



Universidad de las Ciencias Informáticas

Facultad 4

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO INGENIERO EN CIENCIAS INFORMÁTICAS

Diseño e implementación de una API de servicios web para las funcionalidades de gestión de Moodle 1.8.x y 1.9.x.

Autores:

Arleny Acanda Rodríguez
Humberto Cabezas Martínez.

Tutor:

Ing. Ana Delia González Ricardo.

Co-tutor:

MSc. Ismael Armando Nodarse Mora

La Habana, 2011.

“Año 53 de la Revolución”

Declaración de autoría

Declaramos ser los autores del presente trabajo de diploma y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores

Arleny Acanda Rodríguez

Humberto Cabezas Martínez

Firma del autor.

Firma del autor.

Tutores

Ing. Ana Delia González Ricardo

Mcs. Ismael Armando Nordarse Mora

Firma del tutor.

Firma del Cotutor.



*La gloria del mundo es transitoria, y no es
ella la que nos da la dimensión de nuestra
vida, sino la elección que hacemos de seguir
nuestra leyenda personal, tener fe en
nuestras utopías y luchar por nuestros
sueños.*

Paulo Coelho.

Ing. Ana Delia González Ricardo: Graduada en Ingeniería en Ciencias Informáticas en el año 2010. Pertenece al Departamento de Producción de Herramientas Educativas del Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas. Desempeña los roles de: Arquitecta de Información y Administrador de la Calidad en el proyecto Desarrollo de la plataforma Moodle.

Correo: adgonzalez@uci.cu.

M.Sc. Ismael Armando Nodarse Mora: Graduado en Ingeniería en Ciencias Informáticas en el año 2007. Entre sus líneas de investigación está el desarrollo de sistemas informáticos que aplican técnicas de Inteligencia Artificial (IA) a problemas del mundo real y aplicaciones de las Redes Inalámbricas de Sensores y Actuadores. Lidera el Grupo de Investigación y Aplicaciones de la Inteligencia Artificial de la Facultad 4. Cursó la Maestría en Ingeniería del Software e IA por la Universidad de Málaga, alcanzando el grado científico de Máster en Ciencias por la investigación: “Utilización de Redes Inalámbricas de Sensores y Actuadores en Sistemas de Control de Edificios”.

Correo: inodarse@uci.cu



Agradecerco a todas aquellas personas que permitieron el desarrollo de este trabajo, sin el apoyo de ellos el resultado logrado no hubiese sido posible. A Ana Delia, gracias por tu dedicación y paciencia, a Jesús a Juentis y a Yolanda.

A todos muchas gracias.

Arleny.

Quiero agradecer a todas las personas que de una forma u otra me han brindado su apoyo en el desarrollo de este trabajo. En este caso se encuentran los profesores Jesús Juentis Luis Enrique, Ismael y la profesora Ana Delia que no por ser la última deja de ser importante.

Muchas gracias a todos por su dedicación.

Humberto.

A mi madre, por entregarme su apoyo, por la confianza depositada, su amor y dedicación.

A Miru por tu paciencia, gracias por darme la oportunidad de haberte conocido.

A Ray por los años de buena amistad.

A Rey, a Daylen y a Gerardo gracias por ser tan especial.

Arleny.

A mis padres por haberme apoyado en la carrera, gracias por su apoyo y comprensión.

Humberto.



Resumen

La plataforma de teleformación Moodle surge para dar consistencia a la educación a distancia y como herramienta educativa tiene un gran impacto en el mundo del e-learning. Esta permite la gestión de cursos en línea, posibilitando a sus usuarios obtener, utilizar o compartir materiales didácticos. Es por ello que la Universidad de las Ciencias Informáticas en apoyo al proceso de enseñanza-aprendizaje utiliza dos personalizaciones de esta plataforma basadas en las versiones 1.8.6 y 1.9.4. La principal deficiencia detectada en estas versiones se encuentra la imposibilidad de que sus funcionalidades puedan ser consumidas desde otras herramientas. Esto ocurre debido a la ausencia de estándares y protocolos abiertos que faciliten la interoperabilidad entre aplicaciones. A partir de la problemática existente, se elaboró una Interfaz de Programación de Aplicaciones (API) de servicios web para las funcionalidades de gestión más importantes de Moodle, de las versiones 1.8.x y 1.9.x. Para su implementación se utiliza la biblioteca de componentes NuSoap con el objetivo de implementar las funcionalidades sobre servicios web. El resultado de la investigación brinda solución a las deficiencias detectadas y permitirá el desarrollo de comunidades colaborativas de aprendizaje que mejoran el flujo de comunicación entre los usuarios.

Palabras clave:

API, e-learning, interoperabilidad, Moodle, servicios web.

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentación teórica	7
Introducción	7
1. 1 E-learning, educación a distancia.....	7
1.2 LMS, Sistema de Administración del Aprendizaje	7
1.2.1 LMS Moodle	8
1.3 API, Interfaz de Programación de Aplicaciones.....	19
1.3.1 Diferentes tipos de API.....	19
1.4 Arquitectura Orientada a Servicios.....	20
1.5 Servicios web.....	21
1.5.1 Beneficios de integrar sistemas mediante Servicios Web.....	22
1.5.2 Tecnologías de Servicios Web	23
1.5.3 Seguridad en los servicios web	28
1.6 Metodologías de desarrollo de software.....	30
1.7 Tecnologías y herramientas de desarrollo del software.....	31
1.7.1 Lenguaje de Programación	32
1.7.2 Lenguaje de Modelado.....	33
1.7.3 Herramientas de Desarrollo del Software	34
1.7.4 Herramientas CASE para el modelado.....	35
Conclusiones parciales	37
Capítulo 2: Diseño del sistema.....	38
Introducción	38
2.1 Propuesta del sistema.....	38
2.2 Modelo de Dominio	39
2.2.1 Representación del Diagrama de Modelo de Dominio.....	39
2.2.2 Definición de las clases del Modelo de Dominio.....	39

2.3 Especificación de Requisitos.....	40
2.3.1 Requisitos Funcionales	41
2.3.2 Requisitos no Funcionales	42
2.4 Modelo de Casos de Uso del Sistema.....	43
2.4.1 Actores del Sistema	43
2.4.2 Diagrama Casos de Uso del Sistema.....	44
2.4.3 Descripción Textual de Casos de Uso del Sistema	44
2.5 Modelo de Análisis.....	47
2.5.1 Diagrama de Clases del Análisis.....	47
2.5.2 Diagramas de Interacción.....	49
2.6 Modelo del Diseño	51
2.6.1 Diagramas de Clases del Diseño	51
2.7 Estructura de la base de datos.....	53
Conclusiones parciales	55
Capítulo 3: Implementación y prueba del sistema	56
Introducción	56
3.1 Diagrama de Despliegue.....	56
3.2 Diagrama de Componentes	57
3.3 Tipos de pruebas	58
3.3.1 Método de Prueba.....	58
3.3.2 Resultados de las pruebas.....	60
Conclusiones parciales	61
Conclusiones Generales.....	62
Recomendaciones	63
Referencias Bibliográficas.....	64
Bibliografía Consultada.....	69
Glosario de términos.....	70

Introducción

Con el uso de las tecnologías educativas, programas educativos, contenidos digitales y aplicaciones basadas en las Tecnologías de la Información y las Comunicaciones (TICs), se crean nuevos entornos de aprendizaje que posibilitan la comunicación, acción e interacción entre alumnos y profesores. Estos entornos constituyen una opción más dentro de la modalidad de educación a distancia, conocida como e-learning.

El e-learning hace uso de las tecnologías y de los servicios de Internet para llevar a cabo procesos de formación y capacitación del personal, contribuyendo de esta manera, en la mejora del proceso de enseñanza-aprendizaje a través de la red. Constituye además la entrega de material educativo a través de cualquier medio electrónico, incluyendo Intranets, Extranets, audio, vídeo, entre otros medios. (1)

Dentro del e-learning, para desarrollar un programa de formación, se hace uso de plataformas que permiten la interacción y comunicación entre usuarios. Un ejemplo de estas herramientas son los Sistemas de Administración de Contenidos de Aprendizaje (del inglés *Learning Content Management Systems* (LCMS)), utilizados para crear y manejar el contenido educativo y los Sistemas de Administración del Aprendizaje (del inglés *Learning Management Systems* (LMS)), sistema enfocado en el área educativa que permite la gestión de los contenidos y de los usuarios que interactúan dentro de ella. Uno de los LMS más difundidos a nivel internacional por las características que posee, es la plataforma Entorno de Aprendizaje Dinámico Modular Orientado a Objeto (Moodle, por sus siglas en inglés). (1)

La plataforma Moodle es una aplicación web, diseñada para dar soporte a la educación social constructivista¹. (2). Permite la creación de cursos en línea y llevar el control de los contenidos y los usuarios que interactúan dentro de la misma. Para el desarrollo de esta, cuenta con una comunidad oficial en www.moodle.org, donde participan desarrolladores, profesores y estudiantes de, aproximadamente, doscientos doce países. Entre sus objetivos, se encuentra el desarrollo colaborativo de módulos y personalizaciones de esta plataforma.

La plataforma posee un entorno modular, pues consta con módulos de actividades y recursos que permiten construir comunidades colaborativas de aprendizaje acerca de una materia. Dispone además, de

¹Creación colaborativa de artefactos compartidos de un grupo social para otro.

funcionalidades de gestión que permiten el trabajo con los usuarios y los cursos, brindando información relacionada con las actividades que se realicen dentro de la misma.

A nivel internacional, se ha intensificado el uso de este tipo de plataformas para facilitar la formación de los estudiantes mediante la impartición de cursos a distancia. En Cuba, por ejemplo, comienzan a utilizarse desde el año 2000. Microcampus² fue una de las primeras plataformas creadas para favorecer el proceso de enseñanza-aprendizaje a distancia, surgiendo otras a partir de esta, como es el caso de la herramienta Aprendist³ del Instituto Politécnico “José Antonio Echevarría”. En la Universidad Central de Las Villas (UCLV) se comienza a utilizar SEPAD⁴ y EduDist⁵ de la Universidad de Holguín.

La Universidad de las Ciencias Informáticas (UCI), como parte de su proceso enseñanza-aprendizaje, en el año 2005, comienza a utilizar la plataforma Moodle mediante personalizaciones realizadas a la misma. Ejemplo de ellas son: el Entorno Virtual del Aprendizaje (EVA), para enseñanza de pregrado y el Entorno Virtual de Postgrado; ambos basados en las versiones 1.8.6 y 19.4 respectivamente. Otras de las herramientas utilizadas son: el Repositorio de Objetos de Aprendizaje (RHODA), plataforma utilizada para el almacenamiento de los Objetos de Aprendizaje (OA), los cuales son un recurso digital reutilizable, interoperable, recuperable y que facilita el aprendizaje y, la herramienta de autor (CRODA) para la creación de los OA.

La tendencia que existe en la UCI reside en la integración de las herramientas mencionadas con anterioridad. Esto facilitaría la comunicación y el intercambio de información entre ellas y el desarrollo de comunidades colaborativas. En una entrevista realizada a especialistas involucrados en este proceso se identificó que para alcanzar este objetivo, era necesario, tener en cuenta los siguientes aspectos: identificar qué funcionalidades de la plataforma Moodle, de acuerdo a la información que estas brindan, pueden ser consumidas desde otras herramientas; e identificar si existen implementaciones en las

²Plataforma creada por un grupo de trabajo de la Universidad de Alicante, España.

³Plataforma de educación a distancia, basada en el proyecto Mundicampus.

⁴Sistema de Enseñanza Personalizada a Distancia.

⁵Plataforma de Aprendizaje a Distancia de la Universidad de Holguín.

versiones de Moodle utilizadas en la universidad donde se utilicen protocolos y estándares abiertos que faciliten la interoperabilidad entre aplicaciones.

Con el resultado de la entrevista se determinó que el EVA es la plataforma a la cual más usuarios acceden debido al contenido que gestiona. Las funcionalidades de gestión que posee son las más consultadas porque facilitan el trabajo con los usuarios y cursos de la plataforma. También proporcionan información sobre las actividades que se realizan dentro de la aplicación.

Por otro lado, se comprobó, que las versiones sobre las cuales están implementadas las personalizaciones, no fueron desarrolladas teniendo en cuenta el uso de protocolos y estándares que permitan que sus funcionalidades puedan ser consumidas desde otras herramientas; característica que se ve solventada en la nueva versión de Moodle 2.0.

La versión 2.0 de Moodle fue liberada por la comunidad de Moodle internacional en noviembre del año 2010. Tiene como característica principal la implementación de varias de sus funcionalidades haciendo uso de estándares abiertos. Para el logro de una interacción entre las plataformas educativas existentes en la UCI le sería de mucha utilidad una migración a esta nueva versión. Sin embargo, resulta muy complejo, pues existen algunos factores que dificultan el proceso, los cuales se describen a continuación.

Dentro de las nuevas características que posee esta versión se encuentran: el núcleo de Moodle se ha reescrito prácticamente desde cero y la actualización de cada uno de los módulos y actividades incluyó cambios significativos. Esto, unido a que la arquitectura que ha de soportar la nueva versión debe estar considerablemente actualizada (3). Estas son las razones que dificultan una posible migración.

Atendiendo a los aspectos valorados y a las necesidades de los usuarios se plantea el siguiente **problema de investigación** ¿Cómo facilitar la integración de las funcionalidades de gestión de Moodle 1.8.x y 1.9.x con otras herramientas para facilitar el trabajo colaborativo?

Se tiene como **objeto de estudio** la interacción entre herramientas que posibiliten el trabajo colaborativo.

El **objetivo general** es desarrollar una Interfaz de Programación de Aplicaciones (API) para las funcionalidades de gestión de Moodle 1.8.x y 1.9.x sobre Servicios Web (SW) que favorezca el trabajo colaborativo en la UCI.

El **campo de acción** lo constituye: integración de las funcionalidades de gestión de Moodle 1.8.x y 1.9.x con otras herramientas.

Se tiene como **idea a defender**, que con la implementación de una API de SW para las funcionalidades de gestión de Moodle 1.8.x y 1.9.x se facilitaría su publicación y consumo desde otras herramientas, favoreciendo el trabajo colaborativo.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

1. Diseñar una API que permita que las funcionalidades de gestión de Moodle 1.8.x y 1.9.x puedan ser utilizadas a través de SW.
2. Implementar una API para publicar las funcionalidades de gestión de Moodle sobre SW.
3. Realizar pruebas al API para publicar las funcionalidades de gestión de Moodle sobre SW.

Para darle cumplimiento a los objetivos específicos se definen las siguientes **tareas de investigación**:

- Análisis de los conceptos y tecnologías más utilizadas para el desarrollo de SW.
- Análisis de la arquitectura y la base de datos de Moodle.
- Análisis de las funcionalidades de gestión, el manejo de roles y permisos en Moodle. Estudio de cómo se puede adaptar la API de Moodle 2.0 para la publicación de las funcionalidades de gestión sobre SW en las versiones 1.8.x y 1.9.x de Moodle.
- Análisis de la implementación de la API de Moodle 2.0 para la publicación de las funcionalidades de Moodle sobre SW.
- Selección de la metodología y las herramientas de desarrollo a utilizar.
- Diseño de una API que permita gestionar las funcionalidades de gestión existentes en la plataforma Moodle sobre SW para que puedan ser consumidas por otras herramientas.
- Implementación de una API para publicar las funcionalidades de gestión de Moodle 1.8.x y 1.9.x sobre SW.

Los métodos científicos que se utilizaron para el desarrollo de la investigación fueron:

Métodos teóricos:

El **Análisis Histórico-Lógico** para el estudio de la evolución y el desarrollo de las plataformas educativas, así como, la evolución de la integración de estas con otras herramientas para propiciar el trabajo colaborativo.

El **Analítico-Sintético** para el análisis de conceptos y elementos bibliográficos relacionados con el tema para darle solución al problema existente.

La **Modelación** para representar de manera práctica las funcionalidades de gestión sobre servicios web.

Métodos empíricos:

La **Observación** para obtener información relacionada con los últimos trabajos que se han realizado para valorar los resultados y utilizarlos en la propuesta a presentar en esta investigación.

La **Entrevista** para conocer la necesidad real de los usuarios en la UCI y poder redactarla con datos convincentes.

Estructura capitular

Capítulo 1: “Fundamentación teórica”; se abordan, de manera general, temas relacionados con el e-learning. Se realiza un estudio de las funcionalidades de gestión de Moodle. Se describen las metodologías y lenguajes que posibilitan el desarrollo de la propuesta. También se analizan los conceptos fundamentales que se ajustan a la investigación.

Capítulo 2: “Diseño del sistema”; en este capítulo se describe la propuesta de sistema. Se especifican los requerimientos funcionales y no funcionales. Se identifican los casos de uso del sistema con sus respectivas descripciones textuales y su relación con los actores del sistema. Se representan los Diagramas de Clases del Análisis y el Diseño, así como, se describe la estructura de la base de datos.

Capítulo 3: “Implementación y prueba del sistema”; apoyándose en los Diagramas de Componentes y Diagrama de Despliegue, se describe la herramienta propuesta desde el enfoque de la programación. Además, se valida la investigación a partir de casos de pruebas realizados a los casos de usos del sistema.

Capítulo 1: Fundamentación teórica

Introducción

En la actualidad, existe una nueva modalidad de enseñanza: el e-learning, que enriquece la educación a distancia mediante el uso de herramientas, dentro de las que se destacan las plataformas educativas. La integración de estas, beneficia la creación de comunidades donde los usuarios comparten sus conocimientos, permitiendo de esta forma, el trabajo colaborativo. Esta integración está basada en la interoperabilidad entre varias herramientas, que se logra mediante uso de SW, utilizando para ello, protocolos y estándares abiertos.

1.1 E-learning, educación a distancia

La modalidad e-learning es una muestra de cómo ha evolucionado el proceso de enseñanza-aprendizaje en correspondencia con el avance de las nuevas tecnologías, pues ha brindado la posibilidad de favorecer al aprendizaje a distancia a través de ordenadores con el uso de plataformas educativas.

Con el uso del e-learning desaparecen las barreras espacio-tiempo, permite una formación flexible, los contenidos se encuentran actualizados y existe una comunicación constante entre los participantes de las plataformas educativas. Estas, y otras características las posee la nueva modalidad formativa que se define en contraposición con las de la enseñanza tradicional. (4)

Bernardéz lo define como: *“Todas aquellas metodologías, estrategias o sistemas de aprendizaje que emplean tecnología digital y/o comunicación mediada por ordenadores para producir, transmitir, distribuir y organizar conocimiento entre individuos, comunidades y organizaciones”*. (5)

Dentro de las herramientas que propician la aplicación de esta modalidad, se encuentran los denominados LMS, herramienta fundamental para la creación, gestión y distribución de cursos a través de la red.

1.2 LMS, Sistema de Administración del Aprendizaje

Un LMS, también conocido como plataforma de aprendizaje, es un software que automatiza la administración de acciones de formación: gestión de usuarios, gestión y control de cursos, gestión de los servicios de comunicación, etc. (6) Facilita la gestión de cursos, utiliza herramientas de comunicación, le

permite al estudiante el aprendizaje colaborativo⁶, donde el alumno es el protagonista del aprendizaje mediante de la interacción con los demás y de sus propias experiencias. (7)

Los alumnos interactúan a través de una interfaz que posee la plataforma, que le brinda la opción de obtener el contenido mediante los cursos, realizar las actividades programadas y mantener una comunicación entre profesores y alumnos, así como, darle seguimiento a su propio progreso con datos estadísticos y calificaciones.

