez generado el esquema conceptual ontológico, se valida el esquema con los expertos del dominio.

Seguidamente, se pasa a explorar los vocabularios ya existentes para enriquecer al esquema del modelo ontológico de la organización. Por ejemplo, se pueden usar los vocabularios que describen a las personas (FOAF, VCARD, Schema#Person, etc), a la organización (schema.org#Organization), en el Modelo Ontológico de la Organización.

- *Transformar los datos de la organización en los principios de datos enlazados:* en esta fase se transforma los datos en RDF usando el modelo ontológico generado por la tarea anterior. El objetivo de esta fase es la generación de RDF desde las fuentes de información asociadas a la organización, para transformar los datos originales a un formato estándar e interoperable en el contexto de la Web. En la Figura 2 se muestra el flujo de trabajo a usar para la transformación.

Esta transformación se realiza en tres pasos, que se describen a continuación

1. Convertir los datos en Formato CSV (Comma-Separated-Values): Este proceso tiene como entrada las fuentes de datos sin procesar, en distintos formatos. Usualmente, la mayoría de los datos está en base de datos relacionales,

textos, o en archivos XML, y se convierte en formato estándar, en este caso en formato CVS.

2. Seleccionar los Datos: los expertos toman el archivo generado del proceso anterior y analizan los datos. En este proceso se extrae la información que se requiere, es decir, cuáles de los datos se van a publicar con acceso abierto.
3. Generar RDF y enlazado: Este proceso tiene como entrada el archivo generado en el proceso anterior, y los diferentes vocabularios. Esos datos se transforman a RDF explotando esos vocabularios, dando como resultado el archivo rdf. Durante esta tarea se usan archivos OWL¹⁵ (Web Ontology language), RDF, y los vocabularios, para realizar la vinculación de los datos con otros datos.

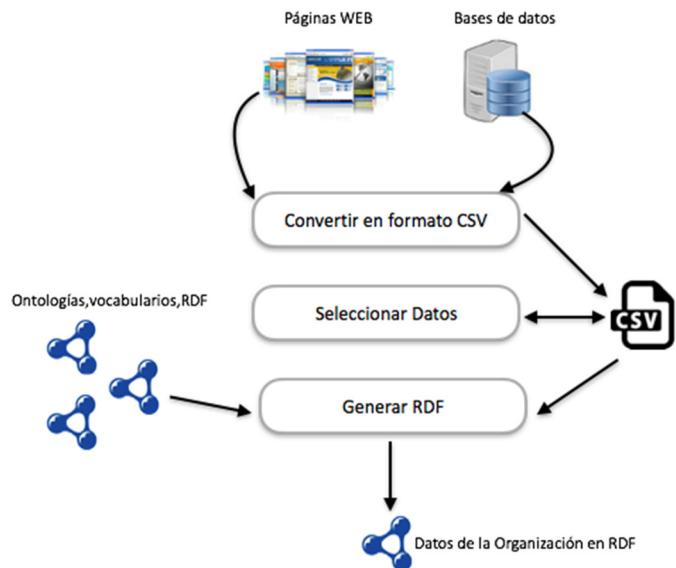


Figura 2. Flujo de Trabajo para la Transformación de los Datos de una Organización en RDF

¹⁵ <https://www.w3.org/OWL>

D. Generar la Aplicación Web

La generación de la aplicación web se desarrolla en dos fases, en la primera fase se crea el servicio web que ofrecerá una serie de métodos o funciones para la gestión de solicitudes y de datos enlazados, que permitirá consumir los datos enlazados de la organización. En la segunda fase se crea la interfaz visual, que servirá como acceso web a los datos, a los usuarios finales y a los agentes maquinas [15]. Las fases detalladas de esta tarea son:

- Crear el servicio web: En esta fase se procede a seleccionar el lenguaje de programación y las librerías para la implementación del servicio web. Luego se generan las operaciones o métodos para la Gestión de Solicitudes que ofrecerá el servicio web, que permitirán consumir el conocimiento publicado por la organización. Por último, se crean las operaciones privadas para la Gestión de los Datos Enlazados. Allí se definen las consultas a los repositorios de los recursos publicados por la organización, a través de la tecnología de Linked Data, y para las consultas se usa el lenguaje Sparql.
- Crear la interfaz: Lo primero que se define en esta fase es el protocolo que se va a usar para consumir el servicio web creado en la fase anterior. Luego, conociendo los métodos disponibles en el servicio web, se implementa la navegabilidad de la interfaz, para obtener las características deseadas.

La ventaja de separar la lógica de negocio en una fase con tecnología de servicio web, radica en que la lógica de negocio pasa a ser un módulo, lo que permite ser invocado por múltiples interfaces y distintas aplicaciones, ya sea de ventana, móviles o web.

IV. CASO DE USO DE MEDAWEDE: PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE LOS ANDES

En esta sección describimos el uso de nuestra metodología, en un caso de estudio. El caso de estudio usa toda la información y datos de los programas del Postgrado en Computación y del Doctorado de Ciencias Aplicadas de la Facultad de Ingeniería de La Universidad de Los Andes.

A. Modelar los Datos de la Organización

El modelado se realiza en dos fases.

Fase 1: Modelo ontológico de la organización

En la Figura 3 se muestra la red de ontologías, que es una colección de ontologías entrelazadas, que incluye el uso de vocabularios existentes, que describe el modelo ontológico del caso de estudio.

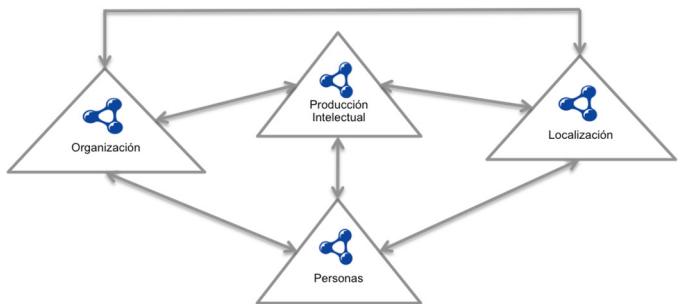


Figura 3. Red de Ontologías Utilizada en el Caso de Estudio

A continuación se describe brevemente la red de ontología del caso de estudio: la *producción intelectual* es producida por *personas*, en este caso las personas son autores y tutores, los autores son los que generan la producción intelectual y los tutores orienta a los autores para la generación de la producción intelectual. Además, la producción intelectual es publicada por una *organización*, y por último; la organización, la producción intelectual y las personas están ubicados en una *localización*.

Para el modelado de los datos de la organización se utilizan vocabularios, tales como los vocabularios dublin core, foaf, Bibo, que contienen varias propiedades y conceptos que requiere nuestro modelo, tales como:

- Bibo: Representa los conceptos y propiedades que permiten describir las citas y referencia bibliográficas, y puede ser usado para describir la producción intelectual de una organización
- DublinCore: Dublin Core Metadata Initiative (DCMI) fue desarrollado para representar recursos y atributos que se utilizan en el ámbito bibliográfico.
- FOAF-Vcard: fue desarrollado para representar/describir atributos y relaciones entre entidades.

A partir del modelo conceptual de la Figura 3, se diseñó la ontología de la organización. En la Figura 4 se muestra parte de la base de conocimiento de la producción intelectual, la cual describe que una tesis es creada por un autor (CodigoTesis dc:creator CódigoAutor), el CódigoAutor es de tipo Autor (CodigoAutor rdf:type Autor), la tesis es tutoriado por un tutor (CodigoTesis dc:contributor CódigoTutor), además, el CódigoAutor tiene los siguientes atributos: Nombre, Primer Nombre, Apellidos etc., al igual que el CódigoTutor. Adicionalmente, indica que el CódigoAutor y CódigoTutor son personas, y CódigoTesis es una TesisMaestria o es una TesisDoctorado, y la TesisMaestria y TesisDoctorados es una Tesis. En el modelado se utilizó el vocabulario Dublin Core que tiene el prefijo dc, y foaf que tiene el prefijo foaf.

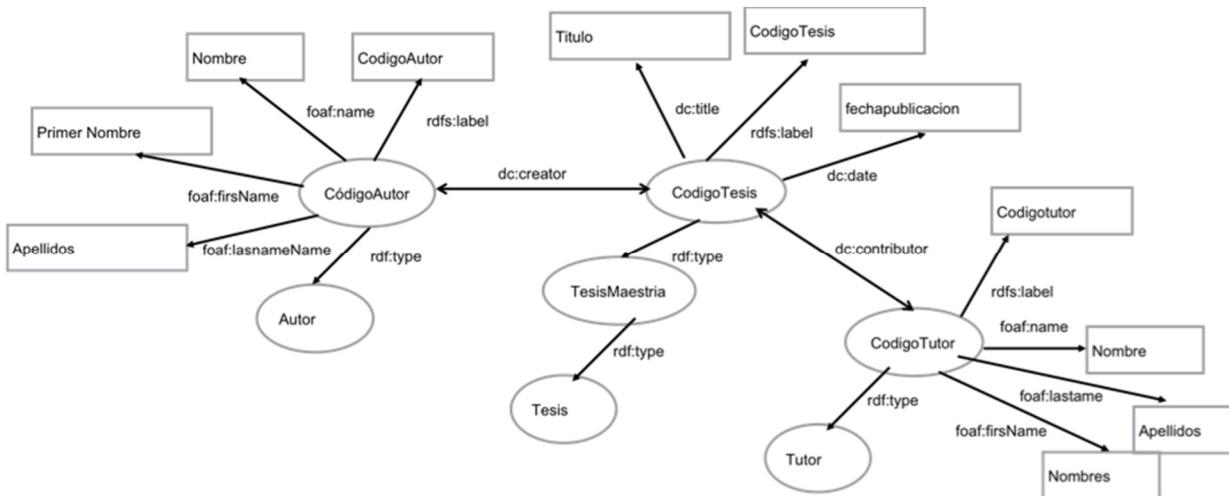


Figura 4. Parte de la Base de Conocimiento de la Producción Intelectual

Luego que se tiene la base de conocimiento, se realiza la segunda fase, que se describe a continuación.

Fase 2: Transformación de los datos de la organización

En esta fase se transforma los datos a RDF, y se realiza el enlazado. En particular, se genera en RDF las fuentes de información asociadas a la producción intelectual, tal que queden transformados los datos originales a un formato estándar e interoperable en el contexto de la Web. Para el proceso de transformación se realizaron los pasos de la Figura 2:

1. Convertir a Formato CSV: Las fuentes de datos sin procesar de la producción intelectual son páginas Web y bases de datos, se procedió a convertir los datos al formato csvs.
 2. Seleccionar los Datos: los expertos toman el archivo generado del proceso anterior y analizan los datos. En este proceso se extraen los datos que se van a publicar con acceso abierto.
 3. Generación de RDF y enlazado: Una vez obtenido los datos, se transforma los datos a RDF con la herramienta OpenRefine (<http://openrefine.org>), usando los vocabularios respectivos, generando el archivo RDF. En la Figura 5 se muestra el esqueleto de RDF para los autores.



Figura 5. Esqueleto RDF para el Autor de Tesis

Todos los datos son transformados (tutores, las tesis, etc.) siguiendo el modelo conceptual de la Figura 4, dando como resultado un archivo como el que se muestra en la Figura 6.

Una vez obtenidos los datos transformados en RDF, se almacenan los datos en un repositorio de datos RDF. En nuestro caso, utilizamos OpenLink Virtuoso y Fuseki. Ya terminado esta etapa, se pasa a la fase de explotación de la información, en la que por ejemplo, se puede generar una aplicación web.

```

@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://ProdIntecULA/Tesis/Autor_1> a <http://ProdIntecULA/Tesis/Autor> ;
    rdfs:label <http://ProdIntecULA/Tesis/Autor_1> ;
    foaf:name "Juan Luis Chaves Sanabria" ;
    foaf:firstName "Juan Luis" ;
    foaf:lastName "Chaves Sanabria" .

<http://ProdIntecULA/Tesis/Tesis_1> a <http://ProdIntecULA/Tesis/TesisMaestria> .

<http://ProdIntecULA/Tesis/Autor_1> dc:creator <http://ProdIntecULA/Tesis/Tesis_1> .

<http://ProdIntecULA/Tesis/Autor_2> a <http://ProdIntecULA/Tesis/Autor> ;
    rdfs:label <http://ProdIntecULA/Tesis/Autor_2> ;
    foaf:name "Hungria Zulema Berbesi Quintero" ;
    foaf:firstName "Hungria Zulema" ;
    foaf:lastName "Berbesi Quintero" .

<http://ProdIntecULA/Tesis/Tesis_2> a <http://ProdIntecULA/Tesis/TesisMaestria> .

<http://ProdIntecULA/Tesis/Autor_2> dc:creator <http://ProdIntecULA/Tesis/Tesis_2> .

<http://ProdIntecULA/Tesis/Autor_3> a <http://ProdIntecULA/Tesis/Autor> ;
    rdfs:label <http://ProdIntecULA/Tesis/Autor_3> ;
    foaf:name "Rosita del Socorro Bruscino López" ;
    foaf:firstName "Rosita del Socorro" ;
    foaf:lastName "Bruscino López" .

<http://ProdIntecULA/Tesis/Tesis_3> a <http://ProdIntecULA/Tesis/TesisMaestria> .

<http://ProdIntecULA/Tesis/Autor_3> dc:creator <http://ProdIntecULA/Tesis/Tesis_3> .

<http://ProdIntecULA/Tesis/Autor_4> a <http://ProdIntecULA/Tesis/Autor> ;
    rdfs:label <http://ProdIntecULA/Tesis/Autor_4> ;
    foaf:name "Emiro Antonio Coronado Cabrera" ;
    foaf:firstName "Emiro Antonio" ;
    foaf:lastName "Coronado Cabrera" .

```

Figura 6. Datos Transformados de los Autores de las Tesis del Postgrado en Computación

B. Generación de la Aplicación Web

Fase 1: Crear Servicio Web

Esta fase se apoya en el paradigma SOA (ver Figura 7). En este caso en particular, se necesita que la aplicación tenga métodos que permitan mostrar el listado de tesis, artículos, autores y tutores, como también, que permitan ver la información relacionada a cada uno de ellos.

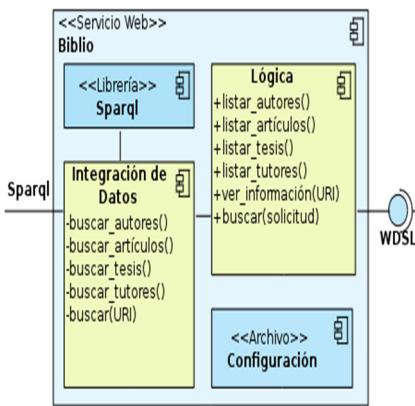


Figura 7. Servicio Web

Un aspecto a considerar para el diseño de la aplicación, es su capacidad para consumir los recursos publicados de la organización bajo la tecnología de datos enlazados. En la Figura 8 se muestra una representación gráfica de los datos de la organización, donde se puede observar de color verde los autores, de azul los tutores, y en forma de documento los artículos y tesis. También se observan las relaciones entre ellos.

