Proyectos Sociotecnológicos

Una Guía para su Desarrollo en el Marco del Programa Nacional de Formación en Informática

Elías Oswaldo Cisneros Arocha
Equipo de Investigación Antonio José de Sucre
Los Teques, Venezuela
cisnerose@cantv.net

Juan Vicente Cisneros Arocha
Equipo de Investigación Antonio José de Sucre
Los Teques, Venezuela
juanv.cisneros@gmail.com

***Resumen*—El Programa Nacional de Formación en Informática (PNFI) es una iniciativa gubernamental para homologar los programas de estudios universitarios públicos en el área de la informática a nivel nacional, el eje fundamental del PNFI son los Proyectos Sociotecnológicos (PST). Se propone una guía de referencia para el desarrollo de PST del Trayecto III, se indican recomendaciones generales para profesores y estudiantes para el abordaje de los PST. Así mismo, se describen los roles, documentos y herramientas a emplear en cada trimestre tomando como referencia la Metodología de la Red Nacional de Integración y Desarrollo de Software Libre, el Diagnóstico Social Participativo y el programa de estudios del PNFI. Se indican un conjunto de estrategias, tareas y recursos para guiar el proceso de desarrollo del PST. Se muestran los resultados de la experiencia docente con dos grupos de la Unidad Curricular de PST entre los años 2011 y 2012. Uno de los principales problemas encontrados es mantener la cohesión del desarrollo del PST con el resto de las unidades curriculares del Trayecto. Finalmente, las orientaciones indicadas en este artículo deben ser evaluadas y ajustadas tomando en consideración la naturaleza de cada proyecto, las pericias del equipo y otras variables que el docente considere importantes.**

Palabras clave—Proyecto Sociotecnológico, desarrollo de software, marco de trabajo, Programa Nacional de Formación en Informática.

I. Introducción

El Programa Nacional de Formación en Informática (PNFI) es creado en 2008 por Estado Venezolano [1], entre los objetivos del PNFI destacan la construcción de una red de conocimiento y aprendizaje para la generación, transformación y apropiación social del conocimiento en el área de la informática en el nivel universitario. El eje fundamental de este programa de estudios son los Proyectos Sociotecnológicos (PST), estos son unidades curriculares de integración de saberes y contraste entre teoría y práctica, además comprenden espacios de formación y vinculación social [1]. La motivación principal para el desarrollo de esta contribución es la escasa cantidad de publicaciones arbitradas referidas a la gestión de los PST en el contexto del PNFI. En la actualidad es esencial el desarrollo de software con calidad, de forma constante se realizan investigaciones y propuestas en este campo [2][3][4][5][6], este artículo describe un modelo de referencia basado en recomendaciones específicas para el abordaje de los PST del Trayecto III, se describen roles, documentos, herramientas, estrategias y tareas para cada trimestre con el fin de generar soluciones con calidad.

El método para recolectar la información consistió en la revisión de documentos oficiales y la observación directa del campo en estudio. La población estudiada consistió en dos cursos de la Unidad Curricular de Proyecto Sociotecnológico (UCPST) del Colegio Universitario de Caracas (CUC) matriculados en el Trayecto III de régimen nocturno entre los años 2011 y 2012.

II. Proyecto Sociotecnológico III

La estructura del plan del estudios PNFI está compuesta por un Trayecto Inicial y cuatro Trayectos ordinarios, la duración de un Trayecto ordinario es de 36 semanas, equivalente a un año de estudio, dividido en 3 trimestres de 12 semanas cada uno. La UCPST tiene la misma duración que un Trayecto ordinario y establece una carga horaria de 8 horas semanales, usualmente los equipos de trabajo lo conforman entre 2 y 4 estudiantes. Los PST se evalúan al final del Trayecto, y deben contar con el aval de los usuarios del sistema, los estudiantes que participaron, el profesor responsable de UCPST y un comité de profesores.

Según cada Trayecto, la UCPST tiene un objetivo académico particular, los estudiantes que ingresan al Trayecto III deben haber desarrollado en el Trayecto anterior las competencias de desarrolladores de software y de modeladores de bases de datos de baja o mediana complejidad. En el Trayecto III se exige la identificación de un problema en el ámbito comunitario, la descripción de los factores que lo generan y el desarrollo de un sistema informático que mejore la situación problemática, cumpliendo el tiempo establecido y con atributos de calidad [1]. Para cumplir los objetivos académicos los estudiantes aplican los conocimientos teóricos y prácticos obtenidos en cada Unidad Curricular (UC) del Trayecto III. Las Unidades Curriculares (UUCC) que aportan los conceptos teóricos y prácticos para alcanzar estos objetivos se observan en la Tabla I.