Dentro de los LMS más difundidos, se encuentra Moodle, una plataforma educativa que es utilizada a nivel internacional por las ventajas que le brinda a los educandos.

1.2.1 LMS Moodle

Moodle es una plataforma que permite la creación de cursos, favoreciendo el constructivismo⁷. Su principal objetivo está basado en contribuir a la mejora de los métodos de enseñanza mediante la educación a distancia y permite la gestión y organización de los recursos educativos para que puedan ser consultados por los alumnos. (8)

Una de las ventajas que posee esta plataforma es que puede soportar textos, imágenes, videos, entre otros. Permite fomentar el auto-aprendizaje y el aprendizaje colaborativo, realizar exámenes online y la evaluación de las tareas de los estudiantes. Cuenta, además, con una gran variedad de módulos para la creación de cursos, incluyendo los de recursos y actividades.

En la actualidad, está traducido a 70 idiomas e incluye más de 15.000 sitios registrados en todo el mundo, se distribuye de forma gratuita bajo la licencia GNU GPL⁸. Está escrito en PHP (del inglés Preprocessor Hypertext Pages), lenguaje de programación muy utilizado para la creación de contenidos web. Soporta varios tipos de bases de datos, especialmente MYSQL⁹ y PostgreSQL¹⁰. Asegura un alto rendimiento

⁶ Interacción entre alumnos y un docente online utilizando la tecnología para comunicarse a distancia.

⁷ Construcción de nuevos conocimientos a medida que la persona interactúa con su entorno.

⁸ Licencia orientada para proteger la libre modificación, distribución y uso del software.

⁹ Sistema de Gestión de Base de Datos.

combinando una buena elección del hardware, sistema operativo y sistema de bases de datos. (2)

Dentro de las principales características que posee Moodle se pueden observar las siguientes (9):

- **La alta disponibilidad:** Debe de ser lo suficientemente robusto para satisfacer las diversas necesidades de miles de estudiantes, administradores, creadores de contenidos y profesores simultáneamente.
- **Escalabilidad:** La infraestructura debe poder ampliarse o escalar para resolver el futuro crecimiento, tanto en términos de volumen de contenidos educativos como del número de estudiantes.
- **Facilidad de uso:** Apoyar un conjunto de servicios automatizados y personalizados, tales como aprender a ritmo individual y perspectivas específicas de aprendizaje, el acceso, la entrega y la presentación de materiales deben ser fáciles de utilizar y muy intuitivos.
- **Interoperabilidad:** Para admitir contenido de diferentes fuentes, y soluciones de equipos de cómputo o programas de diversos proveedores, el LMS debería intercambiar información utilizando estándares abiertos de la industria para implementaciones Web.
- **Estabilidad:** La infraestructura del LMS puede soportar de manera confiable y efectiva una implementación productiva a gran escala las 24 horas del día, los 7 días de la semana.
- **Seguridad:** El LMS puede limitar y controlar selectivamente el acceso de su diversa comunidad de usuarios a los contenidos en línea, recursos y funciones del servidor tanto interna como externamente.

Una de las principales características de Moodle es que posibilita la gestión de cursos y de usuarios, como ya se ha mencionado. Las actividades que se utilizan para realizar acciones sobre ambos componentes se encuentran implementadas en la plataforma dentro de las denominadas funcionalidades de gestión.

¹⁰ Sistema de Gestión de Base de Datos.

1.2.1.1 Funcionalidades de gestión de Moodle

La plataforma Moodle brinda la posibilidad de interactuar con funcionalidades de gestión que permiten el trabajo con los usuarios, los cursos y las matriculaciones de los estudiantes en los diferentes cursos. Posibilita, además, el trabajo con el calendario y las actividades recientes, que aportan informaciones de las actividades que se realizan dentro de la plataforma. A continuación se describen las mismas.

Calendario

El bloque Calendario permite gestionar todas las actividades con fecha límite dentro de la plataforma Moodle, estas actividades constituyen eventos dentro del bloque. Cuando se crea fecha límite para cuestionarios, foros, preguntas rápidas, entre otros, se crean como eventos que se muestran en el Calendario.

Los principales eventos que presenta el bloque Calendario son (10):

- **Sitio:** (evento visible por todos los grupos - creado por el administrador)
- **Curso:** (evento visible por los participantes del curso - creado por el profesor)
- **Grupo:** (evento visible por los miembros del grupo - creado por el profesor)
- **Usuario:** (evento visible por el usuario - creado por el usuario)

Principales funciones que se realizan mediante el Calendario:

- Permite agregar un evento.
- Editar preferencias.
- Visualizar los eventos por grupos, cursos, usuarios, etc.

Para la edición de las preferencias en el Calendario se encuentran modificaciones que son posibles realizar (11):

- **Formato de hora:** Esta preferencia controla como aparecen las horas en el calendario, que puede ser entre los formatos de 12 y 24 horas. Si se elige la opción "por defecto", entonces el formato se seleccionará automáticamente de acuerdo con el idioma que se utilice en el sitio.

- **Primer día de la semana:** Esta preferencia configura la forma en que se muestran todos los calendarios mensuales.
- **Número máximo de eventos próximos:** Esta preferencia establece el máximo número de eventos próximos que serán mostrados en el calendario al mismo tiempo y aparecerán listados en el bloque “Eventos próximos” si está activo.
- **Plazo de comienzo de eventos próximos:** Esta preferencia establece el máximo número de días que faltan para que un evento ocurra y que pueda ser considerado como un evento próximo. Los eventos que comiencen en una fecha por encima de ese valor no serán considerados como próximos.

El uso de este bloque permite la privacidad de determinados eventos como lo pueden ser fechas de exámenes. Posibilita los recordatorios de actividades de enseñanza-aprendizaje. Además, puede ser útil para las convocatorias si tienen plazo límite y permite que los eventos puedan ser vistos por usuarios individuales, sus grupos definidos y por sus cursos.

Gestión de Usuarios

Un usuario en Moodle es un participante que tiene permisos dentro de la plataforma, lo que está dado mediante la asignación de roles. Los principales roles de usuarios que se asignan son los de administrador, profesor y estudiante. Un ejemplo de un listado de usuarios en esta plataforma son los Participantes y los Usuarios en Línea.

Participantes

Esta opción de Participantes en Moodle permite tener una visión de los miembros de un grupo y favorece la comunicación entre los mismos. Proporciona acceso a la lista de usuarios que participan en el curso. En esta lista, cada nombre es un enlace a través del cual se puede obtener información adicional acerca del usuario. (12)

Nos ofrece información sobre:

- Lista de usuarios.
- Mis cursos.

- Mostrar usuarios que han estado inactivos durante un período indicado (días, semanas o meses).

Dentro de las principales funcionalidades de Participantes se encuentran (12):

- Muestra un listado de los participantes de un curso. Primero se muestran los profesores del curso, y a continuación los estudiantes ordenados inicialmente por el último acceso al curso. Los profesores ocultos no aparecen en el listado a la vista de los alumnos o invitados.
- Permite seleccionar el grupo y grado de detalle representado en la lista de participante.
- Ofrece la opción de ordenar los participantes por Nombre, Ciudad, último acceso al curso, etc.
- Muestra de un participante su "Información personal".
- Permite el envío de mensajes instantáneos a varios usuarios seleccionados.

Usuarios en Línea

Otra opción que visualiza un listado de usuarios en Moodle, son los Usuarios en Línea, este bloque muestra los usuarios que se han registrado en el curso actual por un período de tiempo fijado por el administrador del sitio (por defecto son 5 minutos) y brinda la posibilidad de enviar mensajes privados si se selecciona uno de los usuarios que esté conectado. Al mostrar este bloque en la plataforma, los Usuarios en Línea podría ser un indicador importante y útil para abrir una sesión de chat que posibilite la comunicación entre usuarios. (13)

Gestión de Cursos

Una de las funcionalidades que presenta la plataforma Moodle es la gestión de cursos. Los cursos en Moodle tienen un formato social, por temas o semanal. Dentro de las principales funcionalidades del bloque Curso están: mostrar los cursos que estén habilitados en la plataforma en los que un determinado usuario se encuentre matriculado o sea profesor (opción **Mis cursos**) y brinda la posibilidad de acceder a la página principal de otro curso. (14)

Para mostrar un listado de cursos, una opción es, listar todos los cursos que estén disponibles en el sitio. Otra elección es la de buscar cursos, esta permite al usuario la búsqueda de un curso a partir del nombre del mismo. (15)

Actividades Recientes

Las actividades recientes permiten visualizar una lista abreviada de los acontecimientos ocurridos desde el último acceso al curso, incluyendo los nuevos mensajes y los nuevos usuarios. Esta opción puede ser utilizada para tener una visión rápida del desarrollo del trabajo realizado. (16)

La lista de acontecimientos puede contener (16):

- **Nuevos usuarios del curso:** tanto si se trata de profesores como de alumnos.
- **Nuevos elementos del curso:** aparece un vínculo a los nuevos elementos incorporados por el docente al curso (tarea, cuestionario, etc.).
- **Nuevos mensajes en los foros:** también aparece un hipervínculo que lleva directamente al foro y al mensaje en cuestión.

Eventos Próximos

El bloque de Eventos Próximos muestra una lista de los acontecimientos próximos en el calendario, con enlaces al contenido del acontecimiento señalado. El rango de días a incluir en la lista lo determina el administrador del sitio. Los eventos son generados automáticamente por el calendario o por la fecha de finalización de las actividades. También, incluye dos enlaces para ir al calendario y poder agregar nuevos eventos. Al realizar una búsqueda en una fecha, se mostrará la vista para ese día del calendario. Si el título del evento es un enlace, cuando se realiza búsqueda en él, se muestra ese evento. (17)

Matriculaciones

Para las matriculaciones en Moodle, se puede asignar una contraseña de acceso a los cursos para que se inscriban los propios alumnos, o puede ser posible que los estudiantes puedan inscribirse por sí mismos o por otros métodos habilitados por el administrador del sitio Moodle. Las matriculaciones en Moodle se realizan en los cursos por diferentes métodos (18):

- **Matriculación Authorize.net Credit Card Gateway:** El módulo **Authorize.net** permite configurar cursos pagados por comerciantes.
- **Base de datos externa:** Puede utilizar una base de datos externa para controlar las matriculaciones.

- **Matriculación interna:** Esta es la forma de matriculación por defecto, en la cual un estudiante puede matricularse de dos formas: cuando el profesor o administrador lo matricula manualmente o cuando un curso dispone de una contraseña de matriculación.
- **Autenticación LDAP:** Las cuentas de acceso pueden verificarse en un servidor.
- **Paypal:** El módulo Paypal permite entrar en cursos que requieren un pago.
- **IMS Enterprise:** Es un estándar internacional de formato de archivo XML (Extensible Markup Language) que puede utilizarse para especificar matrículas/anulaciones en los cursos, además de información del curso e información del usuario.

Estándares utilizados en las matriculaciones

IMS (Instruction Management Systems) es un consorcio que agrupa a vendedores, productores, implementadores y consumidores de e-learning, y que se enfoca completamente a desarrollar especificaciones en formato XML. Dentro de las especificaciones que han sido liberadas y trabajadas por IMS se encuentran las siguientes: IMS Enterprise Services (IMS-ES) e IMS Learner Information Package (IMS-LIP), estándares que son utilizados en la plataforma Moodle para la matriculación de los estudiantes. A continuación se muestran características especiales de los mismos. (19)

IMS Enterprise Services (IMS-ES): define un formato estándar basado en XML para mantener información acerca de los estudiantes, los cursos y las inscripciones en los cursos para permitir el intercambio de esta información con otras plataformas educativas. Además, es la definición de cómo gestionar el intercambio de información que describe personas, grupos y compañías en el contexto de aprendizaje. La especificación Enterprise Services se basa en los conceptos de (20):

- **Interoperabilidad:** Esencialmente para centrarse en el intercambio de información entre los sistemas de la empresa.
- **Orientada a Servicios:** Esta especificación define el intercambio de información en términos de servicios que se suministran por la colaboración de otros sistemas.
- **Basado en Componentes:** En este, el conjunto de los servicios serán ofrecidos de manera que se pueden combinar para formar una gama de servicios.
- **Múltiples Enlaces:** El modelo de información de Enterprise Services se define utilizando el

Lenguaje Unificado de Modelado (UML). Esto posibilita el mapeo de la información en el modelo a través de diferentes enlaces. Los enlaces son de importancia inmediata para el Lenguaje de Descripción de Servicios Web. (WSDL).

- **Adopción:** Enterprise Services se basan en la especificación original del modelo de datos de la empresa. Si bien hay cambios significativos, en el modelo de datos subyacente, se mantiene la esencia y las estructuras.

IMS Learner Information Package (IMS LIP) es otro de los estándares utilizados para las matriculaciones en Moodle, permite el almacenamiento de información sobre los alumnos para su procesamiento, mantenimiento e interoperabilidad en distintos sistemas de gestión del aprendizaje. Permite su aplicación en distintos contextos, la especificación plantea un formato digital estándar para representar información relativa a todo el proceso de formación del alumno, incluyendo su historial educativo, experiencia profesional, calificaciones y certificados obtenidos, objetivos educativos y habilidades adquiridas. (21) En la Figura 17 que se encuentra en los anexos del trabajo (Anexo 1) se puede observar la estructura de datos básica de un paquete IMS LIP. En el que se debe especificar los siguientes aspectos (21):

- **Identificación (identification):** En este se coloca la biografía y los datos demográficos relevantes para el aprendizaje.
- **Objetivo (goal):** Sobre el aprendizaje, la carrera y otros objetivos y aspiraciones.
- **Títulos, Certificaciones y Licencias (qcl):** En este se sitúan las calificaciones, certificaciones y licencias concedidos por autoridades reconocidas.
- **Actividad (activity):** Cualquier actividad relacionada con el aprendizaje en cualquier estado de finalización. Podría ser auto-reporte. Incluye la educación formal e informal, formación, experiencia laboral, y el servicio civil o militar.
- **Transcripción (transcript):** Un registro que se utiliza para proporcionar un nivel institucional basado en un resumen de los logros académicos. La estructura de este registro puede tomar muchas formas.
- **Interés (interest):** Información que describe los pasatiempos y actividades recreativas.
- **Competencia (competency):** Habilidades y conocimientos adquiridos en el desarrollo cognitivo, afectivo y / o dominios psicomotor.

- **Afiliación (affiliation):** Miembro de las organizaciones profesionales, etc.
- **Accesibilidad (accessibility):** Accesibilidad general de la información de los estudiantes, tal como se define a través de las capacidades del lenguaje, discapacidad y el aprendizaje cognitivo incluyendo preferencias (por ejemplo, cuestiones de estilo de aprendizaje), las preferencias de física (por ejemplo, una gran preferencia por la impresión), y preferencias tecnológicas (por ejemplo, una preferencia para una plataforma de computadora).
- **Llave de seguridad (securitykey):** El conjunto de contraseñas y claves de seguridad asignadas a los alumnos para las transacciones con los alumnos, los sistemas de información y servicios.
- **Relación (Relationship):** El conjunto de las relaciones entre los componentes básicos. El núcleo no dispone de estructuras dentro de los mismos identificadores que enlazan con las principales estructuras. En lugar de todas esas relaciones se plasman en un único núcleo de la estructura, de modo que, los vínculos sean más simples de identificar y gestionar.

Una vez descritas las funcionalidades de gestión, se realiza un análisis de sus características principales y, por la información que ofrecen a los usuarios, se destacan las siguientes: matriculaciones, listado de cursos, listado de usuarios, calendario y actividades recientes. Estas funcionalidades permiten obtener información específica de los cursos, usuarios y de algunas de las actividades que se realizan dentro de Moodle. Es por ello, que se consideran importantes y se seleccionan para incluir dentro de la propuesta, para que puedan ser consumidas desde diferentes herramientas.

Analizar las características de algunas de las versiones de Moodle, es uno de los aspectos más relevantes a tener en cuenta para identificar aquellas particularidades o deficiencias que no permiten la interoperabilidad con otras herramientas.

1.2.1.2 Características de las versiones de Moodle 1.8.6, 1.9.4 y 2.0

Para el análisis de las características de algunas de las versiones de la plataforma Moodle, se tendrán en cuenta las versiones 1.8.6 y 1.9.4 sobre las cuales existen personalizaciones en la UCI (el EVA y el Entorno de Postgrado). No obstante, como esta plataforma permanece en constante evolución ya se encuentra vigente la versión 2.0, la cual se tendrá en cuenta para un realizar un estudio de aquellas funcionalidades que se le han incorporado. A continuación se presentan características que distinguen a

estas tres versiones.

Moodle 1.8.6

Atendiendo a las peculiaridades de las versiones anteriores, se crea la 1.8.6, que posee modificaciones importantes. Con respecto a la implementación de aquellas funciones que pueden permitir la interoperabilidad entre herramientas se encuentran las siguientes (22):

- **Red Moodle (Network Moodle):** Esta característica permite a un administrador de Moodle establecer un enlace con otro Moodle, y compartir algunos recursos del mismo. Está acompañado de un nuevo *plugin* de autenticación que hace posible el inicio de sesión único (*single signon*) entre sitios Moodle. Esta Red Moodle hace uso del protocolo XML-RPC.
- **API de servicios web:** El código de la Red Moodle incluye un servidor XML-RPC que puede exportar toda la API de Moodle a equipos remotos.
- **Mejoras en el módulo Pasarela de pago Authorize.net (*plugin* de matriculación):** Los administradores de pagos pueden obtener un código de autorización por teléfono del banco del cliente, si la tarjeta de crédito no puede ser obtenida directamente desde Internet.

Moodle 1.9.4

En la versión 1.9.4 surgieron algunos cambios importantes en la plataforma, pero con respecto a una forma de integración de Moodle con otras aplicaciones sólo se posee la siguiente: (23):

- **Red Moodle – Interoperabilidad con Mahara:** Moodle 1.9 y el e-portfolio Mahara v0.9 ahora soportan "Single SignOn" - registro único que permite a los usuarios conectarse automáticamente tanto a sus cuentas de Mahara como en Moodle. Los estudiantes pueden mantener su E-portfolios en Mahara.

Moodle 2.0

Moodle 2.0 fue la última versión liberada por la comunidad de Moodle, la cual posee nuevas características, pues se llevó a cabo prácticamente todo un nuevo desarrollo de la plataforma. Dentro de las más relevantes se pueden destacar: los *community hubs*; brindan la posibilidad de crear directorios de cursos públicos para el bien de las comunidades. Mediante los repositorios se pueden integrar los

contenidos externos hacia Moodle desde otras aplicaciones, con las que interactúa la plataforma como es el caso de Mahara, Flirck¹¹, entre otros. Se le adicionaron nuevos bloques (24):

- **Bloque de comentarios:** Permite comentarios que pueden ser agregados a cualquier página. Ideal para opinión de estudiantes.
- **Mi bloque de archivos privados:** Permite fácil acceso a los archivos propios en un repositorio de archivos privado en Moodle (con soporte de cuotas).
- **Bloque de comunidad:** Mantiene registro de cursos externos en los que un usuario está interesado.
- **Bloque de estado de completado de curso:** Reporta el estado de avance de los cursos de un usuario.