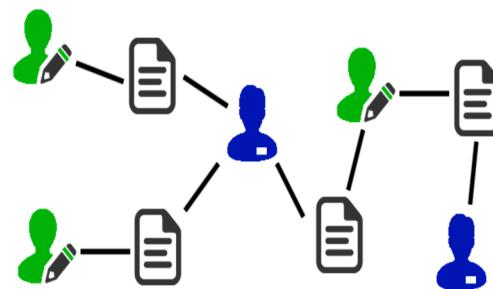


Figura 8. Representación de los Datos Enlazados de la Organización

Para este caso de estudio, se usó Python¹⁶ como lenguaje de programación para implementar el Servicio Web. Las librerías usadas son las siguientes: spyne¹⁷ para el Servicio Web y sparqlwrapper¹⁸ para el uso de Sparql. En la Figura 9 se observa parte de los protocolos configurados, los puntos de montaje, y la creación del servicio web.

```

yaml = Application([service], tns=tns,
    in_protocol=HttpRpc(),
    out_protocol=YamlDocument())

rest = Application([service], tns=tns,
    in_protocol=HttpRpc(), out_protocol=HttpRpc())

xml = Application([service], tns=tns,
    in_protocol=HttpRpc(), out_protocol=XmlDocument())

html = Application([service], tns=tns,
    in_protocol=HttpRpc(), out_protocol=HtmlMicroFormat())

soap = Application([service], tns=tns,
    in_protocol=Soap11(validator='lxml', pretty_print=True),
    out_protocol=Soap11())

root = WsgiMounter({
    'rest': rest,
    'xml': xml,
    'html': html,
    'soap': soap,
    'json': json,
    'jsoni': jsoni,
    'jsonl': jsonl,
    'jsonnl': jsonnl,
    'yaml': yaml,
})

from wsgiref.simple_server import make_server
server = make_server(host, port, root)
Debug()
server.serve_forever()

```

Figura 9. Servicio Web con spyne

Luego se definen los métodos o funciones de la Gestión de Solicitudes, que servirán como interfaz para la solicitud de información. Entre los métodos definidos están los siguientes: listar_autores, listar_artículos, listar_tesis, listar_tutores, ver_información y buscar. En la Figura 10 se observa un ejemplo de la definición de los métodos.

```

class ULA_Biblio(ServiceBase):

    @srpc(_returns=Iterable(String))
    def listar_autores():
        return buscar_autores()

```

Figura 10. Definición de los Métodos o Funciones del Servicio Web

¹⁶ <https://www.python.org>

¹⁷ <http://spyne.io>

¹⁸ <https://rdflib.github.io/sparqlwrapper>

Por otro lado, se diseña un módulo de integración de los datos, donde se definen los métodos privados que permiten buscar la información usando Linked Data. Entre estos métodos se tiene los siguientes: buscar_autores, buscar_artículos, buscar_tesis, buscar_tutores, buscar por la URI, y en cada uno de estos métodos se especifican las consultas en lenguaje sparql que permiten obtener la información de los datos enlazados, como se observa en la Figura 11 el método buscar_autores.

```
from SPARQLWrapper import SPARQLWrapper, JSON
def buscar_autores():
    sparql = SPARQLWrapper("http://127.0.0.1:8890/sparql")
    sparql.setQuery('''
        SELECT ?uri WHERE {
            ?uri a <http://ProdIntecULA/Tesis/Autor> .
            ?uri foaf:lastName ?lastname
        }
    ''')
    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()
    for result in results["results"]["bindings"]:
        yield { 'uri': result["uri"]["value"], 'nombre': result["lastname"]["value"] }
```

Figura 11. Consultas en Lenguaje sparql del Servicio Web

Finalmente, se observa en la Figura 12 dos resultados obtenidos usando el Servicio Web de este caso de estudio. El primer resultado en la parte superior, es una respuesta SOAP con la especificación del servicio (WSDL) en formato XML, dicha especificación es muy importante para la construcción de la interfaz que consume este servicio, ya que se indica los métodos disponibles, con sus parámetros. El segundo resultado que se muestra en la parte inferior, es una respuesta REST en formato JSON, donde se muestran todos los autores devueltos por el método listar_autores.

```
[[{"nombre": "Adarme Jaimes", "uri": "http://ProdIntecULA/Tesis/Autor_83"}, {"nombre": "Albornoz Espinel", "uri": "http://ProdIntecULA/Tesis/Autor_84"}, {"nombre": "Altamiranda P\u00f3ncolerez", "uri": "http://ProdIntecULA/Tesis/Autor_79"}, {"nombre": "Alzur\u00f3n Cortez", "uri": "http://ProdIntecULA/Tesis/Autor_69"}, {"nombre": "Anaya Salazar", "uri": "http://ProdIntecULA/Tesis/Autor_53"}]]
```

Figura 12. Respuesta de la Fase Servicio Web

Fase 2: Crear la Interfaz Web

Aquí se implementa la o las interfaces visuales que sean necesarias para facilitar la visualización de la información de la organización, que esta publicada como datos enlazados.

Para implementar la interfaz, primero hay que definir cómo se va a consumir el servicio web. Las dos formas más comunes

son implementarla con PHP¹⁹ (PHP: Hypertext Preprocessor), o con el lenguaje de programación Python.

En la Figura 13 se muestra el código en PHP que invoca/usa el método listar_autores del servicio web, allí se implementa el uso del protocolo REST con respuesta en formato de intercambio Json.

```
<?php
$url="http://127.0.0.1:8000/json/listar_autores";
$contenido = file_get_contents($url);
$contenido = utf8_encode($contenido);
$resultados = json_decode($contenido);

echo "<h2>Lista de Autores</h2>";
foreach ($resultados as $resultado) {
    echo "<b>URI</b>: ".$resultado->uri."<br><b>Nombre:</b> ". $resultado->nombre. "<br><br>";}
```

Figura 13. Implementación de la Interfaz Web con PHP

La segunda implementación que se muestra en la Figura 14 hace uso del mismo método lista_autores, pero en este caso usando python. Esta implementación es un poco más elaborada, usa el protocolo SOAP con un formato de intercambio XML, por lo que necesita implementar un cliente web que lea la descripción del servicio (WSDL) y se comunique a través del protocolo SOAP. Para este fin, se usa la librería suds²⁰, y para ver el resultado como una interfaz web, se usa la librería Flask²¹, que es un microFramework para Python que despliega un servidor web que permite mostrar la interfaz desarrollada en este lenguaje.

```
import suds
from flask import Flask
app = Flask(__name__)

@app.route("/")
def listar_autores():
    url = "http://localhost:8000/soap/?wsdl"
    cliente = suds.client.Client(url)
    resultados = cliente.service.listar_autores()

    salida="<h2>Lista de Autores</h2>"
    for resultado in resultados[0]:
        salida += "<b>URI</b>: " + resultado[1] + "<br><b>Nombre:</b> " + resultado[0] + "<br><br>"

    return salida

if __name__ == "__main__":
    app.run()
```

Figura 14. Implementación de la Interfaz Web con Python

Como se observa en la Figura 15, ambas implementaciones generan los mismos resultados al mostrar la lista de autores, aunque se estén usando distintos protocolos de intercambio de mensajes, lenguajes de programación y/o librerías.

Finalmente, en la Figura 16 se muestra la interfaz de la aplicación usando los distintos métodos requeridos en la fase del desarrollo del Servicio Web, que permite la navegabilidad usando los datos enlazados. Allí se observa un menú desplegable con los distintos tipos de listas, acompañado de un buscador de información relacionada a una URI de la organización, y en el cuerpo principal se muestra la lista de autores a través del método listar_autores, donde cada autor posee un enlace a su respectiva información. En la Figura 17 se muestra la informa de una URI a través del método ver_información, la cual retorna todas las relaciones que existe

¹⁹ <https://secure.php.net>

²⁰ <https://pypi.python.org/pypi/suds>

²¹ <http://flask.pocoo.org>

con dicha URI. En este caso, se muestra la información de una tesis, y allí se observa el título de la tesis, el tutor, la fecha, entre otras cosa más. Hay que recalcar que cada información que es mostrada esta enlazada a otros datos, característica fundamental en los datos enlazados.

localhost/ula/cliente.php	127.0.0.1:5000
Lista de Autores REST con PHP	Lista de Autores SOAP con Python
URI: http://ProdIntecULA/Tesis/Autor_83 Nombre: Adarme Jaimes	URI: http://ProdIntecULA/Tesis/Autor_83 Nombre: Adarme Jaimes
URI: http://ProdIntecULA/Tesis/Autor_84 Nombre: Albornoz Espinel	URI: http://ProdIntecULA/Tesis/Autor_84 Nombre: Albornoz Espuel
URI: http://ProdIntecULA/Tesis/Autor_79 Nombre: Altamiranda Pérez	URI: http://ProdIntecULA/Tesis/Autor_79 Nombre: Altamiranda Pérez

Figura 15. Resultados de las Distintas Implementaciones del Consumo del Servicio Web

localhost/ula/listar_autores.php

UNIVERSIDAD DE LOS ANDES VENEZUELA

LISTAR POR: Buscar Buscar

Lista de: Autores

- Autores
- Tutores
- Tesis
- Articulos

http://ProdIntecULA/Tesis/Autor_1
Chaves Sanabria
http://ProdIntecULA/Tesis/Autor_10
Ojeda Subaran
http://ProdIntecULA/Tesis/Autor_100
Figueroa Linares

localhost/ula/ver_informacion.php?uri=http://ProdIntecULA/Tesis/Autor_10

Figura 16. Interfaz Mostrando la Lista de Autores

localhost/ula/ver_informacion.php?uri=http%3A%2F%2F

UNIVERSIDAD DE LOS ANDES VENEZUELA

LISTAR POR: Buscar

http://www.w3.org/2007/05/powder-s#describedby
http://localhost:8890/about/id/entity/http://localhost/ProduccionIntelectualULA
http://purl.org/dc/elements/1.1/date
2016-03-08
http://purl.org/dc/elements/1.1/contributor
http://ProdIntecULA/Tesis/tutor_080
http://purl.org/dc/elements/1.1/title
Escaneo de Redes IEEE 802.11: Un enfoque práctico para la Internet del Futuro

localhost/ula/ver_informacion.php?uri=http://ProdIntecULA/Tesis/tutor_080

Figura 17. Interfaz Mostrando la Información de una Tesis

V. CONCLUSIONES

Este trabajo presentó una metodología para explotar los datos enlazados de una organización. Además, en particular, define una Metodología para el desarrollo de Aplicaciones Web usando datos enlazados. Este Metodología tiene dos grandes tareas, la primera es el modelado de los datos de la organización, y la segunda es la generación de la aplicación web usando los datos enlazados, aprovechando la fortaleza de los servicios web a nivel de interoperabilidad e independencia de plataformas.

La metodología se probó en un caso de estudio, vinculado a la producción intelectual de una universidad, usando los principios de datos enlazados y el estándar RDF, para así lograr que la producción intelectual que se produce en la universidad sea más accesible y visible en la Web.

Entre los trabajo futuros se tiene la incorporación de toda la producción intelectual generada en la Universidad de Los Andes. También, se aspira utilizar la metodología en proyectos más complejos.

AGRADECIMIENTO

Dr. Jose Aguilar ha sido parcialmente financiado por el Proyecto Prometeo del Ministerio de Educación Superior, Ciencia, Tecnología e Innovación de la República de Ecuador).

REFERENCIAS

- [1] M.C. Suárez Figueroa. "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse." PhD Thesis. Universidad Politécnica de Madrid, 2010.
- [2] A. Gómez-Pérez, M. Fernández-López, O. Corcho. "Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web." Springer. 2004.
- [3] T. Berners-Lee, J. Hendler, O. Lassila. "The Semantic Web," Scientific American, Vol. 284, No. 5, pp. 28-37, 2001.
- [4] O. Hartin, A. Langegger. "A Database Perspective on Consuming Linked Data on the Web." <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.2688&rep=rep1&type=pdf>, 2010.
- [5] R. Míguez Pérez, J. M. Santos Gago, V. M. Alonso Rorís, L. M. Álvarez Sabucedo y F. A. Mikic Fonte. "Linked Data como Herramienta en el Ámbito de la Nutrición". Nutrición Hospitalaria., Vol. 27. No. 2, pp. 323-332, 2012.
- [6] C. Hernández. "La Aplicación de Linked Data en la Agrupación Datos Bibliográficos" MSc. Tesis. Universidad de Salamanca. Salamanca, 2012.
- [7] A. Rotem-Gal-Oz. "SOA Patterns", Manning, 2012.
- [8] M. Doerr, S. Gradmann, S. Hennicke, A Isaac, C. Mehini y H. Van de Sompel. "El Modelo de Datos de Europeana (EDM). World Library and Information Congress: 76th IFLA General Conference and Assembly, Gotemburgo, Suecia, 2010.
- [9] D. Vila-Suero, B. Villazón-Terrazas y A. Gómez-Pérez. "datos.bne.es: a Library Linked Data Dataset". Semantic Web. IOS Press. 00. 1-6. 2012.
- [10] T. Rodríguez, J. Aguilar, "Aprendizaje Ontológico para el Marco Ontológico Dinámico Semántico", DYNA, Vol. 81, No. 187, pp. 56-63, 2014.
- [11] D. Wood, M. Zaidman and L. Ruth with M. Hausenblas. "Linked Data Structured Data on the Web", Manning Publications Co. 2014.
- [12] T. Heath and C. Bizer. "Linked Data: Evolving the Web into a Global Data Space". Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Clapool Publishers series. 2011.
- [13] C. Rangel, J. Altamiranda, J. Aguilar; "Semantic Mining based on Graph Theory and Ontologies. Case Study: Cell Signaling Pathways", CLEI Electronic Journal, Vol. 19, No. 2, pp. 7-7, 2016.
- [14] J. Aguilar, O. Portilla, "Framework Basado en ODA para la Descripción y Composición de Servicios Web Semánticos (FODAS-WS)", Latin American Journal of Computing, Vol. 2, No. 2, pp. 15-24, 2015.
- [15] V. Bravo, J. Aguilar, F. Rivas, M. Cerrada, "Diseño de un Medio de Gestión de Servicios para Sistemas Multiagentes", XX Conferencia Latinoamericana de Informática, pp. 431-439, 2004.

Multiobjective Optimization Model using Genetic Algorithms for Oil Wells based on Electro Submersible Pumps

Edgar Camargo¹, Egner Aceros², Lice Guzmán¹, Hector Lozada¹, Jose Aguilar³

¹PDVSA, Distrito Socialista Tecnológico –AIT, Mérida, 5101, Venezuela

{camargoea, guzmanlp, lozadah}@pdvsa.com

²Facultad de Ingeniería, División de Postgrado

Universidad del Zulia Maracaibo, 4001, Venezuela

egner.aceros@cantv.net

³CEMISID, Facultad de Ingeniería, Universidad de los Andes

Mérida, 5101, Venezuela

aguilar@ula.ve

Abstract— The processes of optimization of oil wells involve an objective function that maximizes the production and minimizes the energy of lift. For the solution of this type of multi-objective problem, in the last decade the evolutionary technologies have demonstrated to be an effective and efficient tool. In this work is presented an evolutionary approach, in order to improve the performance of the Industrial Production in petroleum wells based on electro submersible pumps (ESP). The production is modeled using Numerical Methods based on Ordinary Differential Equations (ODE).

Keywords—evolutionary computation, multi-objective problem, production of well.

I. INTRODUCTION

The electric submersible pump (ESP), a tried and true workhorse of artificial lift systems, has a long track record for improving production and recovery rates in oil fields worldwide. As the industry has moved into more challenging frontiers and pushed to boost production from brownfields, ESP technology has evolved with the development of increasingly robust mixed flow systems that can handle varied flow rates and provide expanded operating ranges for a wide variety of hydrocarbon environments [1].

The decline of “easy” oil, however, has raised the bar even higher for artificial lift, including ESP performance. Many production companies are finding that the typical for ESP is a limitation, especially in complex, remote locations where intervention costs are high and lead to poor field economics. In these intervention constrained markets, the expense of ESP replacement, including workover costs and deferred production. The risks of early ESP failure and inconsistent performance in high intervention cost environments in challenging fields are the reasons many producers often choose alternative mechanism such as gas lift for artificially lifting their wells that typically do not deliver the highest recovery rates. That conundrum has led operators to put the

need for a consistently reliable, longer lasting ESP systems at the top of their wish lists as they strive to maximize efficiency and reduce their total cost of ownership [2].

In response to that, this article presents the design and implementation of a computational tool for the optimization of systems lift of oil. Particularly, the oil and gas companies carry out constant efforts to optimize their systems of production. These efforts are directed to maximize the total daily production of hydrocarbons, minimizing the environmental costs, between others aspects, that extends the run life over conventional ESPs.

On the other hand, the Evolutionary Algorithms (EA) are technologies of optimization and search inspired by the natural evolution. There are other techniques in the literature that can be compared with EA [12], [13]. These algorithms define a population of individuals, each of them representing a possible solution to the proposed problem. Every individual of the population is defined genetically, this way, the best individuals are those that possess better genes. The best individuals are combined between them so that they form new solutions that possess better genes than their antecessors [3]. These new solutions generate a new population of individuals replacing the previous one ([3], [4], [11]).

The Evolutionary Algorithms (EA) is robust and effective methods for the resolution of problems of optimization. In [4] is presented an example of application in the petroleum industry, for the development of the concept “intelligent fields”, in which it is used for the analysis and interpretation of a great quantity of information, combined with neural networks. So, A concept of industrial automation for control and optimization of the production of hydrocarbon based on genetic algorithms is presented [4].