TABLA I: UNIDADES CURRICULARES DEL TRAYECTO III

Trayecto III	***Trimestre III***	Investigación de Operaciones	*Electiva*	Todo el Trayecto: - Proyecto Sociotecnológico - Ingeniería del Software -Formación Crítica
	Trimestre II	Matemática aplicada II	Modelado de Bases de Datos	
	Trimestre I	Matemática aplicada I	Sistemas Operativos	

Cada UC realiza aportes teóricos y prácticos durante las fases específicas del proyecto, en el caso particular del Trayecto III, se esperan aportes en cuanto a enfoques socio-críticos, habilidades de razonamiento lógico, habilidades informáticas y habilidades para la gestión de proyectos, estas se desarrollan respectivamente en las siguientes Unidades Curriculares:

- UC Formación Crítica (FC): aportes en la vinculación social del proyecto, su importancia en el marco del desarrollo de las capacidades técnicas nacionales, la conveniencia del uso de tecnologías de estándares abiertos y/o software libre. Indicar orientaciones hacia el desarrollo tecnológico nacional y el impulso de la soberanía tecnológica.
- UC Matemática aplicada (MA): aportes sobre la selección de métodos estadísticos, probabilísticos, modelos matemáticos y/o teoría de grafos aplicados a la naturaleza del proyecto. Orientaciones sobre técnicas computacionales para el procesamiento de datos.
- UC Investigación de operaciones (IO): aportes sobre el uso de modelos matemáticos, estadística y algoritmos con objeto de apoyar el proceso de toma de decisiones. Estudiar sistemas reales, con la finalidad de mejorar su funcionamiento mediante los métodos de maximización o minimización.
- UC Ingeniería de Software (IS): aportes teóricos y prácticos sobre modelos, herramientas y métodos para el desarrollo de software de calidad, paradigmas de programación, mejores prácticas de arquitectura de software según la naturaleza de cada proyecto.
- UC Sistemas operativos (SO): aportes sobre la elección de sistemas operativos, sobre la infraestructura del proyecto en cuanto a arquitecturas distribuidas, clientes servidor, entre otras. Mecanismos de seguridad de los sistemas operativos empleados en la infraestructura del proyecto.
- UC Modelado de bases de datos (MBD): aportes sobre modelos de persistencia de datos según los requerimientos del proyecto, diseño, normalización y optimización del modelo de datos del proyecto.
- UC Proyecto Socio-Tecnológico: habilidades asociadas a la gestión de proyectos y asesoramiento metodológico.

Esta guía está alineada al enfoque de cohesión entre las UC del Trayecto III (Fig. 1), para ello se debe establecer un espacio periódico de discusión entre los profesores del Trayecto para mantenerse articulados sobre los avances de cada equipo de PST y las dificultades particulares que presentan. Se recomienda vincular los documentos requeridos en la UCPST con algunas de las asignaciones del resto de las UC, de esta manera se podrá garantizar los aportes en estos documentos por parte de cada profesor.

III. MeRinde y el Diagnóstico Social Participativo

En concordancia con los lineamientos de soberanía tecnológica nacional [7][8] se consideró una metodología para proyectos el desarrollo de software creada en el país, esta es la Metodología de la Red Nacional de Integración y Desarrollo de Software Libre (MeRinde), esta es producto de la integración de mejores prácticas entre varias metodologías y marcos de trabajo del área de desarrollo de software [5]. Esta metodología es el resultado de un proyecto venezolano auspiciado por el Estado [9], propone un estándar para el proceso de desarrollo de software que puede ser empleado y adaptado según los requerimientos de cualquier comunidad u organización. MeRinde provee un conjunto de plantillas que sirven de punto partida para los documentos utilizados en proyectos de desarrollo de software.

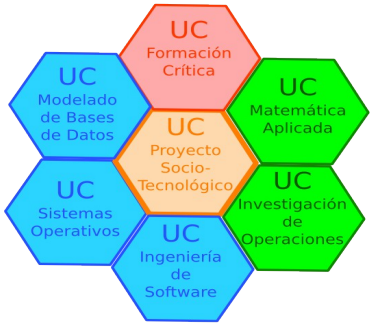

Fig. 1. Enfoque de cohesión entre las unidades curriculares III

Para vincular la metodología MeRinde con el Trayecto III se toma en cuenta el tiempo que se dispone para desarrollar el PST, este es de 36 semanas, dividido en 3 trimestres de 12 semanas cada uno. Por otra parte, MeRinde establece 4 fases, las cuales son Inicio, Elaboración, Construcción y Transición. Para aplicar MeRinde se propone el esquema planteado en la Fig. 2. Al finalizar el Trimestre I se debe obtener el objetivo del Proyecto y la arquitectura candidata, ambos son hitos de las fases Inicio y Elaboración respectivamente. Al finalizar el Trimestre II se espera que la solución propuesta tenga una capacidad operacional con la mayoría de los requerimientos implementados y finalmente en el Trimestre III la solución debe estar operativa para los usuarios, adicionalmente se espera que la solución desarrollada sea liberada con alguna licencia que permita su reutilización.

Por otra parte, el Diagnóstico Social Participativo (DSP) [10] es un método que permite el abordaje de una realidad social a través de la identificación de los problemas que afectan a la población, con el fin de actuar sobre ella para ocasionar cambios, tomando en cuenta la participación activa de la población y las instituciones vinculadas a la realidad social estudiada. El DSP es aplicado durante la Fase de Inicio en el Trimestre I y proporciona una guía para el abordaje de la comunidad para la cual se desarrolla el PST, enfocando los esfuerzos para identificar el contexto social, geográfico y económico. Adicionalmente se toma su planteamiento para priorizar los problemas que inciden en la comunidad, sus causas y efectos; así como su enfoque para identificar el objetivo del proyecto, sus medios y fines. Se generó una plantilla llamada “Identificación del Proyecto” en la cual se incluyen aspectos relevantes del DSP relacionados con el planteamiento del problema en estudio.