Uno de los aportes principales y más importantes que presenta la plataforma, es la aparición de SW con funciones accesibles mediante protocolos estándar que permiten integrar Moodle con sistemas de información externos (como las bases de datos de matrícula) o crear aplicaciones, por ejemplo, para dispositivos móviles que accedan a los contenidos y actividades de un curso. Otros interesantes aportes son: que los módulos Wiki y Taller están completamente renovados y los blogs en Moodle admiten ahora comentarios y se pueden vincular y sincronizar con blogs externos mediante RSS¹². (24)

La arquitectura que ha de soportar esta nueva versión de Moodle debe estar considerablemente actualizada, exigiendo que para poder instalar o actualizar, se ha de tener disponible algunos de los siguientes gestores de bases de datos: MySQL 5.0.25, PostgreSQL 8.3, Oracle 10.2 o MS SQL 2005. El servidor de páginas web ha de poder servir páginas activas PHP, teniendo instalado y configurado al menos la versión PHP 5.2.8. (25)

Analizadas las características de las tres versiones, se identifica como principal deficiencia, que las versiones 1.8.6 y 1.9.4 no poseen suficientes funcionalidades implementadas sobre SW, lo que posibilitaría la interoperabilidad con otras aplicaciones. La versión 2.0 le brinda solución a este problema, pero teniendo en cuenta las modificaciones realizadas a la misma, en la UCI una migración a esta versión,

¹¹ Red social.

¹² Really Simple Syndication, Formato para compartir contenido en la web.

resulta un proceso complejo, por lo que es mucho más conveniente la búsqueda de otras soluciones. Una de las variantes que resultaría más efectivas para darle solución a este problema sería: crear una interfaz para las versiones 1.8.6 y 1.9.4 que utilice los protocolos y estándares necesarios para establecer comunicación entre aplicaciones.

1.3 API, Interfaz de Programación de Aplicaciones

Varios han sido los conceptos emitidos por diferentes autores sobre API (traducida del inglés como Application Programming Interface) ejemplo de ello son lo que se mencionan a continuación:

Francis y Dominique, en su libro **La alquimia de las multitudes**, la define como: *“puertas que los creadores de los programas abren de forma voluntaria para permitir que otros creadores puedan apropiarse de los elementos que interesen de dichos programas y añadir sus propios servicios.”* (26)

Juan Desongles, en conjunto con varios autores, definen a una API como: *“un conjunto de rutinas, protocolos y herramientas para construir aplicaciones de interfaz.”* (27)

Otra de las definiciones de una API la brinda Jorge y Marcell, en su libro **Fundamentos de la Telemática**, donde la definen como: *“un conjunto de convenios de llamada en programación que definen cómo se invoca un servicio a través de la aplicación.”* (28)

Analizados los conceptos anteriores se asume para la presente investigación la emitida por Jorge y Marcell pues se ajusta a la investigación debido a que una API está compuesta por métodos y funciones que definen una interfaz a un servicio, donde estas funciones pueden ser invocadas a través de varias aplicaciones.

1.3.1 Diferentes tipos de API

Una API detalla solamente la forma de llamar a cada función y la tarea que esta desempeña, sin importar cómo se lleva a cabo dicha tarea. La API toma la información y hace que todo el trabajo específico de comunicación sea transparente para la aplicación. Hay cuatro tipos de APIs que posibilitan el intercambio de datos entre diferentes aplicaciones de software en plataformas individuales o distribuidas (29):

- **Remote Procedure Calls (RPCs) o Llamadas a Procedimientos Remotos:** Con las RPCs, los programas pueden comunicarse a través de procedimientos (o tareas) que actúan en buffers compartidos de información.
- **Standard Query Language (SQL):** Un lenguaje de acceso de datos sin procedimientos que permite el intercambio de datos entre aplicaciones, accediendo a bases de datos comunes.
- **Transferencia de archivos:** Habilita el intercambio de datos enviando archivos formateados entre aplicaciones.
- **Envío de mensajes:** Provee el intercambio de datos por comunicaciones de interprogramación directa, a través de pequeños mensajes formateados entre aplicaciones emparejadas.

Las APIs pueden desarrollarse para cualquier plataforma y sistema operativo o para todos estos sistemas al mismo tiempo. También, en los últimos años, se vienen desarrollando como forma de acceder a SW.

El consumo de SW, siguiendo una Arquitectura Orientada a Servicios (SOA) que posibilita el uso de estándares y protocolos, permite alcanzar la interacción entre plataformas. Mediante el uso de estos servicios se proporcionan mecanismos de comunicación entre diferentes aplicaciones para que interactúen entre sí y permiten presentar información dinámica al usuario.

1.4 Arquitectura Orientada a Servicios

La necesidad de interacción entre plataformas para el intercambio de información es uno de los diversos problemas que se presentan en la actualidad. La Arquitectura Orientada a Servicios (SOA) supone una estrategia para eliminar la falta de comunicación entre componentes que conforman las Tecnologías de la Comunicación: sistemas, aplicaciones y datos. (30)

La compañía Microsoft define a SOA como: “*un enfoque de software de diseño en el que las funciones claves se construyen como componentes reutilizables que implementan estándares interoperables de la industria de las comunicaciones.*” (31)

La orientación a servicios, es un medio para la integración a través de diversos sistemas, utiliza protocolos basados en estándares y pueden desarrollarse utilizando cualquier tecnología orientada a servicios, por lo general, se implementan a través de SW. Debido a que se basan en este tipo de servicios, son empleados

estándares como: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), entre otros para su implementación, lo que permite que esta sea libre de cambiar sin afectar cómo se consume el servicio. (30)

La arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes, de manera que, desde la red, pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella. (30)

1.5 Servicios web

La adopción de una solución de diseño basada en SOA no exige implementar SW, pero es la forma más habitual de hacerlo. Los SW, permiten a las organizaciones el intercambio de datos entre diferentes aplicaciones de distintas fuentes. Estos servicios proporcionan mecanismos de comunicación que permiten presentar información dinámica al usuario. Su uso posibilita la interoperabilidad entre plataformas por medio de estándares y protocolos abiertos y, al hacer uso de estos estándares y protocolos se hace más fácil acceder a su contenido y entender su funcionamiento. (32) Varios han sido los especialistas que han emitido conceptos sobre este término.

Calvo y Munilla, en su libro ***E-business Colaborativo***, definen a los SW como: *“aplicaciones modulares auto-descriptivas que se pueden publicar, ubicar e invocar desde cualquier punto de red o desde el interior de una red local, basados en estándares abiertos de Internet. No es necesario que el proveedor y el usuario de un SW tengan el mismo sistema operativo y utilicen el mismo lenguaje de programación, dado que se basan en estándares aceptados plenamente por la industria como, Extensible Markup Language (XML), HyperText Markup Language (HTML) y Simple Mail Transfer Protocol (SMTP, Protocolo para la transferencia de Correo).”* (33)

Dewit, en su libro ***ASP.Net***, describe a los SW como: *“una biblioteca de clases accesible vía HTTP (Protocolo de transferencia de hipertexto). Su interfaz está descrita en XML, de forma estándar e independiente del lenguaje de implementación subyacente”* (34)

La W3C¹³ lo define como: “*un sistema de software diseñado para apoyar la interacción interoperable máquina a máquina sobre una red. Tiene una interfaz descrita en un formato procesable por máquina (específicamente WSDL). Otros sistemas interactúan con el servicio Web de una manera prescrita por su descripción usando mensajes SOAP, típicamente transmitido a través de HTTP con una serialización XML en conjunción con otras normas relacionadas con la Web*”. (35)

La existencia de múltiples definiciones de los SW, muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Para la investigación, se asume el concepto de Calvo y Munilla, pues se ajusta a la explicación que se desea brindar sobre este tipo de servicios. Una posible definición sería hablar de ellos como: un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web, que utilizan estándares para el intercambio de comunicación entre sistemas de cualquier plataforma para lograr una integración entre ellas. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios.

1.5.1 Beneficios de integrar sistemas mediante Servicios Web

Los SW aportan interoperabilidad entre aplicaciones de software y fomentan los estándares y protocolos basados en texto. A continuación se describen otras de las ventajas que estos aportan (32):

- **Simplicidad:** Los SW son fáciles de diseñar.
- **Estándares abiertos:** Factor importante para su adopción masiva, eliminando la necesidad de invertir en el soporte de nuevos protocolos. Estos estándares tienen la flexibilidad suficiente para poder elegir la capa de transporte que mejor se adapte a la situación.
- **Flexibilidad:** Los SW son flexibles por el acoplamiento entre el proveedor y consumidor del servicio, ofreciendo como resultado ahorro de costes y bajos tiempos de desarrollo.
- **Reusabilidad:** Los SW permiten segmentar las aplicaciones en pequeñas unidades con lógica independiente.
- **Dinamismo:** Presenta interfaces dinámicas.

¹³ World Wide Web Consortium. Comunidad internacional para el desarrollo de estándares de la web.

Los SW se basan en un conjunto de estándares de comunicación por ejemplo: XML para la representación de datos, SOAP para el intercambio de datos y el lenguaje WSDL para describir las funcionalidades de un SW, y otros que resuelven aspectos como la interoperabilidad y la seguridad.

1.5.2 Tecnologías de Servicios Web

Los estándares señalan cómo se deben cursar las peticiones de servicio a servidores remotos, la forma en la cual éstos deben enviar los resultados, y cómo se pueden publicar o dar a conocer los servicios que están accesibles a través de un servidor web. Ninguno de estos estándares trata la forma en la que debe implementarse o programarse el servicio en sí mismo. (36)

Dentro de las principales especificaciones que podemos encontrar en los SW están WSDL, SOAP, Universal Description, Discovery and Integration (UDDI) entre otros.

1.5.2.1 XML, eXtensible Markup Language

XML es un lenguaje muy adecuado y ampliamente utilizado para la transferencia de datos. Utilizado en los SW por ser un estándar abierto, soportado o entendido por cualquier aplicación que pueda leer documentos XML, e independiente de cualquier protocolo de transporte. Sólo necesita de un protocolo que pueda transferir textos o documentos simples, con estas características, dentro de los más conocidos, se encuentran: HTTP y SMTP. (37)

Se entiende por XML como lo define la W3C: *“un lenguaje con una importante función en el proceso de intercambio, estructuración y envío de datos en la Web. Describe los datos de tal manera que es posible estructurarlos utilizando para ello etiquetas, como lo hace HTML pero que no están predefinidas, delimitando de esta manera los datos, a la vez que favoreciendo la interoperabilidad de los mismos.”* (38)

XML se complementa con otras tecnologías que representan una forma más avanzada de compartir datos y permitir la compatibilidad entre los sistemas. Permite compartir información de forma segura, fiable y fácil pues, el trabajo con los datos, la validación de los mismos y su estructura, están a cargo del lenguaje especificado en el estándar. (39)

Describe un conjunto de reglas para especificar etiquetas semánticas que organizan un documento en diferentes partes, es un metalenguaje que explica la sintaxis utilizada para definir otros lenguajes de

etiquetas estructurados. XML es uno de los estándares utilizados en la implementación de SW que permite representar documentos, donde estos documentos tienen una estructura lógica y física. (37)

1.5.2.2 SOAP, Simple Object Access Protocol

El Protocolo de Simple Acceso a Objetos (SOAP) es un protocolo de comunicación, basado en el lenguaje XML, simple y extensible. Es un formato para el envío de mensajes y la comunicación vía Internet a través de HTTP. Importante en el desarrollo de aplicaciones para permitir la interacción entre programas. Define cómo dos objetos en procesos diferentes pueden comunicarse por medio de intercambio de datos XML. (40)

SOAP provee una forma de comunicación entre aplicaciones de diferentes sistemas operativos que utilizan tecnologías y lenguajes de programación distintos. No se encuentra fuertemente asociado a ningún protocolo de transporte, pues, la especificación de SOAP, no describe como deberían asociarse los mensajes con HTTP. Un mensaje SOAP es un documento XML que puede transportarse utilizando cualquier protocolo capaz de transmitir texto. Permite la interoperabilidad entre múltiples entornos, pues, su desarrollo, se basa en estándares existentes como: XML para la codificación de mensajes. Utiliza las especificaciones definidas en este estándar y el medio de transporte se puede asociar tanto a los protocolos HTTP como SMTP. (40)

La especificación SOAP menciona que las aplicaciones deben ser independientes del lenguaje de desarrollo, por lo que, las aplicaciones cliente y servidor pueden estar escritas con HTML, DHTML¹⁴, Java¹⁵, Visual Basic¹⁶ u otras herramientas y lenguajes disponibles. Las peticiones con el uso del protocolo HTTP emplean el comando POST para transmitir información entre el cliente y el servidor. (41)

Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí. Estos mensajes constan de tres secciones: envelope, header y body. Dónde: (41):

¹⁴ HTML Dinámico. Protocolo para la creación de sitios web interactivos.

¹⁵ Lenguaje de programación.

¹⁶ Lenguaje de programación.

- **Envelope:** Es el elemento raíz del mensaje para describir su contenido y la forma de procesarlo. Es el elemento superior del documento XML que representa el mensaje SOAP.
- **Header:** Es la información de identificación del contenido. Un grupo de reglas de codificación para expresar las instancias de tipos de datos definidos por la aplicación.
- **Body:** Es el contenido del mensaje. Una convención para representar las llamadas y las respuestas a procedimientos remotos.

Para la implementación de este protocolo, en ocasiones, se utilizan librerías que ayudan y aportan funcionalidades a este proceso, algunos ejemplos son: Microsoft.Net, NuSoap, WSO2, entre otras. A continuación se describen algunas de los ejemplos antes mencionados.

Microsoft.Net

Microsoft .NET Framework, proporciona un medio completo y eficaz de crear, implementar y administrar la gama de SW que utilizan XML y otras tecnologías relacionadas. Con las herramientas que ofrece los programadores pueden crear aplicaciones para Windows que combinan SW XML con aplicaciones tradicionales. Otras tecnologías incluidas permiten compatibilidad con SOAP como base para la transferencia de mensajes XML en Servicios de Message Queue Server y la integración de la arquitectura de componentes con los SW XML. (42)

A pesar de las ventajas que posee, presenta algunas desventajas pues: .NET no es multiplataforma, ya que sólo está disponible para la familia Windows, es un código cerrado, no hay licencias libres y existen otros productos que ofrecen mucha más portabilidad que .NET. Solo está preparada para ejecutarse sobre plataformas Microsoft.

NuSoap

NuSoap, es una rescritura de SOAPx4, desarrollada por NuSphere y Dietrich Ayala. Proporciona una interfaz SOAP a aplicaciones Web. Está compuesta por un conjunto de clases PHP que permiten a los desarrolladores crear y consumir SW basados en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1. Este kit se

distribuye bajo la licencia LPGL¹⁷. Algunas de las ventajas que posee son: que se encuentra en una fase madura de desarrollo, no necesita de la instalación módulos adicionales y es fácil de utilizar (43)

WSO2

El WSO2 Web Services Framework para PHP, es una extensión de PHP que ofrece formas de trabajo con lo SW. Permite el trabajo tanto con el protocolo REST como con SOAP, además, de posibilitar la generación del WSDL. Brinda soporte para SOAP 1.1, SOAP 1.2, MTOM, WS-Addressing, WS-Security UsernameToken. Es una herramienta libre, lo que posibilita aún más su uso por parte de los desarrolladores. (44)

En la instalación del framewok, dependiendo del Sistema Operativo, este posee algunas dependencias de varias librerías. Por ejemplo para Windows requiere la instalación de Libxml2, la distribución binaria de OpenSSL, además de Mysql Server 5, entre otras. Esta característica puede imposibilitar en ocasiones su uso. (45)

Analizados los ejemplos de bibliotecas que pueden ser utilizadas para la implementación de SW mediante el protocolo SOAP, se decide utilizar NuSoap. Con respecto al estudio realizado de los restantes ejemplos, NuSoap posee ventajas en cuanto a su fácil instalación y uso, además de no requerir de la instalación de módulos adicionales para su utilización. Permite su uso desde cualquier Sistema Operativo, pues sólo requiere de guardar la carpeta que contiene las clases PHP en un servidor web.

1.5.2.3 UDDI, Universal Description, Discovery and Integration

Universal Description, Discovery and Integration (UDDI), es un método estándar utilizado por las empresas para descubrir de forma dinámica los SW e invocarlos. Cuando un SW es creado se necesita definir la manera en que se dará a conocer, para que los clientes interesados en él, puedan descubrirlo fácilmente y puedan utilizarlo en sus aplicaciones. El mecanismo ideal que cumple con estos requisitos es UDDI, que permite el descubrimiento de SW con todo tipo de tecnologías y plataformas. (41)

¹⁷ Licencia orientada para proteger la libre modificación, distribución y uso del software.

UDDI se basa en estándares existentes en el desarrollo de los SW como son: XML y SOAP. Es un registro público diseñado para almacenar de forma estructurada la descripción de los servicios y la integración de estos utilizando Internet. Conformar un directorio para almacenar información relacionada con el SW y consiste, en la interfaz del directorio del servicio descrita por el WSDL, de este modo sirve como infraestructura para una colección de software basado en SW. (46)

1.5.2.4 WSDL, Web Services Description Language

El Lenguaje de Descripción de los Servicios Web (por sus siglas en inglés WSDL), es un documento escrito en lenguaje XML, comúnmente utilizado para realizar la descripción funcional y técnica de los SW. Este documento proporciona información necesaria al cliente para interactuar con el SW. (41)

Un documento WSDL, describe ciertas características propias de un SW, especifica las operaciones o métodos que presenta el servicio, así como su localización. Para la descripción de un SW en el documento WSDL utiliza algunos elementos como es el caso de (41):

- **Types:** describe los tipos de datos utilizado en el SW.
- **Message:** describe los mensajes utilizados en el SW.
- **PortType:** describe las operaciones realizadas por el SW.
- **Binding:** describe los protocolos de comunicación utilizados en el SW.
- **Operation:** descripción de una operación admitida por el servicio.
- **Port:** punto final único que se define como la combinación de un enlace y una dirección de red.
- **Service:** colección de puntos finales relacionados.

1.5.2.5 XML-RPC

En la actualidad, uno de los protocolos más empleados para los SW es: XML-RPC (XML-Remote Procedure Call o Llamada a Procedimiento Remoto-XML). En su sitio oficial se define como: *“una especificación y un conjunto de implementaciones que permiten al software que se ejecute en sistemas operativos y entornos diferentes para realizar llamadas de procedimiento a través de Internet.”* (47)

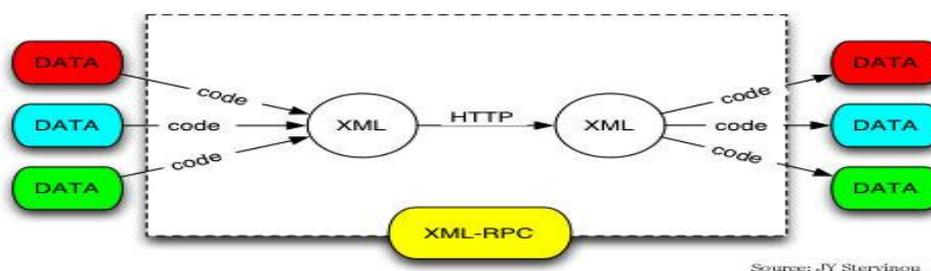


Figura 1 Estructura del protocolo XML-RPC. (47)

Se trata de una especificación y de un conjunto de implementaciones que permiten que aplicaciones heterogéneas (distinto lenguaje de programación, sistema operativo, entre otros.) puedan hacer llamadas a procedimientos remotos a través de Internet usando una especificación común. Para llevar a cabo las llamadas usa HTTP como mecanismo de transporte y XML para codificarlas. Ha sido diseñado con el objetivo de ser lo más simple posible, permitiendo el envío, procesado y retorno de estructuras complejas de datos. El protocolo XML-RPC, define un conjunto de tipos de datos y estructuras donde, cada cliente o servidor que lo implementa puede mapear los tipos de datos que utilice. (47)

Luego de analizados algunos de los protocolos más utilizados para la implementación de los SW, SOAP, como protocolo de comunicación, es uno de los más importantes, pues es un formato basado en el lenguaje XML que permite el envío y la comunicación entre las aplicaciones, utilizando para ello el protocolo HTTP y WSDL, como lenguaje fundamental para que puedan ser descritos los SW.