So, in this work we propose a system for the optimization of wells using the Evolutionary Algorithms, specifically for wells based on electro submersible pumps (ESP). In this case,

we have determined two objectives to optimize: maximize the production of hydrocarbons and minimize the energy lift, which generates a zone of negotiation that allows finding the ideal production. The optimization system is composed of one first phase (component) of well model generation (using techniques of mass and energy balance), and one second phase (component) of optimization of the productive process for the identified scenario [5]. This paper is structured as follows: Theoretical aspects about Evolutionary Algorithms and the Production Process of well are presented in Section 2. The design of the Evolutionary Optimization System and the Model of the Well is presented in Section 3, while the results are shown in Section 4. The paper ends with conclusions.

II. THEORETICAL FRAMEWORK

A. Evolutionary Algorithms

The Evolutionary Algorithms (EA) simulates the process of natural evolution [4]. They consist of an iterative technology that applies operators over a set of individuals of the population with the intention of improving their "fitness", a measure related to the objective function of the problem in question. Every individual of the population represents a potential solution of the problem. Initially, the population is generated randomly, and then it evolves by means of operators', which include recombination of individuals, crossover and mutations. This evolution is guided by a strategy of selection of the individuals most adapted to the resolution of the problem, according to their values of fitness. The Figure 1 presents a generic scheme of an EA. The intention of the Selection is to emphasize in the most capable individuals of the population, hoping that their children have fitness better than that of the parents. The crossover operator combines several individuals to generate new individuals, and the mutation is a random process where a gene is replaced by other for producing a new genetic structure. The Replacement is a process that substitutes individuals of the population (normally the worst) for the new individuals created (normally the best). As soon as a new population has been produced, the "fitness" of the individuals in the new population can be determined and restarts the process ([3], [4], [11]).

```

Begin (P(0))
generation=0
Test (P(0))
While (no CriterionStop) do
    Father = Selection (P(generation))
    Sons = Operators of Reproduction (Father)
    NewPop = Replacement (Sons, P(generation))
    generation ++
    P(generation) = NewPop

```

Figure 1. Scheme of an AE

B. Electrical Submersible Pump Method

ESPs are turbo machines that transfer energy to fluids by means of stages conformed by one impeller and one diffuser. These devices, as any centrifugal devices, are highly affected by changes in the viscosity of the fluid that is being pumped. For a centrifugal pump, as the viscosity of the fluid increases, the required brake horsepower increases as well; whereas, head capacity and efficiency decrease. There are tools that allow

engineers to evaluate the effects of viscosity of fluids in the pump with a reasonable accuracy. The pump of an ESP system may have a significant number of stages spinning at high velocities. As a result, a volume of fluid is exposed to a considerable amount of shear force that induces increments on the fluid temperature. Consequently, and according to the liquids that are being pumped, there might be a considerable difference between the viscosity of the fluid at the inlet of the pump and viscosity of the fluid at the outlet of the pump. Therefore, the stages of pumps should not have the same performance, resulting in an increase of the capacity to generate pressure in the upper section of the system ([1], [2], [10]).

So, the ESP process of transforming mechanical energy into hydraulic energy is characterized by the pump performance curves Figure 2, given at a operational speed that is often expressed in terms of frequency (f). The hydraulic energy that an ESP system is able to deliver is expressed in Height (ft) and Flow Rate(Q); The Energy that a pump demands of the motor is represented by the Brake Horsepower Curve (Potency,Hp), and the Curve of Efficiency (η ,%) is a representation of how much efficient this energy transformation.

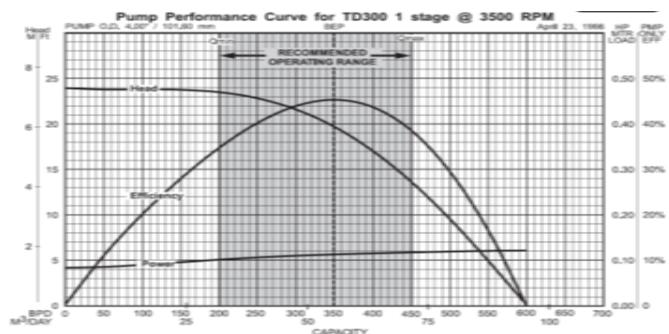


Figure 2. Curve of Performance of Pump a One Stage in 3600 RPM (60 Hz)

III. DESIGN OF THE EVOLUTIONARY OPTIMIZATION SYSTEM

A. First Phase: Generation of the Well Production Model

The mathematical models presented in this work are inspired on ([5], [6], [10]), and basically consist of checking the pressure profile from the head (Pwh) to the bottom (Pbh) of the well, to determine the current production capacity (Q), which exhibits the well through the pumping system. For this purpose, the method called nodal analysis is used ([5], [6], [7]). Thus, a simple model of production by artificial lifting by electric submersible pump is proposed: the inflow of crude oil and gas of the reservoir is modeled by the use of the productivity index (relationship between the production rate (Q) and the differential Between the reservoir pressure (Pr) and the flowing pressure at the bottom of the well (Pbh), using Equation 1, which determines the capacity of the oil reservoir. It is normal for this capacity to decrease over time, for the reduction of the permeability in the vicinity of the well, and for the increase in the viscosity of the well. This Equation 1 is inflow referred to as the energy supply curve, or flow of fluids, that the reservoir delivers to the well (Pbh vs q).

$$\frac{dp_{bh}}{dt} = \frac{\beta_1}{V_1} (q_r - q) \quad (1)$$

As for the outflow, described by Equation 2, the pump transfers kinetic energy to the fluid, reducing the weight of the column, increasing the head pressure in the well, allowing to establish a certain production rate in which the fluid supply capacity of the reservoir equals the fluid extraction capacity of the well.

$$\frac{dp_{wh}}{dt} = \frac{\beta_2}{V_2} (q - q_c) \quad (2)$$

Thus, the production of the system responds to a balance of energy in the form of pressure, between the capacity of energy reservoir and the lifting system, expressed in Equation 3 ([2], [5]).

$$\frac{dq}{dt} = \frac{1}{M} (P_{bh} - P_{wh} - (\rho_1 gh_1 + \rho_2 gh_2) + \Delta P_p - \Delta P_f) \quad (3)$$

Where:

Pbh = Bottom Pressure

Pwh = Head Pressure

β_1, β_2 = Volumetric Module

V1, V2 = Volume Multiphase

qr = Input Flow

q = Flow of well

qc = Flow of Valve

ρ_1, ρ_2 = Density of flow

g = gravity

h1, h2 = Vertical Distances

ΔP_p = Differential of Pressures

ΔP_f = Total Pressure Loss in the Well

So, in the case of the pump, the production curve is used in function of the efficiency indicated by the manufacturer, to which a polynomial adjustment is applied, to determine the Differential equation (ED), which describes the production curve in terms of [6]: the efficiency of the pump to proper operating conditions of the well (variables surface and bottom), API grades (measure of how heavy or light a petroleum liquid is compared to water) and kinetic viscosity (v , is the ratio of the dynamic viscosity μ to the density of the fluid ρ). Thus, API and v are data obtained in crude sample at the laboratory level, Equation 4.

$$H_0(q) = aq^5 + bq^4 + cq^3 - dq^2 + eq - f \quad (4)$$

Where a, b, c, d, e and f are constants characterized for a specific pump.

So, to model the production lift system according to the above (Equations 2, 3, 4), we will use numerical methods based on Adams - Bashfort of fourth order predictor and Adams - Moulton corrector, with initial values obtained with the Runge - Kutta method of the fourth order [8].

B. Second Phase: Optimization of the Production Process

The optimization problem of ESP wells consists of increasing the production of oil and minimizing the energy, based on three variables: Height (H,ft), Production (Q,BPD) and Efficiency (η %). This optimization problem is described by the objective function of the Equation 5, with the respective restrictions of the process. The production pipe is modeled with the pressure gradients "Pressure Drop in the Reservoir" and "Pressure Drop in the Production Pipe", through the well model presented in the first phase. The union of the pressure gradients is modeled as a "Node at pumping of the bottom well", as it was previously explained.

The restrictions are contextualized in the operational scenarios and the reservoir conditions. We assume that: ρ, g are constant, due to the slow dynamics of the reservoir. From the well model, we establish the maximum production capacity that a reservoir can contribute, Qmax. Below, we present the objective function with its respective restrictions:

$$\eta(\%) = \frac{\rho g Q H}{\text{Potency power for the pump * 1,000 W/KW}} \quad (5)$$

Where:

$$H_{min} < H < H_{max} \rightarrow 10.000(\text{ft}) < H < 14.000 (\text{ft}).$$

$$\text{Potency}_{min} < \text{Potency} < \text{Potency}_{max} \rightarrow 20(\text{Hp}) < \text{Potency} < 40(\text{Hp}).$$

$$Q_{prod\ min} < Q_{prod} < Q_{prod\ max} \rightarrow 6000(\text{BPD}) < Q_{prod} < 8000(\text{BPD}).$$

The intervals regarding the restrictions in, Equations 4 and 5, depend on the identified operational scenario, which are characterized in the Equation 6. These ranges will be used by the optimization technique. In order to solve that problem, a genetic algorithm is used, which presents the following components.

Structure of individuals: is coded in real numbers composed of two fields, H and Hp, Equation 6. These variables are used, because they are related to the bottom behavior and surface level production, and they can be manipulated at an operational level with a field instrumentation arrangement. This is important, can be adjusted in terms of the optimum values recommended by the genetic algorithm, and thus achieve the best performances of the producing well.

Number of individuals: random between 2 and 10. Number of generations: 99, Objective function: Equations 4 and 5, including its respective restrictions. Crossover operator: single point cross with 0.7 probabilities. Mutation operator 30%: random with 100 probabilities. Space for search: a population of individuals was gathered with the set of values allowed to variables H and Hp, according to the operational scenario identified in the restrictions of the Equation 6.

That means, the population of individuals will be specific to the operational scenario identified in the previous phase, so that the genetic algorithm may establish the optimum value of the Equation 5 for that operational scenario.

IV. RESULTS

A. First Phase: Generation of the Well Production Model

This section presents the first result, which describes the effectiveness of the tool developed to determine the performance curve of an electric submersible pump (flow vs. efficiency) [6], that will generate the polynomial of the pump under study, following the same behavior of the Curve of the Fabricant, as well as the Curves of the Potency and Height, as a function of the flow rate (see Figure 3). So, the polynomial characteristic of the pump corresponding to the study is (see Equation 6)

$$H_0(q) = 9.3072e^{-11}q^5 + 4.7302q^4 + 0.0005q^3 - 0.004q^2 + 0.150q - 0.054 \quad (6)$$

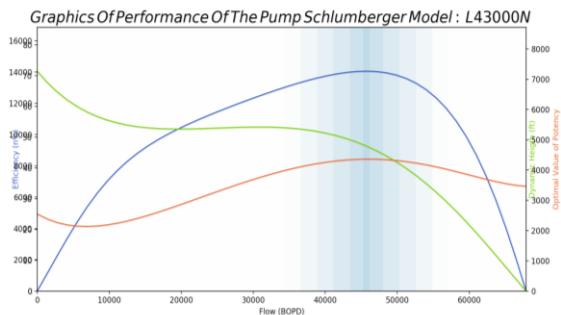


Figure 3. Experimental Curve of the Performance of the Electric Submersible Pump

B. Second Phase: Optimization of the Production Process

This section presents initially an inference of the operational parameters of the bottom (Pbh), head (Pwh) pressures and of the production rate (Q), as a function of the measured variables with instrumentation of field (frequency, f) and pressure in the (production line, Pts). Thus, f and Pt, are real conditions of the well to study object, taking a history of 6900 instances, evaluated in the system developed based on Adams - Bashfort of fourth order predictor and Adams - Moulton corrector, with initial values obtained with the Runge - Kutta method of the fourth order [8].

Thus, the Figure 4 represent the behavior of the classic systems of an artificial lift system. So, for example when the bottom pressure decreases (Pbh) the rate of production (Q) must increase, likewise when the bottom pressure (Pbh) increases there is a significant drop in production.

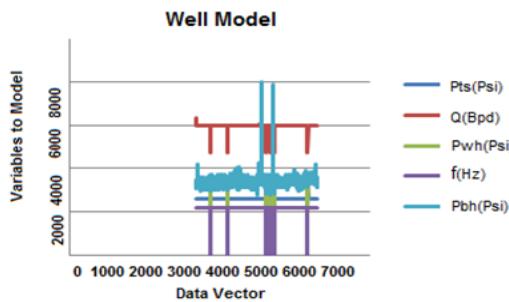


Figure 4. Modeling the Well to Study Object

Following, the Genetic Algorithm was applied for one of the operational scenarios identified in the first phase (see Equation 6). The final population given by the Genetic

Algorithm for that operational scenario is shown in Equation 5, which objective is the value of $\eta(\%)$. According to the results of the Figures 3 and 4, the production system presents an optimum behavior at a height 5000 to 6000 ft, Potency 35 to 45 Hp with an associated production of 7000 to 8000 BPD. So, according to Figure 3, the maximum production of the well is approximately 7500 BPD, with an efficiency level of the order of 70%. Thus, the genetic algorithm obtains a maximum efficiency value of 67%, in 55 generations, the Figure 5, which represents a production values of 7700 BPD (see Figure 6).

The second experiment is performed. Thus, the genetic algorithm obtains a maximum efficiency value of 68%, in 88 generations, Figure 7, which represents a production value of 3500 BPD. This result represents the effectiveness of the implemented algorithms (see Figure 8).

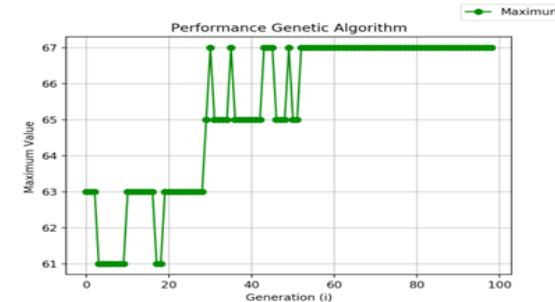


Figure 5. Performance Genetic Algorithm, First Experiment

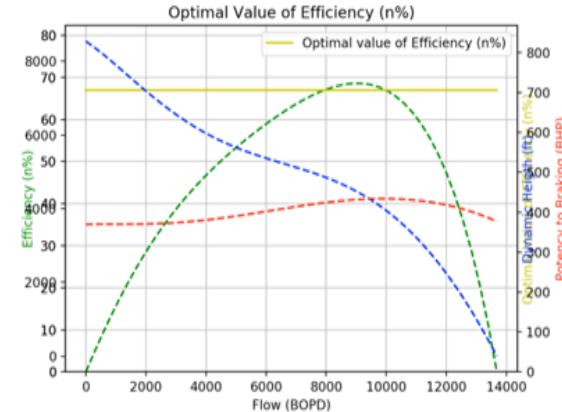


Figure 6. Modeling the Well to Study Object with Genetic Algorithm, First Experiment

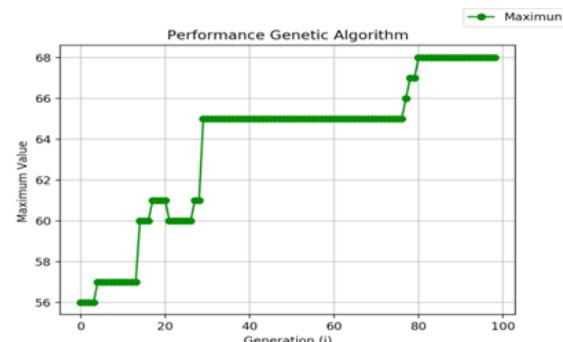


Figure 7. Performance Genetic Algorithm, Second Experiments

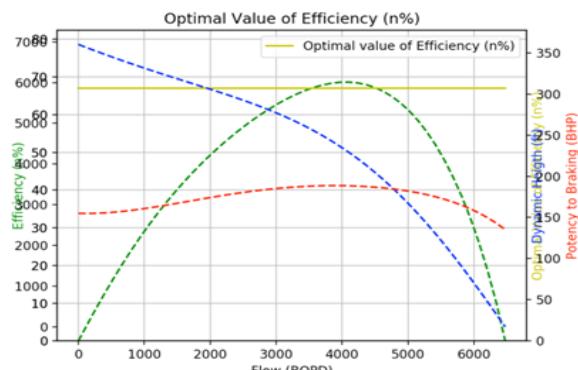


Figure 8. Modeling The Well to Study Object with Genetic Algorithm, Second Experiments

V. CONCLUSION

The production model obtained by using the characterization of the ESP Process using Nodal Analysis allows the prediction of the production rate that the well can produce. These models show the behavior from the “inflow” variables of the reservoir and surface pipe flow for any fluid. In our case, the Intelligent System allows obtaining similar results at reservoir and wellhead levels, in field operational.