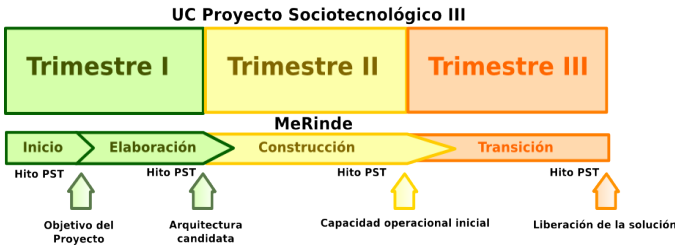

Fig. 2. Relación entre Trimestres de PNFI y fases de MeRinde

A. *Estimación del alcance de un Proyecto Sociotecnológico*

Continuamente se realizan investigaciones en el área de la planificación de proyectos de software que proponen diversos métodos para abordarlos [11][12][13]. En el caso de esta experiencia se consideró tomar como referencia la cantidad de horas académicas que se disponen en la UCPST (8 horas semanales) y la cantidad de integrantes del equipo de PST. En la Tabla II se observan la cantidad de horas que se disponen según el escenario para 2, 3 y 4 integrantes por equipo de PST.

TABLA II: INTEGRANTES Y HORAS ACADÉMICAS PARA DESARROLLAR EL PST

Integrantes del equipo	Horas semanales	Semanas del Trayecto	Total Horas
2	16	36	576
3	24	36	864
4	32	36	1.152

Esta estimación permite formar un criterio al momento de establecer el alcance de un PST, además se deben considerar la cantidad de requerimientos, el tipo de infraestructura tecnológica, el área o dominio del negocio, la actualidad de la temática, las pericias de cada equipo, entre otras. Según la experiencia de los autores en proyectos del PNFI se consideró que el esfuerzo total del PST se corresponde con 9% en tareas la fase de Inicio, 25% en tareas de la fase de Elaboración, 33% en tareas de la fase de Construcción y 33% en tareas de la fase de Transición, sin embargo estos criterios pudieran ser ajustados según el contexto de cada PST. Siguiendo la mejor práctica de "demostrar resultados iterativamente e incrementalmente" propuesta por MeRinde, se recomienda establecer iteraciones de aproximadamente 3 semanas de duración, se espera que en cada trimestre se logren realizar 4 iteraciones. En la Tabla III se observan la cantidad de horas estimadas que se dedican a cada fase de MeRinde durante la duración de la UCPST.

TABLA III: HORAS SEGÚN INTEGRANTES DEL PST Y LA FASE DE MERINDE

Integrantes del equipo / horas	Inicio	Elaboración	Construcción	Transición	Total
2	52	144	190	190	576
3	78	216	285	285	864
4	104	288	380	380	1.152

En la Fig. 3 se observa el énfasis de las disciplinas de MeRinde en cada iteración, así mismo se indican la cantidad de iteraciones en cada trimestre.

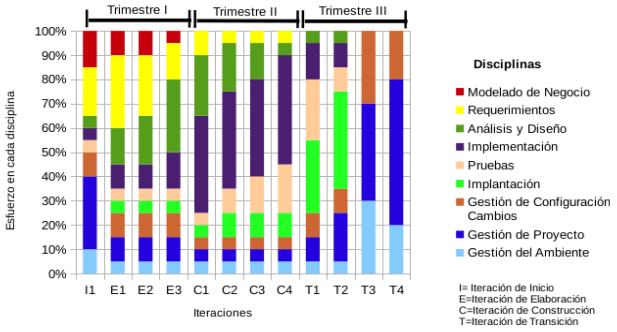

Fig. 3. Disciplinas e iteraciones en el contexto del Trayecto III

A continuación se indican las disciplinas de MeRinde y sus aportes al desarrollo del PST:

- Modelado de Negocio: analizar los procesos y la identificación de los casos de uso del negocio, adicionalmente se incorpora el Diagnóstico Social Participativo con el propósito de identificar causas y efectos del problema, actores involucrados, y finalmente medios y fines para solucionar el problema.
- Requerimientos: capturar los requerimientos funcionales y no funcionales, analizarlos, priorizarlos. El objetivo principal de esta disciplina es identificar las funciones del sistema.
- Análisis y Diseño: transformar los requerimientos a una especificación que describa cómo implementar el sistema. El análisis consiste en obtener una visión del sistema a desarrollar. Por otro lado, el diseño es un refinamiento que toma en cuenta como el sistema cumple sus objetivos.
- Implementación: convertir los elementos del diseño en elementos de implementación, dichos elementos son códigos fuentes, ejecutables, binarios, entre otros.
- Pruebas: evaluar la calidad del producto que se está desarrollando, mediante la aplicación de pruebas para validar que las suposiciones hechas en el diseño y los requerimientos se estén cumpliendo satisfactoriamente, esto quiere decir que se verifica que el producto funcione como se diseñó y que los requerimientos son satisfechos. Probar y certificar la solución en el ambiente operativo del cliente.
- Implantación: distribuir e instalar el sistema desarrollado y asegurar la disponibilidad del producto para los usuarios finales.
- Gestión de Configuración y Cambios: mantener la integridad de todos los objetos que se crean en el proceso y controlar los cambios. Se debe identificar elementos de configuración, restringir y auditar los cambios a esos elementos.
- Gestión de Proyecto: realizar estudio de factibilidad, alcanzar los objetivos propuestos, administrar el riesgo y superar las restricciones para desarrollar un producto que sea acorde a los requerimientos de los usuarios.