1.5.3 Seguridad en los servicios web

La publicación de los SW supone varias ventajas, entre ellas la interoperabilidad entre plataformas que permiten el intercambio de información, la cual se presenta de forma dinámica al usuario. Esto puede convertirse también, en una amenaza a la seguridad de la información que se maneja. Para ello, se debe tener en cuenta, la aseguración de las claves de autenticación, garantizar la integridad y confidencialidad de la información. Para la seguridad de los datos existen diferentes protocolos y estándares, ejemplo de ello: WS-Security, Secure Socket Layer (SSL), entre otros. A continuación, se presentan algunas de las características que poseen los mismos.

WS-Security

WS-Security, es un estándar que ofrece funciones para proteger las aplicaciones de SW, pues se basa en estándares establecidos respecto a la criptografía y firmado de XML. Permite la especificación para firmar y cifrar los mensajes SOAP. Es recomendable su uso cuando es necesaria la autenticación, integridad y confidencialidad en varios puntos y saltos de comunicación y no solo de extremo a extremo como lo hace SSL. Proporciona un requisito especializado de seguridad: **UsernameToken**, que es un dispositivo simple y robusto para la comunicación de la autenticación básica de un usuario a un servicio. Para firmar un mensaje, ofrece un documento que permite verificar la autenticidad del mismo en el momento que es recibido y procesado. Puede ser almacenado además, con fines de auditoría, manteniendo intacta la garantía de autenticidad. (48)

Sin embargo presenta algunas desventajas, pues puede añadir un numeroso volumen a las cabeceras de los mensajes de SOAP, lo que puede tener un impacto significativo en el rendimiento cuando se envían los datos entre el cliente y el servidor en una red. (48) Además, no es un estándar soportado por la librería NuSoap, que fue la seleccionada para la implementación de los SW en este trabajo.

SSL

Secure Socket Layer (SSL), es un protocolo que administra la seguridad de las transacciones que se realizan a través de la red. Se basa en proceso de cifrado de certificado de clave pública, el cual garantiza la seguridad de los datos que se publican a través de la red. Establece un canal de comunicación seguro entre el cliente y el servidor después de ejecutada la fase de autenticación. Es independiente del protocolo utilizado pues puede asegurar las transacciones realizadas a través de HTTP, FTP¹⁸, POP¹⁹ e IMAP²⁰. (49)

Dentro de las ventajas que proporciona su uso se encuentran: brinda la posibilidad de alcanzar altos niveles de rendimiento, pues si el servicio es contactado directamente por los clientes y no a través de

¹⁸ Protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP.

¹⁹ Protocolo que permite obtener los mensajes de correo electrónico almacenados en un servidor remoto.

²⁰ Protocolo de red de acceso a mensajes electrónicos de un servidor.

varios servidores proporciona garantías de confidencialidad de la información a un costo de rendimiento mucho más bajo que WS-Security. Para mantener la seguridad requiere de certificados del cliente como pruebas de identidad del establecer la conexión entre el cliente y el servidor, sin embargo esta es una de sus desventajas pues este tipo de garantía de autenticidad es mucho más débil que el firmado digital de los mensajes. (48)

Una vez analizadas, ambas tecnologías, las cuales permiten garantizar la seguridad de la información que se brinda mediante SW, se decide utilizar para la propuesta al protocolo SSL. La principal ventaja que proporciona a los SW que se implementarán en el desarrollo de la propuesta, es que permitirá un mayor rendimiento y rapidez de respuesta, pues estos servicios se localizarán en un mismo servidor. Esta característica ofrece ventajas de uso para este protocolo, pues es muy recomendable cuando las conexiones son de extremo a extremo, ya que permite a una mejor solución para lograr valores ínfimos de costo de rendimiento.

1.6 Metodologías de desarrollo de software

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto de software que reúna los requisitos del cliente. Un proceso de software detallado y completo suele denominarse “Metodología”. Una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. (50)

Existen varias metodologías que se pueden clasificar en dos grupos, si se toma en cuenta, como criterio, las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, reciben el apelativo de Metodologías Tradicionales. Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños e involucran activamente al cliente en el proceso. (50) A continuación se describe una metodología utilizada por el “Centro de Desarrollo de Tecnologías para la Formación” (FORTES) del Departamento de Producción de Herramientas Educativas en la UCI para el desarrollo de sus proyectos de software.

Proceso Unificado de Rational

Proceso Unificado de Rational (del inglés, Rational Unified Process, RUP) es una infraestructura flexible de desarrollo del software para la Ingeniería de Software Orientada a Objetos, está basado en el Lenguaje Unificado de Modelación (UML). RUP proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo de software. Define, para cada etapa y cada disciplina, el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen. (51)

RUP unifica todo el equipo de desarrollo de software, lo que favorece la comunicación, asegurando la asignación de recursos en forma eficiente, la entrega de los artefactos correctos y el cumplimiento de los tiempos límite.

Mantiene una estructura bien definida del ciclo de vida de un proyecto y, de forma general, está caracterizado por los siguientes aspectos: guiado por los casos de uso, centrado en la arquitectura e iterativo e incremental. Según la metodología RUP el ciclo de vida de un proyecto se divide en las siguientes fases: fase de concepción, de elaboración, de construcción, de transición. (51)

Para el desarrollo de la propuesta, se decide utilizar la metodología RUP, por de los beneficios que aporta. Además, teniendo en cuenta, sus características el proyecto “Desarrollos en Moodle 1.8.x y 1.9.x” perteneciente al “Centro de Desarrollo de Tecnologías para la Formación” (FORTES) del Departamento de Producción de Herramientas Educativas, definió su uso para llevar a cabo cualquier desarrollo que se produzca entorno a la plataforma Moodle. RUP proporciona las mejores prácticas de desarrollo. Se adapta a las necesidades de los proyectos. Permite integrar rápidamente a los miembros del equipo y poner en acción el proceso personalizado.

1.7 Tecnologías y herramientas de desarrollo del software

En el proceso de desarrollo del software se utilizan tecnologías y herramientas que hacen posible el perfeccionamiento de los artefactos que se obtengan durante todo su ciclo de desarrollo. Para ello, se asumen diferentes lenguajes de programación, se identifica una metodología de desarrollo específica y se elige la herramienta para obtener los artefactos.

1.7.1 Lenguaje de Programación

Los lenguajes de programación son herramientas que permiten la creación de programas y software. Facilitan la tarea de programación porque disponen de formas adecuadas que permiten ser escritas y leídas por las personas. Es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación, es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (52)

A continuación se describe PHP, lenguaje interpretado utilizado en la implementación de la plataforma Moodle, utilizado en el desarrollo de sitios web dinámicos, y útil además, en la implementación de SW.

PHP

En el sitio oficial php.net, el término PHP (*acrónimo de: Hypertext Preprocessor*) se define como: “es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML.” (53)

PHP es un lenguaje que está centrado en programación de **script** del lado del servidor, generando HTML y enviándolo al cliente. Puede ser utilizado por cualquiera de los sistemas operativos existentes y soporta la mayoría de los servidores web como Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet y muchos otros. Es un lenguaje que permite elegir el sistema operativo y el servidor web que se necesite utilizar y, además, tiene la posibilidad de usar programación procedimental o la programación Orientada a Objeto. (54)

PHP brinda la posibilidad de crear imágenes y archivos PDF; puede presentar resultados XHTML y cualquier otro archivo de tipo de ficheros XML, además, permite autogenerar estos archivos y almacenarlos en el sistema de archivos creando un caché en el lado del servidor para contenido dinámico. Dentro de la característica que más lo destaca y hace que se convierta en uno de los lenguajes más potentes, está su soporte a una gran cantidad de base de datos: Adabas D, Empress, Hyperwave, IBM DB2, MySQL, ODBC, Oracle (OCI7 y OCI8), PostgreSQL, Unix dbm. PHP cuenta con soporte para

comunicarse con otros servicios usando protocolos como LDAP (Protocolo Ligero de Acceso a Directorios), IMAP, SNMP²¹, NNTP²², POP3, HTTP y muchos otros. (54)

1.7.2 Lenguaje de Modelado

UML

UML surgió con el objetivo de servir de apoyo al proceso de análisis de un problema y se ha convertido en un estándar para modelar y representar la información con la que se trabaja en las fases del análisis y especialmente, en las de diseño.

Proporciona un vocabulario y una regla para permitir la comunicación. Se centra especialmente en la representación gráfica de un sistema. Permite representar de una medida u otra todas las fases de un proyecto informático: desde el análisis con los casos de uso, los Diagramas de Clases, Objetos, hasta la implementación y configuración con los Diagramas de Despliegue. (55)

Los objetivos de UML se pueden sintetizar en sus funciones (55):

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Se selecciona este lenguaje de modelado pues brinda gran variedad de formas para dar soporte a una metodología de desarrollo de software como RUP. Útil, además, para definir el sistema que se desea obtener y detallar cada uno de los artefactos que se generen.

²¹ Protocolo para la administración de redes.

²² Protocolo para la transferencia de noticias en red.

1.7.3 Herramientas de Desarrollo del Software

Una de las claves para obtener el éxito de un proyecto de desarrollo de software, es tener en cuenta, la utilización de tecnologías, estándares y herramientas libres que garanticen la independencia de plataformas y aseguren la posibilidad de mantenimiento futuro de las aplicaciones. Seleccionar el IDE (Entorno de Desarrollo Integrado/Integrated Development Environment), programa informático compuesto por un conjunto de herramientas de programación, es de vital importancia para el desarrollo de un software. Los IDE pueden aplicarse en exclusiva a un solo lenguaje de programación o a varios.

En la universidad dentro de los IDEs más utilizados para programar en lenguaje PHP son: ZendStudio y Netbeans. A continuación, se presentan algunas de las principales características que estos poseen.

Zend Studio

Zend Studio es un entorno de desarrollo completo integrado para el lenguaje de programación PHP. Está escrito en Java y es multiplataforma, pues se encuentra disponible para plataformas Microsoft Windows, GNU/Linux y Mac OS X. Diseñado para maximizar la productividad de los programadores, pues le permite resolver problemas de aplicación de forma rápida y mejora el trabajo en equipo. (56)

Dentro de sus características se puede encontrar que posee soporte para el trabajo con el control de versiones usando CVS o Subversion. Permite la navegación en base de datos y ejecución de consultas SQL. Ofrece soporte básico para otros lenguajes web como: HTML, JavaScript y XML. (56)

Netbeans

Netbeans IDE es un entorno de desarrollo, producto libre y gratuito sin restricciones de uso. Herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. (57)

Es mucho más que un Java IDE, está diseñado como una herramienta de desarrollo modular para una amplia gama de tareas de desarrollo. La base del IDE incluye un editor avanzado en varios idiomas, depurador y la integración de perfiles y archivo de control de versiones. Permite el uso de un **plugins** de Administración para agregar, quitar o actualizar conjuntos de características de PHP, UML, Java EE,

incluso el desarrollo de SOA. El IDE provee herramientas para trabajar con SW. Utiliza asistentes para crear servicios REST²³ y generar el código para invocar SW (tanto en REST como en SOAP). (58)

Se decide utilizar para la implementación que requiere la propuesta el IDE Netbeans, como herramienta para el desarrollo de este trabajo. Este es una herramienta de libre distribución, desventaja que posee el ZendStudio. Otras de las ventajas de la herramienta seleccionada es que ofrece formas de trabajo para la implementación de SW.

1.7.4 Herramientas CASE para el modelado

Computer Aided Software Engineering (CASE) o Ingeniería de Software Asistida por Computación, es la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación. Las herramientas CASE representan una forma que permite Modelar los Procesos de Negocios de las empresas y desarrollar los Sistemas de Información Gerenciales. (59)

Algunos de los componentes de las herramientas CASE permiten:

- Confeccionar la definición de requerimientos de los usuarios.
- Mejorar el diseño de los sistemas.
- Mejorar la eficiencia en la programación (por su generación automática de códigos).
- Otorgar a la administración un mejor soporte en la documentación.

Visual Paradigm

Visual Paradigm es una herramienta UML (Unified Modeling Language) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño Orientados a Objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de Diagramas de Clases, código inverso, generar código desde diagramas y generar documentación. (60)

Dentro de sus principales características se encuentran (60):

- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor del negocio, Documento.

²³ Protocolo utilizado en los servicios web.

- Generación de código - Modelo a código, Diagrama a código.
- Diagramas de flujo de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Plataforma Java (Windows/Linux/Mac OS X):
 - SDE para Eclipse
 - SDE para NetBeans
 - SDE para Sun ONE
 - SDE para Oracle JDeveloper
 - SDE para JBuilder
 - SDE para IntelliJ IDEA
 - SDE para Web Logic Workshop

Rational Rose

Los productos Rational Rose incluyen soporte UML. Para el ambiente de modelado, soporta la generación de código a partir de modelos en C++, CORBA, Java, Visual Basic, entre otros. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad. Permite la publicación web y generación de informes para optimizar la comunicación dentro del equipo. Presenta cuatro Vistas: la Vista de Casos de Uso, en la que se representa el comportamiento deseado del sistema, la Vista Lógica que muestra la estructura y el comportamiento del sistema, la de Componentes y por último la Vista de Despliegue, en la que se modela la distribución o despliegue de los nodos de procesamiento del sistema. Posee librerías para la compatibilización de ingeniería inversa sobre Java, XML_DTD, CORBA y Visual Basic. (61)

El Rational Rose Enterprise Edition, presenta ciertas desventajas que influyen de manera negativa en su uso, pues es un software propietario, lo que dificulta su adquisición. Solo cuenta con versiones en inglés, por lo que sus manuales de ayuda a pesar de ser muy útiles se encuentran en el mismo idioma.

Para el desarrollo de este trabajo se decide utilizar Visual Paradigm, pues permite trabajar con precisión y

calidad. La documentación que propone, contribuye a lograr un mejor entendimiento del sistema por parte del equipo de desarrollo, y la misma servirá de referencia para posteriores trabajos sobre Moodle. Además, el uso de la misma está permitido de forma libre, característica que no posee Rational Rose, por ser un software propietario.

Conclusiones parciales

Las versiones 1.8.x y 1.9.x de Moodle no tienen las funcionalidades implementadas sobre SW lo que imposibilita su uso desde otros entornos.

Para poder consumir funcionalidades de Moodle desde otros entornos puede implementarse una API sobre SW que contenga dichas funcionalidades.

Para llevar a cabo la implementación de una API que contenga las funcionalidades de gestión de Moodle 1.8.x y 1.9.x sobre SW, se determina utilizar las siguientes tecnologías: SOAP como protocolo de comunicación entre plataformas, WSDL, para describir el servicio y el lenguaje XML, para la transferencia de datos, además del uso de la biblioteca NuSoap para la implementación de estos servicios pues permite el trabajo con el protocolo SOAP.

Para el diseño e implementación del API se determina el uso de: la metodología RUP para llevar a cabo el proceso de desarrollo del producto, PHP como lenguaje de programación, Visual Paradigm como herramienta y UML como lenguaje de modelado.

Se decide utilizar PHP como lenguaje de programación por ser el lenguaje sobre el cual se encuentra implementado Moodle. La plataforma Netbeans, ya que ofrece formas de trabajo para la implementación de SW.

Capítulo 2: Diseño del sistema

Introducción

Atendiendo a las necesidades planteadas por los usuarios de la UCI, se determina desarrollar una API que contenga las funcionalidades de gestión de Moodle 1.8x y 1.9.x sobre SW. La misma estará enfocada en permitir que los usuarios gestionen e interactúen con la información que brindan estas funciones sobre los cursos, usuarios y actividades que se realizan dentro de la plataforma. Las funcionalidades de gestión que contendrá la API son las más utilizadas por los usuarios de la universidad y están relacionadas con: el Calendario, la Gestión de usuarios, la Gestión de cursos, las Actividades recientes y las Matriculaciones.

Para que estas funcionalidades puedan ser consumidas desde otros entornos, se implementará una API de SW utilizando la biblioteca NuSoap. La misma proporciona una interfaz SOAP y permite generar los WSDL que contienen la descripción de qué parámetros exige cada funcionalidad. Se tendrá en cuenta además, las restricciones de acceso a la información según los roles predefinidos por los desarrolladores de la plataforma, para de esta forma garantizar la integridad de la información que se pondrá a disposición de los usuarios. Para la seguridad se utilizará el protocolo SSL para asegurar el canal de comunicación cliente-servidor.

Con el objetivo de llevar a cabo un mejor proceso de desarrollo de la API y de documentar con claridad las necesidades de los usuarios, se exponen los artefactos más representativos generados durante la aplicación de la metodología de desarrollo RUP.

2.1 Propuesta del sistema

En la UCI, existe la necesidad de lograr una mayor interacción entre las plataformas educativas existentes. Para dar cumplimiento a este objetivo, se decidió la implementación de las funcionalidades más importantes de Moodle sobre SW, para que puedan ser consumidas desde otras aplicaciones. Para el desarrollo de la propuesta se elaborará una API que se implementará teniendo en cuenta el uso de la librería NuSoap, la cual va a contener las funcionalidades de gestión, que serán brindadas mediante SW.

Desarrollar esta API sobre SW permitirá que los usuarios, según los privilegios que posean, puedan gestionar o interactuar con la información que brindan estas funciones sobre los cursos, usuarios y actividades que se realizan dentro de la plataforma Moodle.

2.2 Modelo de Dominio

Un Modelo de Dominio (MD) representa una parte visual de las clases y sus interacciones con entorno real del proyecto. Este se centra en una parte del negocio relacionada con el ámbito del proyecto y se aplica cuando no es posible identificar con claridad, los procesos, actores y trabajadores que realizan las actividades sobre los procesos del negocio. El diagrama que se elabora en el MD representa los objetos relacionados con los principales conceptos que se trabajarán en el desarrollo de este trabajo. Este diagrama permitirá a los usuarios, desarrolladores e interesados lograr una mayor comprensión de los conceptos que se manejan, permitiéndoles utilizar un vocabulario común para comprender el contexto en el que se encuentra el sistema. Contribuirá, además, a identificar personas, eventos, transacciones y objetos involucrados en el sistema.

