



UNIVERSIDAD SIMÓN BOLÍVAR

Ingeniería de la Computación.

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
MANEJO DEL APRENDIZAJE A DISTANCIA: ÓSMOSIS 2**

Por

ANA GABRIELA DÍAZ
JOSÉ LORENZO RODRÍGUEZ
JOAQUÍN WINDMÜLLER

Proyecto de Grado

Presentado ante la Ilustre Universidad Simón Bolívar
como Requisito Parcial para Optar el Título de
Ingeniero en Computación

Sartenejas, 8 de junio de 2013



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL DEL PROYECTO DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MANEJO DEL APRENDIZAJE A
DISTANCIA: ÓSMOSIS 2**

Presentado por:
ANA GABRIELA DÍAZ
JOSÉ LORENZO RODRÍGUEZ
JOAQUÍN WINDMÜLLER

Este Proyecto de Grado ha sido aprobado por el siguiente jurado examinador:

Prof. Adelaide Bianchini
Jurado

Prof. Elías Tahhan
Jurado

Prof. Fidel Gil
Tutor

Sartenejas, 8 de junio de 2013

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MANEJO DEL APRENDIZAJE A DISTANCIA: ÓSMOSIS 2

Por

ANA GABRIELA DÍAZ
JOSÉ LORENZO RODRÍGUEZ
JOAQUÍN WINDMÜLLER

RESUMEN

Con la masificación del acceso a la tecnología, y en especial a Internet, la labor pedagógica se ha visto en la obligación de ponerse al día con las nuevas herramientas existentes, no solo con la intención de garantizar una máxima asimilación de los contenidos, sino también para extender la labor educativa a personas que, anteriormente, no tenían acceso al conocimiento.

La nueva tendencia en el uso de Internet, denominada Web2.0, apuntan a que los sitios más exitosos serán aquellos que logren aprovechar la inteligencia colectiva para generar contenidos con valor agregado. Pero para alcanzar ese conocimiento colectivo es necesario alcanzar masa crítica, para lograrlo es necesario ofrecer un servicio de alta calidad y en constante evolución.

Implementar esta nueva tendencia en un entorno de aprendizaje en línea es el principal objetivo del presente proyecto de grado. Partiendo de la plataforma existente en la Universidad Simón Bolívar, se plantea desarrollar su reestructuración de modo que se convierta en un repositorio navegable del conocimiento organizacional generado dentro del campus.

La meta de este trabajo fue la implementación de un Sistema de Manejo del Aprendizaje (LMS, por sus siglas en inglés), que ofrezca a la comunidad universitaria un entorno educativo con las características propias de la Web2.0 y a sus administradores un entorno de fácil mantenimiento que permita el rápido crecimiento y evolución por medio del desarrollo de nuevas funcionalidades.

Ósmosis2, como se denomina este proyecto, ofrecerá a los estudiantes y profesores una nueva forma de aprender y de dejar un legado a toda la comunidad.

Índice general

Índice de cuadros

Índice de figuras

Introducción

Con la masificación del acceso a la tecnología, y en especial a Internet, la labor pedagógica se ha visto en la obligación de ponerse al día con las nuevas herramientas existentes, no solo con la intención de garantizar una máxima asimilación de los contenidos, sino también para extender la labor educativa a personas que, anteriormente, no tenían acceso al conocimiento.

La Universidad Simón Bolívar no ha sido la excepción; aprovechando el acceso masivo a Internet como herramienta viable y de bajo costo para implantar una plataforma que sirva como complemento para los cursos que se dictan en sus aulas. En la actualidad, la institución cuenta con un portal educativo basado en un ambiente Web, la cual brinda apoyo al docente en la entrega de material complementario y acercamiento a los alumnos de los cursos que dicta.

Este es el camino que han tomado, con mayor o menor éxito, muchas otras entidades educativas alrededor del mundo. El software que apoya estas plataformas se ha ido desarrollando sobre la marcha, conforme las necesidades hayan ido demandando nueva funcionalidades, y los avances tecnológicos así lo hayan permitido. Pero en gran parte han sido desarrollados sin una concepción previa de cómo deben ser utilizados estos nuevos recursos para sacar el mayor provecho pedagógico a los mismos.

En la actualidad el área de mayor auge en Internet son las comunidades, en donde grupos de usuarios que se dedican a compartir contenidos multimedia los cuales pueden comentar, calificar y clasificar; fenómeno que algunos han llamado “Web 2.0”. Este modelo de interacción, en el cual el usuario es generador de contenidos, se ha probado a si mismo y a sus usuarios como muy útil, puesto que se aprovecha al máximo la capacidad de comunicación, casi instantánea, que brinda la tecnología para la generación de información valiosa a la que todos pueden acceder.

Surge entonces la siguiente pregunta, ¿por qué no usar este modelo para facilitar la labor educativa? Parte de la respuesta contempla el evaluar el uso que se le está dando a las plataformas que ya se poseen e intentar adaptarlas a este nuevo enfoque.

El presente proyecto nace de dicha necesidad de replantear las herramientas utilizadas actualmente por la Universidad Simón Bolívar, de manera que estas se transformen en facilitadores al

servicio de quienes van a utilizarlas y se adecuen lo mejor posible a las estructuras que se manejan en el ambiente educativo, abriendo paso, a su vez, al nuevo paradigma de la educación a distancia.

Para ello se requiere evaluar y seleccionar aquellos componentes que mejor se adapten al nuevo ambiente de aprendizaje y que a su vez estimulen no sólo la adquisición de conocimientos, sino la generación de los mismos. Con esto, es posible plantear un nuevo enfoque que supone hacer más énfasis en la comunidad de usuarios (estudiantes y profesores) que en los contenidos, sin descuidar la calidad de los últimos y que, además, busca reflejar, virtualmente, la interacción física que se observa en las aulas de clase.

Capítulo 1

Planteamiento del Problema

1.1. Antecedentes

La Web lentamente ha pasado de ser un mero mecanismo de entrega de contenidos, a ser una plataforma de intercambio virtualmente presente en cualquier computador. Esto la ha convertido en la opción clara para la implementación de soluciones vinculadas a la creciente necesidad de emplear nuevas tecnologías como medio de apoyo a la educación tradicional.

Sin embargo, la educación a distancia, también llamada *e-learning*, ha evolucionado a un ritmo mucho más lento que otras áreas en la Web, posiblemente por tratar de aplicar a la misma modelos que tradicionalmente han sido válidos para la educación presencial. Los sistemas de educación a distancia, en su mayoría, se encuentran atados a la filosofía lectura/escritura con la cual nació la Internet, haciendo que el intercambio de información sea unidireccional; paradigma que día a día se hace menos atractivo.

A esta tendencia se le agrega, como lo indica un estudio hecho en la USB, el desinterés de los docentes por conocer y aplicar estrategias de educación a distancia. De dicho estudio se concluyó que era necesario “*dar a conocer metodologías y herramientas para la creación de cursos a distancia*”, incluyendo la adopción de estándares que promuevan “*la portabilidad, accesibilidad e interoperabilidad de los recursos educativos [...] trayendo consigo el efecto de mejorar cualquier iniciativa que se desee en la creación de cursos a distancia*” (?)

Este desinterés es un fenómeno generalizado que afecta a las plataformas educativas, y que se hace mucho más evidente cuando se las compara con las actuales herramientas masivas de intercambio de información que han emergido de una nueva cultura en la Internet. Para muchas de estas plataformas la solución ha sido agregar nuevas capacidades para mantener la paridad con las nuevas tendencias. Sin embargo, este crecimiento ha generado una desconexión entre las herramientas y los beneficios pedagógicos que persiguen, y más importante aún, han olvidado mantener la motivación del alumno/docente para utilizar la aplicación.

1.2. Ósmosis 1.5

Para las organizaciones que contemplan al *e-learning* como una parte fundamental de su trabajo, las principales necesidades se encuentran en lograr un proceso sostenido de creación de información, y la evolución de la plataforma ante los requerimientos de sus usuarios. De allí se desprende que las funcionalidades que se requieren para un sistema de este tipo sean la transferencia rápida y efectiva de conocimiento, el enriquecimiento de la información a través de su contexto y la habilidad de fabricar contenidos educativos en menos tiempo y con el menor costo posible ?).

Esas mismas necesidades fueron las que llevaron a la Dirección de Servicios Multimedia (DSM) a iniciar la construcción de una plataforma educativa. A continuación se presenta una sinopsis de la evolución que siguió el proyecto hasta llegar a la plataforma con la que cuenta actualmente.

Este proyecto tiene su origen en una necesidad personal del jefe de la sección Web de la DSM, Hermes Rodríguez, mientras cursaba la Especialización en Telecomunicaciones: habilitar un espacio para compartir la mayor cantidad de información posible con sus compañeros de clase y docentes. Así surge esta primera plataforma de *e-learning*, un desarrollo sencillo con funcionalidad limitada.

Posteriormente se utiliza el sistema de manejo de contenidos Claroline para hacer una propuesta formal, sin embargo luego de realizar un estudio pedagógico y tecnológico, se selecciona Dokeos – un software de código abierto regido por la licencia GPL – como herramienta para desarrollar la plataforma.

El desarrollo de nuevas funcionalidades se basó en los requerimientos específicos de los usuarios. Dichos aditamentos fueron propuestos a los desarrolladores de Dokeos bajo el espíritu de la GNU/GPL. Sin embargo dichos aportes no fueron implementados en la línea principal de desarrollo del paquete de *e-learning*, por lo que la USB terminó, sin querer, con un nuevo producto que lentamente se alejó de la línea de desarrollo principal de Dokeos. Ante esta realidad se decidió bautizar a la nueva herramienta “Ósmosis” en referencia a su origen: facilitar la transferencia de conocimiento entre los actores que la utilizan.

Desde entonces Ósmosis ha sido promocionada desde de la Universidad Simón Bolívar para convertirla en un producto que permita generar redes de conocimiento a nivel nacional.

1.2.1. Aula Virtual de la Universidad Simón Bolívar

La única implantación conocida, en producción, de Ósmosis se encuentra en la Universidad Simón Bolívar y es la plataforma de aprendizaje a distancia, conocida como Aula Virtual, que sirve tanto a los cursos regulares como a los de extensión y a algunos liceos de la zona. En números, a la fecha (29/01/2008) Ósmosis ofrece su servicio a **22.311 usuarios**, de los cuales **647 son docentes**. Se manejan **1.274 cursos** (882 públicos y 392 privados) en **35 categorías**. El **curso más antiguo** fue creado el **13 de Septiembre de 2004** y el mas reciente data del **29 de Enero de 2008**.

La cantidad de usuarios del sistema demuestra que además de los miembros de la comunidad de la Universidad Simón Bolívar también participan otras entidades educativas, por ejemplo, el curso con más estudiantes inscritos (**594**) es Introducción a la Informática y se dicta en la UNEFA (Universidad Experimental Politécnica de la Fuerza Armada Bolivariana).

Deficiencias

A pesar del éxito que ha tenido la implementación de Ósmosis en las Aulas Virtuales, es importante reconocer que Ósmosis tiene su origen en Dokeos, el cual carecía de una buena documentación y ha heredado código e ideas de distintos Sistemas de Gestión del Aprendizaje (SGA o LMS, por sus siglas en inglés) por lo que actualmente presenta un código fuente difícil de mantener.