- Gestión del Ambiente: dar soporte al proyecto con los procesos, métodos y herramientas correctas. Ofrecer las herramientas, plantillas, documentos, lineamientos a seguir para el desarrollo del software y del proyecto.

IV. Roles, Documentos y Herramientas

La conformación del equipo de PST es un asunto importante, cada integrante debe aportar habilidades esenciales para contribuir al proyecto [14][15][16]. Es común permitir la conformación voluntaria de los equipos de proyecto, sin embargo esta práctica puede generar equipos con habilidades desequilibradas. En este sentido, es aconsejable definir un método que permita caracterizar las habilidades de cada estudiante y proponer la conformación de equipos de trabajo equilibrados, en los cuales sus integrantes complementen sus fortalezas y debilidades. Algunos criterios pueden ser las habilidades técnicas en modelado de requerimientos, análisis y diseño, programación, dominio de sistemas operativos, entre otras. También son importantes habilidades como la capacidad para trabajar en equipo, redacción, gestión del tiempo, liderazgo, entre otras. A continuación se indican los roles más importantes y su participación a lo largo del PST.

A. Roles propuestos

A medida que el proyecto avanza los miembros del equipo asumen distintos roles, la decisión de la asignación de los roles puede ser consensuada entre el equipo de PST y el Profesor, tomando en cuenta las fortalezas de los integrantes del equipo y el desempeño que cada uno demuestre en las tareas del proyecto. A continuación se indican los roles que se consideran indispensables en el contexto del PST del Trayecto III:

- Analista de Calidad: velar por el cumplimiento de las mejores prácticas, se encarga de revisar los documentos que reflejan el avance del proyecto (diagrama Gantt, actas de reunión, reporte de pendientes, y otras afines al control y seguimiento del proyecto), y de verificar que los objetivos se cumplan.
- Analista de Producto: dirigir el proceso de captura de requerimientos, definir los actores y casos de uso y estructurar el modelo de casos de uso, estableciendo la forma en que funcionará el sistema y cuáles son las restricciones del mismo.
- Arquitecto de Software: definir la arquitectura del sistema y de la refinación de la misma en cada iteración; debe construir prototipos necesarios para probar aspectos riesgosos desde el punto de vista técnico; define los lineamientos generales del diseño y la implementación.
- Desarrollador: codificar los componentes en código fuente en algún lenguaje de programación; debe elaborar y ejecutar las pruebas unitarias realizadas sobre el código desarrollado; es responsable de los componentes que ha desarrollado debiendo documentarlos, actualizarlos ante cambios y mantenerlas bajo el control de configuración mediante alguna herramienta.
- Probador: realizar las pruebas definidas previamente, utilizando las instrucciones, métodos y herramientas necesarias para esta tarea.
- Analista de infraestructura: configurar y administrar los recursos de infraestructura para el funcionamiento del sistema que está en desarrollo, esto incluye los aspectos de telecomunicaciones, procesamiento y almacenamiento de datos.

Existen otros roles importantes, el involucrado, este rol agrupa a las personas que requieren la solución, estos pueden ser usuarios y/o clientes. También están los profesores del Trayecto III, estos asumen el rol de Mentores, realizan acompañamiento a los equipos de trabajo mediante revisiones de los documentos y haciendo recomendaciones para mejorar los mismos, cada profesor del Trayecto III hace aportes según la UC asignada y las disciplinas que les correspondan, estos complementan aspectos específicos según las orientaciones del Profesor de la UCPST. Estos Mentores están en la capacidad de aclarar dudas que puedan surgir sobre su área de competencia, así como también contribuir en la calidad general del desarrollo del proyecto. A continuación, en la Tabla IV se muestra la participación de cada rol según el trimestre en que se encuentre el PST.

Tabla IV: Participación Trimestral de los Roles del PST

Roles	Trimestre I	Trimestre II	Trimestre III
Analista de Calidad	Media	Baja	Baja
Analista de Producto	Alta	Media	Baja
Arquitecto de Software	Media	Alta	Media
Desarrollador	Baja	Alta	Media
Probador	Baja	Media	Alta
Analista de Infraestructura	Baja	Baja	Media
Involucrado	Alta	Alta	Alta
Mentor (Profesor PST – Profesor PNFI)	Alta	Alta	Alta

B. *Documentos propuestos*

En este apartado se describen un conjunto de documentos y/o artefactos que han sido adaptados al contexto de la UCPST, estos están basados en MeRinde, el DSP y el plan de estudios del PNFI, a continuación se describen:

- Identificación del proyecto: contiene elementos del diagnóstico social participativo, la definición de la situación problemática, objetivo general, específicos, causas, efectos, medios y fines de la propuesta de Proyecto. Se aborda un estimado de costos y un análisis de factibilidad general. Finalmente las autoridades académicas deciden sobre la aprobación de la propuesta de PST.
- Planificación, riesgos y visión: describe el alcance del proyecto, las tareas a realizar, los riesgos identificados y visión general del sistema.
- Especificación de requerimientos: indica los requerimientos del sistema, describe las funciones del sistema, los requerimientos no funcionales, características del diseño, y otros elementos necesarios para proporcionar una descripción completa y comprensiva de los requerimientos.