2.2.1 Representación del Diagrama de Modelo de Dominio

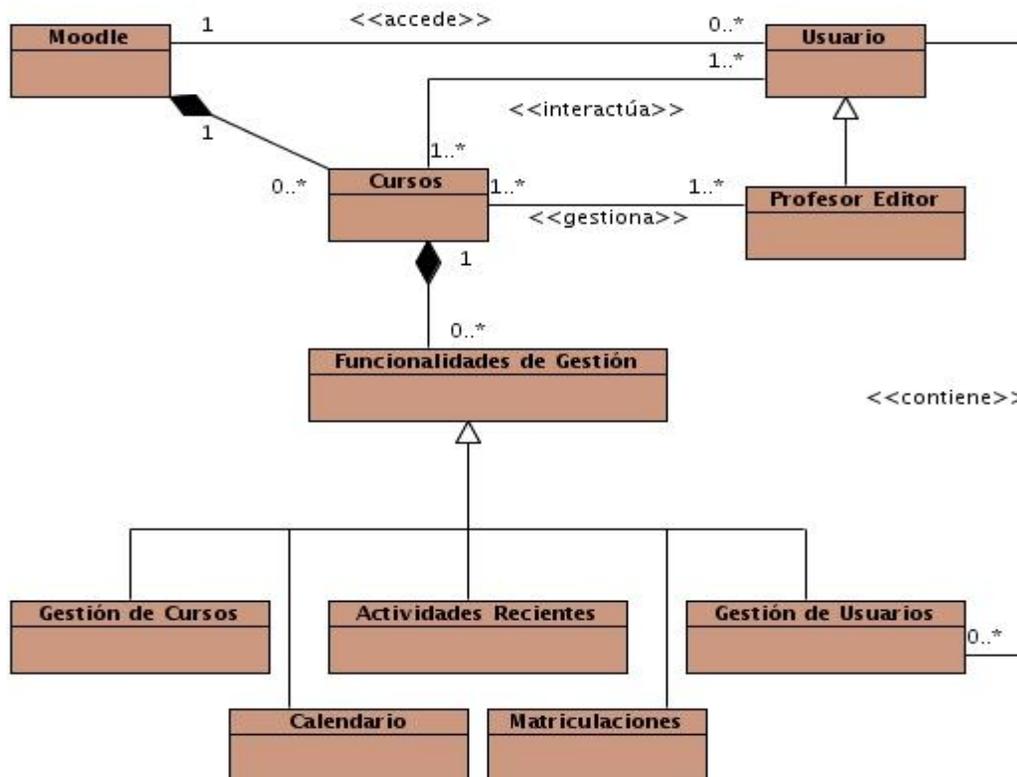


Figura 2 Modelo de Dominio

2.2.2 Definición de las clases del Modelo de Dominio

A continuación, se explica en qué consiste cada una de las clases del Modelo de Dominio.

Moodle: Entorno Virtual de Aprendizaje donde, tanto el profesor editor, como el usuario, pueden acceder a las funcionalidades de las actividades de gestión e interactuar con estas.

Funcionalidades de Gestión: Brindan información relacionada con los listados de los usuarios, los listados de los cursos y algunas de las principales actividades que se realizan dentro de la plataforma Moodle. Permiten, además, el trabajo con el Calendario, las actividades recientes y la matriculaciones en los cursos de la plataforma.

Profesor Editor: Usuario que posee privilegios para gestionar los cursos de la plataforma Moodle.

Usuario: Usuario que posee privilegios para interactuar con los cursos de la plataforma Moodle.

Cursos: Unidad educativa en la que se ofrece un conjunto estructurado de conocimientos teóricos y/o prácticos.

Gestión de Usuarios: Constituye las listas e información de los estudiantes que se encuentran matriculados dentro de los cursos de la plataforma Moodle.

Calendario: Bloque que puede ser utilizado por el usuario y el profesor editor para gestionar los eventos propios del usuario y los cursos, respectivamente.

Gestión de Cursos: Contiene las listas e información de los cursos que se imparten dentro de la plataforma.

Actividad Reciente: Contiene una lista de forma abreviada con los acontecimientos producidos en un curso desde la última visita de un usuario a un curso.

Matriculaciones: Representa las matriculaciones de los diferentes usuarios en los cursos de la plataforma.

2.3 Especificación de Requisitos

Un requisito, es una descripción de las necesidades y aspiraciones que se tienen respecto a un producto. El objetivo principal de la definición de los requisitos es identificar lo que realmente se necesita. La definición y descripción de estos, ayuda a que exista una mayor comunicación entre el cliente y el equipo de desarrollo. Existen dos tipos de requerimientos, los requisitos funcionales y los no funcionales.

2.3.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, describe lo que este debe hacer.

Para cumplir con los objetivos propuestos, se espera que los SW que integrarán la API a implementar para la plataforma Moodle cumplan con los requerimientos funcionales siguientes:

RF1 Matricular estudiante en curso: Este SW de Moodle debe brindar la posibilidad de matricular estudiantes en un curso mientras no esté completa la capacidad.

RF2 Dar baja a estudiante de un curso: Este SW de Moodle debe brindar la posibilidad de dar baja a un estudiante que esté matriculado en un curso.

RF3 Mostrar método de matriculación: Este SW de Moodle debe brindar la posibilidad de mostrar los diferentes tipos de matrículas que se encuentren activadas en la plataforma para realizar las matriculaciones.

RF4 Mostrar evento del calendario: Este SW de Moodle debe brindar la posibilidad de visualizar los eventos del Calendario a los usuarios, para así, obtener información relacionada con las actividades que se estén realizando en los cursos a los que estén inscritos.

RF5 Adicionar evento al calendario: Este SW de Moodle debe brindar la posibilidad de que los eventos del Calendario sean adicionados por los usuarios.

RF6 Editar evento del calendario: Este SW de Moodle debe brindar la posibilidad de que los eventos del Calendario puedan ser modificados por los usuarios.

RF7 Eliminar evento del calendario: Este SW de Moodle debe brindar la posibilidad de que los eventos del Calendario puedan ser eliminados por los usuarios.

RF8 Listar usuarios de un curso: Este SW de Moodle debe brindar la posibilidad de visualizar un listado de usuarios que se encuentran matriculados en un mismo curso.

RF9 Mostrar perfil de usuario: Este servicio web de Moodle debe brindar la posibilidad de que un usuario determinado pueda visualizar su perfil.

RF10 Buscar perfil de usuario: Este SW de Moodle debe brindar la posibilidad de buscar el perfil de un usuario determinado.

RF11 Editar perfil de usuario: Este SW de Moodle debe brindar la posibilidad de que los usuarios puedan modificar y editar su perfil.

RF12 Listar usuarios de la plataforma: Este SW de Moodle debe brindar la posibilidad de mostrar todos los usuarios que se encuentran registrados en la plataforma.

RF13 Mostrar todos los cursos: Este SW de Moodle debe brindar la posibilidad de que los usuarios visualicen todos los cursos que se imparten en la plataforma.

RF14 Mostrar mis cursos: Este SW de Moodle debe brindar la posibilidad de que los usuarios visualicen todos los cursos en los que están matriculados.

RF15 Buscar curso: Este SW de Moodle debe brindar la posibilidad de que los usuarios realicen la búsqueda de un curso determinado mediante su nombre.

RF16 Mostrar información general del curso: Este SW de Moodle debe brindar la posibilidad de que los usuarios visualicen toda la información de los cursos que se encuentran en la plataforma.

RF17 Mostrar actividad reciente: Este SW de Moodle debe brindar la posibilidad de que los usuarios puedan visualizar una lista de forma abreviada con los acontecimientos producidos en el curso desde la última visita.

RF18 Autenticar Usuario: Este SW debe brindar la posibilidad de que los usuarios puedan autenticarse y poder tener acceso a los demás servicios que brinda la plataforma Moodle, según los privilegios que tengan estos.

2.3.2 Requisitos no Funcionales

Los requisitos no funcionales son restricciones de los servicios, cualidades o propiedades que el producto debe tener. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, como cuán usable, seguro, conveniente o disponible puedan ser estos.

Disponibilidad

- Se les garantizará el acceso a la información las 24 horas a los usuarios autorizados.

Confidencialidad

- La información manejada por el SW debe estar protegida ante el acceso no autorizado y la divulgación.

Restricciones de diseño

- Para la implementación de los SW se requiere utilizar el lenguaje de programación PHP, utilizando además la biblioteca de componentes NuSoap.

Rendimiento

- Garantizar tiempos de respuestas generalmente rápidos, al igual que la velocidad de procesamiento de la información, en menos de 5.0 segundos, en el 90 por ciento de los casos.

Legales

- La plataforma a la que se le desarrollarán SW está basada en la licencia GNU/GPL.

Software

- Navegador Web (Internet Explorer 7 y Mozilla Firefox 3.5.5).
- Moodle 1.9.4 y 1.8.6.
- PHP 5.2.8.

Hardware

- 512 MB de memoria RAM.
- 80 GB de disco duro.
- Microprocesador 500 MHz.

2.4 Modelo de Casos de Uso del Sistema

Un Diagrama de Casos de Uso (CU) del Sistema es la representación gráfica de los procesos y su interacción con los actores. Este modelo está formado por los CU del sistema y sus actores. Un actor es cualquier individuo, grupo, o sistema de información que interactúa con el sistema, por otra parte, un CU no es más que una secuencia de acciones que producen un valor observable para un actor concreto. (62)

2.4.1 Actores del Sistema

Actor	Descripción
Profesor Editor	Cumple la función de administrador del sistema, el cual va a tener privilegios para editar, buscar y administrar los SW que brindará la plataforma Moodle. Además, podrá ver los métodos de matriculación empleados por la plataforma.

Usuario	Cumple con los roles de estudiante y profesor no editor, los cuales solamente tienen privilegios para editar, buscar y administrar algunos de los SW que brindará la plataforma Moodle.
----------------	---

Tabla 1 Actores del Sistema

2.4.2 Diagrama Casos de Uso del Sistema

El Diagrama de Casos de Uso del Sistema que aparece a continuación muestra los casos de usos y actores definidos, con una asociación entre cada par actor/caso de uso que interactúan.

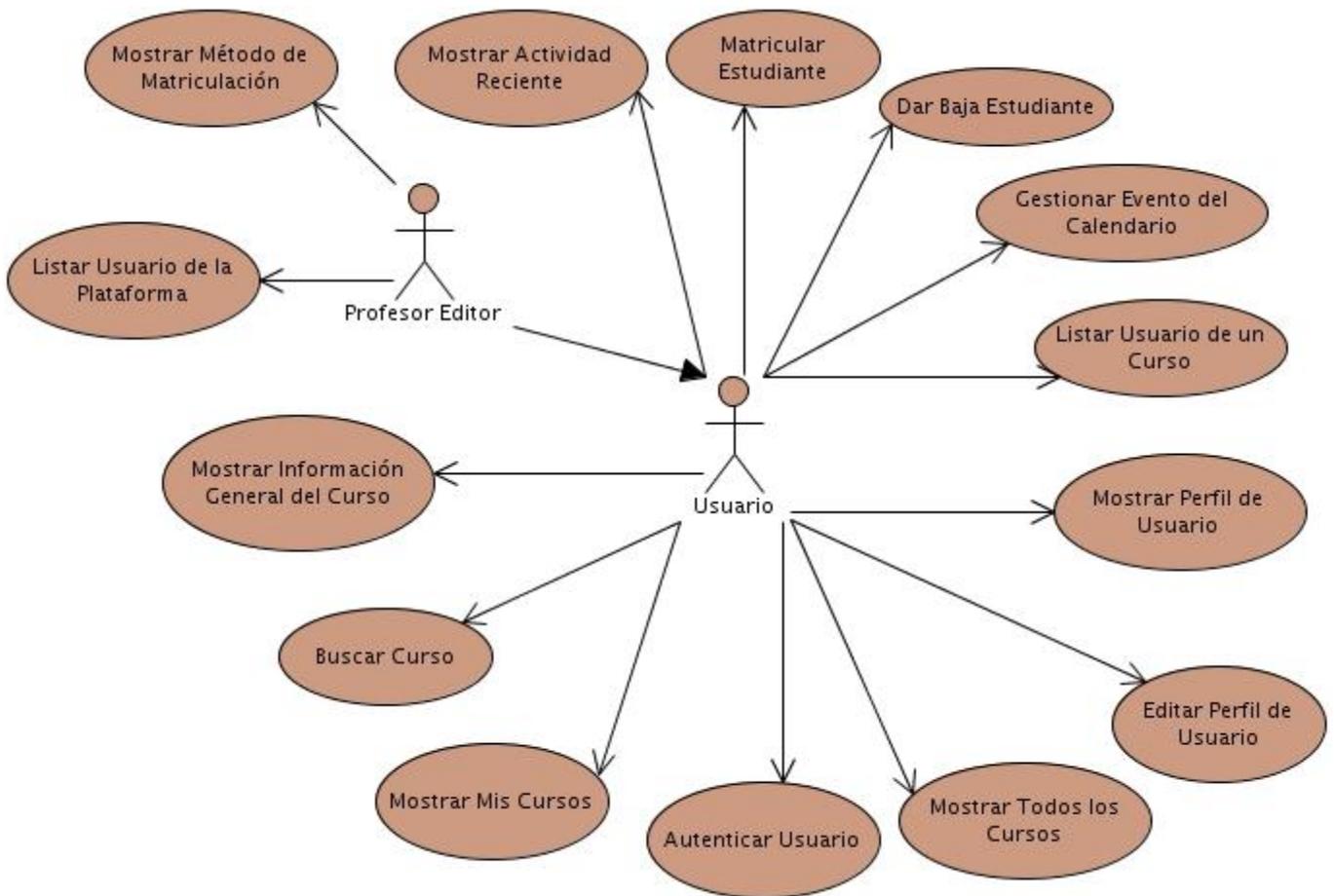


Figura 3 Diagrama de Casos de Uso del Sistema

2.4.3 Descripción Textual de Casos de Uso del Sistema

En este epígrafe se describen detalladamente los CU presentes en el Diagrama de Casos de Uso del Sistema. Se situará el propósito general de cada CU, el actor que le da inicio al mismo, así como, las precondiciones y las poscondiciones para su funcionamiento.

CU Matricular Estudiante

Tabla 2 Descripción textual del CU Matricular Estudiante

Caso de Uso:	Matricular Estudiante
Actores:	Usuario (Inicia)
Resumen:	El CU inicia cuando el Usuario solicita la opción de matricularse en un curso de la plataforma Moodle y el SW le permite realizar la acción. El CU finaliza cuando el usuario se matricula en un curso.
Precondiciones:	El Usuario debe estar autenticado.
Referencias	RF1
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El CU inicia cuando el actor solicita la opción de matricular en un curso.	2. Brinda la posibilidad de insertar el identificador del curso en el que desea matricular.
3. Inserta el identificador del curso en el que desea matricular y selecciona la opción de matricular.	4. Valida los datos de entrada. 5. Verifica en la base de datos si existe el identificador del curso. 6. Verifica si el actor se encuentra matriculado en el curso. 7. Muestra un mensaje de confirmación preguntando si el actor desea matricular en el curso.
8. Selecciona la opción de aceptar.	9. Actualiza la base de datos. 10. Muestra un mensaje indicando que el usuario se ha matriculado en el curso.
Flujos Alternos	
Flujo Alternativo 4a "Campos vacíos"	
	4a.1 Muestra un mensaje de error indicando que han quedado campos vacíos.
Flujo alternativo 4b "Datos Incorrectos"	

	4b.1 Muestra un mensaje de error indicando que existen datos incorrectos.
Flujo Alternativo 5a "Mensaje de aviso"	
	5a.1 Muestra un mensaje indicando que el curso no existe en la plataforma.
Flujo Alternativo 6a "Mensaje de verificación"	
	6a.1 Muestra mensaje indicando que el usuario se encuentra matriculado en el curso.
Flujo Alternativo 9a "Opción cancelar"	
	9a.1 No matricula al actor en el curso.
Poscondiciones	Los usuarios quedan matriculados en los cursos existentes en la plataforma Moodle.

CU Autenticar Usuario

Tabla 3 Descripción textual del CU Autenticar Usuario

Caso de Uso:	Autenticar usuario	
Actores:	Usuario (Inicia)	
Resumen:	El CU inicia cuando el Usuario desea autenticarse en la plataforma Moodle, para ello debe introducir su usuario y contraseña y el servicio web le permite realizar este proceso. El CU finaliza cuando el usuario queda autenticado.	
Referencias	RF18	
Prioridad	Crítica	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El CU inicia cuando el actor solicita el servicio de autenticación.	2. Muestra la opción para realizar la autenticación del usuario, donde debe recoger los siguientes datos: <ul style="list-style-type: none"> • Usuario. • Contraseña.

3. Introduce los datos y selecciona la opción de entrar.	5. Valida los datos. 6. Actualiza en la base de datos los datos del usuario autenticado.
Flujos Alternos	
Flujo alternativo 5a“Campos vacíos”	
	5a.1 Muestra un mensaje de error indicando que existen campos vacíos. Brinda la opción de reintentar la operación.
Flujo alternativo 5b“Datos Incorrectos”	
	5b.1 Muestra un mensaje de error indicando que existen datos incorrectos. Brinda la opción de reintentar la operación.
Poscondiciones	El usuario queda autenticado.

El resto de las descripciones de los CU identificados para las funcionalidades de gestión sobre SW se localizan en los anexos del trabajo. (Anexo 2 – Anexo 13)

2.5 Modelo de Análisis

El Modelo de Análisis proporciona la estructura a la vista interna, es descrito por el lenguaje del programador. Está estructurado, además, por clases y paquetes estereotipados, utilizado fundamentalmente, por los desarrolladores, para comprender cómo debería ser diseñado e implementado el sistema. Sirve de apoyo para una primera aproximación del diseño.

2.5.1 Diagrama de Clases del Análisis

Un Diagrama de Clases de Análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. La propuesta de RUP para clasificar las clases que representan a este diagrama es: **Entidad**, para modelar información que posee larga vida y es a menudo persistente, **Interfaz**, para modelar la relación entre el sistema y los actores y **Control**, para coordinar la realización de uno o unos pocos CU coordinando las actividades de los objetos que implementan la funcionalidad del CU.