A nivel técnico, Ósmosis está desarrollado en PHP, sin la utilización de algún *framework*, y carece de una separación clara entre las capas lógica/datos/vista lo cual, en el mejor de los casos,

hace que el desarrollo de nuevas funcionalidades o la solución de problemas sea ineficiente.

1.3. Ósmosis2

Este proyecto, al que se denominará indiscriminadamente Ósmosis2 y Ósmosis2, surge como consecuencia de lo expuesto anteriormente, con la finalidad de contemplar y llevar a cabo el diseño e implementación de una nueva versión de Ósmosis, en la cual sea posible incorporar nuevas tecnologías e integrarlas con los recursos pedagógicos que los docentes demandan, esto con la finalidad de que la plataforma sea empleada como herramienta principal para el aprendizaje.

Así mismo, y en función del crecimiento de Ósmosis, uno de los requisitos para esta nueva versión es lograr un código mantenible, lo que implica la menor cohesión posible entre sus capas. Es decir, la reingeniería de la plataforma para adaptarla a las necesidades pedagógicas y tecnológicas actuales.

A continuación se presentan los objetivos generales y específicos del proyecto en cuestión:

1.3.1. Objetivos Generales

- Diseñar y desarrollar un Sistema de Gestión del Aprendizaje fundamentado en corrientes pedagógicas y en los principios de la Web 2.0.
- Desarrollar una plataforma modular de aprendizaje a distancia que se adapte a las necesidades pedagógicas actuales.
- Evaluar las características de la versión actual de Ósmosis.

1.3.2. Objetivos Específicos

- Realizar una investigación referente a la definición de LMS y sus características.
- Estudiar los enfoques pedagógicos de mayor uso contemplados en los LMS
- Analizar y comparar los distintos LMS existentes.

- Realizar el levantamiento y análisis de los requerimientos de la plataforma, incluyendo estándares existentes en el diseño de contenidos para la educación.
- Establecer, a partir del análisis de los LMS y los requerimientos detectados, las funcionalidades que contemplará la plataforma de Ósmosis2
- Evaluar y seleccionar una plataforma de desarrollo sobre la cual se desarrollará el LMS (lenguajes, frameworks, manejadores de bases de datos, etc).
- Diseñar una interfaz gráfica que permita reflejar la nueva visión de Ósmosis

Capítulo 2

Marco Teórico

2.1. Educación en Línea

La educación en línea se refiere a a educación virtual, distribuída a través de Internet, la Web o cualquier otra forma de educación desarrollada con la mediación de las computadoras. Se caracteriza por:

- Separación del maestro y el aprendiz, lo que la diferencia de la educación presencial.
- La influencia de una organización educativa, lo que la diferencia del aprendizaje autodidacta y de la tutoría privada.
- El uso de una red de computadores para presentar o distribuir los contenidos educativos.
- La disponibilidad de una vía de comunicación a través de una red de computadores de modo que los estudiantes se beneficien de la comunicación entre ellos y con el profesor.

A pesar de que el término *e-learning* se usa como sinónimo de educación en línea, es importante aclarar que son distintos y que “educación en línea” es más amplio: *e-learning* sólo se refiere a la entrega de los contenidos y la retroalimentación automatizada ante las actividades desarrolladas por el estudiante y no a la comunicación entre aprendiz y tutor. (?)

2.1.1. Learning Management Systems (LMS)

Learning Management Systems o Sistema de Gestión del Aprendizaje, es un término amplio usado para referirse a sistemas que organizan y proveen acceso a servicios de aprendizaje en línea

para estudiantes, educadores y administradores. Estos servicios incluyen control de acceso, fuente de contenidos de aprendizaje, herramientas de comunicación y organización en grupos de usuarios. También se les conoce como plataformas de aprendizaje (?).

Objetivos

Según ?, algunos de los objetivos de un Sistema de Gestión del Aprendizaje (LMS) son:

- Permitir la reutilización y redistribución de los contenidos.
- Ofrecer alta disponibilidad, y facilidad de uso mientras se mantiene la escalabilidad, la seguridad y la estabilidad.
- Integrar aprendizaje en los salones con aprendizaje en línea.
- Medir la efectividad de las iniciativas de aprendizaje mediante el seguimiento del progreso de los estudiantes y su desempeño en los distintos tipos de actividades.
- Servir de contenedor de cursos.
- Facilitar la comunicación y el trabajo colaborativo entre profesores y estudiantes.

(?)

Características

En cuanto a las características básicas que debe tener un LMS son destacables:

- Administración de usuarios, roles, profesores, herramientas y generación de reportes.
- Calendario del curso.
- Mensajes y notificaciones a estudiantes.
- Autoevaluaciones y pruebas que permitan manejar el desempeño del estudiante antes y después de las evaluaciones
- Puntuación de actividades del curso, individual y por grupos, incluyendo lista de espera.
- Mostrar calificaciones y actas de notas.

- Distribución del curso a través de la Web o por medio de la misma.

(?)

2.2. Corrientes Pedagógicas

Las corrientes pedagógicas son distintas teorías de pensamiento, surgidas a lo largo de la historia, que intentan explicar el proceso de aprendizaje y formación del ser humano, las más importantes son:

- Conductismo
- Cognitivismo
- Constructivismo
- Constructivismo social
- Conectivismo

2.2.1. Conductismo

Según ¹, en el conductismo la experiencia precede al conocimiento, en otras palabras, define el aprendizaje como un cambio observable en el comportamiento producto de una relación “estímulo - respuesta”. Los procesos internos tales como el pensamiento y la motivación son incognoscibles (?), no pueden ser observados ni medidos directamente por lo que no son relevantes a la investigación científica del aprendizaje. El aprendizaje se constata por medio de la observación de un cambio en el comportamiento. Si no hay cambio observable no hay aprendizaje.

El conductismo aprueba el uso de refuerzos para fortalecer conductas apropiadas y su desuso para debilitar las no deseadas. La asignación de calificaciones, recompensas y castigos son también aportaciones de esta teoría.

Los principios de las ideas conductistas pueden aplicarse con éxito en la adquisición de conocimientos memorísticos que suponen niveles primarios de comprensión, como por ejemplo el aprendizaje de las capitales del mundo o las tablas de multiplicar. Sin embargo esto presenta una limitación importante: la repetición no garantiza asimilación de la nueva conducta, sino sólo su ejecución. Por ejemplo, una persona puede aprender las tablas de multiplicar pero no sabrá cuando debe hacer uso de ellas. Esto indica que la situación aprendida no se puede extrapolar fácilmente a

otras situaciones.

En este paradigma, el estudiante es un individuo cuyo aprendizaje se puede modificar por medio de un reajuste de los recursos educativos de tal manera que éste adquiriera las actitudes académicas deseadas.

El maestro es el encargado de ejecutar dichos reajustes así como aplicar los estímulos necesarios para la enseñanza.

2.2.2. Cognitivismo

El paradigma cognitivista sustenta al aprendizaje como un proceso en el cual se sucede la modificación de significados de manera interna, producido intencionalmente por el individuo como resultado de la interacción entre la información procedente del medio y el sujeto activo. Dicha perspectiva surge a finales de 1960 como una transición entre el paradigma conductista y las actuales teorías psicopedagógicas.

“Al cognitivismo le interesa la representación mental y por ello las categorías o dimensiones de lo cognitivo: la atención, la percepción, la memoria, la inteligencia, el lenguaje, el pensamiento y para explicarlo puede, y de hecho acude a múltiples enfoques, uno de ellos es el de procesamiento de la información; y cómo las representaciones mentales guían los actos (internos o externos) del sujeto con el medio, pero también cómo se generan (construyen) dichas representaciones en el sujeto que conoce.” (?)

El cognitivismo es, de manera simplificada, el proceso independiente de decodificación de significados que conduzcan a la adquisición de conocimientos a largo plazo y al desarrollo de estrategias que permitan la libertad de pensamiento, la investigación y el aprendizaje continuo en cada individuo, lo cual da un valor real a cualquier cosa que se desee aprender. De aquí entonces se desprende el paradigma del Cognitivismo, “un marco global de referencia para el crecimiento y desarrollo personal” (?)

2.2.3. Constructivismo

Esta corriente pedagógica afirma que el conocimiento de todas las cosas es un proceso mental del individuo, que se desarrolla de manera interna conforme obtiene información y se relaciona con su entorno.

Para el constructivismo, el aprendizaje es un proceso en el cual el estudiante construye activamente nuevas ideas o conceptos basados en conocimientos presentes y pasados. En otras palabras, “el aprendizaje se forma construyendo nuestros propios conocimientos desde nuestras propias experiencias” (?)

A pesar de ser el aprendiz quien construye su aprendizaje, el profesor juega un rol importante en el proceso, actuando como facilitador y animando a los estudiantes a solucionar problemas reales o simulaciones. Por lo general el aprendizaje se hace por medio de un proceso social de colaboración entre los estudiantes y bajo esta óptica, el proceso de aprendizaje no es de “todo o nada” sino que los estudiantes desarrollan nuevos conocimientos a partir de los que ya poseen. Esto significa que el docente debe constatar que el aprendizaje efectivo logrado por el aprendiz es el esperado así como fomentar la conexión entre los estudiantes y plantear situaciones para estimular el razonamiento.

El constructivismo propone una metodología de aprendizaje, más no propone los medios para llevar a cabo dicho aprendizaje. Es decir, describe cómo ocurre pero no cómo se logra el aprendizaje.

2.2.4. Constructivismo social

El constructivismo social en educación y teoría del aprendizaje estudia la forma en que el ser humano aprende a la luz de su situación social y de la comunidad en la que aprende.

Uno de los puntos focales del constructivismo es descubrir las maneras en las que los individuos y grupos participan en la creación de la percepción de su realidad social. El estudio de la aparición de fenómenos sociales, su institucionalización y como posteriormente se tornan en tradiciones. La realidad construida socialmente, por los individuos y su interpretación de sus conocimientos, es un proceso dinámico y constante.

En décadas recientes, los teóricos del constructivismo han extendido en enfoque tradicional del aprendizaje para referirse a las dimensiones colaborativas y sociales del aprendizaje.

?, propone extender el constructivismo social de modo que se tome en cuenta la sinergia entre los más recientes avances de la tecnología de la información. Introduce el término Constructivismo Comunal, en el cual “los estudiantes no solamente pasan a través de un curso, como el agua a través de una tubería; sino que dejan su propia huella en el proceso de aprendizaje, en su escuela o universidad e idealmente en su disciplina. Esto resultará en un beneficio para el curso o la institución, pero más importante es el beneficio que genera a los estudiantes.” Avances como los blogs, wikis y podcasts, aumentan el potencial comunicacional de los individuos, lo cual permite que el conocimiento construido individualmente redunde en un beneficio a los demás aprendices.(?)

El constructivismo social expone que el ambiente de aprendizaje óptimo es aquel donde una interacción dinámica entre los instructores y los alumnos, con actividades que proveen oportunidades para los alumnos de crear su propia verdad gracias a la interacción con lo demás. Esta teoría, por lo tanto, enfatiza la importancia de la cultura y el contexto para el entendimiento de lo que está sucediendo en la sociedad y para construir conocimiento basado en este entendimiento (?).