- Documento de arquitectura: indica las especificaciones principales del diseño, proporciona una descripción de la arquitectura del sistema y sirve como medio de comunicación entre el arquitecto de software y otros miembros de equipo del proyecto con respecto a las decisiones arquitectónicas que se han tomado en el proyecto. Contiene varias vistas que muestran aspectos distintos del sistema como son: vista de casos de uso, vista lógica, vista de implementación, vista del proceso, vista de implantación y vista de persistencia.
- Plan de pruebas: especifica los casos de prueba y procedimientos de prueba, este artefacto incluye el propósito de las pruebas, elementos a probar, herramientas a utilizar y recursos. Al tener el resultado de las pruebas se puede comparar lo obtenido con lo esperado.
- Plan de implantación: describe el conjunto de tareas para poner en funcionamiento el sistema en las instalaciones de los usuarios. Las actividades descritas en este documento abarcan temas referentes a la instalación del sistema, instrucciones específicas sobre la sustitución de antiguos sistemas, compatibilidad del sistema, estrategias de migración y adaptación al nuevo sistema, así como planes de contingencia en caso de fallos que garanticen el menor impacto a la continuidad del negocio.
- Manual de usuarios: describe las funcionalidades del sistema desde la perspectiva de los usuarios, está enfocado como un documento de ayuda a las personas que manipularan directamente el sistema.
- Memoria descriptiva: es un requisito académico del PNFI que contiene la descripción de las actividades más relevantes realizadas durante el desarrollo del PST.

A continuación, en la Tabla V se muestran los documentos y su evolución a lo largo del Trayecto III.

TABLA V: ELABORACIÓN Y EVOLUCIÓN DE LOS DOCUMENTOS

Documento entregable	Trimestre I	Trimestre II	Trimestre III
Identificación del Proyecto	Iniciar-Refinar-Terminar		
Planificación, Riesgos y Visión	Iniciar-Refinar	Refinar-Terminar	
Especificación de Requerimientos	Iniciar-Refinar	Refinar-Terminar	
Documento de Arquitectura	Iniciar	Refinar	Terminar
Plan de Pruebas		Iniciar-Refinar	Refinar-Terminar
Plan de Implantación		Iniciar-Refinar	Refinar-Terminar
Manual de Usuarios			Iniciar-Terminar
Memoria Descriptiva		Iniciar-Refinar	Refinar-Terminar

En la Tabla VI se muestra el esquema de revisiones por parte de los profesores de cada UC, las revisiones de cada profesor se deben enfocar hacia su área de competencia. Las revisiones de los documentos son coordinadas por el profesor de la UCPST, para ello se debe establecer un espacio de coordinación entre los profesores del Trayecto III. Existen un grupo de documentos complementarios que se consideran de gran valor para el proyecto, a continuación se describen:

- Portafolio de evidencias: compendio de minutas, fotografías, avales de asistencia a jornadas técnicas, cursos, talleres, entre otros.

TABLA VI: DOCUMENTOS Y REVISIONES POR PARTE DE LAS UC DEL TRAYECTO III

Documento entregable / Unidad Curricular	FC	MA	IS	SO	IO	MBD
Identificación del Proyecto	X	X	X			
Planificación, Riesgos y Visión	X	X	X	X		
Especificación de Requerimientos		X	X	X		
Documento de Arquitectura			X	X	X	X
Plan de Pruebas			X	X	X	
Plan de Implantación	X		X			X
Manual de Usuarios	X		X			
Memoria Descriptiva	X					

- Vídeo testimonial: material audiovisual para mostrar las actividades desarrolladas durante la realización del PST, es importante reflejar el testimonio de los involucrados, la situación problemática y los resultados obtenidos con la implantación del sistema.
- Avances del Trimestre I: documento con los avances más significativos del Trimestre I.
- Avances del Trimestre II: documento con los avances más significativos del Trimestre II.
- Sitio informativo: sitio web disponible en internet con los aspectos más relevantes del PST, debe incluir objetivo del proyecto, logros, capturas de pantallas, identificación del equipo de proyecto, información de contacto, documentación de interés.
- Aval de divulgación pública: certificado de presentación del proyecto en una jornada de divulgación de conocimientos o mediante una publicación en alguna revista de acceso público.
- Encuesta de valoración de usuarios: contiene las valoraciones de los usuarios luego de utilizar el sistema con el propósito de conocer la aceptación del sistema y mejorar sus atributos de calidad.

A continuación, en la Tabla VII se observa la evolución de cada uno de los documentos complementarios a lo largo del Trayecto III.

