







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 (SCORM<sup>3</sup> que agrupa LOM<sup>4</sup> e IMS<sup>5</sup>, 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]): *abierto y*

*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 2.0<sup>6</sup> de la AR obtenida en [8].

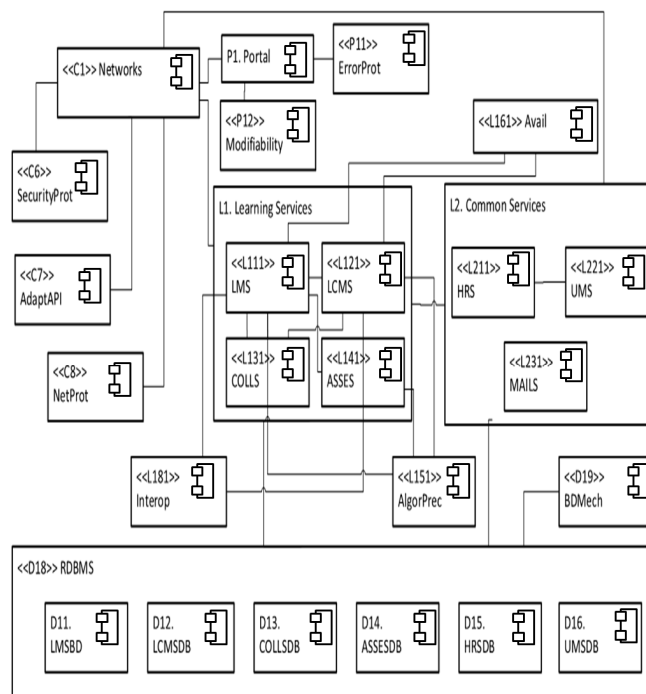


Figura 2. AR para Sistemas E-Learning [8]

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

<sup>3</sup> Sharable Content Object Reference Model

<sup>4</sup> Learning Object Metadata, XML

<sup>5</sup> Global Learning Consortium, implementa LOM

<sup>6</sup> 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 LAMP<sup>7</sup>.

**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, L1RC1
<<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

<sup>7</sup> 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 proveer los rápidos avances tecnológicos; esto le permitirá incluir componentes opcionales 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, <<L111>> *LMS*, <<L121>> *LCMS*, <<L131>> *COLLS*, <<L141>> *ASSES*, <<L211>> *HRS*, <<L221>> *UMS*, <<L231>> *MAILS*; <<L161>> *Avail* agrupa *L16*, *LMS copy* y *L17*. *LCMS copy*, <<L151>> *AlgorPrec* agrupa algoritmos de cálculo, <<L181>> *Interop* agrupa *L18*, *AICC* y *L19*. *SCROM*; finalmente <<P111>> *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.

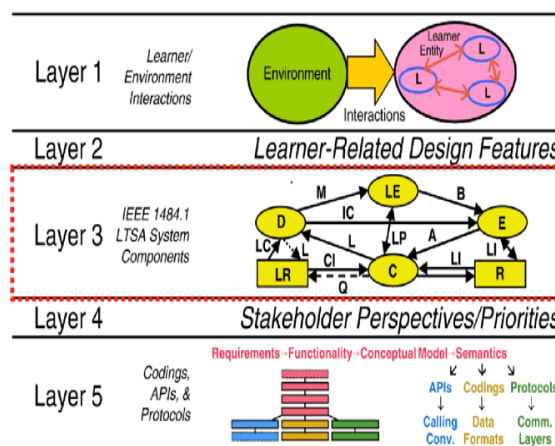


Figura 3. LTSA: IEEE P1484-1 Standard [12]