Según ?, el constructivismo social plantea dos principios fundamentales cuya aplicación tiene consecuencias tanto en el estudio del desarrollo cognitivo como en la práctica educativa:

- El conocimiento no se recibe pasivamente sino que es construido activamente por el sujeto cognitivo.
- La función del aprendizaje es adaptable. Sirve para la organización del mundo de la experiencia y no para el descubrimiento de una realidad ontológica (?)

2.2.5. Conectivismo

El conectivismo es una teoría del aprendizaje para la era digital que ha sido desarrollada por George Siemens basado en el análisis de las limitaciones del conductismo, el cognitivismo y el constructivismo, para explicar el efecto que la tecnología ha tenido sobre la manera en que actualmente vivimos, nos comunicamos y aprendemos. (?)

El conectivismo es la integración de los principios explorados por la teorías del caos, redes neuronales, complejidad y auto-organización. El aprendizaje, según ?, es un proceso que ocurre dentro de una amplia gama de ambientes que no no están necesariamente bajo el control del individuo, es por esto que el conocimiento puede residir fuera del ser humano, por ejemplo dentro de una organización o una base de datos, y se enfoca en la conexión especializada en conjuntos de información que nos permita aumentar cada vez más nuestro estado actual de conocimiento.

Esta teoría es conducida por el entendimiento de que las decisiones están basadas en las transformación acelerada de los basamentos. Continuamente nueva información es adquirida dejando obsoleta la anterior. En este contexto, es vital adquirir la capacidad para discernir entre lo importante y lo trivial, así como la capacidad para reconocer cuando esta nueva información altera las decisiones tomadas en base a la información pasada.

El punto de inicio del conectivismo es el individuo. El conocimiento personal se hace de una red, que alimenta de información a organizaciones e instituciones, que a su vez agregan nueva información en la misma red, lo cual termina proveyendo nuevo aprendizaje al individuo. Este ciclo de desarrollo del conocimiento permite a los aprendices a mantenerse actualizados en el campo en el cual han formado conexiones.

Principios del conectivismo

- El aprendizaje y el conocimiento yacen en la diversidad de opiniones.
- El aprendizaje es el proceso de conectar nodos o fuentes de información.
- No solo los humanos aprenden y el conocimiento puede residir fuera del ser humano.
- La capacidad de aumentar el conocimiento es más importante que la información conocida.
- Nutrir y mantener las conexiones es necesario para facilitar el aprendizaje continuo.
- La habilidad para ver las conexiones entre los campos, ideas y conceptos es primordial.
- La información actualizada y precisa es la intención de todas las actividades del proceso conectivista.

- La toma de decisiones es en sí misma un proceso de aprendizaje. Escoger qué aprender y el significado de la información entrante es visto a través del lente de una realidad cambiante: es posible que lo que hoy es correcto mañana esté errado bajo la nueva información que se recibe.

Con respecto al primer punto, ? indica que una red de aprendizaje puede sostener puntos aparentemente contradictorios dado que lo que hoy es cierto mañana podría no serlo ante algún cambio en la base. Y es esta diversidad la aumenta la posibilidad de realizar decisiones acertadas. (?)

2.2.6. Learning Objects

Según ?, existen 3 definiciones prominentes de qué son los learning objects:

- “Cualquier entidad, digital o no, que puede ser usado para el aprendizaje, educación o entrenamiento.”
- “Cualquier recurso que puede ser reusado para dar soporte al aprendizaje”
- “Una entidad digital que puede ser usada, reusada o referenciada durante el aprendizaje apoyado por la tecnología.”

En términos generales, sus principales características son (?):

- **Auto-contenido y reusable:**

Cada learning object es independiente y puede ser compartido y reusado, en distintas ocasiones y para distintos fines, como una entidad indivisible.

- **Entidad de aprendizaje reducida:**

Al contrario de una clase completa, los learning object contienen recursos más breves de aprendizaje de unos 2 a 15 minutos.

- **Pueden ser agregados:**

Los learning objects pueden ser agrupados en colecciones más grandes, incluso en estructuras tradicionales de educación.

- **Rastreados con metadatos:**

Cada learning object contiene información descriptiva sobre la cual se pueden realizar búsquedas.

2.3. La Web 2.0

El término *Web 2.0* fue usado por primera vez por Dale Dougherty, vice-presidente de O'Reilly Media, en una sesión de ideas sobre el estado de la web luego de la ruptura de la “burbuja .com”, en 2001. Dougherty notaba que la web era más importante que nunca y que las compañías que habían sobrevivido tenían muchos aspectos en común, dichas observaciones le hacían pensar que el colapso marcado por la ruptura de la burbuja determinó un punto de inflexión y que un llamado a la acción era necesario.

El término cobró interés desde entonces, pero hasta el momento la comunidad de desarrolladores web mantiene un desacuerdo del verdadero significado. Para zanjar de dicha diatriba, ? publicó un artículo en el que aclaraba mucho de lo discutido en esa sesión de ideas que dio origen al término. De dicho artículo son destacables los siete principios de la Web 2.0:

2.3.1. La web como plataforma

Netscape Communications, creadores del navegador Netscape, fue la pionera en usar la web como plataforma aprovechando su dominio en el mercado de los navegadores (siendo el estándar de facto). Ofrecía un entorno (*webtop*) de servicios de alto costo hospedados en el propio navegador. Al final los navegadores y los servidores se convirtieron en simple mercancía y el verdadero valor se depositó en los servicios entregados sobre la plataforma web.

En contraste, usando esta filosofía de entrega de servicios surgió Google. Que sin requerir instalaciones, actualizaciones o parches, ofrecía al usuario un servicio en constante mejora que generaba ganancias de manera directa o indirecta de su uso.

La web como plataforma se utiliza para entregar software en forma de servicios en vez de recibir servicios a través de un software.

2.3.2. Aprovechando la inteligencia colectiva

Frases como la de Eric Raymond, originalmente acuñada para referir al software de código abierto, “con suficientes ojos, cualquier error es leve” (?) resuenan en la Web 2.0 potenciada por los usuarios: sistemas de calificación de los contenidos como el de Digg.com implementan a cabalidad

la citada frase, mientras que sitios insignia de la Web 2.0 como del.icio.us y flickr.com permiten a los usuarios etiquetar los contenidos (enlaces y fotografías respectivamente) generando estructuras flexibles para la clasificación de los contenidos, denominadas folcsonomías, en contraste con las tradicionales taxonomías definidas por un administrador.

2.3.3. Los datos son el próximo Intel Inside

Todos los sitios web significativos de la actualidad se basan en el manejo de datos. Esta es la principal competencia de las iniciativas Web 2.0: el manejo de datos. La frase de Hal Varian “SQL es el nuevo HTML” resume la importancia que tiene el manejo de los datos.

Pero no sólo se trata de tener el control sobre el contenido, sino generar valor agregado a partir de dichos datos. Un ejemplo de ello es Amazon.com y Barnesandnoble.com, ambas compañías obtuvieron su catálogo de un mismo proveedor, sin embargo Amazon se dedicó desde un comienzo a mejorar los datos agregándoles información obtenida de las casas editoriales como imágenes de las portadas, tabla de contenidos, índice y material de muestra. No sólo eso, sino que aprovecharon a sus usuarios para agregar notas a los datos de tal manera que, luego de 10 años, es Amazon y no su proveedor original quien es considerado la principal fuente para referencias bibliográficas. Adicionalmente, Amazon introdujo su propio identificador que permite referir libros que no tienen ISBN.

La carrera consiste en controlar ciertos tipos de datos: localización, identidad, fechas de eventos públicos, identificadores de productos y otros. En algunos casos el costo de crear y recopilar los datos permitirá al beneficiado mantener el control sobre ellos y licenciarlos, construyendo así lo que se denomina el Intel Inside. En los demás casos, el ganador será aquel que alcance el punto crítico por medio de sus usuarios, generando un servicio con valor agregado.

2.3.4. El fin del Ciclo de Liberación del Software

El hecho de que el software sea entregado como servicio y no como un producto conlleva dos cambios fundamentales:

- **Las operaciones deben convertirse en una competencia primordial** convertir el software en un servicio requiere que diariamente sea evaluado y mantenido. Ya sea para actualizar los

datos, generar nuevos derivados de ellos o simplemente mejorar la calidad de la respuesta. Esto explica que lenguajes dinámicos (también conocidos como de scripting) sean fundamentales dentro de las empresas Web 2.0 ya que dan cabida al cambio diario requerido por el software.

- **Los usuarios deben ser tratados como co-desarrolladores** así como el paradigma del Software Libre promovía la frase “libere temprano, libere a menudo”, el nuevo paradigma es el “beta perpetuo” en el que el software es desarrollado continuamente y nuevos servicios son habilitados constantemente. En este sentido, se debe mantener una vigilancia constante del patrón de uso de los servicios para facilitar la toma de decisiones sobre nuevos desarrollos.

2.3.5. Modelos de desarrollo ligeros

Desde el momento en que los servicios web se pusieron en boga, las grandes compañías se pusieron al corriente y habilitaron complejos servicios web para permitir entornos de desarrollo estables para aplicaciones distribuidas. Sin embargo, la complejidad de dichos modelos fue sustituida por opciones mas sencillas como RSS (XML).

La Web 2.0 empieza a concebir a los sistemas como consumidores de la información ofrecida libremente bajo formatos de fácil manejo, en contraste con los servicios web tradicionales que están diseñados para un alto acoplamiento. Tradicionalmente, los desarrolladores procuraban resguardar con celo sus creaciones mientras que en la actualidad, son los servicios abiertos y re-utilizables son los que se tornan más interesantes. Los servicios más exitosos son los que permiten a los usuarios mezclar los datos en maneras innovadoras.

La frase “algunos derechos reservados”, popularizada por la organización *Creative Commons* para contrastar con la tradicional “todos los derechos reservados”, es la guía para la nueva era de la información abierta.

2.3.6. El Software independiente del dispositivo

La nueva plataforma no está restringida a la PC, los usuarios ahora acceden a Internet a través de teléfonos celulares, tablet PC, Blackberry y otros.

El beneficio de diseñar software independiente del dispositivo es que permitirá, no sólo el consumo desde mayor número de dispositivos sino la generación de contenidos desde más lugares: automóviles que reportan el tráfico y el periodismo ciudadano son dos ejemplos pioneros de las nuevas capacidades de la red.

2.3.7. Una experiencia rica para el usuario

Desde 1992, por medio del navegador *Viola*, se había intentado ofrecer al usuario una experiencia con contenidos activos dentro del navegador usando “applets”. Posteriormente otras opciones fueron apareciendo: Java, Javascript y Flash, las cuales permitieron al desarrollador crear aplicaciones del lado del cliente y así ofrecer experiencias más ricas para los usuarios.

Pero no fue hasta que Google presentó Gmail y Google Maps que estas capacidades de la plataforma fueron realmente sacadas a relucir. El uso de AJAX (Javascript asíncrono y XML) permitió crear interfaces de usuario tan ricas como las disponibles en las aplicaciones de escritorio pero con los beneficios agregados de la plataforma web: acceso consistente y desde cualquier locación.

2.4. Accesibilidad

La accesibilidad se refiere a la facilidad con la cual algo puede ser usado o accedido por cualquier personas, especialmente por aquellas que poseen algún tipo de discapacidad.