TABLA VII: EVOLUCIÓN DE LOS DOCUMENTOS COMPLEMENTARIOS

Otros documentos complementarios	Trimestre I	Trimestre II	Trimestre III
Portafolio de evidencias	Iniciar	Refinar	Terminar
Vídeo testimonial	Iniciar	Refinar	Terminar
Avances del Trimestre I	Terminar		
Avances del Trimestre II		Terminar	
Sitio informativo			Terminar
Aval de divulgación pública			Terminar
Encuesta de valoración de usuarios			Terminar

C. *Herramientas propuestas*

Durante el desarrollo del PST se utilizan varias herramientas de apoyo, los profesores del Trayecto III deben orientar a cada equipo sobre las herramientas más adecuadas según la naturaleza de cada PST, a continuación se describen algunas categorías de estas herramientas y se sugieren algunas que respetan los lineamientos de soberanía tecnológica nacional:

- Generador de diagramas UML: crear los diagramas de ingeniería de software como casos de uso, de clases, de actividades, de despliegue, de estados, entre otros. Se sugieren las herramientas Modelio, Umbrello y StarUML.
- Diseñador de prototipos: crear prototipos de pantallas y validar los requerimientos funcionales. Se sugiere la herramienta Pencil Project.
- Planificador de Proyectos: reflejar la planificación del proyecto, tareas, documentos, responsables, entre otros. Se sugiere la herramienta OpenProj.
- Documentador de código fuente: documentar el funcionamiento del código fuente, se genera documentación detallada sobre los módulos, clases, atributos, métodos, entre otros. Se sugieren las herramientas PHPDoc y PyDoc.
- Gestor de versiones de software: mantener el control sobre las versiones desarrolladas, es de utilidad para equipos de trabajo que deben reutilizar componentes de software. Se sugiere la herramienta Subversion y Git.
- Gestor de incidencias de software: documentar sistemáticamente las fallas, problemas y soluciones encontradas en el software. Se sugiere la herramienta Mantis BT.
- Monitor de recursos de hardware: obtener información sobre el desempeño del software y el hardware al momento de realizar las pruebas de rendimiento.
- Generador de pruebas de carga: diseñar y aplicar escenarios de pruebas de carga de datos, como ejemplo conexiones a la base de datos, peticiones a los servidores, entre otras. Se sugiere la herramienta Apache Jmeter.
- Gestor de contenidos para la web: difundir en internet los avances del proyecto y captar el interés de parte de potenciales usuarios. Se sugieren las herramientas Drupal y Wordpress.
- Gestor de máquinas virtuales: implementar una infraestructura de desarrollo y de pruebas empleando técnicas de virtualización, así como implementar infraestructuras con varios sistemas operativos. Se sugiere la herramienta VirtualBox.

En la Tabla VIII se muestra el énfasis que se emplea en cada herramienta según el trimestre del Trayecto III.

V. Estrategias, Tareas y Recursos

A continuación se presentan un conjunto de estrategias, tareas y recursos que pueden ser empleados por los profesores de la UCPST para el desarrollo de sus clases, estos aspectos son referenciales y deberán ser evaluados por cada profesor según las características de cada PST y su contexto académico.

Tabla VIII: Uso de las Herramientas en cada Trimestre

Generador de diagramas UML	X	X	X
Diseñador de prototipos	X	X	
Planificador de Proyectos	X	X	X
Documentador de código fuente	X	X	X
Gestor de versiones de software	X	X	X
Gestor de incidencias de software		X	X
Monitor de recursos de hardware		X	X
Generador de casos de pruebas de carga		X	X
Gestor de contenidos para la web			X
Gestor de máquinas virtuales	X	X	X

A. *Estrategias*

Se sugieren un conjunto de estrategias a lo largo los trimestres que guíen el desarrollo de las actividades, en la Tabla IX se describen estas estrategias.

Tabla IX: Estrategias Aplicadas en la UCPST del Trayecto III

Trimestre	Estrategias
Trimestre I	Investigar la comunidad para conocerla, describirla y detectar necesidades. Investigar sobre el campo de acción del proyecto. Investigar sobre la metodología de investigación y sobre la metodología de desarrollo de Software.
Trimestre II	Aplicar un enfoque modular de la arquitectura del software. Aplicar técnicas para pruebas de software.
Trimestre III	Socializar la experiencia mediante la publicación en revistas de acceso público, jornadas de divulgación, congresos e Internet.

Común a todos los trimestres se consideran las siguientes estrategias:

- Asignar roles y responsabilidades en función de las fortalezas del equipo de proyecto.
- Realizar reuniones periódicas con los involucrados.
- Participar en las asesorías con los profesores del resto de las UC del Trayecto III.
- Realizar reuniones de asesoría con especialistas de plataforma, bases de datos, desarrollo de software para el diseño de la Infraestructura Tecnológica.
- Aplicar un enfoque iterativo e incremental para la implementación de las funcionalidades del sistema.
- Asistir a jornadas técnicas y académicos para conocer sobre temas de actualidad.
- Revisar los contenidos y lecturas complementarias indicadas por el profesor de la UCPST.

B. *Tareas*

El PST se evalúa al finalizar el Trayecto III, sin embargo, para facilitar el seguimiento del proyecto a lo largo los trimestres se describen un conjunto de tareas de referencia, en la Tabla X se describen estas tareas.