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 ()		
			Requiere	Proporciona	Restricciones
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 ()		Restricciones
			Requiere/Proporciona	indican las relaciones entre los componentes	
			Requiere	Componentes que Proporcionan	
Modifiability>>	Sí (*)	Provee separación entre UI y la capa de proceso	- modificabilidad	Plataforma PHP - MVC <sup>8</sup>	
L1. Learning Services	No	Agrupar 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-doceo	- 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;
<<L121. LCMS>>	Sí (*)	Sistemas de autoría: creación de contenidos; permite a los autores registrar, ensamblar, administrar y publicar/exportar contenidos para ser entregados via 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: <i>Evolution, ForceTen, Eduslide</i>  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	- 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>> L16. LMS copy	Sí	Garantiza la disponibilidad en caso de falla	Subsistema	Componentes	Opcional
L17. LCMS copy	Sí	Garantiza la disponibilidad en caso de falla	Subsistema	- Espejo, espejo con replicación	Opcional
<<L181. Interop>> L18. AICC	Sí	Garantiza importación/exportación	API	API	Obligatorio
L19. SCROM	Sí	Garantiza importación/exportación	API	API	Obligatorio

<sup>8</sup> 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 ()		Restricciones
			Requiere/Proporciona indican las relaciones entre los componentes	Requiere Componentes que Proporcionan	
L2. Common Services	No	Agrupar 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. MAILS>>	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	Agrupar 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. Satellite	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	obligatoria
C2. ODBC	Sí	Garantiza portabilidad	API	Mecanismo	obligatoria
C3. JDBC	Sí (*)	Garantizan propiedades de la comunicación variantes: múltiples, dependen del mercado	Protocolos	Mecanismos, servicios Web	obligatorias
<<C8. Network 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
<<C6. Security protocols>>	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	indican las relaciones entre los componentes Requiere Componentes que Proporcionan	
		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	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;
D19. DB mechanisms	Sí (*)	Garantizan propiedades de los RDBMS variantes: específicas para cada RDBMS	RDBMS	Componentes	

(\*) 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, MAILS: 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	HRS, MAILS DBMS	Supervisor Administrador	ASSESDB LMSDB, LCMSDB, COLLSDB
R: Register	DBMS	Administrador	UMSDB, HRSDB

### 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 componente 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*

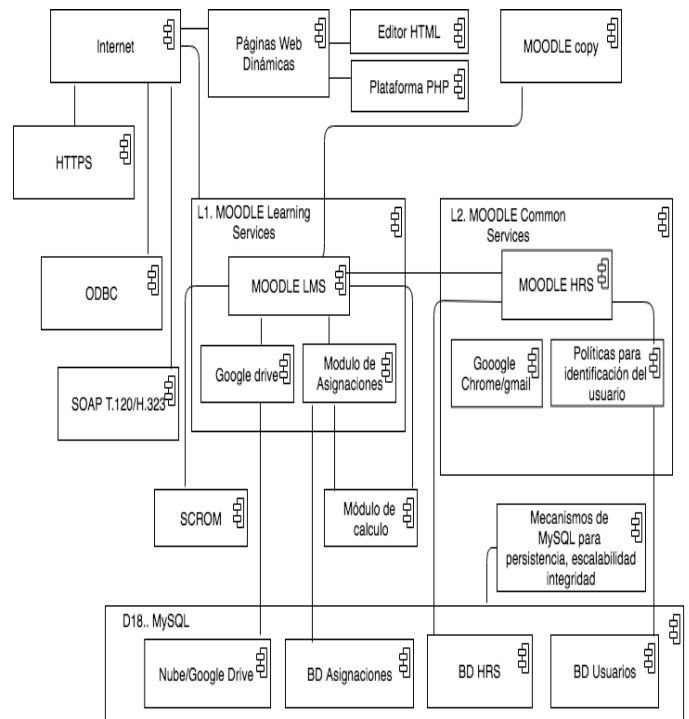
• *Capa de Presentación:*

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

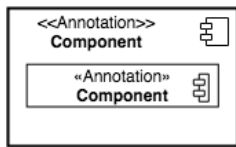
• *Capa de Datos:*

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 <<C1. 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 línea 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](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](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.
-