The production of the ESP method was optimized in terms of the integrated subsoil and surface information, which will allow guaranteeing the best distribution of the energy in maximizing the production of oil. The subsoil-surface integrated approach is innovative in the sense that it integrates the reservoir/wellhead infrastructure behavior. This is done through an objective function, with the respective restrictions of the process, which allows contextualizing such objective function in the operational scenario and the reservoir conditions identified in the supervision scheme. The genetic algorithm establishes the optimum production and efficiency value for the identified operational scenario from the relationship of the two productive processes: reduce the energy and optimize the efficiency and maximize of productions.

Finally, our multiobjective optimization model system must be proved using other method of lift, compared with other intelligent techniques [9], [12], [13] and parallelized [11], in

order to analyze and improve its system performance (at level of the results quality and execution time).

REFERENCES

- [1] A. Al-Jasmi, K. Goel, G. Moricca, and A. Carvajal. ESP Smart Flow Integrates Qualiy and Control Data for Diagnostics and Optimization in Real Time, SPE 163809. 2013.
- [2] C. Bravo, J. Aguilar, J. Aguilar-Martin, F. Rivas, A. Rios, and L. Saputelli. An implementation of a Distributed Artificial Intelligence Architecture to the Integrated Production Management, Journal of Natural Gas Science & Engineering, Vol. 3, No. 6, pp. 735-747, 2011.
- [3] R. Leal, J. Aguilar, L. Trave-Massuyes, A. Rios, and E. Camargo. Genetic Algorithm Approach for Diagnosability Analysis, International Journal of Engineering Development and Research, Vol. 2, No. 4, pp. 3786-3799, 2014.
- [4] E. Camargo, J. Aguilar, A. Ríos, F. Rivas, and J. Aguilar-Martin. Optimization Model based on Genetic Algorithms for Oil Wells, Proceeding 9th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, pp. 173-178, 2010.
- [5] E. Camargo, J. Aguilar, A. Ríos, F. Rivas, and J. Aguilar-Martin. Nodal Analysis-Based Design for Improving Gas Lift Wells Production, WSEAS Transactions on Informations Science & Applications, Vol. 5, No. 5, pp. 706-715, 2008.
- [6] E. Camargo, J. Aguilar, E. Aceros, and H. Lozada, Implantación de un Sensor Virtual en Procesos Operacionales, Sociedad Chilena Mecánica Computacional, Vol. 14, No. 1, pp. 68-77, 2017.
- [7] E. Camargo, J. Aguilar, and E. Aceros, Intelligent Well Systems, Proceeding IEEE Asia-Pacific Conference on Computer Aided System Engineering, pp. 13-18, 2015.
- [8] R. Burden and D. Faires. Análisis Numéricos, Séptima Edición, International Thomson Editores, S.A., 2002.
- [9] E. Camargo and J. Aguilar. Advanced Supervision of Oil Wells Based on Soft Computing Techniques, Journal of Artificial Intelligence and Soft Computing Research, Vol. 4, No. 3, pp. 215-225, 2014.
- [10] E. Camargo, J. Aguilar, A. Ríos, and F. Rivas. Un Modelo de Producción de Pozos por Levantamiento Artificial Utilizando Análisis Nodal. Revista Ciencia e Ingeniería, Vol. 30, No.1, pp. 23-28. 2009.
- [11] F. Hidrobo and J. Aguilar. Toward a Parallel Genetic Algorithm Approach Based on Collective Intelligence for Combinatorial Optimization Problems, Proceeding IEEE International Conference on Evolutionary Computation, pp. 715-720, 1998.
- [12] J. Aguilar. A General Ant Colony Model to Solve Combinatorial Optimization Problems. Revista Colombiana de Computación. Vol. 2, No. 1, pp. 7-18, 2001.
- [13] J. Aguilar. Definition of an Energy Function for the Random Neural to Solve Optimization Problems, Neural Networks, Vol. 11, No. 4, pp. 731-738, 1998.

Diseño del Modelado Matemático para el Diagnóstico de Pozos de Extracción por Bombeo de Cavidades Progresivas

Edgar Camargo
PDVSA, Distrito Socialista Tecnológico -AIT
Mérida, 5101, Venezuela
camargoea@gmail.com

Egner Aceros
Facultad de Ingeniería, División de Postgrado
Universidad del Zulia Maracaibo, 4001, Venezuela
egner.aceros@cantv.net

Lice Guzmán
PDVSA, Distrito Socialista Tecnológico -AIT
Mérida, 5101, Venezuela
guzmanlp@pdvsa.com

Mariuska Briceño
PDVSA, Distrito Socialista Tecnológico -AIT
Mérida, 5101, Venezuela
bricenomh@pdvsa.com

Resumen— Con el objetivo de hacer más eficientes los procesos involucrados en la optimización de sistemas de producción de hidrocarburos, la industria petrolera nacional se ha apoyado en diversas herramientas, especialmente en aquellas que proveen soluciones integrales e innovadoras. En este artículo se presenta el diseño de un sensor virtual híbrido, para el diagnóstico de un pozo con Bombeo de Cavidades Progresivas aplicando inteligencia artificial, a través de modelado difuso y del modelo matemático obtenido para la inferencia de parámetros operacionales. Los resultados demuestran el buen desempeño y la exactitud en la predicción de parámetros críticos como la tasa de producción del crudo y la carga axial.

Palabras Clave: *Bombeo por Cavidades Progresivas, Lógica difusa, Sensor Virtual, Estado operacional, Diagnóstico.*

I. INTRODUCCIÓN

En la producción de hidrocarburos el objetivo fundamental es maximizar el valor de un activo (petróleo) que decrece en forma permanente, al menor costo posible, es por ello que los métodos de levantamiento artificial se utilizan en un alto porcentaje en todo el mundo, con el propósito de mantener y/o extender la fase de extracción de crudo en los pozos de producción. Así, basados en metas de producción, los expertos en el área se han visto en la necesidad de aplicar técnicas y métodos de recuperación secundaria de crudo, entre los cuales destaca el Bombeo de Cavidades Progresivas (BCP) como sistema de levantamiento artificial.

Para una toma de decisiones acertada y pertinente del desempeño del sistema, se requiere de numerosa información vital: curva de declinación, balance de materia, curva de comportamiento del pozo, simulaciones y modelos del yacimiento; todo esto en tiempo real o casi real. El uso de técnicas eficientes y confiables del filtrado de la información (minería de datos, inteligencia artificial, etc) y la asimilación de estos datos es indispensable para manejar la gran cantidad de información que se genera a partir del monitoreo continuo a

tiempo real, con integración efectiva de sensores remotos para actualizar los datos del reservorio y con ello optimizar el proceso, a través de un seguimiento adecuado y eficiente de las condiciones del pozo, fases y tasas de flujo, presión del reservorio y monitoreo de la arenas, etc.

Es importante considerar que, de los datos obtenidos, depende la calidad de las decisiones tomadas. En ese sentido, se requiere la implementación de herramientas tecnológicas adaptadas al contexto de la industria petrolera venezolana especialmente debido al esquema actual de explotación de reservas no convencionales de crudos pesados y extra-pesados

En este artículo se presenta el diseño y la implantación de una herramienta computacional para tarea de predicción, denominada Sensor Híbrido, el cual permite la estimación de parámetros operacionales en procesos petroleros, combinando las ventajas de los sistemas difusos con modelos matemáticos para la estimación de variables ([1],[2]). Dicho sistema usa datos operacionales, a los cuales se les realiza un tratamiento previo para limpiarlos, filtrarlos, analizar la correlación entre ellos, entre otras cosas, para que los resultados obtenidos sean satisfactorios. La validación del modelo del sensor híbrido se realiza a través de la comparación con el escenario real objeto de estudio. Específicamente, se propone la predicción de carga axial de una bomba BCP utilizando modelos matemáticos [2] y la predicción de la tasa de producción utilizando una técnica difusa jerárquica, debido a la importancia operacional de estas variables apoyando el diagnóstico de problemas y el análisis para su resolución.

II. FUNDAMENTOS TEÓRICOS

A. *Pozos por Levantamiento Artificial por Bombeo Cavidad Progresiva*

Se utilizan técnicas de levantamiento artificial cuando los yacimientos no cuentan con suficiente energía como para

producir petróleo o gas en forma natural para conducirlos a la superficie, o cuando las tasas de producción no cumplen con los objetivos deseados. El sistema BCP es una tecnología que ha demostrado ser la más eficiente en levantamiento artificial para la producción de petróleos con elevada viscosidad y en pozos de difícil operación (alta temperatura, presencia de gas y porcentajes de agua elevados), ya que los componentes de este sistema son especialmente diseñados para trabajar eficazmente en condiciones extremas. Una bomba de cavidad progresiva (BCP) consiste en una máquina rotativa de desplazamiento positivo, la cual consiste en un engranaje helicoidal enroscado extremo simple (rotor), que rota excéntricamente dentro de un engranaje helicoidal enroscado (estator). El principio de funcionamiento es por transmisión de movimiento rotacional a través del motor hacia una sarta de cabillas por medio de distintos engranajes, esta sarta de cabillas hace girar al rotor, formando cavidades progresivas ascendentes y el crudo se desplaza hasta la superficie por efecto del rotor que gira dentro del estator fijo. El estator es el componente estático de la bomba y contiene un polímero de alto peso molecular con la capacidad de deformación y recuperación elástica llamado elastómero [3].

Los componentes del sistema de BCP se dividen en dos conjuntos, los equipos de superficie y el equipo de subsuelo. A nivel de superficie, se dispone de un motor eléctrico, un cabezal que dispone de un sistema de freno, un sistema de poleas y correas, y un stuffing box o prensaestopas, que sirve para evitar la filtración de fluidos a través de las conexiones de superficie. El sistema de transmisión, transfiere la energía desde el motor hasta el cabezal de rotación. La barra lisa o pulida va ubicada al final de la sarta de cabillas y su principal función es transmitir el movimiento de rotación del cabezal a la sarta de cabillas y el rotor y la Te de producción, es un dispositivo necesario en las instalaciones de bombas de cavidades progresivas, para dirigir el fluido de producción del tubing a la línea de producción de superficie (ver Figura 1)

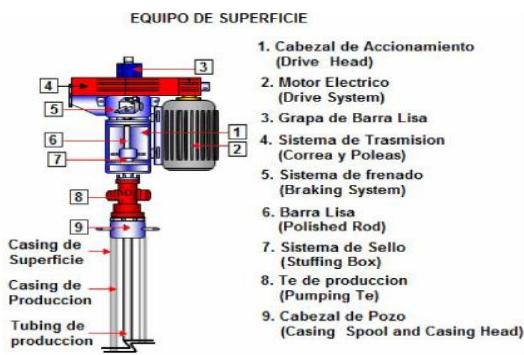


Figura 1. Partes del Equipo de Superficie del Sistema BCP

El equipo de subsuelo está conformado por la bomba, sus accesorios y la sarta de cabillas, esta última es la encargada de transferir la energía rotacional al rotor de la bomba y es la parte del sistema que soporta las fuerzas torsionales y axiales del

movimiento rotacional del motor y la columna de crudo que es desplazada hasta la superficie. La sarta de cabillas seleccionada debe estar en capacidad de soportar una cierta carga axial (peso aparente de la sarta y peso de la altura hidráulica) y de transmitir un torque determinado desde la superficie hasta el fondo del sistema, compuesto por la fuerza para levantar el crudo y los esfuerzos de fricción por el movimiento giratorio de las cabillas. También posee un pin de paro que sirve como punto de referencia para el espaciamiento del rotor dentro del estotor durante la instalación del sistema.

El estotor también constituye parte de los equipos de subsuelo y son tubos de acero con cuerpos de elastómeros pegados internamente. Debido al movimiento rotatorio del rotor, se corre un riesgo potencial de desenroscar la tubería de producción en cualquier punto susceptible de la misma, sobre todo en aquellas aplicaciones donde eventualmente se producen incrementos de torque, por lo que se utilizan anclas; y finalmente la tubería de producción, es el equipo de subsuelo formado por un conjunto de tubos que transportan el fluido desde el fondo del pozo hasta la superficie y al mismo tiempo sirve de guía a la sarta de cabillas (ver Figura 2)

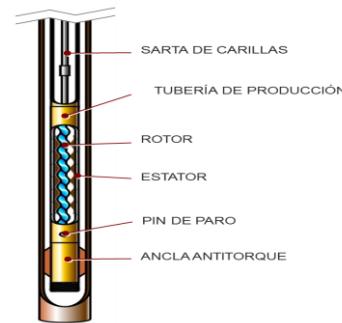


Figura 2. Partes del Equipo de Subsuelo del Sistema BCP

B. Variables Operacionales del Sistema BCP

Los hidrocarburos a través de las arenas productoras en el yacimiento, dependen de la presión de burbuja (P_b) y las condiciones de temperatura en fondo (T_{wf}), esta presión varía con el tiempo y va disminuyendo a lo largo del tiempo y es aproximadamente igual la presión de entrada a la bomba (P_{IP}). Es de suma importancia contar con mediciones de la presión de fondo ya que la bomba BCP requiere de una presión de entrada y de descarga, mediante las cuales se estima su eficiencia en cuanto a producción. La Presión de descarga de la bomba (P_{DP}), es la presión a la salida de la bomba y es generada por la acción de la bomba y el nivel de sumergencia (NS), esta presión se ve reflejada en la presión de superficie en cabezal y es equivalente a la presión de cabezal (THP). El nivel de sumergencia (NS), es el nivel de la columna de fluido en la tubería de revestimiento (Casing) con respecto a la superficie, este nivel es determinante para el funcionamiento de la bomba BCP, ya que a mayor nivel de sumergencia, mayor será la presión de entrada y mayor deberá ser el trabajo de la bomba para elevar la columna de fluido, por esto es necesario que el nivel de sumergencia permanezca en valores bajos, pero que sea lo suficiente para que la bomba no se achique o pierda

fluído de entrada. El torque, es la fuerza rotacional aplicada a la sarta de cabillas por el cabezal y transferida al rotor de la bomba BCP, la cual es inferida por el variador de frecuencia en superficie, y es de suma importancia para el diagnóstico del estado operacional de la bomba. Finalmente, la carga axial es la fuerza que actúa a lo largo del eje longitudinal de un miembro estructural, aplicada al centroide de la sección transversal del mismo produciendo un esfuerzo uniforme, esta fuerza en el sistema BCP representa la carga soportada por la sarta de cabillas, que incluye la fuerza del peso de la sarta y la columna de fluido en la tubería de producción. A través del modelo matemático que representa esta fuerza se infiere la carga axial y a partir de la misma se estima la producción del sistema BCP [3].

C. Sistemas Difusos

La Lógica Difusa (LD) constituye una técnica de gran importancia en el modelado de sistemas, pues los modelos lógicos - difusos pueden ser usados para describir fenómenos complejos difíciles de modelar mediante métodos formales [2]. Entre las ventajas de la lógica difusa esta la posibilidad de hacer un razonamiento o procesamiento con información imprecisa o borrosa. Un modelo lógico-difuso (sistema difuso) consta fundamentalmente de una base de conocimientos, conformada por un conjunto de reglas del tipo Si-Entonces con proposiciones imprecisas expresadas en lenguaje natural, y de un mecanismo de inferencia que permite obtener una conclusión a partir de una entrada determinada. Las reglas permiten establecer las relaciones entre las diferentes variables, llamadas entradas, para generar ciertas conclusiones ([4],[5]). El conocimiento codificado en la base de reglas se deriva de la experiencia y de la intuición humana, además del conocimiento práctico y teórico que se disponga del sistema. Cada una de las reglas de la base de conocimientos esta caracterizada por una parte “Si” llamada el antecedente y una parte “Entonces” llamada el consecuente. El antecedente de una regla contiene un conjunto de condiciones y el consecuente contiene una conclusión. Cada regla opera de la siguiente forma: si las condiciones del antecedente son satisfechas, entonces la conclusión del consecuente es aplicada. Para un modelo difuso las variables incluidas en el antecedente corresponden a las entradas del sistema y las variables incluidas en el consecuente corresponden a las salidas del sistema [5]. Las condiciones contenidas en el antecedente y las conclusiones del consecuente de las reglas consisten en proposiciones de la forma F es X_0 , donde F es una variable y X_0 es un conjunto difuso definido sobre el universo de discurso de la variable F .