La accesibilidad está ligada al diseño universal, que es un concepto relativamente nuevo donde se busca que el proceso de diseño se realice teniendo en cuenta a personas con distintos niveles de habilidad. El diseño universal propone 7 principios: (?)

1. Uso equitativo
2. Flexibilidad en el uso
3. Uso simple e intuitivo
4. Información perceptible
5. Tolerancia a errores
6. Esfuerzo físico bajo
7. Tamaño y espacio apropiado para el uso

2.4.1. Accesibilidad a los contenidos en la Web

“La accesibilidad Web significa que personas con algún tipo de discapacidad van a poder hacer uso de la Web. En concreto, al hablar de accesibilidad Web se está haciendo referencia a un diseño Web que va a permitir que estas personas puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto mermadas sus habilidades a consecuencia de la edad” (?)

La labor de desarrollar aplicaciones web accesibles es ardua debido a la naturaleza del ámbito: cualquier persona podría acceder al sistema utilizando, potencialmente, infinidad de dispositivos distintos como navegadores, lectores de pantalla, etc. Para lograr estandarizar el proceso de desarrollo web se creó el *Consortio World Wide Web* (W3C por sus siglas en inglés), el cual es el encargado de desarrollar dichos estándares (?). Entre dichos estándares Web están las directrices

para asegurar y mejorar la accesibilidad a los contenidos por parte de los usuarios.

Facilitar el acceso a los contenidos en la Web es importante ya que esta una fuente de información que se usa en muchos contextos como la educación, el comercio, gobierno y más (?).

En el desarrollo web, las pautas para determinar el nivel de accesibilidad de un sitio son definidas bajo 3 niveles de prioridad:

1. **Prioridad 1** reglas que **deben** ser cumplidas por los desarrolladores web.
2. **Prioridad 2** reglas que **deberían** ser cumplidas por los desarrolladores web.
3. **Prioridad 3** reglas que **pueden** ser cumplidas por los desarrolladores web.

Cumplir las reglas de cada uno de estos niveles asegura al desarrollador que las barreras de acceso serán cada vez menores. Las pautas a seguir están disponibles en el apéndice ??

2.5. Metodología de Desarrollo: Extreme Programming

Extreme Programming (XP) es una metodología de desarrollo de software cuyo éxito radica en centrar los esfuerzos en la satisfacción del cliente y la mitigación de riesgos. Ésta metodología es recomendada para equipos de trabajo pequeños (de 2 a 12 desarrolladores) y hace énfasis en el trabajo en equipo. Para XP, es imperativo que el equipo de trabajo tenga a su disposición, en cualquier momento, la retroalimentación del cliente (?).

Según ?, XP potencia los proyectos de desarrollo de software en cuatro aspectos esenciales:

- **Comunicación:** existe una comunicación constante entre los clientes y los programadores.
- **Simplicidad:** se busca mantener un diseño simple y limpio.
- **Retroalimentación:** se obtiene al realizar las pruebas al software desde el primer día. Se realizan entregas parciales del sistema tan pronto como sea posible y se implementan los cambios sugeridos.
- **Coraje:** las características anteriores le permiten a los desarrolladores responder de forma efectiva a los cambios constantes.

2.5.1. Flujo de Trabajo

El flujo de trabajo de la metodología, como se indica en la figura ??, inicia con la redacción de **historias de usuario**, de las que se puede proceder a **planificar las entregas**. XP recomienda desarrollar un **esbozo de la arquitectura** que implemente de manera somera las historias de usuario que representen mayor dificultad técnica. De esta manera la estimación del tiempo de desarrollo de cada historia, durante la planificación, será más acertada (?).

Una vez realizada la planificación de las entregas se procede al desarrollo iterativo en el cual el cliente selecciona las historias de usuario a desarrollar y los desarrolladores diseñan las **pruebas** correspondientes. Como resultado de una iteración se obtiene la **velocidad del proyecto** que a su vez sirve de límite para la selección de las próximas historias a desarrollar.

XP recomienda que se realicen **entregas pequeñas**, de manera continua, de modo que el cliente pueda ofrecer la retroalimentación necesaria.

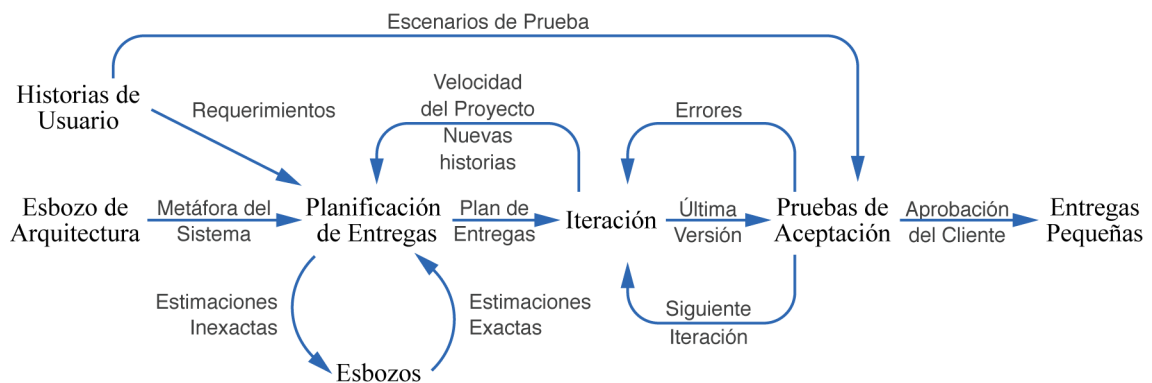


Figura 2.1: Flujo de Trabajo en Extreme Programming. (?)

2.5.2. Etapas del Flujo de trabajo en XP

Xtreme Programming, como lo indica la Figura ??, contempla varias etapas dentro del flujo de trabajo, las más destacables son:

1. **Historias de usuario:** son similares a los casos de uso, sin embargo no están escritas en un lenguaje técnico dado que es el cliente el encargado de redactarlas, en aproximadamente tres oraciones, basandose en sus necesidades específicas. Deben contener un nivel de detalle razonable de modo que los desarrolladores puedan hacer las estimaciones de tiempo, de manera más acertada, para las entregas. Las historias sirven como fuente de inspiración para la creación de pruebas de aceptación automatizadas (?).
2. **Selección de la metáfora del sistema:** a partir de la creación del esbozo de la arquitectura y del conocimiento de las historias de usuario se enmarca el desarrollo del proyecto en una metáfora. La selección de dicha metáfora permite que el equipo de trabajo maneje el mismo criterio, de manera consistente, para los nombres de clases y métodos. Logrando así, que los desarrolladores agilicen la tarea de determinar la existencia de alguna funcionalidad o clase. (?)
3. **Planificación de Entregas:** previo al inicio del desarrollo en iteraciones se lleva a cabo una reunión para crear el plan de entregas. El objetivo de dicha reunión es estimar cada historia del usuario en términos de semanas de programación ideales. Una semana ideal se refiere a cuánto tiempo estima el desarrollador que le tomará implementar cada historia y sus pruebas trabajando con dedicación exclusiva al proyecto. Luego de hacer las estimaciones, el cliente

procede a seleccionar las historias a implementar de acuerdo a su prioridad (?)

La filosofía para lograr un consenso durante la reunión de planificación de entregas debe girar en torno a cuatro variables: alcance, recursos, tiempo y calidad.

- Alcance: cuánto se tiene por hacer.
- Recursos: cuántas personas están disponibles.
- Tiempo: cuándo estará listo el proyecto o la entrega.
- Calidad: qué tan bueno será el proyecto y que tan bien probado será.

De estas variables, la gerencia puede seleccionar sólo tres. Los desarrolladores manejarán la última. Por lo general la planificación resultante se mantendrá mientras la velocidad del proyecto sea estable y no surjan nuevas historias de usuario (?).

4. **Iteración:** una iteración se alimenta de tres fuentes de información: el plan de entregas, la velocidad del proyecto calculada en iteraciones anteriores y los errores reportados en las pruebas de aceptación. El desarrollo de una iteración en XP, contempla en sí varias fases como se describe en la figura ??:

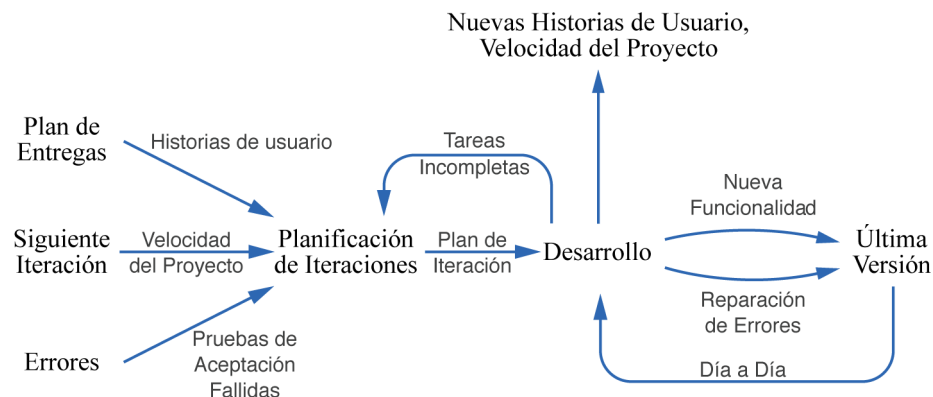


Figura 2.2: Desarrollo iterativo - Iteración. (?)

- **Planificación de iteración:** la planificación de cada iteración se realiza al principio de la misma, debe evitarse planificar por anticipado más allá de la iteración en curso, así mismo es importante mantenerse dentro de la planificación y no agregar funcionalidades que no esten planificadas. La estimación de tiempo de cada tarea debe realizarla el desarrollador asignado a dicha tarea; la estimación no debe ser modificada durante

la iteración ya que de ello depende lograr mejores estimaciones en las subsiguientes iteraciones, de la misma manera debe respetarse el límite de duración de la iteración en curso (?).

- **Desarrollo:** el desarrollo dentro de la iteración describe un día de trabajo bajo la óptica de XP. Al inicio del día se realiza una reunión corta (?) en la que se reportan problemas con sus soluciones y se promueve la concentración en el proyecto. Luego, cada desarrollador, procede a seleccionar una de las tareas asignadas (o una prueba que falle) para desarrollarla en pareja con un compañero.

Según ?, el **desarrollo en parejas** es el punto focal del desarrollo, esta metodología promueve un mayor conocimiento del código (funcionalidades y pruebas) por parte de cada miembro del grupo. Esta forma de trabajo está rodeada de cuatro aspectos importantes: **el intercambio de parejas** que suele ocurrir cuando alguna pareja requiere ayuda (?), **integración continua** de nuevo código al proyecto al desarrollar nuevas funcionalidades o pruebas unitarias (?), **reestructuración sin piedad** (refactoring) para simplificar partes complejas del código (?) y por último **creación de pruebas unitarias** antes de desarrollar nuevas funcionalidades. (?)