Diagrama de Clases del Análisis del CU Autenticar Usuario



Figura 4 Diagrama de Clases del Análisis del CU Autenticar Usuario

Diagrama de Clases del Análisis del CU Matricular Estudiante

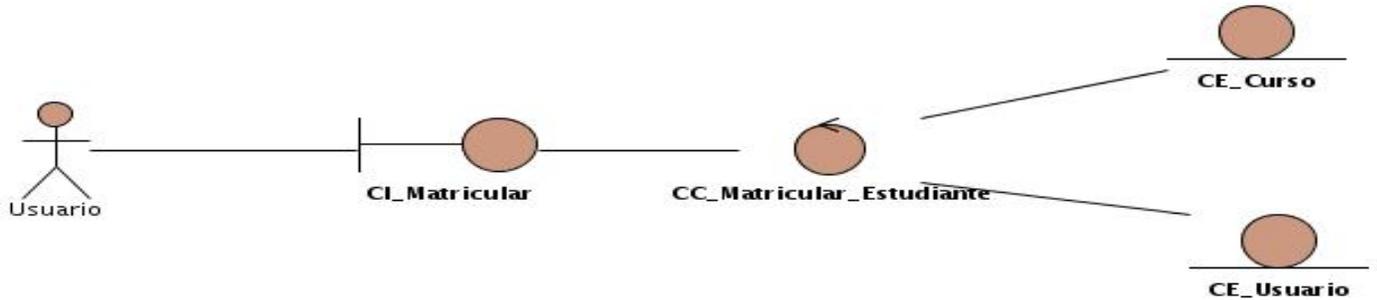


Figura 5 Diagrama de Clases del Análisis del CU Matricular Estudiante

Diagrama de Clases del Análisis del CU Gestionar Evento del Calendario

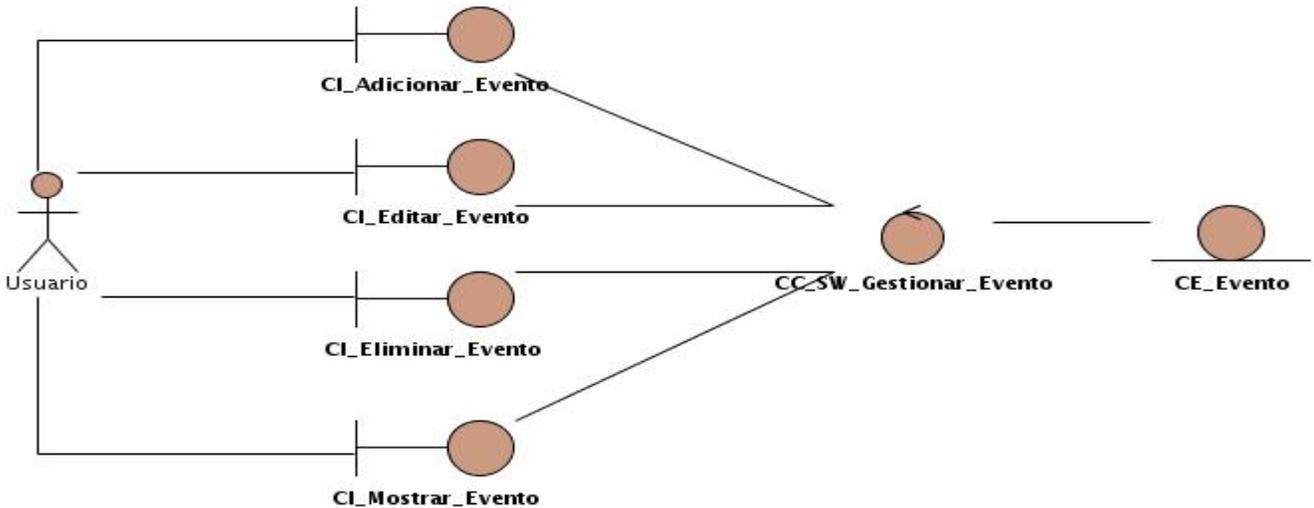


Figura 6 Diagrama de Clases del Análisis del CU Gestionar Evento del Calendario

El resto de los Diagramas de Clases del Análisis para las funcionalidades de gestión de Moodle se encuentran en los anexos del trabajo. (Ver Anexo 14 - Anexo 24)

2.5.2 Diagramas de Interacción

Se trata de un término genérico que se aplica a varios tipos de diagramas que hacen hincapié en las interacciones entre objetos. El patrón de interacción entre objetos se muestra en un Diagrama de interacción. Estos tienen dos formas de representarlos, basadas en una misma información subyacente pero, resaltando cada una un punto de vista de la misma: Diagramas de Secuencia, Diagramas de Colaboración. (63)

Un Diagrama de Secuencia muestra una interacción que está organizada como una secuencia temporal. En particular, muestra los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que intercambian, organizados en forma de una secuencia temporal. (64)

Un Diagrama de Colaboración modela los objetos y los enlaces significativos dentro de una interacción. Muestra una interacción organizada en torno a los objetos que efectúan operaciones. Su uso es para mostrar la implementación de una operación. (65)

Diagrama de Colaboración del CU Autenticar Usuario

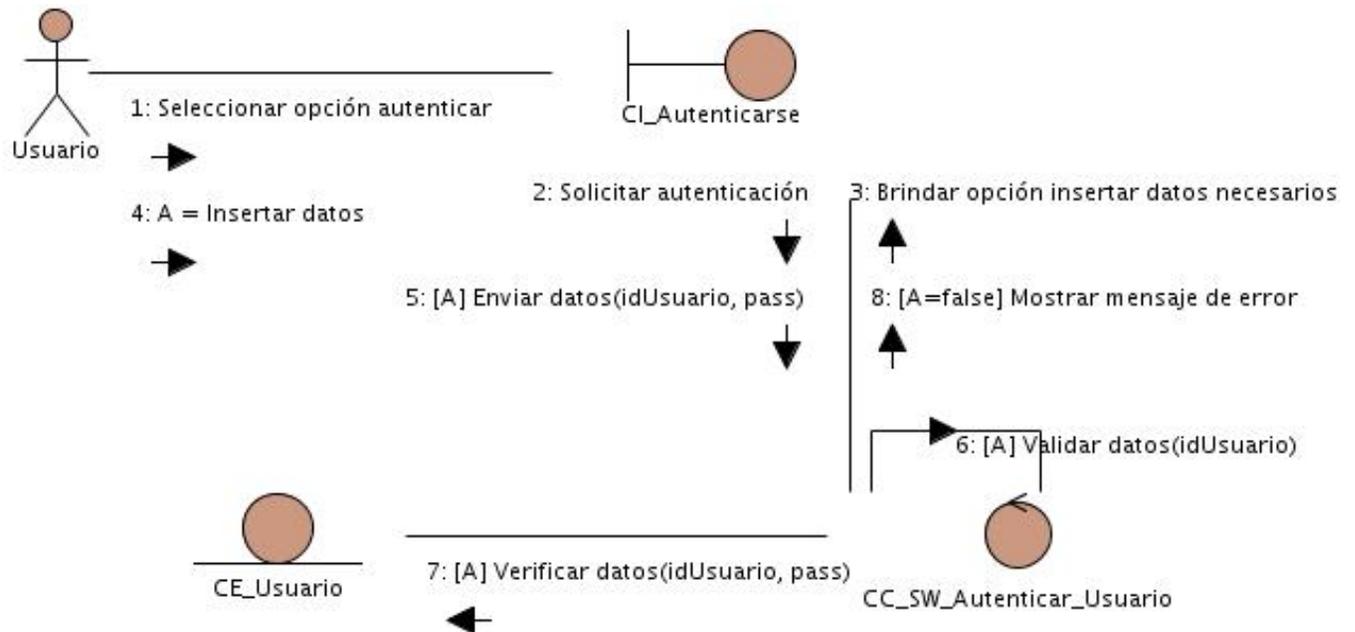


Figura 7 Diagrama de Colaboración del CU Autenticar Usuario

Diagrama de Colaboración del CU Buscar Curso

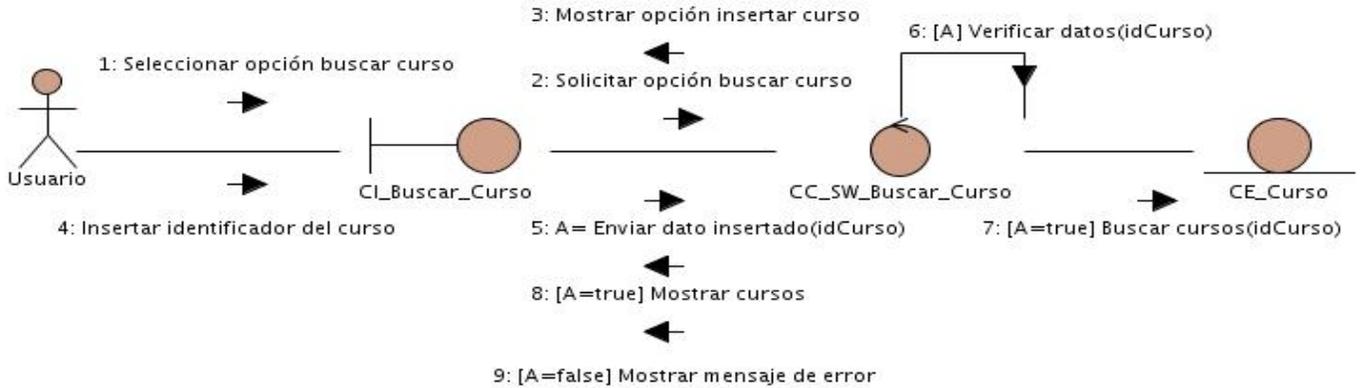


Figura 8 Diagrama de Colaboración del CU Buscar Curso

Diagrama de Colaboración del CU Mostrar Actividad Reciente

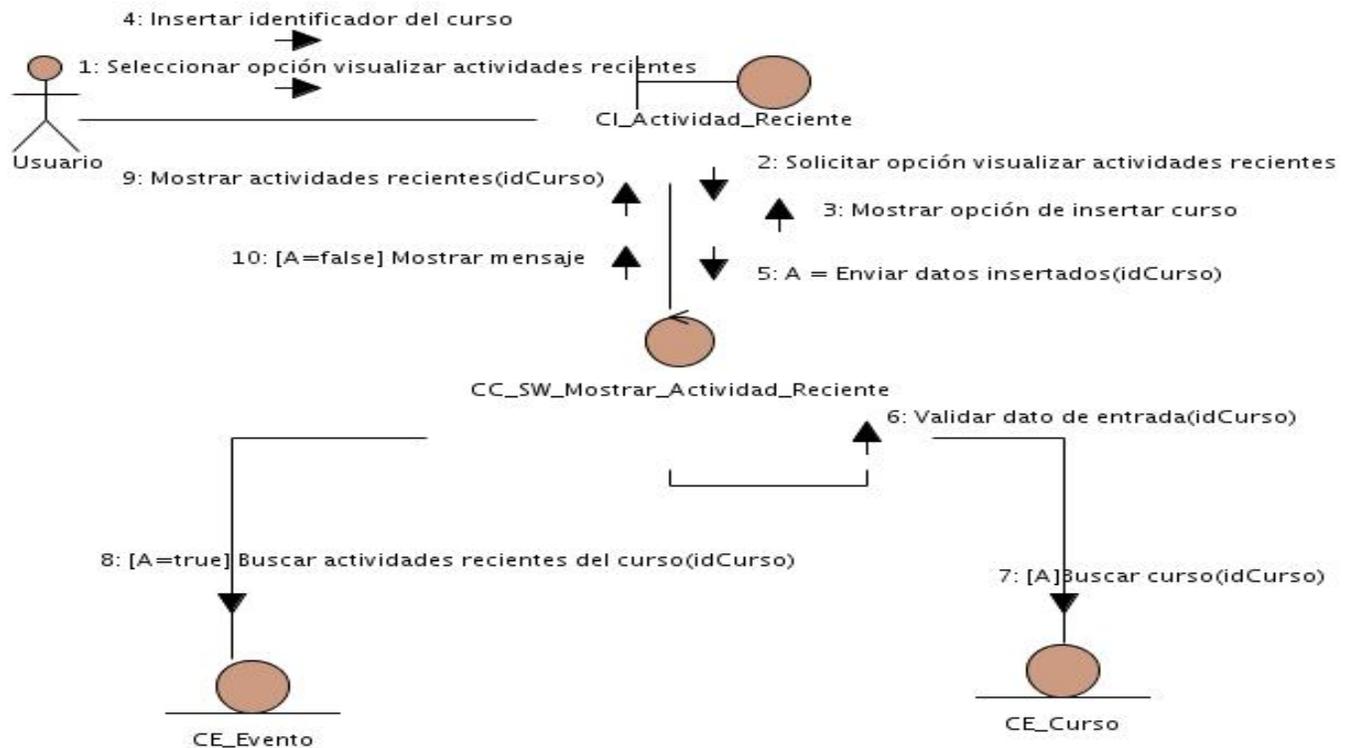


Figura 9 Diagrama de Colaboración del CU Mostrar Actividad Reciente

El resto de los Diagramas de Colaboración para las funcionalidades de gestión de Moodle se encuentran en los anexos del trabajo. (Ver Anexo 25 – Anexo 38)

2.6 Modelo del Diseño

El diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución y que prepara para la implementación y prueba del sistema. Pretende crear un plano del Modelo de Implementación, por lo que el grueso del esfuerzo está en las últimas iteraciones de elaboración y las primeras de construcción. (66)

2.6.1 Diagramas de Clases del Diseño

Los Diagramas de Clases del Diseño describen gráficamente las especificaciones de las clases de software y de las interfaces. Los Diagramas de Clases también son la base para un par de diagramas relacionados: los Diagramas de Componentes y de Despliegue. Este tipo de diagrama contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias.

Diagrama de Clases del Diseño del CU Autenticar Usuario

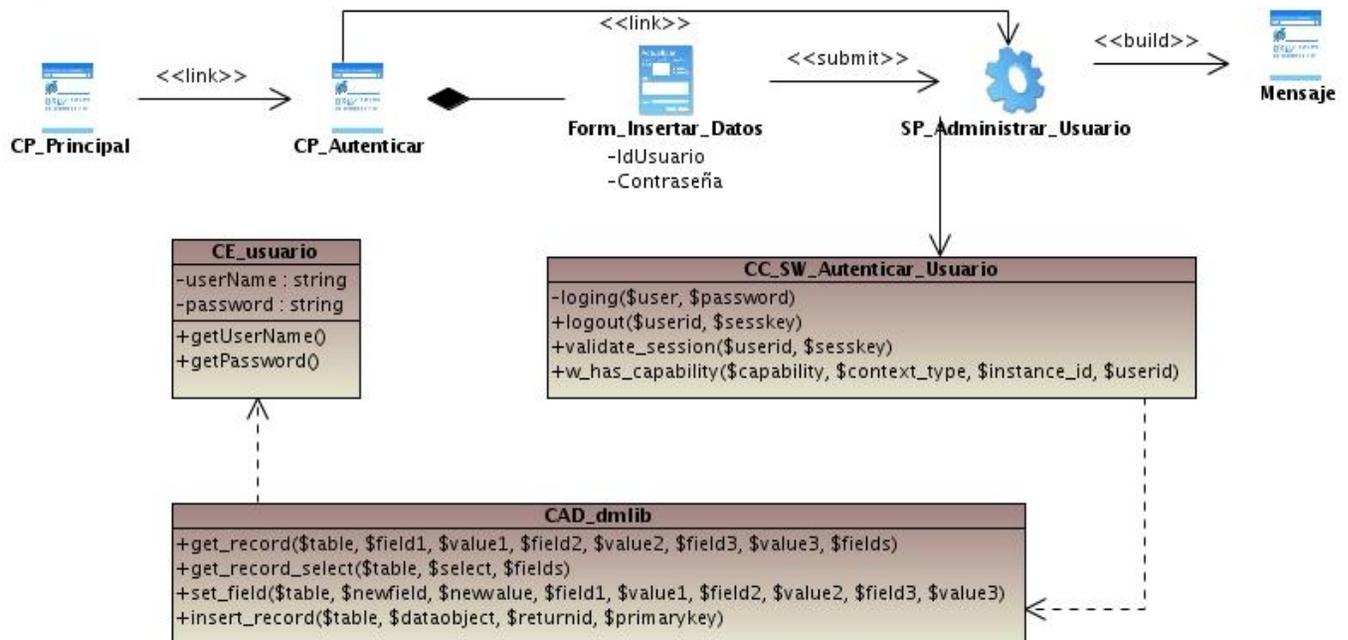


Figura 10 Diagrama de Clases del Diseño del CU Autenticar Usuario

Diagrama de Clases del Diseño del CU Buscar Curso

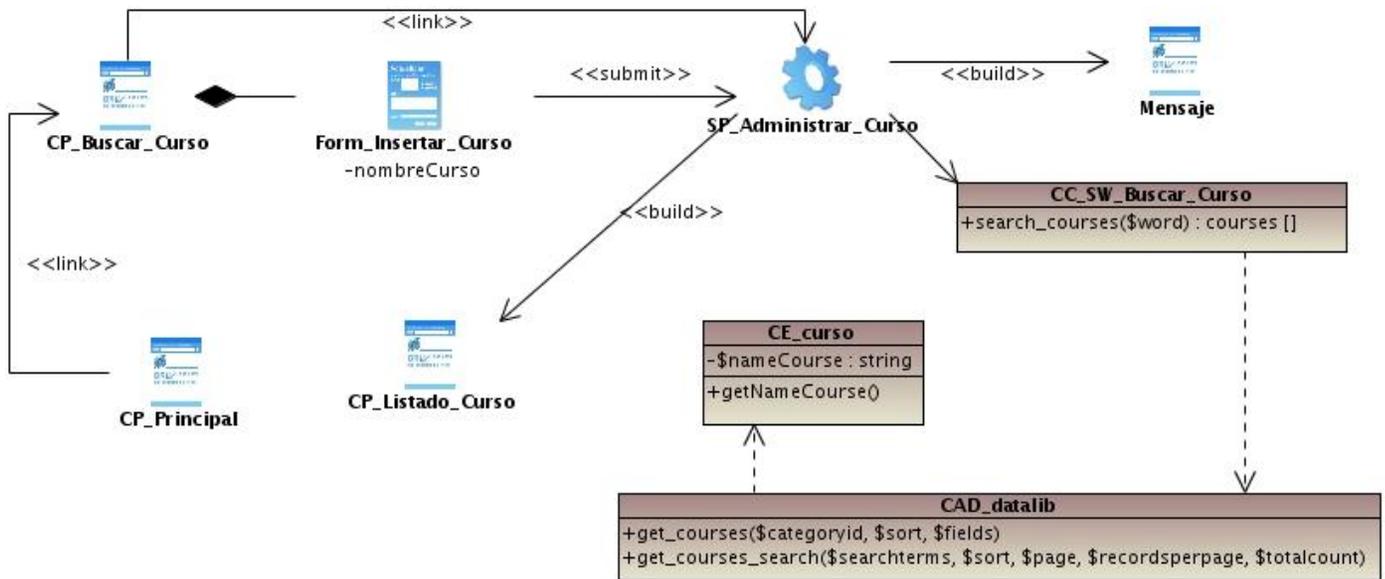


Figura 11 Diagrama de Clases del Diseño del CU Buscar Curso

Diagrama de Clases del Diseño del CU Mostrar Actividad Reciente

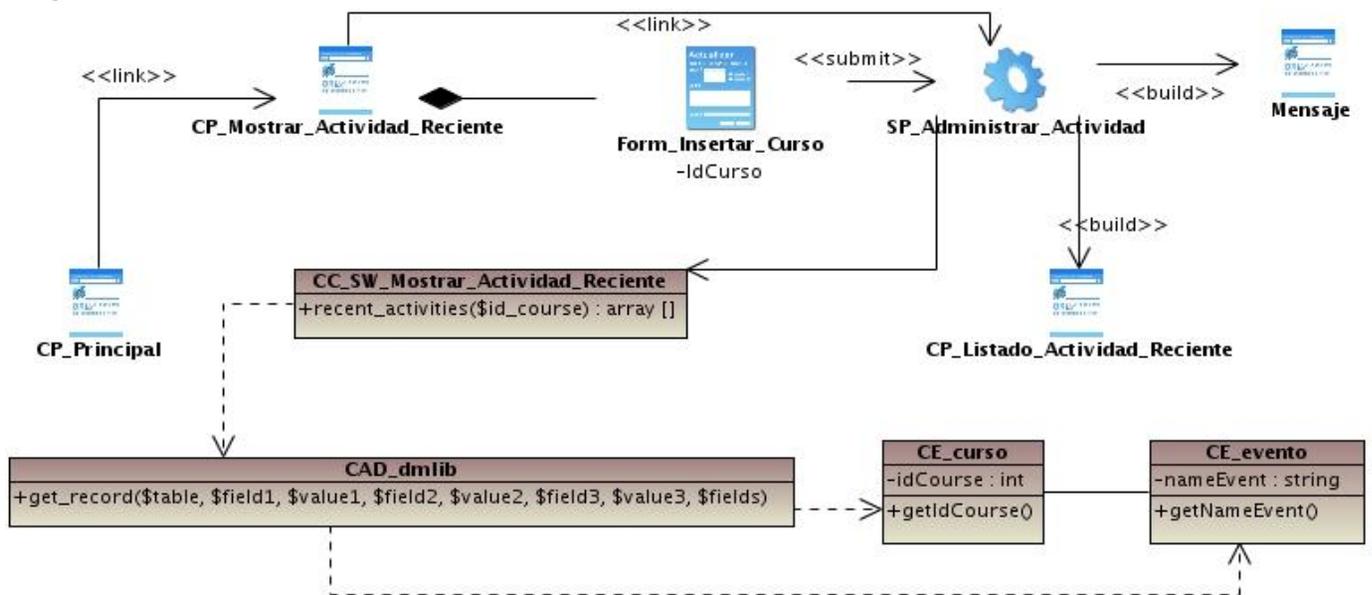


Figura 12 Diagrama de Clases del Diseño del CU Mostrar Actividad Reciente

El resto de los Diagramas de Clases del Diseño para las funcionalidades de gestión de Moodle se encuentran en los anexos del trabajo. (Ver Anexo 39 – Anexo 50)

2.7 Estructura de la base de datos

Para el desarrollo de la propuesta el trabajo se realizará con las bases de datos de las versiones 1.8.6 y 1.9.4 de Moodle las cuales cuentan con 186 y 198 tablas, respectivamente. Cada tabla de Moodle posee un campo numérico autoincremental (id), como clave primaria. Las tablas principales contienen instancias de cada módulo, poseen un identificador propio y un nombre para cada una de las instancias del módulo, lo que permite el correcto funcionamiento de la plataforma.