TABLA X: TAREAS EN LA UCPST DEL TRAYECTO III

Trimestre	Tareas
Trimestre I	Conformar el equipo de trabajo. Abordar al menos una comunidad e identificar las necesidades y oportunidades de mejora aplicando un diagnóstico social - participativo. Formular al menos una propuesta de proyecto, elaborar el planteamiento de la situación problemática y sus antecedentes. Definir los objetivos, alcance, factibilidad del proyecto. Identificar los riesgos. Definir las bases de teóricas según el campo de acción. Definir los requisitos funcionales y no funcionales. Realizar el análisis y diseño del sistema (diseñar prototipos no funcionales, modelo de persistencia, modelo de lógico). Configurar la plataforma de desarrollo. Elaborar la arquitectura candidata (implementar los casos de uso críticos relacionados a la arquitectura).
Trimestre II	Refinar la arquitectura del sistema. Implementar las funcionalidades definidas en el alcance. Realizar las pruebas funcionales y de rendimiento.
Trimestre III	Planificar la implantación del sistema. Elaborar la documentación para los usuarios. Configurar la plataforma de producción. Implantar el sistema. Obtener los indicadores de mejora de procesos. Divulgar los resultados del proyecto. Hacer de acceso público el código fuente del sistema con algún licenciamiento que permita su reutilización. Presentar los resultados ante las autoridades académicas y la comunidad beneficiada.

Común a todos los trimestres se consideran las siguientes tareas:

- Mitigar los riesgos identificados.
- Hacer seguimiento al avance del proyecto.
- Actualizar y registrar los cambios de todos los artefactos generados.
- Realizar una presentación de avances.

En la Tabla XI indican un conjunto de contenidos, herramientas y plantillas que sirven de apoyo al desarrollo de las tareas indicadas en cada trimestre.

TABLA XI. RECURSOS EMPLEADOS EN LA UCPST DEL TRAYECTO III

Trimestre	Recursos
Trimestre I	Diseñador de prototipos.
Trimestre II	Gestor de máquinas virtuales. Documentador de código fuente. Gestor de versiones de software. Gestor de incidencias de software. Monitor de recursos de hardware.
Trimestre III	Gestor de máquinas virtuales. Generador de casos de pruebas de carga. Gestor de contenidos.

Común a todos los trimestres se consideran las siguientes herramientas y recursos:

- Generador de diagramas UML.
- Plantillas de documentos y/ artefactos.
- Bibliografía especializada.
- Contenidos generados por el Profesor.

Finalmente, en la Fig. 4 se observa el engranaje de los roles principales, documentos y herramientas en los trimestres del Trayecto III, de esta forma se establece un modelo de referencia para el abordaje de los PST con un enfoque donde se integra las distintas disciplinas de la Ingeniería de Software con los aportes de las áreas de conocimiento de todas las UC que conforman el Trayecto III del PNFI.

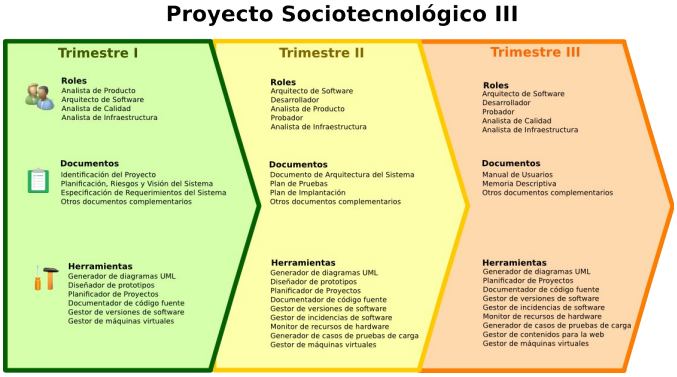

Fig. 4. Trimestres, roles, documentos y herramientas

VI. CONCLUSIONES

En este artículo se define una guía de referencia para el abordaje de los PST del Trayecto III del PNFI, este modelo detalla un conjunto de actividades referenciales para guiar el desarrollo del PST, incluye un conjunto de plantillas que permiten organizar el proceso de documentación, indica algunas herramientas de apoyo a la gestión de proyecto y plantea un esquema para vincular las Unidades Curriculares del Trayecto III con la UCPST. Este modelo fue aplicado a dos secciones de la UCPST del Trayecto III del PNFI entre los años 2011 y 2012 en el CUC, con un total de 56 estudiantes y 17 equipos de PST, los resultados obtenidos con la aplicación de estas orientaciones fueron los siguientes:

- El 70% de los equipos de PST (12) finalizaron la UCPST y fue aprobado el PST por parte del Comité Evaluador.
- El 24% de los equipos de PST (4) finalizaron la UCPST pero el PST no fue aprobado por el Comité Evaluador.
- El 6% de los equipos de PST (1) se retiraron del curso antes de terminar la UCPST.

En la Fig. 5 se muestra la relación entre los PST aprobados, no aprobados y los no finalizados.

El PNFI propone un cambio de paradigma en el contexto universitario venezolano, por ello es necesario explicar detalladamente a los estudiantes y profesores el objetivo de la UCPST, el método de evaluación, el perfil técnico del PST y demás aspectos metodológicos. Así mismo se debe explicar el carácter flexible de los enfoques aplicados para abordar los

PST los cuales permiten a los profesores seleccionar aquellos métodos que mejor se adapten a cada contexto de proyecto. Las orientaciones expuestas en este artículo deben ser evaluadas y ajustadas según el contexto de cada PST, tomando en consideración la naturaleza de cada proyecto.