5. **Velocidad del proyecto:** es un indicador de cuan rápido se está desarrollando el proyecto. Se calcula a partir del número de historias de usuario (o tareas) completadas durante cada iteración. Luego se emplea para determinar cuántas y cuáles historias de usuario (o tareas) serán implementadas durante la reunión de planificación de cada iteración. También es útil cuando se desea determinar cuantas iteraciones serán necesarias para finalizar una entrega mayor (?).
6. **Pruebas de aceptación o funcionales** Las pruebas de aceptación son creadas a partir de las “historias de usuario”. En el transcurso de una iteración las “historias de usuario” seleccionadas en la reunión de planificación de la iteración serán traducidas a pruebas de aceptación. El cliente especifica los escenarios en los cuales se probará si una historia de usuario, que puede tener una o varias pruebas, ha sido implementada correctamente. Las pruebas de aceptación son pruebas de caja negra al sistema, cada una de ellas representa algún resultado esperado del sistema. Los clientes son responsables de verificar la correctitud de la prueba y revisar los resultados de la misma, para así decidir cuál prueba fallida, en caso de que la haya, tiene ma-

por prioridad. La implementación de una historia de usuario no será considerada completa mientras no pase todas las pruebas de aceptación asociadas a la historia (?).

2.5.3. Retroalimentación y Planificación

La figura ?? muestra el ciclo donde se indican los tiempos aproximados de planificación, al igual que los tiempos en que se obtiene retroalimentación del trabajo completado.

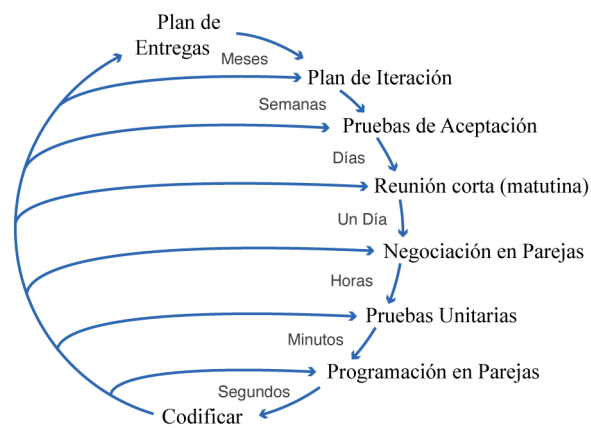


Figura 2.3: Ciclo de Planificación y Retroalimentación en Xtreme Programming

2.6. Patrón de Arquitectura MVC

Modelo-Vista-Controlador es un patrón de ingeniería del software que permite crear independencia entre la capa de interfaz, la capa de negocios y la capa de datos, facilitando el modificar la apariencia de la aplicación o las reglas del negocio sin afectar al resto. En el MVC el Modelo representa la información o la data de la aplicación y las reglas del negocio empleadas para manejarla, la Vista corresponde a los elementos de la interfaz del usuario como cuadros de texto, checkbox, etc. y el Controlador maneja los detalles correspondientes a la comunicación con el modelo de la aplicación.

2.6.1. Elementos del patrón MVC

- **Modelo:** representa los datos y las reglas del negocio que manejan el acceso a esos datos. Generalmente el modelo se emplea como una aproximación al proceso original de diseño de la base de datos, por lo que las técnicas de modelado también aplican en la definición del modelo (?).
- **Vista:** muestra el contenido del modelo. Accede a los datos del negocio, a través del modelo, y especifica que data debe ser mostrada. Es responsabilidad de la vista el mantener la consistencia en las presentaciones cuando el modelo cambia (?).
- **Controlador:** lleva a cabo la traducción de la interacción con la vista a acciones que pueden ser ejecutadas por el modelo. Las acciones realizadas por el modelo incluyen el activar procesos del negocio o cambiar el estado del mismo. En base a la interacción del usuario y la salida de las operaciones del modelo, el controlador responde seleccionando una vista adecuada (?).

El flujo de control para este patrón es (?):

1. El usuario interactúa con la interfaz.
2. El controlador toma el evento de entrada de la interfaz con el usuario.
3. El controlador notifica al modelo de la acción ejecutada por el usuario, que posiblemente resulta en un cambio en el estado del modelo.

4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar una interfaz que se adapte a los cambios del mismo. La vista toma los datos del modelo pero este no tiene conocimiento directo sobre la vista.
5. Se esperan nuevas interacciones en la interfaz de usuario que inicien el ciclo nuevamente.

2.6.2. Ventajas

- **Permite la re-utilización de los componentes del modelo**

La separación entre el modelo y la vista permite emplear varias vistas para un mismo modelo. Como consecuencia de esto los componentes del modelo son más fáciles de implementar, probar y mantener, ya que todos los accesos se realizan a través de estos componentes.

- **Mayor facilidad para el soporte de nuevos tipos de clientes**

Sólo es necesario crear nuevas vistas, incorporar alguna lógica en el controlador e integrarlo con la aplicación existente.

- **Facilidad en el diseño**

Las aplicaciones web se han hecho más difíciles de diseñar que las aplicaciones tradicionales, el patrón MVC está siendo usado como una solución a esa dificultad.

2.6.3. Desventajas

- **Incrementa la complejidad del diseño**

El patrón introduce algunas clases extras para la separación del modelo, la vista y el controlador.

- **Cierta cohesión entre la vista y el controlador**

Cualquier modificación en la vista suele significar un cambio en el controlador.

Capítulo 3

Diseño

3.1. Metáfora del Sistema

Siguiendo las indicaciones de la metodología, la metáfora seleccionada para el desarrollo de **Ósmosis2** es “el aula de clases tradicional”, en la cual los estudiantes atienden a las clases del profesor. Esto permite que muchas de las herramientas planificadas se integren fácilmente dentro de esta metáfora. Por ejemplo las palabras lecciones, foros, conversaciones (chat), encajan sin mucho esfuerzo.

3.2. Roles de usuario

También, dentro de la metáforas, los actores son obvios: estudiante, profesor, ayudante (asistente) y oyente. Esta jerarquía de “super-roles” permite asignar a cada usuario su función específica dentro de un curso (un profesor en el curso A puede ser un estudiante en el curso B). Adicional a estos cuatro roles se define uno adicional, el cual no está superditado a un curso en particular, que es el administrador de la plataforma y tiene acceso privilegiado a todas las funciones del sistema.

Sin embargo, por lo extenso del proyecto y para agregar mayor semántica dentro de cada herramienta, se han desarrollado “sub-roles” que permiten definir las funciones de cada usuario dentro de la herramienta y definir qué super-roles pueden ser asignados dentro de dichos sub-roles.

Para conocer sobre las funcionalidades de cada una de las herramientas mencionadas a conti-

nuación consultar el **Apéndice B**, en el cual se describen las características del sistema en función de las herramientas y recursos que lo conforman.

Foro

- *Moderador*: el moderador de un foro es el encargado de mantener las conversaciones dentro de un ambiente de respeto y cerrar las conversaciones cuando lo considere pertinente.
Super-Roles: instructor y ayudante.
- *Forista*: es la persona que escribe sus comentarios, sobre un tema específico, en el foro.
Super-Roles: estudiante.

Blog

- *Escritor*: el escritor de un blog es aquella persona que plasma ideas o comentarios en él para compartirlos con el resto de la comunidad.
Super-Roles: instructor, ayudante y estudiante.
- *Lector*: es el lector de los comentarios del blog de otra persona.
Super-Roles: instructor, ayudante, estudiante y visitante.

Wiki

- *Supervisor*: es el encargado de supervisar la calidad de información colocada en el wiki, que sea adecuada.
Super-Roles: instructor.
- *Editor*: son aquellas personas que participan en el wiki agregando información relacionada a los temas de los cursos o que considere de interés para otros estudiantes o instructores.
Super-Roles: ayudante y estudiante.
- *Lector*: es el lector de la información contenida en el wiki. Puede ver el historial o comparar versiones del contenido allí escrito.
Super-Roles: visitante.

Chat

- *Moderador*: al igual que en el foro, el moderador de un chat es el encargado de mantener las conversaciones dentro de un ambiente de respeto, así mismo puede amonestar a un participante si lo considera necesario.

Super-Roles: instructor y ayudante.

- *Participante*: son las personas que participan en conversaciones sobre distintos temas empleando el chat como medio para intercambiar ideas en tiempo real.

Super-Roles: estudiante.

Mensajería

- *Remitente*: es la persona que envía un mensaje (e-mail) a una o más personas.

Super-Roles: instructor, ayudante y estudiante.

- *Destinatario*: es la persona que recibe un mensaje. Puede realizar las opciones típicas de un servicio de mensajería (leer el mensaje, eliminarlo, ver la bandeja de entrada, etc.).

Super-Roles: instructor, ayudante y estudiante.

Evaluaciones

- *Evaluador*: es la persona que crea, aplica y corrige las evaluaciones o actividades asignadas a un grupo de personas.

Super-Roles: instructor y ayudante.

- *Evaluable*: es la persona que responde una evaluación aplicada por el evaluador.

Super-Roles: estudiante.

Agenda

- *Planificador*: es la persona que planifica y agrega un nuevo evento a la agenda.

Super-Roles: instructor y ayudante.

- *Lector*: es la persona que observa los eventos plasmados en la agenda.

Super-Roles: instructor, ayudante, estudiante y oyente.

Lecciones

- *Tutor*: es la persona que proporciona o dicta la lección
Super-Roles: instructor y ayudante.
- *Aprendiz*: es aquel que utiliza la información contenida en la lección.
Super-Roles: estudiante.

Casillero

- *Propietario*: es aquella persona que es dueña del casillero
Super-Roles: instructor, ayudante y estudiante.
- *Lectores*: son las personas que pueden visualizar archivos almacenados en los casilleros de otros usuarios
Super-Roles: instructor, ayudante, estudiante y oyente.

Portafolio

- *Dueño*: el dueño del portafolio es el único que tiene capacidad de modificarlo y decidir qué usuarios pueden acceder a él.
Super-Roles: instructor, ayudante y estudiante.

3.3. Historias de Usuario

A continuación se presentan, organizadas por herramienta, las historias de usuario según redactadas según las recomendaciones de la metodología XP.

Manejo de Usuarios

- **Registrar usuario individuales y por lotes**

El administrador tiene la opción de registrar a los usuarios del sistema de forma individual o por grupos (registrar a todos los estudiantes del curso). De igual manera, cada estudiante puede registrarse por su cuenta y de forma individual.

- **Iniciar sesión**

El usuario ya registrado, introduce el login y el password para tener acceso al sistema y sus funcionalidades.

- **Cerrar sesión**

El usuario cierra la sesión a través de la cual ha estado trabajando y abandona el sistema.

- **Manejo de permisos**

El sistema debe permitir asignar roles a los usuarios con respecto a toda la plataforma (administrador, miembro, visitante) y dentro de cada curso (profesor, estudiante, asistente).

- **Eliminar usuario**

Se pueden eliminar los usuarios del sistema de forma individual o por grupos de modo que sus datos no sean válidos para acceder al sistema, sin embargo debe ser posible conservar toda la información generada por este. Esta eliminación debe poder realizarse en lotes.

Foros

- **Crear tema**

Los profesores del curso tienen la potestad de crear temas para delimitar las discusiones del foro.

- **Iniciar discusión**

Cualquier miembro del curso puede iniciar una discusión dentro de cualquiera de los temas existentes del foro.

- **Responder hilo de discusión**

Los usuarios pueden responder en las discusiones iniciadas en el foro.

- **Editar mensaje**

Cualquier mensaje en los hilos de discusión deben ser modificables por sus respectivos dueños.

- **Puntuar un comentario**

Los usuarios del foro, podrán asignar una evaluación de la calidad de los mensajes de sus compañeros.