Con la implementación del CU Autenticar Usuario, uno de los SW implementado para las versiones de 1.8.6 y 1.9.4 de Moodle utilizadas en la universidad, se incorpora de manera automática una nueva tabla a la base de datos de la plataforma, la cual posee como nombre: **mdl_wservices_sessions**. En esta nueva tabla se insertan datos cuando los usuarios se autentican para obtener permiso a acceder a alguno de los SW implementados que requieran autenticación.

Tabla 4 Descripción de la tabla mdl_wservices_sessions

Nombre: mdl_wservices_sessions		
Descripción: Registra información referente a las sesiones que han sido creadas cuando los usuarios acceden a un SW.		
Atributo	Tipo	Descripción
id	bigint(10)	Identificador de la tabla.
Created	bigint(10)	Fecha de creación de la sesión.
expiry	bigint(10)	Fecha en la que el usuario se desconecta de la sesión.
Sessionkey	Varchar(32)	Identificador de la sesión.
Userid	bigint(10)	Identificador del usuario.
Verified	tinyint(1)	Verifica si el usuario que se identifica posee una sesión creada. Este campo puede tener los siguientes valores: <ul style="list-style-type: none">• “0, si la sesión no ha sido creada.”• “1, si la sesión ya ha sido creada.”

Para la implementación de los SW para plataforma Moodle, se utilizaron las tablas que están predefinidas para cada una de las funcionalidades de gestión (Calendario, Actividades Recientes, Listado de Usuarios,

Listado de Cursos y Matriculaciones) que posee la plataforma. Algunas tablas utilizadas fueron: User, la cual brinda información acerca de los usuarios inscritos en la plataforma; Event, permite obtener aquellos datos relacionados con los eventos creados por los usuario; Course, contiene información referente a los cursos que se encuentran en Moodle. Estas tablas se encuentran relacionadas con otras entidades que permiten el correcto funcionamiento de la base de datos de la plataforma, ejemplo de estas entidades: Role-Assignments, Role, Context y Course-Display. A partir de la relación de cada una de las entidades utilizadas, se genera el siguiente Modelo de Datos:

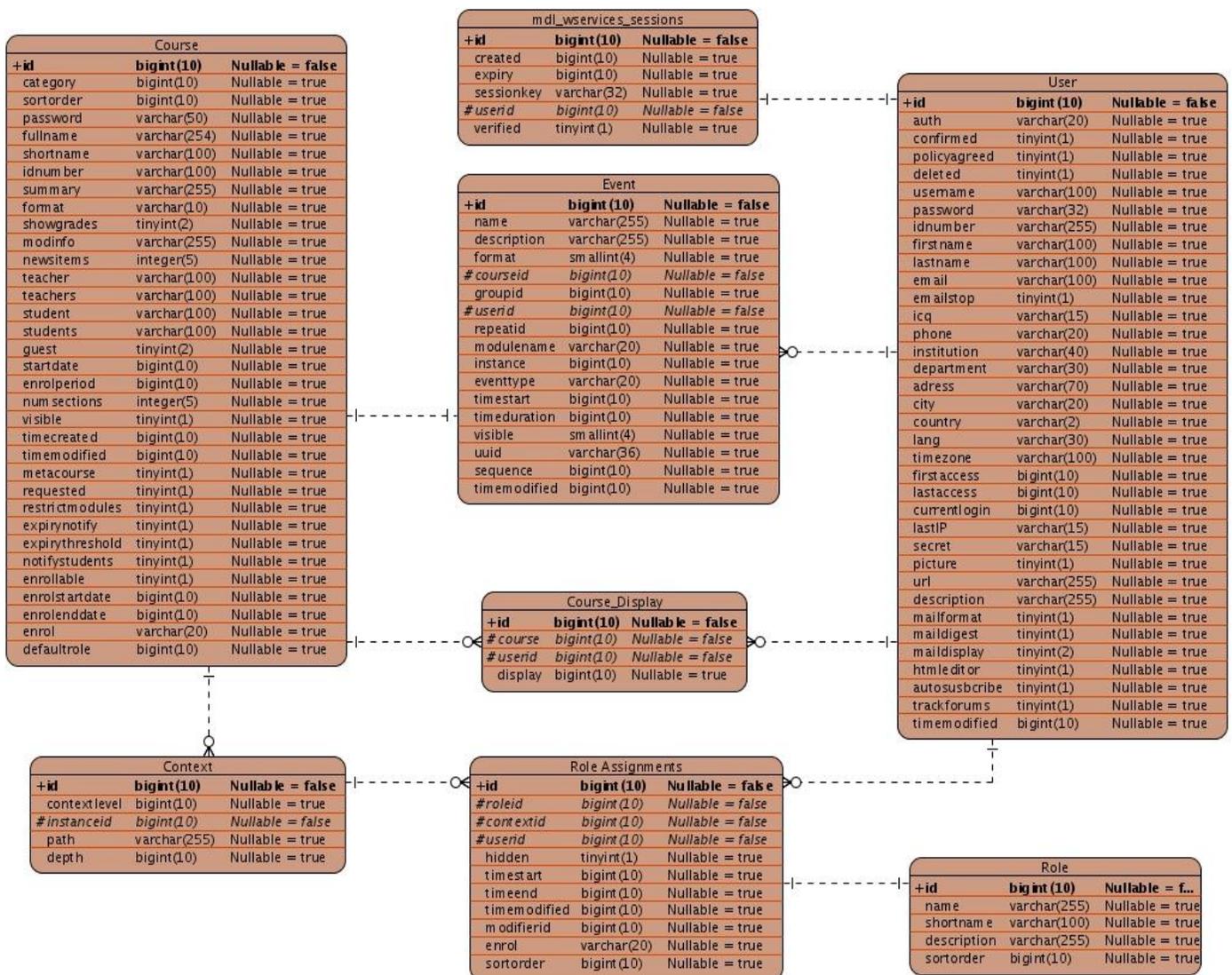


Figura 13 Modelo de datos

Las tablas que describen a cada uno de los atributos de las entidades utilizadas en la implementación de los SW para las funcionalidades de gestión de Moodle se encuentran en los anexos del trabajo. (Ver Anexo 51 – Anexo 54)

Conclusiones parciales

El Modelo de Dominio, permitió ofrecer una visión de los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde se desarrollarán los SW de Moodle. Constituyó además, la base para identificar los requisitos funcionales con los cuales debe cumplir la propuesta.

Se determinaron aquellas restricciones con las cuáles deben cumplir los SW de Moodle mediante la definición los requisitos no funcionales.

Los artefactos modelados durante el Análisis y el Diseño constituyen la base de la implementación exitosa del producto.

El Modelo de datos, permitió obtener un mayor conocimiento de las relaciones entre las clases persistentes que posee la plataforma Moodle para el desarrollo de la API de SW.

Capítulo 3: Implementación y prueba del sistema

Introducción

En este capítulo se describen los elementos necesarios a tener en cuenta para la implementación de los SW de las funcionalidades de gestión de Moodle, a partir del resultado obtenido del diseño. Para el logro este objetivo, se deben generar artefactos que hagan posible este proceso, Diagrama de Despliegue y Diagrama de Componentes, artefactos que permiten conocer la estructura de los SW implementados para la plataforma Moodle. Otro aspecto a tener en cuenta para el desarrollo de este capítulo son las Descripciones de Casos de Prueba para cada uno de los CU identificados, comprobando de esta forma los requerimientos con los que deben cumplir.

3.1 Diagrama de Despliegue

El Modelo de Despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los cuales representan objetos físicos con recursos computacionales. Se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. Estas conexiones se etiquetan con un estereotipo que expresan el tipo de conector o protocolo de comunicación o de red utilizado.

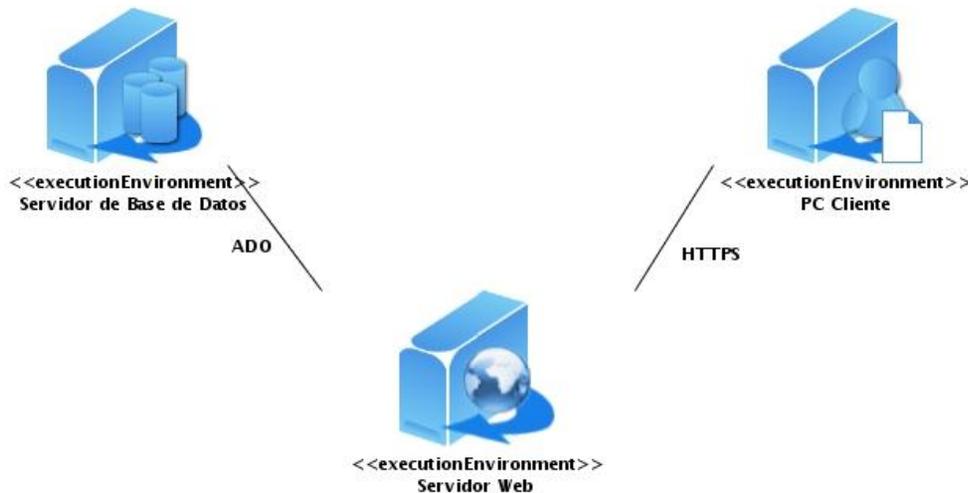


Figura 14 Diagrama de Despliegue

El Diagrama de Despliegue está compuesto por los nodos necesarios para obtener la información necesaria brindada mediante los SW de las funcionalidades de gestión de la plataforma Moodle. Para ello, cuenta con un servidor web Apache, donde se encuentra montado Moodle, el cual está conectado a servidor de base de datos PostgreSQL o MySQL, mediante el protocolo ADO (del inglés ActiveX Data Objects), donde su principal función es, la de realizar la solicitud de los datos a la base de datos. Los

usuarios podrán acceder a estos servicios mediante una computadora cliente que le permitirá el consumo de los mismos implementados utilizando el protocolo SOAP, que va a permitir la conexión al servidor web a través del protocolo HTTPS.

3.2 Diagrama de Componentes

Los Diagramas de Componentes muestran un conjunto de elementos del modelo de implementación tales como: componentes, subsistemas de implementación y sus relaciones. Se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias lógicas entre un conjunto de componentes de software, sean éstos: componentes de código fuente, librerías, binarios o ejecutables.

Los SW implementados para las funcionalidades de gestión de la plataforma Moodle, se encuentran agrupados en paquetes que ayudan a un mejor entendimiento de la estructura de los mismos. Dentro de estos paquetes se encuentran el de Gestión y Autenticación (Auth), los cuales, utilizan funciones de diferentes librerías que estaban definidas en la misma plataforma. A continuación se muestra el Diagrama de Componentes para los SW de Moodle, especificando cada una de las relaciones con las librerías utilizadas para la implementación de los mismos:

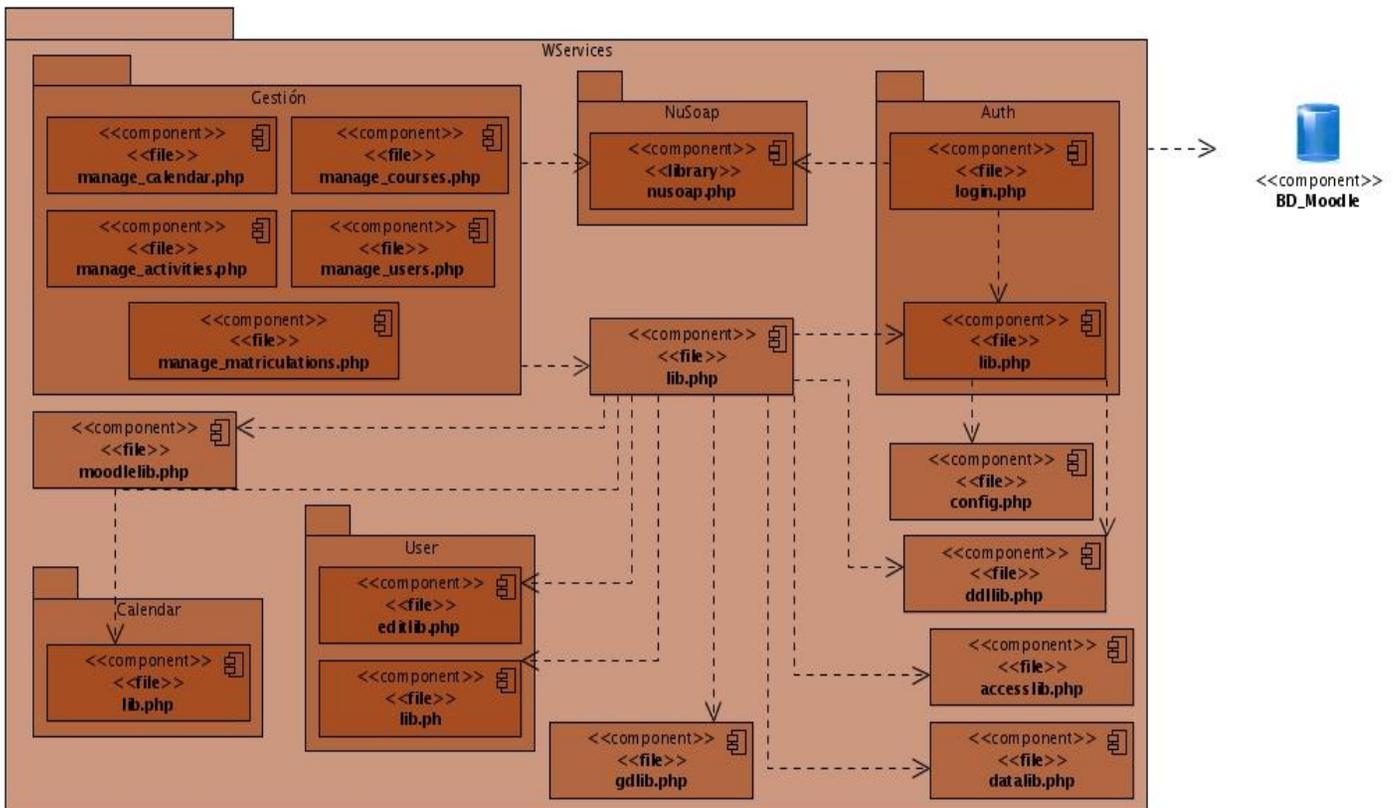


Figura 15 Modelo de Componentes

A continuación se describen cada uno de los paquetes que confeccionan el Modelo de Componentes:

Gestión: Este paquete contiene los ficheros que brindan las funciones de los SW y los ficheros que describen las características de estos.

Auth: Este paquete contiene los ficheros que permiten la autenticación de los usuarios, mediante un SW, para brindarles el acceso a los SW de las funcionalidades de gestión Moodle.

NuSoap: este paquete contiene los ficheros de la librería NuSoap que permite generar el WSDL de los SW de las funcionalidades de gestión de Moodle.

User: Este paquete contiene las librerías que permite acceder a algunas de las funciones relacionadas con el perfil de los usuarios de la plataforma Moodle.

Calendar: Este paquete contiene la librería que permite acceder a algunas de las funciones del calendario de la plataforma Moodle.

3.3 Tipos de pruebas

Para la comprobación del correcto funcionamiento de los SW para las funcionalidades de gestión de Moodle, se tuvo en cuenta el nivel de prueba de aceptación. Esta permite verificar que el software está listo y que puede ser utilizado por los usuarios finales, para ejecutar aquellas tareas y funciones para las cuales, fue construido el software. Dentro de los tipos de pruebas que se tuvo en cuenta para aplicarle al producto final, se encuentra la de funcionalidad, que toma en cuenta los siguientes requisitos:

Función: Pruebas que fijan su atención en la validación de las funciones, métodos, servicios, CU.

Seguridad: Asegurar que a los datos o a el sistema solamente es accedido por los actores deseados.

Volumen: Enfocada en verificar las habilidades de los programas para manejar grandes cantidades de datos, tanto como entrada, salida o residente en la BD.

Definido el nivel y los tipos de pruebas a utilizar, se decide utilizar el método de Caja Negra el cual se explica a continuación.

3.3.1 Método de Prueba

El método de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Estos pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Capítulo 3: Implementación y prueba del sistema

Una de las técnicas utilizadas en este método es la de Partición de Equivalencia: esta divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Para realizar las pruebas se elaboran las Descripciones de Casos de Prueba (DCP) que ayudan a que los probadores se cercioren del buen funcionamiento de los servicios implementados. A continuación se describen los DCP de cada uno de los CU identificados en el desarrollo de la propuesta:

Tabla de Descripción del Caso de Prueba del CU Autenticar Usuario

Tabla 5 Tabla de Descripción del Caso de Prueba del CU Autenticar Usuario

Id del escenario	Escenario	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1	Autenticar usuario	V	V	El usuario es autenticado y le permite entrar a la aplicación.	Seleccionar el servicio Autenticar usuario -> Insertar usuario y contraseña-> Seleccionar botón Entrar.
EC 2	Campos vacíos	V	I	Muestra un mensaje de error indicando que existen campos vacíos. Brinda la opción de reintentar la operación.	Insertar datos -> Seleccionar botón Entrar.
		I	V		
EC 3	Datos incorrectos	V	I	Muestra un mensaje de error indicando que existen datos incorrectos. Brinda la opción de reintentar la operación.	Insertar datos -> Seleccionar botón Entrar.
		I	V		

Tabla 6 Tabla de descripción de las variables de la DCP del CU Autenticar Usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Usuario	Campo de texto	No	El usuario debe ser válido y registrado en la base de datos.

2	Contraseña	Campo de texto	No	La contraseña debe ser válida y registrada en la base de datos.
---	------------	----------------	----	---

Las restantes DCP, así como las descripciones de las variables, para cada uno de los CU identificados para la implementación de los SW de las funcionalidades de gestión de la plataforma Moodle se localizan en los anexos del trabajo. (Ver Anexo 55 – Anexo 78)

3.3.2 Resultados de las pruebas

La construcción de un software tiene como objetivo satisfacer las necesidades de los clientes, por lo que es recomendable la evaluación del mismo mediante iteraciones de pruebas que se realicen, comprobando la calidad de este.