Para desarrollar un clasificador difuso basta en muchos de los casos utilizar alguna de las funciones de similitud ya establecidas, tales como la función triangular o la trapezoidal, es decir, no es necesario concebir una nueva función de pertenencia difusa, el esfuerzo se enfoca entonces en conocer los datos de entrada y con base en la experiencia que se tenga, saber cuál es la más apropiada para modelar y clasificar (ver Figura 3)

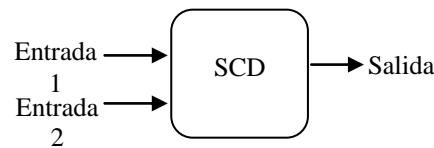


Figura 3. Modelo de un SCDM Clásico

Por otra parte, un Sistema Difuso Multicapa (SDM) consiste en un número de sistemas difusos distribuidos de forma jerárquica, los cuales poseen la ventaja de que el número total de reglas de su base de conocimiento es menor, y son más simples, que un sistema difuso convencional. El tipo más tradicional de SDM es aquel en el que cada módulo es un sistema difuso (SD) completo que relaciona a un conjunto reducido de variables, que pueden ser variables de entrada del sistema global o variables internas generadas como salidas de otros módulos [5]. Existen otros enfoques como el que consiste en identificar partes comunes del conjunto de reglas y definir módulos para esas partes comunes [5]. Nuestro trabajo usa el primer enfoque, que se puede ver en la (ver Figura 4)

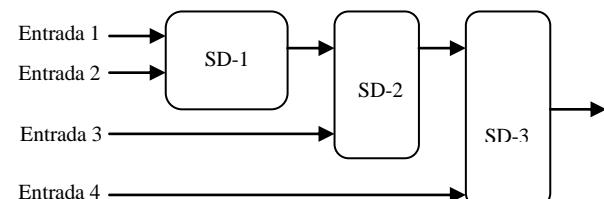


Figura 4. Modelo de un SCDM Jerárquico

III. DISEÑO DEL SISTEMA HÍBRIDO

El diseño del sistema híbrido está compuesto por varias fases: 1) una fase (componente) de generación del modelo del pozo, en la cual se proponen utilizar técnicas de balance de masa y energía; 2) una fase (componente) de identificación del escenario operacional del pozo, empleando lógica difusa; y 3) una fase (componente) de optimización del proceso productivo para el escenario identificado, proponiendo el uso de técnicas evolutivas basadas en restricciones del proceso y el costo operacional de producción [3]. Además, tiene un componente de instrumentación que le permite capturar las variables del sistema, en este caso en particular las variables de inyección y producción.

A. Fase Modelado de la Carga Axial Sistema BCP

Una de las variables que describe el funcionamiento y estado operacional del sistema BCP es la carga axial. Esta variable puede ser medida con una celda de carga instalada en el cabezal del sistema BCP o puede ser inferida a través de un modelo matemático, a partir de las mediciones de variables con sensores existentes y los datos de diseño del sistema de levantamiento por BCP. El modelo matemático de la Ecuación 1 para la inferencia de la carga axial parte de la definición de fuerza, la ecuación que define el nivel de sumergencia y el diferencial de presión de la bomba, más la aplicación de un

diagrama de cuerpo libre en la sarta de cabillas, obteniéndose un sistema de ecuaciones, que por medio de sustitución y despeje permite obtener el modelo matemático que describe el comportamiento de la carga axial a partir de la medición de dos presiones a nivel de superficie, en este caso la presión de casing y presión de cabezal.

$$Ca = \frac{Ab * Ac}{Ab - Ac} * \left(\frac{wr * (0,88) * Gesp * L + Pc * Ar}{Ac} + THP \right) \quad (1)$$

Donde:

Ca: Carga axial

Ab: Área de la bomba

Ac: Área transversal del casing

wr: Densidad de peso de las cabillas

Gesp: Gravedad específica del crudo

L: Longitud de la sarta de cabillas

Pc: Presión del casing

Ar: Área transversal del rotor

THP: Presión del cabezal equivalente a la presión de descarga de la bomba

El modelo matemático de la carga axial fue programado y probado en Python, a partir de los datos históricos de la presión de casing y presión de cabezal medidas en campo, se comprobó que el modelo sigue el comportamiento físico real. Luego de comprobar el funcionamiento del sensor virtual de carga axial con los datos históricos, se determinó el caudal de producción estimado, a partir de la curva de afluencia del pozo (IPR), de la Ecuación 2 la cual muestra el aporte del yacimiento en barriles, según la presión de fondo. La curva IPR se establece a través de métodos matemáticos definidos de acuerdo a las características del yacimiento, en específico a la presión de burbuja (Pb), que estipula un punto de referencia en la elección del método a aplicar para la determinación de la IP, al ser comparado con la presión de fondo fluyente del pozo (Pwf). Existen tres métodos, Vogel, Darcy y lineal. El método de Vogel es el más aplicado ya que describe el aporte del pozo para presiones de fondo menores y mayores a la presión de burbuja, por lo que se considera un método estándar. El modelo matemático de Vogel para obtener la curva IP se muestra a continuación.

$$IP = \frac{Q}{Pwf - Pb + \left(\frac{Pb}{1,8} \right) * \left(1 - 0,2 * \left(\frac{Pwf}{Pb} \right) - 0,8 * \left(\frac{Pwf}{Pb} \right)^2 \right)} \quad (2)$$

Donde:

IP: Índice de Productividad

Q: Caudal de Producción

Pwf: Presión de fondo fluyente

Pb: Presión de Burbuja

B. Identificación de Escenarios Operacionales

Para el desarrollo del diseño del modelado matemático y la herramienta de diagnóstico se aplicó un sistema difuso jerárquico. Luego se determinaron los parámetros a emplear en las funciones de membresía, para aplicar lógica difusa en la estimación de la producción del pozo según la carga axial calculada y la presión de entrada a la bomba, lo cual conforma el primer bloque del desarrollo de la herramienta de modelado mostrado en la (ver Figura 5)

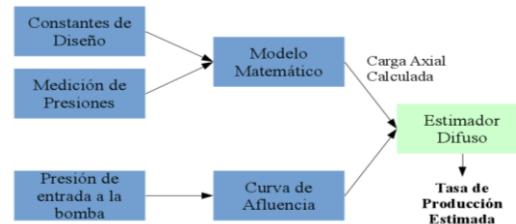


Figura 5. Primer Bloque Difuso de la Herramienta de Modelado Matemático

Este bloque se desarrolló en Python basados en los parámetros de los rangos para las funciones de membresía se procedió al diseño del estimador difuso y la inclusión de las reglas difusas, tipo Mandami, las cuales se muestran en la Figura 6. Con los datos de la carga axial estimada con el modelo matemático y la curva IPR obtenida, se estima la tasa de producción.

1. If (Caxial is bajo) and (IPR is bajo) then (Produccion is bajo) (1)
2. If (Caxial is bajo) and (IPR is medio) then (Produccion is medio) (1)
3. If (Caxial is bajo) and (IPR is alto) then (Produccion is bajo) (1)
4. If (Caxial is medio) and (IPR is bajo) then (Produccion is alto) (1)
5. If (Caxial is medio) and (IPR is medio) then (Produccion is medio) (1)
6. If (Caxial is medio) and (IPR is alto) then (Produccion is medio) (1)
7. If (Caxial is alto) and (IPR is bajo) then (Produccion is alto) (1)
8. If (Caxial is alto) and (IPR is medio) then (Produccion is alto) (1)
9. If (Caxial is alto) and (IPR is alto) then (Produccion is medio) (1)

Figura 6. Reglas Difusas del Primer Bloque de la Herramienta de Modelado en MATLAB

Una vez completado este bloque se procedió al desarrollo de la segunda etapa del sistema difuso jerárquico, con la que se estimó el estado operacional del sistema BCP (ver Figura 7)

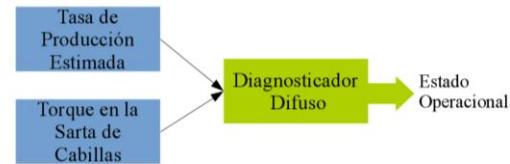


Figura 7. Esquema del Segundo Bloque del Sistema Difuso Jerárquico

Las variables de entradas de este segundo bloque son la tasa de producción estimada en el primer bloque del sistema difuso jerárquico, y el torque en la sarta de cabillas del sistema BCP. Siendo el torque una variable crucial a la hora de diagnosticar el estado operacional de la bomba, ya que las variaciones en el torque representan fallas tales como: rotura de la sarta de cabillas, fallas en el elastómero de la bomba, desgaste en la bomba, cambios en las características del flujo, errores en el diseño del sistema, entre otros. Por esto se toma como variable

junto con la tasa de producción estimada del pozo para determinar el estado operacional del sistema BCP. El estado operacional fue considerado en un rango de 0% a 100%; siendo 0% correspondiente a operatividad baja o sistema fuera de su rango operacional, y 100% el funcionamiento ideal y más eficiente del sistema BCP. Las reglas difusas aplicadas a este bloque se muestran en la (ver Figura 8)

1. If (QEstimada is bajo) and (Torque is bajo) then (EdoOperacional is bajo) (1)
2. If (QEstimada is bajo) and (Torque is medio) then (EdoOperacional is medio) (1)
3. If (QEstimada is bajo) and (Torque is alto) then (EdoOperacional is bajo) (1)
4. If (QEstimada is medio) and (Torque is bajo) then (EdoOperacional is alto) (1)
5. If (QEstimada is medio) and (Torque is medio) then (EdoOperacional is medio) (1)
6. If (QEstimada is medio) and (Torque is alto) then (EdoOperacional is medio) (1)
7. If (QEstimada is alto) and (Torque is bajo) then (EdoOperacional is alto) (1)
8. If (QEstimada is alto) and (Torque is medio) then (EdoOperacional is alto) (1)
9. If (QEstimada is alto) and (Torque is alto) then (EdoOperacional is medio) (1)

Figura 8. Reglas del Segundo Bloque del Sistema Difuso Jerárquico

C. Características del Pozo a Estudiar

Para el diseño del modelado matemático se identificaron datos del pozo y sus características en la Tabla I.

TABLA I. DATOS DE DISEÑO DEL POZO

Área de la bomba (ft ²)	3.5
Área de las cabillas (ft ²)	1.125
Densidad de peso de las cabillas (lb/ft)	5.8
Longitud de la sarta de cabillas(ft)	5430
Área efectiva del rotor (ft ²)	2.21
Radio Casing (ft)	0.557
Radio Tubing (ft)	0.1875

Las propiedades del fluido que la bomba BCP maneja se muestran en la Tabla II.

TABLA II. DATOS DE YACIMIENTO DEL POZO

Gravedad específica del crudo	0.43
Presión de reservorio (psi)	1600
Temperatura de reservorio (°F)	158
Factor Volumétrico del crudo	10.799
Viscosidad del crudo (cp)	9000
Producción por día promedio (BNPD)	384.82

IV. RESULTADOS

En esta sección se muestra una posible implementación del sensor virtual híbrido (Svh) para pozos BCP, así como los resultados de las pruebas realizadas para la validación del mismo, usando datos provenientes de un pozo de la industria petrolera venezolana. El propósito de este prototipo es la predicción de las variables Carga Axial y Producción . En el caso de Carga Axial se usa la ecuación 1 de autoría propia. Esta herramienta fue implantada en un controlador lógico

programable, usando el Lenguaje Python a nivel del cabezal del pozo. Dicha herramienta permite al operador estudiar ciertos escenarios, para poder tomar previsiones cuando en algunos de ellos ocurran fallas, y minimizar de esta manera sus efectos en las caídas de producción [2].

Los resultados de las simulaciones en MATLAB, a partir de los datos históricos de las variables operacionales, y la estimación de la carga axial determinada por el programa desarrollado en Python se muestran en las figuras a continuación. Donde se puede observar que en la Figura 9 la carga axial sigue el comportamiento físico del sistema con respecto a la presión de cabezal, debido que la presión del casing es constante.

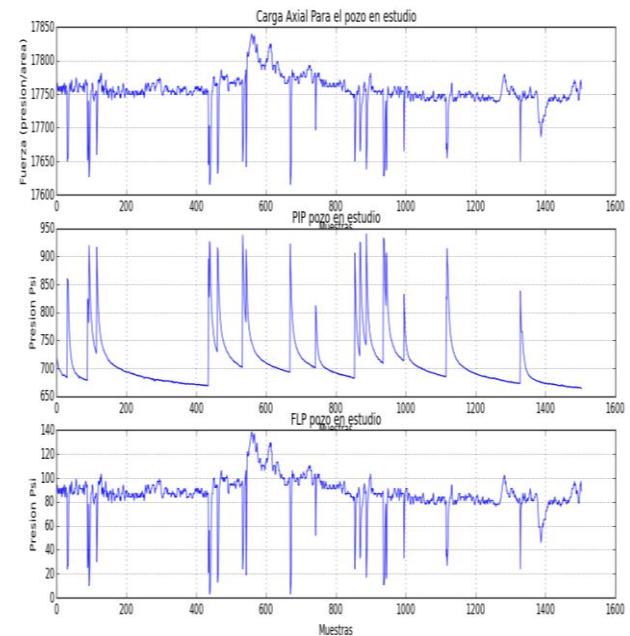


Figure 9. Carga Axial Estimada, PIP y FLP (equivalente a THP), Extraídas del Programa en Python

Por otra parte la tendencia que puede observarse en la Figura 10 de la producción estimada con respecto a la producción calculada a través de la curva de IPR es similar en ciertos puntos, ya que al existir baja de presión de fondo teóricamente debería subir la producción, pero este estado no lo garantiza, siempre y cuando la carga axial se mantenga en un rango alto y una presión de fondo baja se puede asegurar que hay una producción elevada.

Además se observa en la Figura 11 el estado operacional del sistema BCP en donde se perciben oscilaciones en el sistema debido a la falta de optimización del mismo, en donde no hubo un control eficiente en el variador de frecuencia del motor del sistema BCP, la toma de decisiones claves a partir del censado de presiones de fondo y cabezal, así como la no existencia de la estimación de la carga axial y el control inteligente del pozo a través de esta variable tan importante.

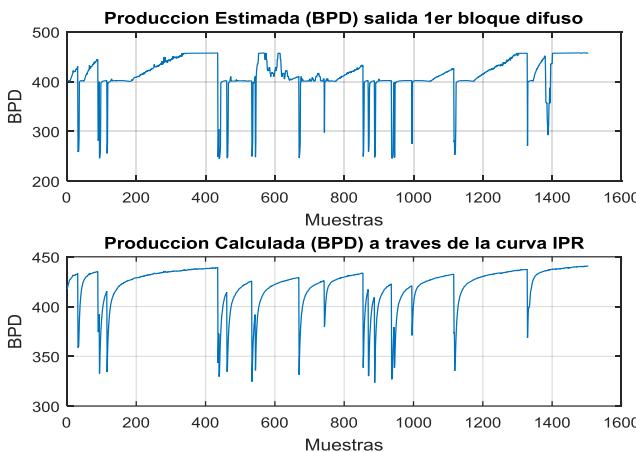


Figura 10. Producción Estimada y Producción Calculada por IPR

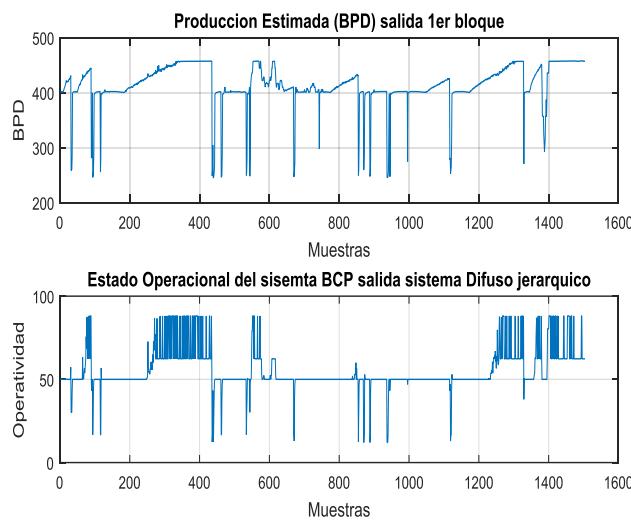


Figura 11. Salida del Primer Bloque Difuso y del Sistema Difuso Jerárquico

Se puede observar que la producción BPD(Bariles de Producción Diaria), la estimada difiere con la producción calculada ya que la producción estimada toma en cuenta la carga axial, la cual representa la columna de crudo que desplaza el sistema BCP hacia la superficie, por lo tanto cuando existe baja carga axial la producción también cae, así el yacimiento aporte en fondo de pozo. Por otra parte también se puede observar el estado operacional del sistema BCP en una escala de 0% a 100%, representando la operatividad del sistema BCP, tomando en cuenta la producción estimada y el torque en la sarta de cabillas, se calculó que el sistema estuvo, en promedio, a un 55,41% de su operatividad en el lapso de tiempo en el que fueron tomados los datos de las variables medidas. También se observa una oscilación debido a la falta de control a nivel del variador de frecuencia para controlar el torque en las cabillas en función de la producción estimada, por esto se hace necesario la implementación de un sistema de control inteligente que mantenga la operatividad del sistema en niveles aceptables sin llegar a provocar daños en el sistema BCP. Como se puede apreciar en la Figura 11 de la salida del

sistema difuso jerárquico, en cuanto al estado operacional del sistema BCP, se observa que el sistema mantuvo una operatividad oscilante, esta operatividad va enfocada al desempeño del sistema en cuanto a producción y estado de la bomba en fondo de pozo, este estado se ve reflejado directamente en el torque.