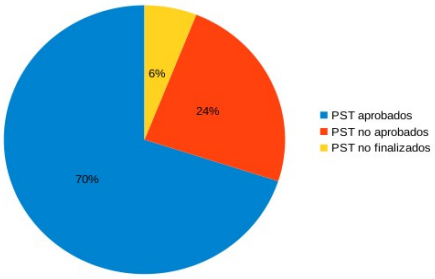

Fig. 5. Resultados finales de los PST durante el período de estudio

También es importante recalcar la función del profesor de la UCPST como director de los proyectos, con el propósito de evaluar y facilitar la incorporación de los aportes provenientes de las demás Unidades Curriculares. En este sentido, las principales dificultades se encontraron en la alineación de criterios entre la UCPST y el resto de las Unidades Curriculares del Trayecto III, fue necesario esfuerzos importantes para mantener el enfoque de cohesión de las actividades de la UCPST, se realizaron múltiples reuniones entre los profesores que atienden el resto de las Unidades Curriculares, en algunos casos no se alcanzó la receptividad esperada, sin duda este es uno de los desafíos más importantes para lograr implantar el modelo propuesto en este artículo.

Finalmente, para futuras experiencias en la UCPST se planteará conformar un repositorio de acceso público de Proyectos, empleando algún licenciamiento que permita su reutilización con alguna licencia compatible con la filosofía del Software Libre.

REFERENCIAS

[1] MPPEU. Programa Nacional de Formación en Informática. Ministerio del Poder Popular para la Educación Universitaria, Caracas, Venezuela. 2008.

[2] E. Paikari. "Model-based planning and control for balancing functionality and quality in iterative software construction" (Order No. NS23817). 2013. Available from ProQuest Dissertations & Theses Full Text: The Sciences and Engineering Collection. (1518961409). Retrieved from http://search.proquest.com/docview/1518961409?accountid=50437 .

[3] J. Alvarez, S. Solé . "Experiencia en aseguramiento de calidad en el ámbito de software libre". Revista Electrónica Conocimiento Libre y Licenciamiento. No 8. Año 4. 2014. ISSN: 2244-7423 . Disponible en: http://convite.cenditel.gob.ve/files/2015/01/ExperienciaRevistaCLIC_n8_1.pdf.

[4] M. Stefanovic. "Advanced software engineering models for quality improvement" (Order No. MQ60501). Available from ProQuest Dissertations & Theses Full Text: The Sciences and Engineering Collection. (304745151). 2001. Retrieved from http://search.proquest.com/docview/304745151?accountid=50437

[5] K. Domínguez, M. Pérez, L. Mendoza, K. Santos, C. Marrero, H. Rivero. "Gestión de proyectos de desarrollo de software libre con un enfoque de calidad". XII Seminario Latino-Iberoamericano de Gestión Tecnológica, 2007. Disponible en: http://www.altec-dl.org/index.php/altec/article/view/1456

[6] E. Ichu. "Quality assurance in software development: An exploratory investigation in software project failures and business performance". (Order No. 3404630). Available from ProQuest Dissertations & Theses Full Text: The Sciences and Engineering Collection. (518658207). 2010 Retrieved from http://search.proquest.com/docview/518658207?accountid=50437

[7] *Ley de Infogobierno, Gaceta Oficial de la República Bolivariana de Venezuela No. 40.274.* Caracas, 2013. Disponible en: http://www.cnti.gob.ve/images/stories/documentos_pdf/leydeinfogob.pdf

[8] *Decreto 3.390, Gaceta Oficial de la República Bolivariana de Venezuela No. 38.095.* 2004. Disponible en: http://www.cnti.gob.ve/images/stories/documentos_pdf/decreto3390softwarelibre.pdf

[9] C. Marrero, K. Santos. "Metodología de la red nacional de integración y dsarrollo de software libre (MeRinde). Guía detallada", Centro Nacional de Tecnologías de Información. 2007. Disponible en: http://www.merinde.net/

[10] L. Sánchez. *Diagnóstico social participativo*. Caracas, Venezuela. Fondo Editorial de la Fundación Escuela de Gerencia Social del Ministerio de Planificación y Desarrollo. 2005 . ISBN: 980-6424-52-2.

[11] F. Gharehchopogh, A. Talebi, I. Maleki. "Analysis of use case points models for software cost estimation". International Journal Of Academic Research, 6(3), 118-124. 2014, doi:10.7813/2075-4124.2014/6-3/A.16

[12] R. Rezaei, B. Arasteh. "A new method in software cost estimation with tabu search". International Journal Of Academic Research, 6(5), 41-45. 2014, doi:10.7813/2075-4124.2014/6-5/A.7

[13] S. Ebrahimi, A. Jafarian. "A new model for software cost estimation by using artificial bee colony algorithm". International Journal Of Academic Research, 6(4), 178-184. 2014 doi:10.7813/2075-4124.2014/6-4/A.24

[14] M. André, M. Baldoquín. "Un sistema de soporte a la decisión para la asignación de recursos humanos a equipos de proyectos de software". Revista Investigación Operacional Vol. 31, No. 1, 61-69 2010. Disponible en: http://rev-inv-ope.univ-paris1.fr/files/31110/31110-04.pdf

[15] J. Llorens. *Gerencia de Proyectos de Tecnología de Información. Cómo organizar, planificar, estimar, evaluar y controlar exitosamente proyectos de tecnología de información*. Caracas, Venezuela. Editorial CECSA. 2005. ISBN: 980-388-186-8.

[16] R. Pressman. *Ingeniería del Software, un enfoque práctico*. 6ª ed. Distrito Federal, México. Editorial Mc Graw Hill. 2005 . ISBN: 970-10-5473-3.