A continuación, se muestra un gráfico (Ver Figura 22) con el resultado de los casos de pruebas aplicados a cada CU correspondiente a la API implementada en el cual, se evidencia, la cantidad de No Conformidades (NC) detectadas durante las tres iteraciones que se realizaron en la revisión del producto. Las NC generadas se encuentran documentadas en el Expediente del proyecto nombrado Servicios Web de Moodle.

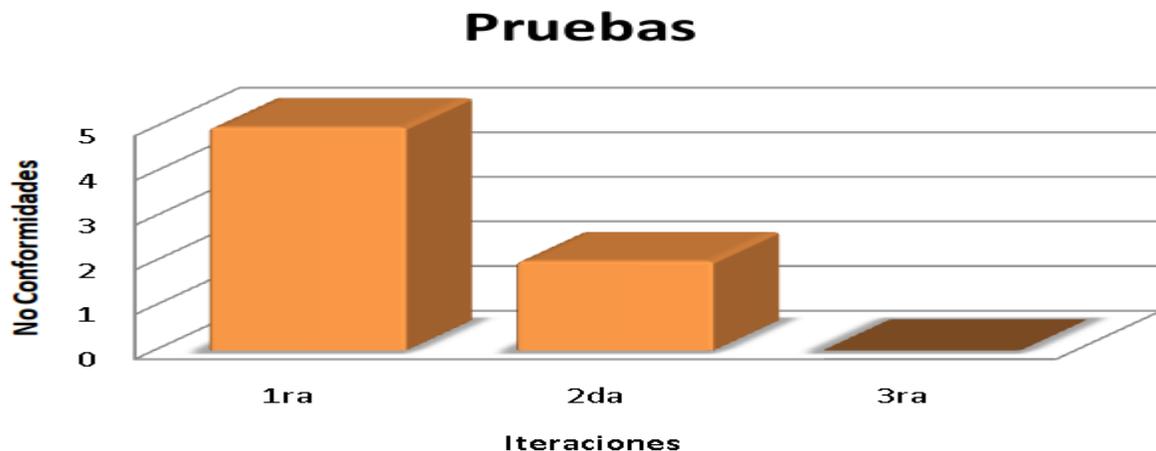


Figura 16 Gráfico de no conformidades.

Como se observa en el gráfico anterior se detectaron un total de 7 NC las cuales fueron resueltas de manera satisfactoria. Algunas de las NC detectadas de manera general se encuentran:

- No se mostraban cada uno de los mensajes que habían sido descritos en las descripciones de los CU identificados para la propuesta.

Capítulo 3: Implementación y prueba del sistema

- Fue necesario validar el cumplimiento de los roles establecidos por la plataforma Moodle para tenerlos en cuenta al brindar un servicio.

Brindarle la solución a las NC encontradas ayudó al mejoramiento de la propuesta para obtener una API libre de errores que satisface las necesidades de los usuarios.

Conclusiones parciales

Con la modelación del Diagrama de Despliegue se obtuvo la estructura de cada uno de los elementos de configuración del procesamiento y las conexiones entre los elementos que poseen los SW implementados.

El Diagrama de Componentes permitió conocer las relaciones de cada uno de los ficheros utilizados en la implementación del código de los SW.

Las DCP para los CU identificados, dio paso al desarrollo de las pruebas realizadas a la API, lo que permitió brindarle una mejor solución a las NC identificadas en las diferentes iteraciones.

Conclusiones Generales

Tras el estudio realizado a las funcionalidades de gestión de la plataforma Moodle, a los SW, entre otras investigaciones para el desarrollo de la propuesta, se concluye con que se cumplieron los objetivos del trabajo.

Además:

- El estudio realizado sobre los aspectos que se tuvieron en cuenta para la implementación de las funcionalidades de Moodle, en las versiones 1.8.x y 1.9.x, corroboró, la no utilización de estándares y protocolos abiertos que permiten la comunicación entre aplicaciones.
- La API diseñada como resultado de la investigación, permite la interoperabilidad entre plataformas y facilita el uso y consumo de las funcionalidades de gestión de Moodle 1.8.x y 1.9.x desde otras aplicaciones.
- Las pruebas desarrolladas a la propuesta dieron paso a la obtención de una API libre de errores y lista para satisfacer las necesidades de los usuarios.

Recomendaciones

Para dar continuidad al presente trabajo se recomienda:

- Poner en práctica la API de SW de las funcionalidades de gestión de Moodle, para las versiones 1.8.x y 1.9.x desarrollada en este trabajo.
- Integrar los resultados obtenidos, con la aplicación, a la comunidad de desarrolladores de Moodle.
- Continuar la implementación de la API de SW para otras funcionalidades que posee la plataforma Moodle.

Referencias Bibliográficas

1. **Mi Portal.** [En línea] 2011. [Citado el: 10 de enero de 2011.] http://www.miportal.edu.sv/Home/Tecnologia_en_el_Aula/sistemas+elearning.htm.
2. **Moodle. org.** [En línea] 4 de agosto de 2007. [Citado el: 14 de diciembre de 2010.] http://docs.moodle.org/es/Acerca_de_Moodle.
3. **ISIT Elearning.** [En línea] 2011. [Citado el: 14 de enero de 2011.] <http://www.grupoisit.com/noticias/moodle/186-ya-suena-a-moodle-20>.
4. **Eumed. net.** [En línea] 2011. [Citado el: 17 de diciembre de 2010.] <http://www.eumed.net/rev/ced/19/msr.htm>.
5. **Bernardez, Mariano L.** *Diseño, producción e implementación de E-learning*. Boomington, Indiana. : AuthorHouse, 2007.
6. **campusFormación. com.** [En línea] 2004. [Citado el: 15 de enero de 2011.] <http://www.campusformacion.com/glosario.asp>.
7. **Technology Evaluation Centers.** [En línea] 2008. [Citado el: 15 de diciembre de 2010.] <http://lms2.technologyevaluation.com/es/>.
8. **Integra-t.** [En línea] 2010. [Citado el: 16 de diciembre de 2010.] <http://www.softwarelibregalicia.es/e-learning/moodle>.
9. **Moodle. org.** [En línea] 25 de octubre de 2006. [Citado el: 15 de diciembre de 2010.] http://docs.moodle.org/es/A_favor_de_Moodle.
10. **Moodle. org.** [En línea] 22 de julio de 2009. [Citado el: 17 de diciembre de 2010.] <http://docs.moodle.org/es/Calendario>.
11. **Moodle. org.** [En línea] 22 de enero de 2009. [Citado el: 15 de diciembre de 2010.] http://docs.moodle.org/es/Detalles_de_las_preferencias_del_Calendario.
12. **Moodle. org.** [En línea] 15 de mayo de 2008. [Citado el: 14 de diciembre de 2011.] <http://docs.moodle.org/es/Participantes>.
13. **Moodle. org.** [En línea] 21 de febrero de 2007. [Citado el: 13 de diciembre de 2010.] http://docs.moodle.org/es/Usuarios_en_l%C3%ADnea.
14. **Moodle. org.** [En línea] 2 de febrero de 2007. [Citado el: 15 de diciembre de 2010.] http://docs.moodle.org/es/Formatos_de_cursos.
15. **Moodle. org.** [En línea] 20 de junio de 2008. [Citado el: 19 de diciembre de 2010.] <http://docs.moodle.org/es/Cursos>.

16. **Moodle. org.** [En línea] 17 de febrero de 2007. [Citado el: 14 de diciembre de 2010.] http://docs.moodle.org/es/Actividad_reciente.
17. **Moodle. org.** [En línea] 22 de julio de 2009. [Citado el: 16 de enero de 2011.] http://docs.moodle.org/es/Eventos_pr%C3%B3ximos.
18. **Moodle. org.** [En línea] 16 de abril de 2006. [Citado el: 14 de diciembre de 2010.] <http://docs.moodle.org/es/Matriculaci%C3%B3n>.
19. **IMS Global Learning Consortion. org.** [En línea] 15 de febrero de 2011. [Citado el: 17 de febrero de 2011.] <http://www.imsglobal.org/>.
20. **IMS Global Learning Consortion. org.** [En línea] 3 de enero de 2011. [Citado el: 16 de enero de 2011.] <http://www.imsglobal.org/es/index.html>.
21. **IMS GLobal Learning Consortion. org.** [En línea] 3 de enero de 2011. [Citado el: 16 de enero de 2011.] <http://www.imsglobal.org/profiles/lipinfo01.html>.
22. **Moodle. org.** [En línea] 3 de diciembre de 2010. [Citado el: 18 de diciembre de 2011.] http://docs.moodle.org/es/Notas_de_Moodle_1.8.
23. **Moodle. org.** [En línea] 29 de octubre de 2010. [Citado el: 17 de diciembre de 2010.] http://docs.moodle.org/es/Notas_de_Moodle_1.9.
24. **Moodle. org.** [En línea] 5 de enero de 2011. [Citado el: 15 de enero de 2011.] http://docs.moodle.org/es/Notas_de_Moodle_2.0.
25. **Directivos. edu.co.** [En línea] 2011. [Citado el: 10 de febrero de 2011.] <http://www.medellin.edu.co/sites/Educativo/Directivos/usodetics/BlogApropiacion/Lists/Entradas%20de%20blog/Post.aspx?ID=19>.
26. **Francis Pisani, Dominique Piotet.** *La alquimia de las multitudes*. Barcelona, España : Paidós Ibérica, S.A, 2009. 978-84-493-2196-2.
27. **Juan Desongles Corrales, Eduardo Antonio Ponce Cifredo, Ma. Luisa Garzón Villar.** *Técnicos de Soporte Informático*. España : Mad, SL, 2006. 84-665-5102-6.
28. **Arguiñiga, Jorge Lázaro Laporta y Marcell Miralles.** *Fundamentos de telemática*. Valencia : Universidad Politécnica de Valencia, 2005. 84-9705-913-1.
29. **elwebaster. com.** [En línea] 2008. [Citado el: 20 de febrero de 2011.] <http://www.elwebmaster.com/referencia/api-interface-de-programacion-de-aplicaciones>.
30. **Microsoft Download Center. com.** [En línea] 2011. [Citado el: 16 de diciembre de 2010.] http://docs.google.com/viewer?a=v&q=cache:xsXmRpr3FckJ:download.microsoft.com/download/c/2/c/2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf+La+Arquitectura+Orientada+a+Servicios+%28SOA%29+de+Microsoft+aplicada+al+mundo+real&hl=es&gl=cu&pid.

31. **Msdn.** [En línea] 2011. [Citado el: 13 de enero de 2011.] <http://msdn.microsoft.com/es-ar/library/bb507204.aspx>.
32. **Scrib. com.** [En línea] 2011. [Citado el: 13 de diciembre de 2010.] <http://www.scribd.com/doc/14464276/Manual-Completo-Servicios-Web>.
33. **Ignacio García Valcárcel, Eduardo Munilla Calvo.** *E-business Colaborativo*. España : Fundación Confemetal. ISBN 84-95428-98-9.
34. **Dewit, Oliver.** *Asp.net Programación Web con Visual Studio y Web Matrix*. s.l. : Ediciones ENI, 2003.
35. **W3C. org.** [En línea] 2011. [Citado el: 13 de enero de 2011.] <http://www.w3.org/TR/ws-gloss/> .
36. **W3C.** [En línea] 2002. [Citado el: 22 de enero de 2011.] <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>.
37. **w3school.com. com.** [En línea] 2011. [Citado el: 15 de enero de 2011.] http://www.w3schools.com/xml/xml_usedfor.asp.
38. **W3C. es.** [En línea] 16 de diciembre de 2008. [Citado el: 15 de enero de 2011.] <http://www.w3c.es/divulgacion/a-z/>.
39. **XML. com.** [En línea] 2010. [Citado el: 15 de enero de 2011.] <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>.
40. **Kalin., Martin.** *Java web services : up and running*. United States of America : Beijing ; Sebastopol, Calif. : O'Reilly, ©2009., 2009. ISBN: 9780596521127.
41. **Jim Webber, Savas Parastatidis y Ian Robinson.** *Rest in practice*. United States of America : Farnham ; Sebastopol, CA : O'Reilly, 2010., 2010. ISBN: 978-0-596-80582-1.
42. **Technet.Microsoft. net.** [En línea] 2011. [Citado el: 3 de junio de 2011.] <http://technet.microsoft.com/es-es/library/cc736697%28WS.10%29.aspx>.
43. **Source Forge. net.** [En línea] 2011. [Citado el: 29 de marzo de 2011.] <http://sourceforge.net/projects/nusoap/>.
44. **WSO2. org.** [En línea] WSO2 Inc, 2011. [Citado el: 8 de junio de 2011.] <http://wso2.org/project/wsf/php/1.2.0/docs/manual.html>.
45. **WSO2. org.** [En línea] WSO2 Inc, 2011. [Citado el: 8 de junio de 2011.] <http://wso2.org/project/wsf/php/1.2.0/docs/manual.html>.
46. **Msdn. com.** [En línea] 2011. [Citado el: 14 de enero de 2011.] <http://msdn.microsoft.com/en-us/library/Aa286530>.
47. **XML-RCP.Com. com.** [En línea] 2011. [Citado el: 16 de enero de 2011.] <http://www.xmlrpc.com/>.

48. **Sosnoski, Dennis.** IBM. *com*. [En línea] 7 de julio de 2009. [Citado el: 8 de junio de 2011.] <http://www.ibm.com/developerworks/ssa/library/j-jws6/>.
49. **Kioskea. net.** [En línea] 16 de octubre de 2008. [Citado el: 8 de junio de 2011.] <http://es.kioskea.net/contents/crypto/ssl.php3>.
50. **Dsic.** [En línea] 2011. [Citado el: 17 de diciembre de 2010.] <http://docs.google.com/viewer?a=v&q=cache:VDccWRFNAB0J:www.dsic.upv.es/asignaturas/facultad/lsi/doc/IntroduccionProcesoSW.doc+proceso+de+desarrollo+de+sw&hl=es&gl=cu&pid=bl&srcid=ADGEESgOMMrxkdXZZc9Pyaj7asmmbhOAMWmodFYRhHaGow5hCQvLafOFeMHZcEFMTyaOQF8vEgs2>.
51. **UTVM.** [En línea] 2011. [Citado el: 15 de diciembre de 2010.] <http://www.utvm.edu.mx/Organolnformativo/orgJul07/RUP.htm>.
52. **Lenguajes de Programación. com.** [En línea] 2009. [Citado el: 22 de enero de 2011.] <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
53. **Php. net.** [En línea] 15 de enero de 2011. [Citado el: 15 de enero de 2011.] <http://www.php.net/manual/es/intro-whatis.php>.
54. **Php. net.** [En línea] 15 de enero de 2011. [Citado el: 15 de enero de 2011.] <http://www.php.net/manual/es/intro-whatcando.php>.
55. **Universidad Politécnica de Valencia.** [En línea] 2011. [Citado el: 20 de febrero de 2011.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
56. **Zend. com.** [En línea] 2010. [Citado el: 29 de enero de 2011.] <http://www.zend.com/en/products/studio/>.
57. **Netbeans. org.** [En línea] 2011. [Citado el: 18 de enero de 2011.] <http://netbeans.org/features/ide/index.html>.
58. **Netbeans. org.** [En línea] 2011. [Citado el: 19 de enero de 2011.] <http://netbeans.org/features/web/web-services.html>.
59. **C&TA.** [En línea] 2011. [Citado el: 13 de enero de 2011.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
60. **Visual Paradigm. com.** [En línea] 2011. [Citado el: 13 de enero de 2011.] <http://www.visual-paradigm.com/product/>.
61. **GSInnova.** [En línea] 2011. [Citado el: 13 de enero de 2011.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
62. **Agile Modeling .** [En línea] 2010. [Citado el: 14 de febrero de 2011.] <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>.

63. **Agile Modeling.** [En línea] 2010. [Citado el: 18 de febrero de 2011.] <http://www.agilemodeling.com/artifacts/interactionOverviewDiagram.htm>.
64. **Agile Modeling .** [En línea] 2010. [Citado el: 18 de febrero de 2010.] <http://www.agilemodeling.com/artifacts/sequenceDiagram.htm>.
65. **Agile Modeling .** [En línea] 2010. [Citado el: 18 de febrero de 2011.] <http://www.agilemodeling.com/artifacts/communicationDiagram.htm>.
66. **eva.uci. cu.** [En línea] 2011. [Citado el: 10 de marzo de 2011.] http://eva.uci.cu/file.php/259/Curso_2010-2011/Semana_1/Conferencia_1/Materiales_basicos/Diseno.pdf.
67. **Moodle. org.** [En línea] 2010. [Citado el: 17 de diciembre de 2010.] <http://moodle.org/sites/>.
68. **Software Engineering Institute.** [En línea] septiembre de 2004. [Citado el: 12 de enero de 2011.] <http://www.sei.cmu.edu/reports/04tn036.pdf>.
69. **GSInnova.** [En línea] 2011. [Citado el: 13 de enero de 2011.] <http://www.rational.com.ar/herramientas/rup.html>.

Bibliografía Consultada

Universidad de Mursia. [En línea] 2011. [Citado el: 16 de enero de 2011.]
<http://www.um.es/ead/red/9/SGA.pdf>.

ired.org. [En línea] 8 de noviembre de 2005. [Citado el: 14 de diciembre de 2010.]
<http://www.ired.org/ev/mod/resource/view.php?id=364>.

Aprende. [En línea] 2011. [Citado el: 14 de enero de 2011.]
<http://welearn.cartagena.es/moodle/mod/book/view.php?id=509>.

Aprende. [En línea] 2011. [Citado el: 10 de enero de 2011.]
<http://welearn.cartagena.es/moodle/mod/book/view.php?id=509&chapterid=11>.

PC Magazine. [En línea] Ziff Davis, 2011. [Citado el: 2011 de enero de 13.]
http://www.pcmag.com/encyclopedia_term/0,2542,t=open+standards&i=48476,00.asp.

World Wide Web Consortion.es. [En línea] 6 de mayo de 2010. [Citado el: 3 de diciembre de 2010.]
<http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.

Moodle. org. [En línea] 17 de mayo de 2066. [Citado el: 17 de enero de 2011.]
http://docs.moodle.org/es/IMS_Enterprise.

Compute-rs.com. com. [En línea] 2011. [Citado el: 17 de diciembre de 2011.] <http://www.compute-rs.com/es/consejos-129101.htm>.

Glosario de términos

Estándar abierto: Es una especificación pública para desarrollar una tarea específica. Ejemplos: TCP, IP, HTML, entre otros.

Interoperabilidad: Se define como la capacidad de los sistemas o productos para trabajar con otros sistemas o productos posibilitando el intercambio de datos entre estos sin restricción de acceso o implementación.

Protocolo: Se denomina protocolo a un conjunto de normas y/o procedimientos para la transmisión de datos que ha de ser observado por los dos extremos de un proceso comunicacional (emisor y receptor). Estos protocolos «gobiernan» formatos, modos de acceso, secuencias temporales, entre otros.

Plugins: Es una aplicación adicional que se relaciona con otra para aportarle una nueva función y generalmente muy específica. Esta aplicación es ejecutada por la aplicación principal e interactúan por medio de la API.

Script: Es un programa simple, que se almacena en un archivo de texto plano. Son archivos interpretados, es decir, es capaz de analizar y ejecutar otros programas que se encuentren escritos en un lenguaje de alto nivel. Facilita la automatización de tareas a través de la creación de pequeñas utilidades.

Servidor web: Es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Servidor web Apache: Servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Es el servidor web por excelencia para Linux, posee un diseño modular, permitiendo a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.