V. CONCLUSION

El uso de sistemas híbridos para la estimación de variables es de gran interés por el bajo costo requerido para su implementación, y además, por la gran disponibilidad actual de sistemas de adquisición de datos y bases de datos históricas, que aportan los datos requeridos para el diseño de estos sistemas.

El sistema de modelado matemático desarrollado para el diagnóstico de pozos de extracción por bombeo de cavidades progresivas, estima la carga axial en la sarta de cabillas como un sensor virtual, y también infiere producción de crudo del sistema y el estado operacional del sistema BCP. Empleando un sistema difuso jerárquico, a partir de las variables medidas en superficie, tales como presión del casing, presión de cabezal, y además con la presión de fondo fluyente y el cálculo del torque.

La aplicación de lógica difusa en la inferencia de variables operacionales tales como producción y el estado operacional del sistema BCP brinda un soporte esencial a la hora de tomar acciones de optimización y control del sistema así como también proporciona una estimación para la realización del diagnóstico de fallas en el sistema a través del conocimiento del personal de campo.

La herramienta inteligente desarrollada tiene una gran aplicabilidad operacional y permite obtener resultados satisfactorios y precisos para el diagnóstico del estado operacional de un sistema BCP a partir de las variables inferidas y medidas en campo.

REFERENCIAS

- [1] R. Ronning, "Automatic Start-up Control of Artificially Lifted Wells" Master of Science in Engineering Cybernetics. Norwegian University of Science and Technology, 2011.
- [2] E. Camargo and J. Aguilar, "Intelligent Well Systems", 2015 Asia-Pacific Conference on Computer Aided System Engineering, pp. 13-18, 2015.
- [3] J. Chirinos, Método de Diagnóstico de Fallas en el Sistema de Levantamiento Artificial por Bombeo de Cavidad Progresiva. Revista arbitrada venezolana del Núcleo LUZ-Costa Oriental del Lago, Universidad del Zulia, Venezuela 2013.
- [4] E. Camargo and J. Aguilar, "Advanced Supervision of Oil Wells Based on Soft Computing Techniques" Journal of Artificial Intelligence and Soft Computing Research, Vol. 4, No. 3, pp.215-225, 2014.
- [5] E. Camargo, J. Aguilar, A. Ríos, F. Rivas and J. Aguilar Martín, Diseño de un Sensor Virtual Aplicado a Procesos de Petróleo. Artículo aceptado en el congreso iberoamericano de ingeniería mecánica, Chile 2016.

Algoritmo Adaptativo para la Estimación por Series de Tiempo Usando Árboles de Decisión

Alfredo José Naime Marcano

Dirección de Automatización, Informática y Telecomunicaciones (AIT)

Petróleos de Venezuela S.A. (PDVSA)

Caracas, Venezuela.

naimea@pdvsa.com

Resumen—Los métodos de series de tiempo tradicionales ofrecen modelos cuyos parámetros permanecen constantes en el tiempo. Sin embargo, los procesos industriales de oferta y demanda requieren decisiones oportunas basadas en una realidad dinámica. Un cambio de configuración, apagado o encendido de una línea de producción o proceso, modifica el problema y las variables a predecir. Los sistemas de soporte a las decisiones deben adaptarse dinámicamente si queremos que respondan rápida y apropiadamente a las operaciones y sus procesos. Esta metodología se basa en obtener para cada período, el modelo que mejor se ajusta a los datos, evaluando un gran número de alternativas y usando técnicas de aprendizaje estadístico. De esta forma el modelo se irá adaptando a los datos en la práctica y tomará decisiones basada en la experiencia. Con tres meses de prueba para la estimación de variables asociadas a procesos de oferta y demanda, se ha obtenido predicciones que difieren menos de 8 centésimas ($<0,08$) o 0,1% del valor medido. Esto indica que la ciencia de datos y el aprendizaje estadístico representan una importante área de investigación para la predicción de variables y la optimización de procesos.

Palabras Clave—modelos de series temporales, estimación, pronósticos, análisis de datos, minería de datos, aprendizaje estadístico, árboles de decisión.

I. INTRODUCCIÓN

Dentro del objetivo de describir y predecir el comportamiento de variables y procesos aleatorios, podemos usar un conjunto de herramientas y metodologías estadísticas, las cuales, combinadas con ciencias de la computación y tecnología de punta para computo, almacenamiento y transmisión de datos, convergen en lo que hoy se denomina la Ciencia de Datos, el Aprendizaje Estadístico o la Minería de Datos. Cuando los datos que disponemos están estrechamente vinculados al tiempo y son observaciones realizadas a intervalos regulares, la forma de trabajarlos, estudiarlos y hacer análisis para predecir escenarios futuros, es a través de modelos de series de tiempo.

Una serie temporal o serie de tiempo, es un conjunto de observaciones x_t , registradas en el tiempo t . Específicamente, una serie de tiempo discreta, es una serie de tiempo cuyas observaciones se realizan a intervalos de tiempo discretos y fijos para todas las observaciones [1]. Podemos encontrar este tipo de series en todos los campos de la ciencia [2], incluyendo economía y otras ciencias sociales, bioestadística y, en nuestro caso, fenómenos físicos asociados a procesos industriales de oferta y demanda de recursos, como volumen, presión, temperatura, caudal, entre otros (Figura 1).

Un modelo de series de tiempo se define como una especificación de distribuciones conjuntas (medias y covarianzas) de una secuencia de variables aleatorias $\{X_t\}$ con $\{x_t\}$ observaciones [1].

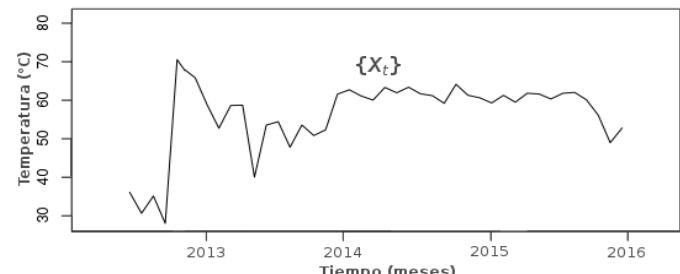


Figura 1. Gráfico de una Serie de Tiempo

Si bien, muchos procesos industriales, y las variables aleatorias asociadas a estos, tienen una fuerte interdependencia, en nuestro caso, analizaremos solamente series univariantes [2] para desarrollar un algoritmo que aplique la metodología de Box y Jenkins [3][4], a un gran y exhaustivo conjunto de modelos posibles para luego seleccionar los que logren mejores resultados.

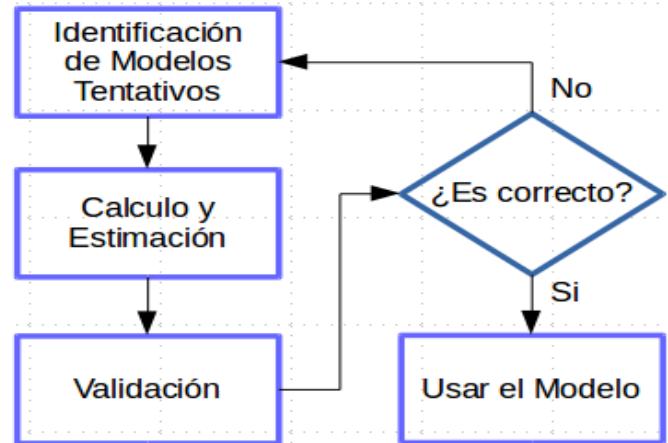


Figura 2. Método de Box-Jenkins

La metodología de Box y Jenkins presenta un conjunto de procedimientos para describir los datos y seleccionar el o los modelos tentativos. Luego, se calculan los coeficientes del modelo, típicamente usando el método de máxima verosimilitud o de mínimos cuadrados. Finalmente, se calculan los residuos del modelo y se analizan para chequear su validez (Figura 2). Sólo los modelos que sean aceptados se usarán para pronosticar [3][4].

Nosotros planteamos algunos cambios para automatizar esta metodología y poder usar el gran poder de cómputo que disponemos actualmente para que una máquina pruebe y seleccione los mejores modelos basado en un conjunto de reglas. Algunas de estas reglas han sido diseñadas usando los métodos tradicionales para validación de modelos de series de tiempo y otras usando técnicas de aprendizaje estadístico con algoritmos de árboles de decisión.

Este método fue diseñado y programado en *R Studio Server* durante el período de octubre 2016 a junio 2017 y está actualmente siendo probado y evaluado en conjunto con las divisiones industriales solicitantes con sus datos operacionales de campo.

En la segunda sección, describiremos el problema a resolver. Seguidamente, describiremos los criterios para establecer las reglas que nos permiten comparar, contrastar, ponderar y seleccionar los mejores modelos. En la cuarta sección describiremos a grandes rasgos el algoritmo diseñado y algunas de las estructuras de datos utilizadas. En la quinta sección, mostraremos algunos de los resultados obtenidos hasta el momento, para después plantear nuestras conclusiones. El documento finaliza con la sección de referencias.

II. PLANTEAMIENTO DEL PROBLEMA

Tradicionalmente, la identificación y selección de los modelos se realiza mediante la descripción de la serie en sus componentes de tendencia y estacionalidad. El científico de datos, calcula y analiza medias, varianzas, funciones de autocovarianzas y autocorrelación simple y parcial [1][2][3][4], para plantear y evaluar un conjunto de transformaciones a la serie. Para determinar los coeficientes de los componentes Autorregresivo (p), grado de diferenciación (d) y medias móviles (q), de un Autorregresivo Integrado de Medias Móviles, *ARIMA* (p,d,q), se analiza la cantidad de rezagos con valores fuera de las bandas en las funciones de autocorrelación simple y parcial, para cada una de las diferenciaciones y transformaciones aplicadas a la serie original y se plantean unos pocos modelos tentativos a evaluar [2][4].

Esto está bien, pero seguramente nos tomará mucho tiempo y posiblemente dejaremos por fuera muchos otros buenos modelos sin considerar, evaluar ni analizar. El análisis predictivo es una tarea que consume tiempo, constancia, dedicación y paciencia. Sin embargo, para los efectos prácticos, el sentido de la oportunidad en la toma de decisiones no es compatible con esta forma de analizar el problema.

Adicionalmente, una vez seleccionados los parámetros (p,d,q) y construido el modelo, los mismos quedan fijos y constantes en el tiempo para todos los pronósticos que se realicen. Pero la contundente realidad nos dice que esto no siempre funciona así. Los fenómenos que queremos modelar cambian en el tiempo y por tanto, los modelos deben irse igualmente ajustando.

Los analistas, ingenieros de procesos y gerentes operacionales requieren tomar decisiones oportunas basadas en una realidad muy dinámica. Un cambio de configuración de una planta o de uno de sus procesos, cambia las condiciones del problema y de las variables a predecir, y los sistemas que apoyan la toma de decisiones deben irse adaptando dinámicamente si queremos que den respuesta oportuna y adecuada a la dinámica de las operaciones.

Para esto, los responsables de controlar estos procesos, necesitan herramientas ágiles y adaptativas, que logren analizar en corto tiempo miles de alternativas y presenten las mejores a los distintos niveles donde se requiere toma de decisiones. Todo esto nos ha motivado a diseñar y desarrollar un sistema de apoyo a la toma de decisiones, basado en análisis de series de tiempo y árboles de decisión para predecir el valor de variables industriales.

Nuestro algoritmo, simplemente probará y evaluará todos los modelos posibles, sin dejar a priori ninguno por fuera. Evidentemente, al explorar un exhaustivo universo de modelos posibles, sin imponer ningún criterio previo, mas allá de algunas transformaciones básicas, estaremos en mejores condiciones para tomar una decisión basada en la evidencia. Pero para esto, debemos determinar las reglas que la máquina usará para comparar y preferir un modelo por encima de otro.

Como hemos mencionado, en todo análisis de series temporales, es común que se realicen transformaciones o diferenciaciones a las series, según sus características. Algunas transformaciones pueden ser no lineales o logarítmicas, diferenciaciones contra períodos anteriores o una mezcla de ellas. En este caso, el experto de datos decidirá que transformaciones aplicar a la serie y el algoritmo probará todos los modelos posibles con la serie original y con todas las transformaciones que se definan. De esta manera, el gran poder de cómputo de la máquina y el algoritmo, ayuda a probar y evaluar sistemáticamente miles de modelos.

Una vez definidas las transformaciones a aplicar y las reglas para la selección que el experto ha entonado, el algoritmo queda listo para ser aplicado con nuevos datos a un área determinada y probar su generalización. El algoritmo preselecciona los modelos que den mejores resultados en las pruebas que el experto ha definido y los presenta al analista o decisor del proceso para su aprobación final. El decisor valida y evalúa los resultados y sugerencias de la máquina y tomará la decisión final sobre cuál modelo utilizar para realizar el pronóstico.

Así la máquina, usando un conjunto de transformaciones y reglas generales que el experto ha programado y engranado, se convierte en una herramienta para analistas, supervisores y gerentes operacionales, tomen decisiones mediante el análisis de una gran cantidad de datos temporales históricos, en un tiempo razonable. Nuestro enfoque busca, en cada período a estimar, los modelos que mejor ajustan los datos y cumplen las restricciones que impone el experto. Así ajustará dinámicamente los parámetros del modelo en el tiempo.

Adicionalmente, cuando disponemos de un histórico considerable de los datos operacionales del proceso a modelar, es posible simular el comportamiento de los pronósticos que realizaría el algoritmo de esos valores, usando los datos conocidos para el momento y compararlos contra los valores reales almacenados. De esta forma, no sólo podemos probar el modelo, sino que podemos construir una serie de reglas basadas en técnicas de aprendizaje estadístico para mejorar los criterios de selección de la máquina para encontrar el mejor modelo dentro del conjunto preseleccionado. Si no contamos con ese histórico, igualmente, es posible ir construyendo esa base de datos de conocimiento para analizarla en el tiempo.

Esta metodología ha sido diseñada y se está probando para estimar valores promedio de ciclo mensual para ciertos procesos industriales del sector energético, apoyando la toma de decisiones tácticas y operacionales. Actualmente, se está estudiando su adaptación para pronósticos de ciclo diario. Respetando el natural criterio de confidencialidad de los datos industriales asociados a los procesos a modelar, este escrito busca documentar y exponer la metodología utilizada para la construcción de este novedoso enfoque que creemos dará mucho que aportar en los próximos años a nuestra industria nacional.

III. CRITERIOS DE SELECCIÓN

En la amplia literatura sobre series temporales, se han descrito un gran número de pruebas para validar la consistencia estadística de los modelos planteados. Sin lograr ser exhaustivos en este punto, a continuación mencionaremos algunos de estos criterios, que nos han permitido construir una base de reglas para que la máquina logre ponderar y seleccionar un modelo en contraposición a otro.

A. Criterio de Información de Akaike (AIC)

El AIC (*Akaike Information Criterion*), es una medida que nos permite comparar la calidad de un modelo con respecto a otro en función de su número de parámetros (k) y el estimador de máxima verosimilitud de la varianza [1].

$$AIC = \ln(\hat{\sigma}_k^2) + \frac{n+2k}{n}, \quad \hat{\sigma}_k^2 = \sum_{t=1}^n (x_t - \bar{x})^2$$

Al comparar dos modelos, tenderemos a favorecer al que tenga un AIC menor, tanto para reducir la varianza como para evitar el sobre ajuste.