- **Programar fecha de cierre**

El sistema permite cerrar un tema o una discusión de modo que ningún participante pueda agregar nuevos mensajes. Este cierre puede ser programado de modo que se ejecute automáticamente llegada la fecha.

Intercambio de Archivos

- **Subir archivo**

El sistema debe permitir a los usuarios subir archivos al sistema, estos archivos serán de acceso público.

- **Modificar datos del archivo**

Los usuarios pueden modificar los datos (título o descripción) del archivo que subió al sistema.

- **Ordenar casillero**

El usuario del casillero debe poder organizar sus archivos como lo desee, creando carpetas y moviendo archivos de lugar.

- **Acceso al casillero**

Todo casillero será accesible públicamente y existirá una carpeta especial para hacer entregas en la que todo usuario podrá dejar archivos que sólo el dueño del casillero podrá ver.

- **Manejar versiones para archivos**

El sistema permite manejar la actualización de versiones de los archivos subidos. Esto facilita el control de los posibles cambios que pueda sufrir un archivo.

- **Detección de virus**

El sistema debe suministrar un antivirus que verifique que los archivos a subir o descargar no posean virus que puedan afectar a los usuarios.

Mensajería Interna

- **Enviar mensaje**

Los usuarios pueden enviar mensajes a otros usuarios del sistema. Los mensajes pueden ser leídos dentro del sistema o pueden ser redirigidos al correo electrónico del usuario.

- **Leer mensaje**

El sistema permitirá a los usuarios leer los mensajes que ha recibido

- **Manejar libreta de direcciones**

- **Agregar contactos**

Los usuarios poseen una “libreta” en la cual pueden almacenar los datos de contacto de otros usuarios del sistema. El sistema permite agregar nuevos contactos con la información de cada uno de ellos.

- **Buscar contacto**

El usuario que posee contactos en su libreta puede realizar una búsqueda de los datos de una persona en particular, ingresando el nombre del contacto a buscar o cualquier otra información relevante.

- **Eliminar contacto**

El usuario que posee una libreta de contactos puede eliminar un contacto y toda la información del mismo.

- **Editar información de contacto**

El usuario puede agregar datos adicionales sobre sus contactos.

- **Exportar contactos**

El sistema permite que los usuarios exporten su lista de contactos a otros formatos digitales que faciliten la impresión o manejo de los mismos fuera del sistema.

Blog**■ Entradas**

El escritor del blog debe se capaz de escribir y modificar las entradas de su blog.

■ Eliminar entrada

El escritor del blog puede, en cualquier momento, eliminar una entrada de su blog.

■ Listar entradas

Se pueden visualizar las entradas registradas en el blog. La cantidad de información mostrada dependerá de las configuraciones seleccionadas por el dueño del blog.

■ Agregar comentario

Dependiendo de las configuraciones definidas por el escritor del blog, los lectores podrán dejar comentarios en las entradas publicadas.

■ Enviar trackback/pingback

El sistema de blogs debería permitir manejar trackbacks y pingbacks para permitir la comunicación con otros blogs dentro y fuera del sistema.

■ Recibir trackback/pingback

Al recibir una petición de trackback o pingbak se debe registrar un comentario en la entrada correspondiente que exprese que ha sido realizada esta acción

■ Importar blog externo

El usuario registrado, escritor de un blog externo, tiene la capacidad de indicarle al sistema donde se aloja dicho blog de manera que tome de ahí las nuevas entradas.

Wiki**■ Crear página**

Los miembros del curso pueden crear y modificar páginas en el wiki.

■ Historia de Modificaciones

El sistema debe llevar un registro de los cambios que se realizan en cada página de cada wiki.

■ Restaurar página

En cualquier momento el usuario podrá restaurar una página del wiki a una versión anterior.

- **Comparar versiones de páginas**

El usuario podrá comparar distintas versiones de una página para detectar los cambios realizados. Debe ser posible saber el autor de los cambios.

- **Manejo de borradores**

El usuario tendrá la posibilidad de almacenar borradores de las páginas que está modificando.

- **Manejo de discusiones**

Cada página debe permitir una sección para que los usuarios discutan los contenidos de dicha página.

Chat

- **Ver historial de conversaciones**

El sistema debe permitir al moderador del curso ver los registros (historial) de las conversaciones.

- **Expulsar usuario**

El moderador de la sala de chat puede expulsar a un participante de manera temporal o permanente.

- **Registrar pregunta desde el chat**

Los participantes tienen la posibilidad de copiar preguntas hechas en el chat (con sus respuestas) a una sección permanente de preguntas importantes.

Pizarra

El instructor o el ayudante pueden configurar una sesión de trabajo en la “pizarra”. En ella podrá mostrar material multimedia (PDF, presentaciones, dibujos y videos) o escrita (chat y pizarra).

Lecciones

- **Crear lección**

El profesor del curso puede subir al curso una lección en formato SCORM.

- **Ver lección** El sistema debe proveer un visor de lecciones SCORM de modo que los miembros del curso puedan revisarla en línea.

Enlaces

- **Agregar enlace**

Cada curso debe tener una sección de enlaces a recursos de interés. Cada miembro que lo desee podrá agregar recursos por esta vía.

- **Modificar enlace**

Cada miembro podrá modificar o eliminar los enlaces que haya agregado.

Calendario/Agenda

Los miembros del curso podrán registrar, modificar y eliminar eventos de la agenda. Dichos eventos serán visibles por el curso.

Trabajo desconectado

- **Agregar Podcast/Screencast**

El blog debe soportar podcasting y screencasting.

- **Descargar contenidos del curso**

El usuario puede descargar el material del curso para usarlo posteriormente sin necesidad de estar conectado con el sistema.

Agregador de Noticias

Cada curso podrá agregar de manera automática noticias generadas en otras páginas.

Grupos de Trabajo

- **Crear grupo**

El profesor o ayudante puede crear grupos de trabajo con los estudiantes del curso.

- **Modificar grupo**

El profesor o ayudante de un curso podrá modificar cualquier grupo de trabajo para agregar o eliminar un miembro del grupo.

- **Eliminar grupo**

El profesor o ayudante podrá eliminar un grupo una vez finalizada la actividad para la cual fue creado.

Comunidad

- **Listar cursos**

El sistema debe exponer a la comunidad todos los cursos disponibles en la plataforma.

- **Listar entradas de blogs**

El sistema debe exponer las entradas de los blogs.

- **Notificaciones Globales**

El sistema debe permitir a los administradores colocar notificaciones globales para todos los usuarios.

- **Mostrar noticias externas**

El sistema debe permitir a los administradores seleccionar una o más fuentes de noticias que se mostrarán como notificaciones globales.

Portafolios

- **Crear perfil**

El usuario registrado (instructor, ayudante o estudiante) puede crear un perfil con sus datos personales e información correspondiente a los estudios cursados, entre otros. Podrá crear un currículum virtual del cual será dueño.

- **Configurar perfil**

El dueño del portafolio podrá editar la información suministrada al crear su perfil inicial para modificar o eliminar datos.

- **Ver perfil**

Los usuarios registrados y los visitantes pueden visualizar los datos del perfil de otro usuario

o de ellos mismos. La cantidad de información suministrada dependerá del nivel de permisos que tenga el usuario que visualiza el perfil.

- **Configurar portafolios**

El dueño del portafolios podrá seleccionar los elementos que desea mostrar públicamente, el portafolio completo estará disponible para que lo envíe personalmente a quienes desee mostrar la información completa.

- **Editar portafolios**

El dueño del portafolio puede agregar información nueva, así como eliminar o modificar la información existente.

- **Exportar portafolios**

El dueño del portafolio podrá exportar su perfil personal con el estilo de un currículum, indicando la información que desea mostrar y cómo desea presentarla.

Pruebas en línea

- **Crear preguntas**

El instructor o el ayudante pueden elaborar preguntas que, posteriormente, les permitan crear una prueba o utilizarlas en otra actividad

- **Modificar preguntas**

El instructor o el ayudante pueden editar las preguntas creadas para cambiar su contenido o adaptarla a algún tipo de actividad. Debe tenerse en cuenta que dichas preguntas pueden haber sido usadas en otras pruebas.

- **Crear pruebas**

El instructor o el ayudante pueden elaborar pruebas para evaluar el contenido de la materia del curso. Estas pruebas tendrán asociadas una puntaje.

- **Modificar pruebas**

El instructor o el ayudante pueden modificar las pruebas creadas. Se podrá agregar una pregunta, eliminarla o cambiar su puntaje.

- **Guardar pruebas**

El instructor o el ayudante pueden guardar las pruebas elaboradas para reutilizarlas en futuros cursos.

- **Evaluar pruebas**

El instructor o el ayudante pueden corregir las preguntas de las pruebas aplicadas a los estudiantes y asignarles el puntaje obtenido. Posteriormente este puntaje se verá reflejado en el acta de calificaciones.

Actividades Evaluadas

El profesor o ayudante podrá asignar una actividad, con una puntuación definida, a los estudiantes.

Control de calificaciones

- **Crear tabla de control de evaluación**

El instructor podrá crear una tabla de calificaciones en la cual se plasmará las notas obtenidas por los estudiantes tanto en las actividades evaluadas como en las pruebas, luego se podrá totalizar el puntaje para cada estudiante del curso.

- **Modificar tabla de control de evaluación**

El instructor podrá editar la tabla de calificaciones para agregar nuevas actividades o pruebas a evaluar, para cambiar el puntaje de las ya existentes o para eliminarlas.

- **Exportar tabla de calificaciones**

El sistema permitirá al instructor generar un acta de las calificaciones del curso con las actividades realizadas y el puntaje obtenido por los estudiantes en cada una de ellas.

Manejo del Curso

- **Crear curso**

El instructor podrá crear un curso que corresponderá a la materia a dictar. Podrá dar la descripción del mismo, los horarios, etc. y agregar a los estudiantes que lo tomarán. Adicionalmente podrá agregar a la plantilla del curso aquellas herramientas, disponibles en el sistema, que se consideren adecuadas para lograr los objetivos del mismo.

- **Programar aparición de recursos**

EL sistema debe permitir a los profesores definir las fechas en las que algún contenido estará disponible para que los estudiantes lo utilicen.

- **Guardar estructura de un curso**

El sistema debe permitir que la estructura de un curso pueda ser guardada para ser utilizada como plantilla de otro curso.

Seguimiento de usuarios

- **Listar usuarios en línea**

Los miembros podrán en cualquier momento conocer qué compañeros se encuentran usando la plataforma.

- **Ver historial de navegación** El profesor tendrá una sección en la que podrá observar el uso que le dan los estudiantes al plataforma, de modo que pueda detectar a los estudiantes poco participativos o con problemas.

Manejo de estadísticas

El sistema debe extraer, a partir de los datos recabados en el seguimiento de los usuarios, datos de interés estadístico sobre el uso de los recursos y herramientas. Estos datos deben ser visibles para profesores y administradores.

Learning Objects

La plataforma debe permitir a los usuarios generar Learning Objects a partir de los recursos disponibles.

3.4. Requerimientos no-funcionales

Las siguientes características y condiciones técnicas que deben cumplirse en Ósmosis2:

3.4.1. Capacidad de mantenimiento

Esta característica engloba los requerimientos relacionados con la sustentabilidad y capacidad de crecimiento que debe tener el sistema en el tiempo. Ósmosis2 debe cumplir con:

1. Modularidad

El código fuente del sistema debe contar con una separación clara de sus capas utilizando la arquitectura MVC, de modo que el crecimiento de la aplicación sea organizado.

2. Plug-and-play

La arquitectura del sistema debe ser de tal manera que cada herramienta se conecte y esté lista para usar. De la misma manera, debe ser igual de fácil desactivar las herramientas.

3. Documentación clara

El sistema debe contar con una documentación del código clara y completa. Adicionalmente, el sistema debe contar con manuales de uso para desarrollador y usuario.

3.4.2. Usabilidad y Accesibilidad

El sistema debe respetar las pautas definidas por la W3C para la accesibilidad en el desarrollo web así como los principios básicos del diseño universal. Debe contar con una interfaz gráfica intuitiva y fácil de usar compatible con cualquiera de los navegadores web de mayor uso (Firefox 2, Internet Explorer 6, Opera 9, Safari 3 y posteriores).

3.4.3. Prestaciones

En esta característica se definen las condiciones que deberá soportar la aplicación a nivel de:

1. Escalabilidad

Es un aspecto de gran importancia ya que la plataforma debe manejar, desde sus primeras versiones, miles de usuarios, cantidad que se espera aumentará progresivamente con la incorporación de nuevas instituciones educativas a la plataforma. Así mismo debe permitir el

crecimiento rápido en los servicios, de modo que se pueda dar respuesta rápida a los nuevos requerimientos que puedan surgir.

2. Rendimiento

El sistema debe ejecutar las operaciones de manera eficiente, para ello debe manejar aspectos como la concurrencia y caching de resultados, de modo que sea posible manejar grandes cantidades de peticiones.

3.4.4. Seguridad

1. Autenticación

El acceso al sistema debe estar restringido por el uso de claves asignadas a cada uno de los usuarios.

2. Listas de acceso

El sistema debe permitir que los usuarios tengan acceso para la modificación de sólo los recursos que su rol les permita. Dicho roles deben definirse en función de cada curso.

3. Trazabilidad

El sistema deberá contar con mecanismos que permitan el registro de las actividades realizadas por cada usuario.

4. Confidencialidad

El sistema debe permitir que cada usuario determine qué información desea revelar dentro de la aplicación.

Capítulo 4

Marco Tecnológico

Este capítulo tiene como principal finalidad presentar las opciones que fueron consideradas para el entorno (lenguaje-framework) que se utilizaría para el desarrollo de Ósmosis2; comparando dichas opciones en diferentes aspectos. Posteriormente se presenta la opción seleccionada y se justifica dicha selección en función de los requerimientos específicos de la aplicación y de otras variables importantes.

4.1. Entorno de desarrollo

Para el desarrollo de Ósmosis2 se buscó un entorno que permitiese solucionar uno de los problemas presentes en la plataforma actual: propiciar la escalabilidad sin sacrificar la facilidad y la velocidad de desarrollo.

Para asegurar la escalabilidad era necesario considerar lenguajes de programación que hubiesen demostrado con anterioridad su confiabilidad. Para cumplir con el requerimiento de facilidad en el desarrollo es necesario seleccionar algún framework que ofrezca al desarrollador un entorno en el que hayan elementos pre-construidos de manera que sólo deba ensamblar, de manera ordenada y estructurada, las partes para lograr la tarea que desea.

Bajo estos requisitos iniciales, la decisión debía ser un balance entre estabilidad del lenguaje (en su historia y a futuro) y la facilidad que ofrece el framework estudiado. La selección inicial de lenguajes se basó en el nivel de conocimientos de los desarrolladores, por ello se consideraron **JSP**

por su robustez, **PHP** por su larga historia soportando exitosamente a muchas aplicaciones web y **Ruby** por el reciente éxito (publicitario) que ha tenido.

Una vez seleccionados los lenguajes, se consideró un solo framework para cada uno:

- **JSP**: Struts
- **Ruby**: Rails
- **PHP**: CakePHP

4.1.1. JSP/Struts

Struts es un framework de código abierto para la construcción de aplicaciones web basadas en Servlets/JSP bajo el paradigma MVC. Su principal característica es que su implementación de MVC tiene como punto medio al controlador que es el que conecta al modelo con la vista. Sin embargo, Struts no ofrece una capa de acceso a datos sino que recomienda algunas como Hibernate, EJB, Cayenne, etc. Situaciones parecidas se presentan en las capas de vista y de lógica del negocio (incluso la validación de datos), en las cuales se deben conectar otros frameworks especializados (?).

En este sentido Struts es altamente flexible, pero dicha flexibilidad suele generar confusión entre los desarrolladores. Uno de los principales problemas que presenta Struts es que frena la velocidad de desarrollo al requerir archivos de configuración en formato XML que debe ser modificado cada vez que se desea implementar una nueva funcionalidad. Muchos defensores de Struts apelan a que dichas configuraciones no son necesario tocarlas si se hace uso de las herramientas automatizadas que ofrecen los IDEs (Integrated Development Enviroment). Sin embargo bajo los requerimientos de Ósmosis, esto se traduce en otra razón en contra.

El gran elemento a favor de Struts es la robustez que ofrece un lenguaje orientado a objetos y las muchas librerías existentes que se pueden usar. Sin embargo, en este proyecto tan importante como la reutilización de librerías, es la facilidad y rapidez en el desarrollo de nuevas funcionalidades.

4.1.2. Ruby/Rails

Rails es un framework MVC completo para el desarrollo de aplicaciones web sobre bases de datos. Por completo se entiende que es un framework que ofrece las funcionalidades necesarias para manejar todos los niveles de una aplicación: acceso a datos, validación y reglas del negocio, etc.

Rails favorece la convención sobre la configuración, lo cual disminuye la cantidad de detalles que son necesarios recordar en comparación con Struts. Adicionalmente ofrece al desarrollador herramientas para minimizar la cantidad de código necesario, potenciando la reutilización de código. Rails al ser completo, ofrece soporte para validación y abstracción de la base de datos.

El rendimiento es uno de los principales problemas de Rails, un ejemplo de este tipo de problemas se presentó con el servicio web Twitter en el cual millones de usuario envían mensajes constantemente lo cual significó constantes caídas del servicio.

4.1.3. PHP/CakePHP

PHP es el lenguaje de mayor uso y presencia en la web, en la actualidad. Es un lenguaje interpretado que, al igual que Ruby, no es fuertemente tipado.

PHP tiene una larga trayectoria en el desarrollo web, con corporaciones como Yahoo! usándolo para potenciar sus sitios. Tal vez unos de los más grandes problemas que han sido destacados por los detractores de PHP son las vulnerabilidades detectadas en muchas aplicaciones como resultado de malas prácticas potenciadas por el lenguaje, sin embargo frameworks de desarrollo como Cake se han encargado de cerrar dichos agujeros.

CakePHP es un framework que toma muchos de los conceptos útiles de Rails. Es también un framework completo que ofrece código útil para el desarrollo de las interfaces hasta la abstracción del acceso a la base de datos y, como Rails, se basa en la convención sobre la configuración.

Según ?, una de las ventajas de CakePHP sobre Rails es el lenguaje sobre el cual se contruye: para PHP está probado que escala bien. Sin embargo, su queja contra PHP se reduce a lo desordenado que puede llegara a ser el código de este lenguaje. (?). Pero, por otra parte, este es uno de los

problemas que CakePHP viene a solucionar.

4.1.4. Sinopsis y selección

El cuadro ?? resume los puntos destacados anteriormente junto con otros aspectos considerados para la decisión.

Criterios	Entornos de desarrollo		
	JSP/Struts	Ruby/Rails	PHP/CakePHP
Validación de datos	X	X	X
Framework completo	-	X	X
Agilidad de desarrollo	-	X	X
Soporte de la comunidad	X	X	X
Documentación	X	X	-
Requerimientos no funcionales			
Módulos Plug-and-Play	-	X	X
Encriptación y Autenticación automatizada	-	X	X
Listas de acceso	-	X	X
Escalabilidad y Rendimiento	X	-	X
Otros aspectos			
Dominio del lenguaje y framework	X	-	X
Total	5/10	8/10	9/10

Cuadro 4.1: Comparativa de los entornos de desarrollo considerados. Se consideran aspectos relacionados con la facilidad de uso, requerimientos no funcionales y familiaridad de los desarrolladores con el mismo, quedando seleccionado **CakePHP**

Capítulo 5

Implementación

5.1. Arquitectura de plugins

Las características funcionales que ofrece Ósmosis son casi en su totalidad proveídas por los plugins, que son aplicaciones separadas que comparten el entorno de ejecución de la plataforma, y que pueden intervenir o participar en cualquier lugar de la misma. Para referirnos a los plugins también utilizaremos el término *herramientas*, siendo este último término específico a los plugins que extienden la funcionalidad de los cursos.

Esta modularidad permite el desarrollo rápido de nuevas características para la plataforma y mitiga el riesgo de dañar funcionalidades existentes debido a la modificación del código principal. Así también posibilita que los docentes elijan las herramientas que mejor se adecuen a su metodología educacional, en lugar de estas ser impuestas globalmente por un administrador.

La arquitectura de plugins de Ósmosis está basada en la ofrecida por CakePHP. En este sentido, un plugin de Ósmosis es, prácticamente, una aplicación CakePHP (?), que presenta nativamente los siguientes atributos:

1. Configuraciones compartidas

La configuración de conexiones a fuentes de datos, lógica de control de acceso, lógica del modelo de datos y clases utilitarias son automáticamente heredadas de la aplicación principal, aunque pueden ser redefinidas o extendidas por el propio plugin.

2. Intercomunicación entre plugins

Cualquier plugin puede comunicarse con el modelo de datos o invocar acciones de algún otro plugin. En este sentido es posible replicar la idea de la arquitectura modular de los plugins para uno en particular.

Sin embargo, Ósmosis extiende las capacidades naturales de un plugin del framework que utiliza para permitirles:

1. Incrustar elementos en la interfaz

Esto permite que cada plugin pueda suscribirse a lugares de emplazamiento anunciados, ya sea por las vistas principales de Ósmosis o por las de cualquier otro de estos módulos, de tal manera que pueda participar en la presentación de información. Una de las maneras en que el sistema aprovecha esta capacidad es, por ejemplo, mostrar un resumen de los eventos sucedidos en un curso. Con esta capacidad es posible, también, que los plugins agreguen secciones fijas en el diseño, por ejemplo la lista de contactos del chat.

2. Engancharse a sucesos

Esto permite que los plugins puedan detectar la ocurrencia de un evento o un cambio de estado en el modelo de datos, como el registro de un nuevo usuario, de modo que puedan realizar acciones adicionales.

3. Definir reglas de acceso propias

Cada plugin puede al momento de su instalación definir sus propias reglas de acceso basadas en los grupos a los que pertenecen los usuarios de la aplicación. De esta manera no se necesita que la autorización sea manejada a nivel del plugin, sino que la lógica se comparte para todos en la aplicación principal.

4. Crear auto-instaladores

Lo cual posibilita que cualquier plugin defina las acciones adicionales que requiere para ponerse en funcionamiento, por ejemplo, cargar las tablas que necesita en la base de datos.