B. Prueba de Box-Pierce

Es una prueba aplicada a los residuos de un modelo de series temporales, típicamente un Autoregresivo de Medias Móviles, ARMA (p,q), para calcular la función de autocorrelación de la serie de residuos [5], la cual debería estar cercana a cero, si el modelo está bien planteado. Por tanto, seleccionaremos los modelos cuyo *p-valor* en esta prueba sea mayor.

C. Normalidad de los Residuos

Aplicamos varias pruebas para comparar la distribución de los residuos contra una distribución *normal* ($0, \sigma^2$), como por ejemplo, comparar los percentiles de ambas distribuciones. De esta forma, tenemos varios criterios que nos ayudan a ponderar los modelos entre sí, en función de la distribución empírica que siguen sus residuos. A partir de esto, podemos realizar pruebas de hipótesis sobre su normalidad, que es una de las premisas que deben cumplir los modelos de series temporales [6].

D. Eficacia del Estimador

Tomamos el último valor conocido de la serie y lo sacamos del modelo. Con los $n-1$ datos restantes calculamos el modelo que queremos probar. Luego realizamos la estimación de un nuevo valor y comparamos contra el valor conocido que sacamos originalmente de la serie.

Esto nos da una visión práctica de cuanto el modelo se acercó a predecir el último valor de la serie con los $n-1$ valores anteriores. Si con los $n-1$ elementos de esta serie, nos acercamos muy bien al elemento n , ¿estaremos en capacidad con ese modelo de acercarnos razonablemente bien al elemento $n+1$? Creemos que sí.

Otro criterio que podemos usar es la varianza de la distribución empírica de los errores o residuos del modelo, tomando los que tengan menor varianza o menor rango. En la Figura 3, se puede observar algunos gráficos de correlación

entre alguno de los criterios usados y en el recuadro rojo la posible zona de selección a aplicar. En este caso, cada gráfico representa la correlación entre cada par de variables. Podemos ver que dentro del recuadro rojo, se encuentran los modelos con menor valor de AIC, menor desviación estándar y mayor valor en la prueba de *Box-Pierce*.

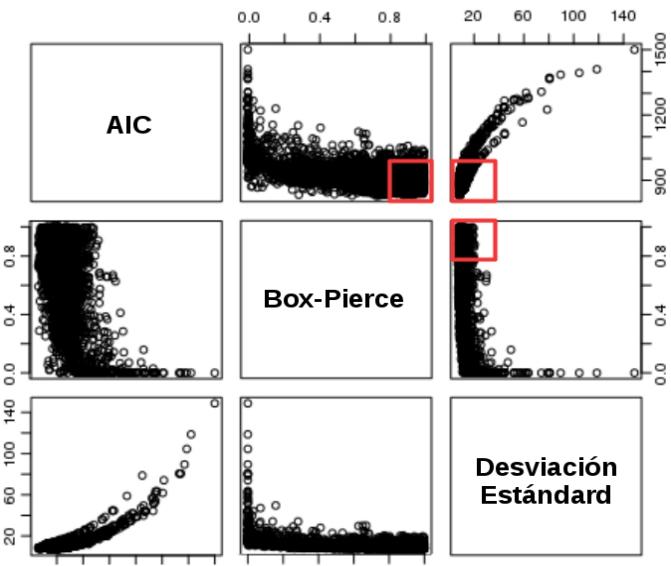


Figura 3. Correlación de Indicadores para la Selección de Modelos

Es importante notar que un criterio por sí sólo, no es la solución al problema. Es justamente la ponderación y el engranaje que realiza el científico, experto en análisis de datos, lo que nos dará el resultado esperado: tener de forma sistemática un conjunto de modelos a disposición del analista operacional o de negocios que le permitan, en base a su experiencia y una formación rápida en análisis de datos, tomar una decisión más informada de forma oportuna.

Por esto, el estudio y mejoramiento de estos criterios y reglas nos hará seguir avanzando para implantar mejores algoritmos y aplicaciones que ayuden a realizar pronósticos y construir escenarios para la toma de decisiones. Este último aspecto es el que estamos realizando con distintas técnicas de aprendizaje estadístico, donde en esta primera etapa hemos implantado algoritmos de árboles de decisión, de los cuales daremos algunos resultados.

IV. METODOLOGÍA GENERAL

Una vez que se ha decidido el conjunto de transformaciones que se desean aplicar a la serie original, según la naturaleza del problema y el comportamiento típico de los datos, para todas ellas se calculan todos los modelos posibles.

Por ejemplo, puede ser que interese trabajar con la serie diferenciada con respecto al mes, trimestre o año anterior y compararlas contra la serie original. También podríamos comparar varias transformaciones no lineales o una mezcla de las mismas en una sola ejecución. Esto no es limitativo, sin embargo, a medida que se determinan los modelos mejor ponderados y los rezagados, se va entonando en la práctica el

árbol de búsqueda para optimizar el algoritmo y dar respuesta en menor tiempo.

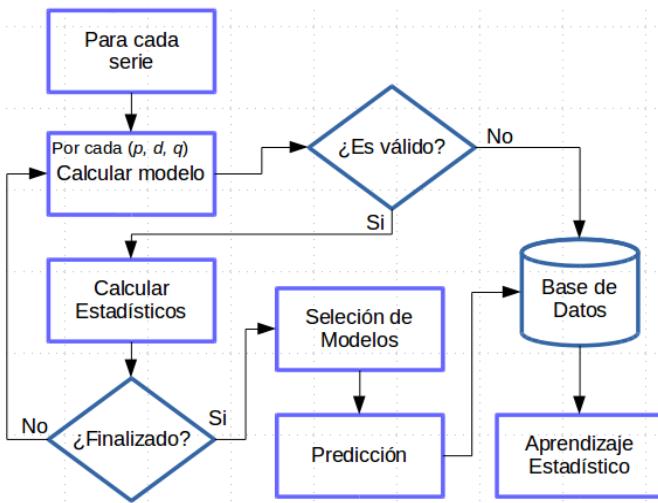


Figura 4. Algoritmo Propuesto e Implantado

A continuación, un fragmento del código donde se construyen en R, una lista de series a evaluar, tomando la serie original y cuatro diferenciaciones a uno, tres, cuatro y doce períodos.

```

Serie<-ts(Datos,start=c(ano,mes),frequency=12)
Series_a_Evaluar <- list(original = Serie,
                           dif1 = diff(Serie, lag=1),
                           dif3 = diff(Serie, lag=3),
                           dif4 = diff(Serie, lag=4),
                           dif12 = diff(Serie, lag=12))
  
```

De cada serie, se tomarán los mejores modelos en cada ejecución, ya sea diaria, semanal, mensual u otro período, dependiendo de las características de los datos y de la serie que tengamos. Por cada período a predecir, los modelos seleccionados de cada serie y sus pronósticos, se almacenan en una base de datos para evaluar su efectividad al comparar retrospectivamente los pronósticos con los datos reales. De esta forma, evaluamos la eficacia de todos los modelos seleccionados por la máquina y los seleccionados por el analista en cada ciclo. Estos datos son los que analizará nuestro algoritmo de aprendizaje para darle mejor información al usuario sobre cual es el mejor modelo para realizar el pronóstico basado en su propia experiencia previa.

A. Búsqueda Exhaustiva

Habiendo definido las series a evaluar, para cada una, calculamos un conjunto de modelos de tipo Autoregresivo Integral de Medias Móviles, ARIMA(p,d,q), para $0 \leq p \leq 12$, $0 \leq d \leq 5$, $0 \leq q \leq 12$. Este conjunto nos da originalmente un total de 1.014 modelos para cada serie, el cual se irá reduciendo según los modelos que la máquina elimine o por decisión del experto a la hora de entonar las reglas de búsqueda del algoritmo para una serie en particular.

También es posible que en base a la experiencia y la evolución que se va notando en cada ciclo, se descarten algunas transformaciones y/o se incorporen otras al conjunto de prueba.

B. Descarte de Modelos

Es posible que para un modelo en particular, es decir, una tripleta (p,d,q) , no se puedan calcular los coeficientes del modelo. En otras palabras, para esas condiciones no hay solución posible, por tanto, ese modelo se descarta para todas las corridas futuras. Esto se realiza con la intención de que a medida que evolucionan los pronósticos, vayamos 'podando' aún más el conjunto de búsqueda.

Finalmente, los modelos que prevalezcan serán los que estarán definidos para todos los períodos calculados desde la primera ejecución para este conjunto de series. Con esta información almacenada en bases de datos, no sólo mejoraremos nuestros algoritmos, sino que tendremos mejores criterios para formar a los analistas operacionales en análisis de datos y el experto podrá evaluar sus hipótesis iniciales y proponer nuevas.

C. Selección Segmentada

Para cada serie evaluada, el algoritmo seleccionará los cuatro mejores modelos, basándose en los criterios anteriormente señalados. En primera instancia, selecciona los que tengan mejor coeficiente AIC. Posteriormente se seleccionan los que resulten mejor en el test de Box-Pierce. Después, de este reducido grupo, podemos interceptar el conjunto de los que resultaron mejor en las pruebas de normalidad de residuos contra los que más se acercaron al último valor de la serie (elemento n , último valor conocido). El resultado de esta lista, si son más de cuatro modelos, podemos ordenarlos según el criterio deseado (por ejemplo, la desviación estándar del estimador o la distancia al elemento n en la prueba de eficacia) y tomamos los cuatro que quedaron mejor posicionados para cada serie.

Para establecer los umbrales de selección, usamos la misma distribución empírica conjunta y/o marginal de los indicadores ya mencionados, de esta forma, el criterio de selección no utiliza valores absolutos, sino selecciona según el percentil determinado por el experto de datos o la posición dentro de la escala de ponderación que este mismo experto ha definido.

D. Selección Final por el Analista de Datos

Finalmente, estos cuatro modelos preseleccionados para cada serie, son presentados al analista de datos para que tome la decisión final y realice los pronósticos. El analista puede evaluar la recomendación de la máquina y validar algunos resultados como, por ejemplo, el histograma de los residuos de cada modelo preseleccionado (Figura 5).

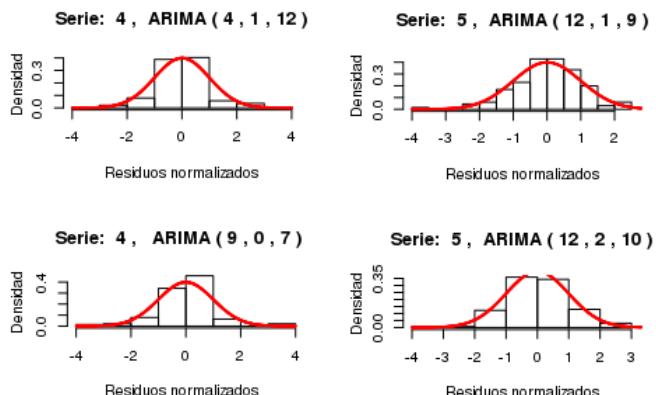


Figura 5. Ejemplo de Histogramas de Residuos para Selección

De todos los modelos seleccionados por la máquina, se guardan, en una base de datos, los resultados de cada prueba realizada para su selección y el pronóstico realizado con ese modelo. Una vez conocido el valor real que tomó la variable en estudio para ese período, éste se almacena en la misma base de datos para evaluar la eficacia de cada uno de los modelos. Con estos datos, el experto pueda realizar un análisis evolutivo posterior y mejorar los algoritmos y criterios de selección a aplicar y es el insumo para establecer la base minable de los algoritmos de aprendizaje estadístico.

El analista puede graficar la curva de la serie y visualizar el pronóstico resultante con ese modelo (Figura 6). Las líneas punteadas en rojo representan los límites del intervalo de confianza construido alrededor del estimador puntual.

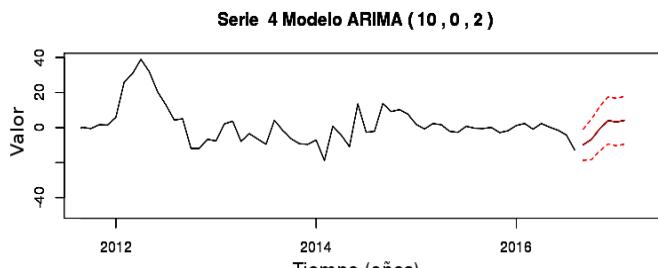


Figura 6. Gráfico de una Serie con Pronóstico

E. Aspectos de Base de Datos

El programa desarrollado, lee los datos de origen desde una base de datos con los valores históricos asociados a los procesos a trabajar y construye a partir de estos el conjunto de series con las transformaciones que ha definido el experto previamente. Estos datos de configuración son también traídos de la base de datos.

A continuación un fragmento de código R donde se muestra la definición de un conector de base de datos para PostgreSQL y una consulta tipo a base de datos para construir un conjunto de datos.

```
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv,
                  dbname="base_de_datos",
                  host="###.###.###.###",
                  port=5432,user="postgres",
                  password="*****")
res <- dbSendQuery(con,
                   'SELECT *
                    FROM esquema.tabla')
datos <- dbFetch(res)
```

Al detectar una tripleta (p,d,q) para la cual no es posible calcular los coeficientes del modelo, se ejecuta una excepción que almacena en la base de datos los valores de p , d y q asociados a la serie particular evaluada para esa variable. De esa manera, el algoritmo no volverá a probar con esa combinación para una próxima ejecución.

A continuación, una porción de código donde se muestra la función de cálculo de coeficientes y el manejo de excepciones en R.

```
tryCatch(modelo<-arima(Series_a_Evaluar[[serie]],
                        order = c(p, d, q)),
        error = function(x) {
          prueba<-Modelo_Fallido(variable,
                                 serie, p, d, q)
        })
```

De todos los modelos preseleccionados para una serie determinada, asociada a alguna variable industrial registrada, se almacenan los resultados de las pruebas y los pronósticos con su respectiva fecha y período al que corresponde dicho pronóstico, sea mensual, semanal o diario,

Periódicamente, se comparan los pronósticos con los valores nuevos que vienen de las mediciones de campo y se cargan en la base de datos para nuevas predicciones. Con estos datos, el experto puede medir la eficacia de cada modelo y realizar los ajustes si son requeridos.

F. Aprendizaje Estadístico

Ya sea que contemos con una gran cantidad de datos históricos que nos permita ejecutar el algoritmo con datos anteriores de entrenamiento, o que construyamos ese histórico con los pronósticos que se realicen desde su puesta en marcha, el uso de técnicas de aprendizaje estadístico para mejorar la selección final de los modelos es un aspecto muy valioso.

En este caso, tomamos diez años de valores conocidos y ejecutamos nuestro algoritmo de estimación desde enero 2010, tomando para cada período los valores conocidos hasta ese momento y el resultado, estimado por cada modelo, se comparó con el valor real.

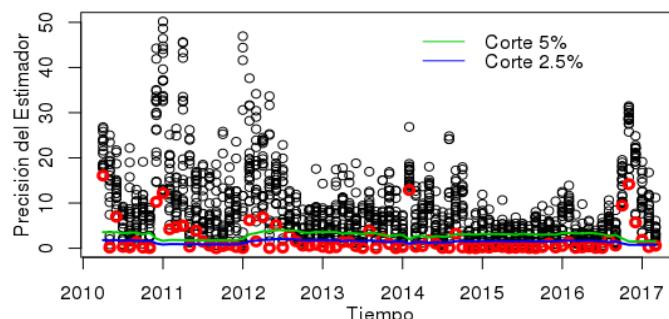


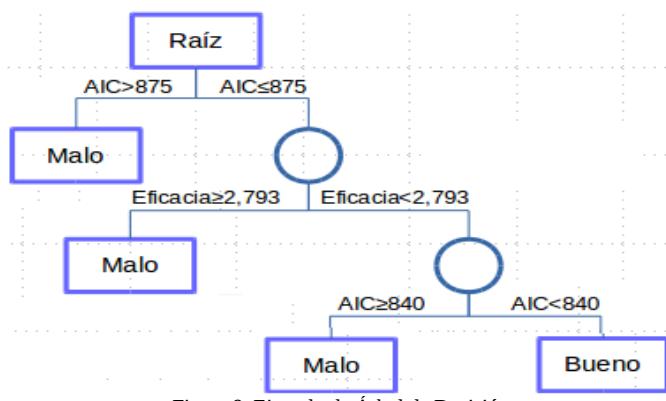
Figura 7. Diferencias de los Pronósticos Durante Aprendizaje

En la Figura 7 se muestran las diferencias entre el valor real y cada una de las estimaciones realizadas, para una misma variable, durante un período de aprendizaje de siete años. Los valores en color rojo representan el mejor pronóstico (más cercano al valor real) realizado por el algoritmo para cada período. Las curvas de color verde y azul delimitan la zona donde la predicción presentó una diferencia menor al 5% y al 2,5% con respecto al valor real de la variable en ese período.

Consideraremos un buen modelo aquellos que posean una diferencia menor del 5% con respecto al valor real y un modelo con una diferencia menor a 2,5% como un modelo excelente. Podemos notar que existen períodos en que ninguno de los modelos pudo estar lo suficientemente cerca, sin embargo, para la amplia mayoría de los períodos se encontraron modelos muy cercanos al valor real.

Basado en este criterio, se preparan dos variables de clasificación que indicarán si el modelo puede ser considerado bueno o no y si puede ser considerado excelente o no. Estas nuevas variables, unidas a los valores de AIC, el error de la prueba de eficacia (pronóstico del elemento n), la desviación estándar de los residuos y del estimador, son utilizadas como datos para construir un árbol de decisión (Figura 8). Las reglas resultantes son usadas en nuevas ejecuciones para darle al decisor una mejor recomendación.

En el ejemplo, un modelo con un AIC mayor a 875 es clasificado de una vez como un mal modelo, así como los que teniendo un AIC menor o igual a 875, tengan una eficacia igual o mayor a 2,793. Sólo se clasifican como modelo “bueno” los que posean eficacia menor a 2,793 y un AIC menor a 840.



V. RESULTADOS

Con esta metodología y el algoritmo desarrollado e implantado se han realizado estimaciones de promedios mensuales para datos operacionales de algunas plantas del sector de energía del país.

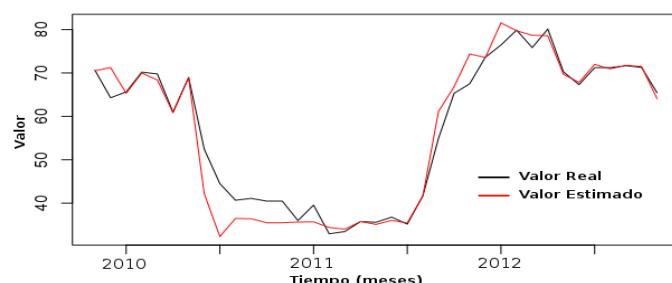


Figura 9. Datos Reales versus Pronóstico Realizados por el Algoritmo

Durante el período de prueba, se han obtenido resultados con diferencias absolutas menores a ocho centésimas ($< 0,08$) y relativas inferiores al 0,1% con respecto al valor real, medido y verificado por el departamento responsable del área operacional correspondiente (ver Figura 9).

En la siguiente tabla (Tabla I), podemos observar algunos períodos del aprendizaje y como el algoritmo fue ajustando los parámetros. Igualmente, se puede ver como unos modelos vuelven a ser seleccionados en el tiempo y en algunos casos son modelos muy cercanos entre sí.

TABLA I. EVOLUCIÓN DE MODELOS DURANTE ENTRENAMIENTO

Año	Mes	Serie	p	d	q	Diferencia	AIC
2016	2	2	11	0	8	1,31683455	839,47944
2016	3	2	11	0	6	0,07919229	846,66450
2016	4	5	3	0	12	0,17304743	792,92478
2016	5	4	5	0	6	0,36668074	845,72376
2016	6	1	12	0	4	0,07119348	868,95186
2016	7	1	12	0	4	0,05324412	874,94216
2016	8	5	11	0	11	1,76432401	823,59321
2016	9	1	7	0	8	0,58715746	886,34540

Esto está dentro de los objetivos que nos hemos planteado para esta primera etapa de prueba y seguimos evaluando y mejorando los métodos cada día.

TABLA II. TABLA PARCIAL DE RESULTADOS PARA SELECCIÓN

Número de modelo	Num. de Serie	p	d	q	AIC	Indicadores Desv. Std.	Box-Pierce	Predicción	Diferencia
2126	3	6	0	7	883,3175	8,110390	0,8881563	52,74089	0,04857977
2336	3	8	1	10	885,6179	7,821447	0,9079015	53,04330	0,25382522
320	1	7	1	11	888,8161	7,568454	0,9262747	52,37180	0,41767004
2139	3	6	0	8	884,3977	8,080934	0,9549042	53,22259	0,43311821

Una vez, pre-seleccionados los modelos, el analista puede observar los distintos resultados obtenidos y compararlos entre sí. Adicionalmente, puede realizar mayores pruebas si lo considera necesario. En la tabla II, puede visualizarse un ejemplo de algunos modelos preseleccionados por el algoritmo y la comparación de los valores resultantes en alguna de las pruebas aplicadas.

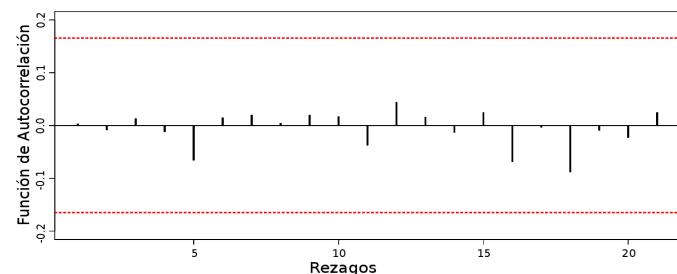


Figura 10. Gráfico de Autocorrelación Parcial de Rezagos

También se puede visualizar las funciones de correlación simple y parcial para los rezagos de un modelo particular que se deseé verificar (Figura 10). En este caso, podemos observar, para este modelo seleccionado, como los valores están distribuidos y muy por debajo de las bandas de confianza.

Para estos modelos resultantes, se pueden aplicar todas las pruebas y el análisis que tradicionalmente se realiza a los datos, tanto para seleccionar los parámetros probables del modelo, como para la comprobación estadística del mismo. Sin embargo, este análisis posterior se realiza no para seleccionar los parámetros del modelo sino para validar los seleccionados por la máquina.

En la Figura 9, se muestra el comportamiento de una variable real medida, contra los pronósticos que el algoritmo fue realizando en el tiempo. Por cada período el algoritmo fue cambiando el modelo para adaptarse a los nuevos datos.

En la Figura 11, podemos observar el árbol de decisión construido para la selección de modelos considerados excelentes para esta prueba de aprendizaje, la cual contó con 1.669 pronósticos en un período de 84 meses. Aquí se puede notar como el valor AIC es una de las principales pruebas para la clasificación, sin embargo, un valor muy bajo no representa una garantía absoluta para la selección.

Igualmente, para el criterio que hemos denominado “Eficacia”, para referirnos a lo cerca que este modelo puede predecir el último valor de la serie con los $n-1$ elementos anteriores, nos da una muy buena referencia. Sin embargo, cuando la desviación estándar de los residuos es menor, demasiado cerca tampoco será bueno.

Por otro lado, cuando la desviación estándar del estimador es menor, estar muy cercano al último elemento de la serie vuelve a ser una muy buena señal. Es importante observar que las reglas que surgen de este aprendizaje estadístico, establecen nuevas relaciones no triviales entre estos valores que pueden ser aplicables a escenarios futuros en este mismo contexto.

Finalmente, para mayor detalle, presentamos el árbol de clasificación, con los criterios detallados en modo texto:

- 1) Raíz
- 2) AIC < 739.4034 ==> Malo
- 3) AIC ≥ 739.4034
- 6) AIC ≥ 874.9885 ==> Malo
- 7) AIC < 874.9885
- 14) Eficacia ≥ 2.793151 ==> Malo
- 15) Eficacia < 2.793151
- 30) Desv.Estd.Residuos < 8.549181
- 60) AIC < 843.4874 ==> Malo
- 61) AIC ≥ 843.4874
- 122) Eficacia < 0.8773045 ==> Malo
- 123) Eficacia ≥ 0.8773045 ==> Excelente
- 31) Desv.Estd.Residuos ≥ 8.549181
- 62) Desv.Estd.Estimador ≥ 12.41628 ==> Malo
- 63) Desv.Estd.Estimador < 12.41628
- 126) Eficacia ≥ 0.9645337
- 252) AIC ≥ 771.5318 ==> Malo
- 253) AIC < 771.5318 ==> Excelente
- 127) Eficacia < 0.9645337 ==> Excelente

VI. CONCLUSIONES

Esta novedosa y práctica metodología nos permite obtener rápidamente modelos predictivos de gran valor y eficacia para la estimación de datos operacionales. Así podemos darle a los analistas, ingenieros y gerentes operacionales mejores herramientas para tomar decisiones oportunas y más informadas.

Aprovechar el gran poder de cómputo que disponemos hoy en día y combinarlo con minería de datos en el sector industrial para la toma de decisiones en áreas operacionales, nos da una herramienta muy valiosa para la optimización de procesos.

Almacenar estos resultados en un histórico de pronósticos, nos brinda mayores posibilidades de análisis para seguir construyendo mejores reglas de negocio y mejores métodos de predicción y optimización en el futuro.

Cuando analizamos lo bien y rápido que el algoritmo se

adapta a cambios en las condiciones del proceso, vemos como esta forma dinámica de calcular el modelo nos puede brindar mejores pronósticos y aprender de la experiencia.

No sólo aprende la máquina, aprende también el analista y el decisor táctico a cargo del proceso. Así las decisiones dependen cada vez menos de la intuición y son basadas mayormente en los datos reales de cada proceso industrial a controlar.

La construcción de reglas usando árboles de decisión, no sólo es adecuada y fácil de implantar dentro de los algoritmos de toma de decisiones, sino que son entendibles por el usuario funcional del modelo y aplicables en escenarios operacionales, ayudando a la creación de nuevas reglas de negocio.

Finalmente, recomendamos seguir estudiando y mejorando la metodología, los algoritmos y los criterios de selección, usando los registros que se tienen hasta el momento y los que se seguirán generando en el futuro en distintos escenarios de aplicación.

REFERENCIAS

- [1] P. J. Brockwell, R. A. Davis. "Introduction to Time Series and Forecasting". Springer-Verlag New York, Inc. pp. 1-219. 2002, Second Edition. ISBN 0-387-95351-5
- [2] M. P. Gonzalez. "Análisis de Series Temporales: Modelos ARIMA". Departamento de Economía Aplicada III (Econometría y Estadística). Facultad de Ciencias Económicas y Empresariales . Universidad del País Vasco. Sarriko-On. 2009.
- [3] T. H. Diem Ngo. "The Box-Jenkins Methodology for Time Series Models". Warner Bros. Entertainment Group, Burbank, CA. Statistics and Data Analysis. SAS Global Forum 2013.
- [4] S. Green. "Time Series Analysis of Stock Prices Using the Box- Jenkins Approach". Electronic Theses & Dissertations. Georgia Southern University. 2011.
- [5] G. E. P. Box, D. A. Pierce, Distribution of Residual Correlations in Autoregressive-Integrated Moving Average Time Series Models. Journal of the American Statistical Association, 65, pp. 1509–1526. 1970.
- [6] R. H. Shumway, D. S. Stoffer. "Time Series Analysis and Its Applications With R Examples". Second Edition. Springer Science. 2006.

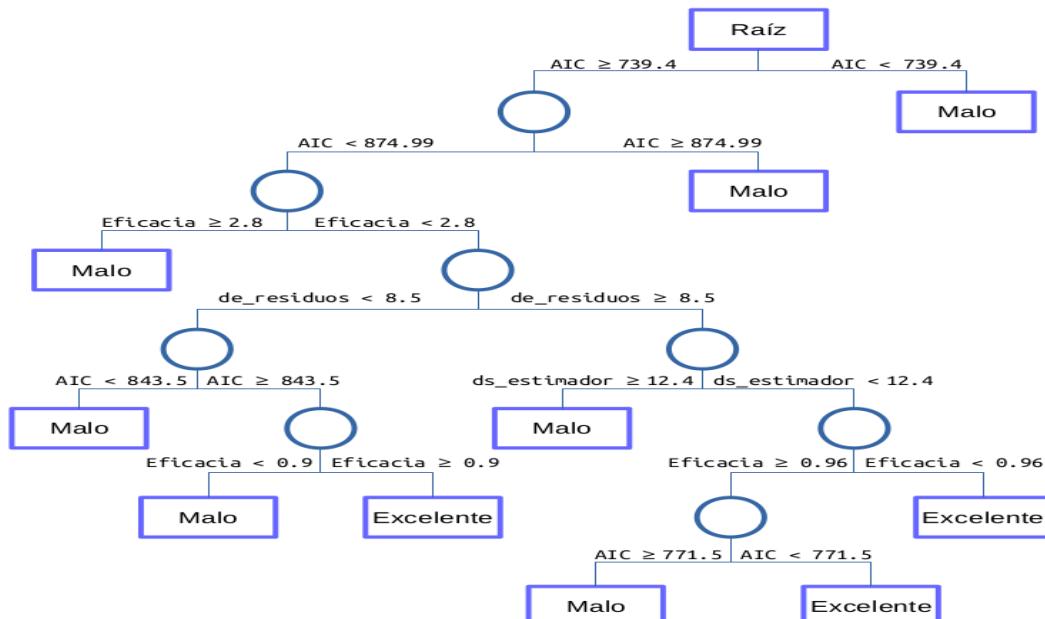
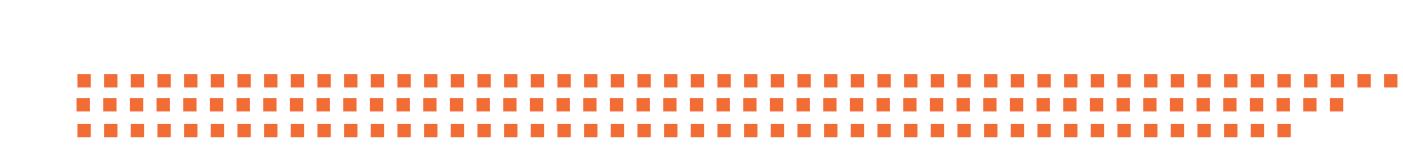


Figura 11. Árbol de Decisión Resultante del Aprendizaje Realizado



Índice de Autores

Índice de Autores

A

Aceros Egner	123, 128
Acosta María	73
Aguilar Jose	114, 123
Aguilera Ronald	73
Albaran Iris	82
Amezquita Colombia	82

B

Bautista Liliana	49
Briceño Mariuska	128

C

Cadenas José	107
Camargo Edgar	123, 128
Carballo Yusneyi	73
Carrasquel Soraya	107
Castañeda Marbella	82
Coronado David	107

D

Delgado Mayela	25
Dos Santos Ricardo	114

E

Esteves Yuly	13
--------------	----

F

Flaviani Federico	94
-------------------	----

G

Gamess Eric	60
Guzmán Lice	123, 128

L

Labbad José	2
Losavio Francisca	13, 25
Lozada Hector	123

M

Matteo Alfredo	25
----------------	----

N

Naime Alfredo	134
---------------	-----

P

Pereira Gustavo	60
-----------------	----

R

Ramírez Esmitt	37
Ramírez Juan	49
Rodríguez Rosseline	2, 107
Rodríguez Taniana	114

S

Sam Daniel	37
Silva Liliana	82

T

Tineo Leonid	2, 107
--------------	--------

CoNCISa 2017

UCAB, Ciudad Guayana, Venezuela - Octubre 2017

Sociedad Venezolana de Computación

Conferencia Nacional de Computación, Informática y Sistemas

La Sociedad Venezolana de Computación está comprometida con el impulso de una nueva generación académica profesional en nuestra área de saber para el desarrollo del país.

Los editores expresan su mayor gratitud a todos los que contribuyeron con sus altos estándares de revisión para garantizar el nivel de excelencia de CoNCISa. También se destacó el valorable y eficiente aporte del Comité Organizador y del Comité de Programa que contribuyeron de forma importante para la edición de estas memorias.

Los conceptos y puntos de vista expresados en los trabajos publicados en este libro representan las opiniones personales de los autores y no reflejan el juicio de los editores, de las universidades participantes, o de la Sociedad Venezolana de Computación.

www.concisa.net.ve

ISBN: 978-980-7683-03-6