Además el núcleo de Ósmosis se encarga de verificar que el plugin que se esté tratando de acceder, o bien, que aquel que desee participar de un evento, esté debidamente activado, no sólo globalmente para la aplicación, sino específicamente para cada curso. Esto hace posible remover características de la plataforma durante su ejecución a distintos niveles y que se puedan fácilmente volver a activar.

5.1.1. Instalación y activación

Un aspecto interesante de la arquitectura de plugins de Ósmosis es que permite su administración de manera sencilla. Los pasos generales para instalar un plugin son:

1. Descomprimir el plugin
2. Colocar la carpeta descomprimida en el directorio plugins de Ósmosis.
3. Activar el plugin en la interfaz administrativa.

Adicionalmente, los profesores tienen la potestad de activar o desactivar las herramientas en sus cursos sin afectar a los demás.

5.1.2. Plugins Implementados

Agenda

La agenda es una herramienta que sirve para planificar y anunciar los eventos que un curso pudiera tener. Se implementó la agenda de tal manera que pudieran etiquetarse los eventos de que se listen en la misma, y así poder rastrear los mismos a través de búsquedas en un futuro.

La agenda muestra una vista principal estilo calendario del mes en curso, en cada casilla correspondiente al día, se muestran extractos de texto referentes a los eventos registrados para el mismo.

Para interactuar con la agenda basta seleccionar los enlaces correspondientes a cada evento para ver los detalles como una descripción ampliada, horario y etiquetas que han sido asignadas por los usuarios. Adicionalmente se puede avanzar o retrasar las vistas de calendario en el tiempo, de modo que es posible conocer con antelación qué está planificado para el curso.

Blog

Da la capacidad, a cada uno de los usuarios, de mantener un diario académico con sus ideas, pensamientos y actividades que realiza con respecto, pero no limitado, a los temas de los cursos en los que participa.

El blog permite también que los demás usuarios dejen comentarios en las entradas, de este modo, se habilita una vía adicional por la cual los estudiantes pueden exponer ideas, intercambiar conocimientos y corregir errores.

Chat

El plugin Chat fue implementado de manera que no fuese dependiente del curso que el usuario está manejando. En lugar de ser una sala de conversación donde profesores y alumnos intercambian mensajes que todos pueden leer, se desarrolló una implementación típica de mensajería instantánea, en la cual la conversación se limita a dos interlocutores.

A pesar de que lo que se conoce como salas de chat pudiera ser más efectivo a la hora de comunicar una misma idea a todos los participantes de la conversación, se prefirió implementar un canal de conversación omnipresente en la plataforma, de forma tal que los interlocutores no tuviesen que coincidir en la misma herramienta para efectuar el intercambio de ideas; sino que, por el contrario, fuese posible ubicar a la persona con la cual se quiere conversar, sin importar qué sección del sistema esté usando.

De igual manera, se implementaron herramientas que se describirán en breve y que cumplen la misma función de una sala de conversación, en el sentido de poder comunicar a todas las personas de un mismo curso una única idea que sea leída por todos; no obstante las herramientas que suplen dicha falta son de carácter asíncrono, y la espera para recibir una respuesta a una idea emitida puede ser un poco más larga.

Para la implementación de este plugin, se recurrió al uso de javascript, comunicación con el servidor a través de peticiones REST (?), y respuestas en formato XML. A continuación se detalla el ciclo de vida de una instancia de este plugin:

- El usuario ingresa a la plataforma y es instanciado el chat, con lo cual se manda una petición al servidor solicitando una marca de tiempo que será usada para pedir los mensajes que estén esperando por ser recibidos.
- El cliente de chat envía una petición solicitando la lista de contactos disponibles.

- La instancia de chat solicita al servidor los mensajes enviados al usuario desde la marca de tiempo recibida.
- El servidor responde con un paquete XML conteniendo los posibles mensajes que faltan por recibir. En caso de haber mensajes, manda una nueva marca de tiempo, sino envía la marca de tiempo anterior.
- El cliente de chat despacha los mensajes a los contenedores XHTML que albergan la conversación en curso, o crea nuevos contenedores si se trata de una nueva conversación.
- La instancia de chat pregunta periódicamente al servidor por nuevos mensajes enviando la última marca de tiempo recibida.

Es de hacer notar que esta implementación usa la técnica de Encuestamiento (Polling), que puede hacer un consumo innecesario de la red. No obstante, se diseñaron los mensajes XML, de tal manera que se enviara lo estrictamente necesario para que el plugin funcione.

En su estado actual, el plugin de chat puede ser extendido con poco esfuerzo para albergar conversaciones grupales, o hacer salas de chat completas.

Foro

Esta herramienta representa un espacio para la discusión de los temas relacionados con el curso. El funcionamiento del foro es bastante sencillo:

1. El profesor define los temas sobre los cuales se puede discutir.
2. Cualquier miembro del curso puede iniciar una discusión dentro de los límites de dichos temas.
3. Cualquier miembro del curso pueden dejar su respuesta a las discusiones existentes.

Adicional a este funcionamiento básico, es posible modificar respuestas y cerrar discusiones o temas completos.

Casillero

Esta herramienta permite a cada miembro contar con un espacio para almacenar archivos. Este es un cambio sustancial con respecto a la implementación de Ósmosis 1.5 en la que era el curso el que contaba con ese espacio. Esto responde a dos aspectos importantes:

1. Mantener la consonancia con la metáfora seleccionada

Son los profesores y alumnos quienes tienen casillero, no los cursos.

2. Aumentar el nivel de apertura en la información

Se deja en manos de la mayoría, los estudiantes, la generación de contenidos.

Metafóricamente, el casillero representa el mismo objeto en la vida real con algunos detalles:

■ Los archivos pueden ser organizados en carpetas

Aunque en un casillero no suelen haber carpetas como en un archivador, se ha seleccionado dicha metáfora (evitando la palabra directorio que no dice nada) como escape para la necesidad de organizar los archivos.

■ Los archivos disponibles en el casillero son de acceso público

Permite controlar el uso indebido de la herramienta y reforzar la generación de información

■ Una carpeta especial funge como buzón de entregas (Drop-box)

El buzón permite la entrega de material que debe mantenerse secreto, por ejemplo para evitar plagios.

Pruebas

Permite al profesor diseñar y aplicar pruebas en línea a los estudiantes del curso, los tipos de pregunta que maneja esta herramienta son:

- Selección (simple y múltiple)
- Apareamiento
- Ordenamiento
- Desarrollo (Texto)

La implementación de esta herramienta se basa en un subconjunto del estándar *IMS Question & Test Interoperability (QTI) v2.1 (?)*. Esta especificación presenta un modelo de datos para la representación de preguntas y de pruebas con sus correspondientes resultados.

QTI define como objeto de evaluación básico a los *items* que contienen una pregunta, las instrucciones en cuánto a cómo debe ser presentada, el proceso de respuesta (cómo debe evaluarse las respuestas) y el feedback que debe darse (incluyendo pistas o soluciones).

Hay muchos elementos de este estándar que han sido omitidos en beneficio de la simplicidad, sin embargo uno de los aspectos más interesantes de este estándar es la posibilidad de intercambio de pruebas completas, distribuidas a través de paquetes XML. De implementarse esta capacidad, Ósmosis podría ser considerado un repositorio abierto de pruebas en línea basadas en dicho estándar.

Con respecto a la implementación, Ósmosis permite crear un repositorio de preguntas reutilizables por cualquier profesor para la creación de sus pruebas, además de una interfaz diseñada para la confección rápida y visual de los mismos.

Lecciones

Esta herramienta habilita al profesor para entregar a los alumnos contenidos navegables en forma de lecciones interactivas.

La implementación de esta herramienta también está basada en la definición de un estándar. En este caso se trata de *Sharable Content Object Reference Model*, también denominado SCORM 2004. 3ra. Edición (?). El objetivo principal de SCORM es la creación de contenidos de aprendizaje reutilizables, bajo el modelo de “objetos instruccionales”. Ósmosis tiene la capacidad de consumir paquetes SCORM y mostrarlos a los usuarios en forma de lecciones haciendo un seguimiento de aquellas secciones que ya han sido estudiadas por el usuario.

El desarrollo de esta herramienta se basa en uno de los libros que definen el estándar, *SCORM Content Aggregation Model (CAM) (?)*, en el cual se describen los componentes usados para la construcción de los objetos de aprendizaje, cómo empaquetarlos para que puedan ser compartidos

entre sistemas, cómo se deben describir para facilitar la búsqueda y la reutilización y la secuencia de la información en dichos componentes, entre otros.

Uno de las deficiencias más importantes en la implementación es que, una vez cargados los contenidos en formato SCORM no es posible descargarlos nuevamente, esto significa que la habilidad de compartir que genera SCORM no está disponible en Ósmosis.

Wiki

La herramienta Wiki permite la creación de páginas web de manera rápida, sencilla y colaborativa. Dado que las implementaciones comunes de sistemas de wiki existentes contemplan el seguimiento de cambios realizados a cada página creada, se procedió a hacer una investigación sobre la mejor manera de almacenar estos cambios.

Con respecto a este tema, (?), considera que la manera más eficiente en cuanto espacio en disco y a rendimiento de los procesadores, es almacenar la copia completa de la versión anterior y generar una página nueva que contenga los cambios deseados. Pero, dado que las revisiones a un artículo son vistas en menor frecuencia que las versiones actuales, es seguro guardar el texto comprimido con el algoritmo Gzip.

Ósmosis, al igual que el software usado por la Wikipedia, almacena copias completas de las revisiones para cada página, a la vez que soporta visualizar las diferencias entre cualquier par de revisiones. De la misma manera se implementó la funcionalidad requerida para restaurar versiones anteriores de una página, en cuyo caso, se guardará la versión actual como una revisión, y será reemplazada por la versión seleccionada.

El Wiki de Ósmosis fue implementado de manera que los usuarios no tuviesen que aprender ningún tipo de sintaxis especial para crear las páginas, de manera que se minimice la curva de aprendizaje para estos. En su lugar se utilizó un editor de texto enriquecido WYSIWYG (Lo que ves es lo que obtienes, por sus siglas en inglés), que emula aplicaciones de escritorio para la edición y creación de textos.

5.2. Autorización de usuarios

La autorización de usuarios en Ósmosis se implementó con un sistema híbrido, que permite agregar reglas a nivel de cada controlador de aplicación, y de usar un repositorio central de permisos representado con listas de control de acceso o *ACL* por sus siglas en inglés (?).

La lista de control de acceso opera a nivel de los roles de usuario, de manera que funciona como un control de acceso basado en roles (?), los cuales no son fijos para cada miembro de la plataforma, sino que cambian según la ubicación en la que estén trabajando. Por ejemplo un usuario podría ser un profesor en un curso, mientras que es un estudiante en otro, o bien no tener acceso al mismo en lo absoluto. Por otro lado, la autorización basada en los controladores permite un control mucho más fino de lo que cada usuario tiene permitido hacer, por ejemplo a este nivel se puede verificar si la persona que desea modificar un registro en particular es la dueña del mismo, por lo cual tiene privilegios especiales.

El subsistema de autorización fue implementado de tal manera que fuese posible intercambiarlo por cualquier otro método de autorización que se desee en un futuro, pues simplemente se tiene que cambiar el objeto de autorización por algún otro que defina los métodos requeridos.

5.3. Seguimiento del usuario

Ósmosis provee un sistema sencillo para hacer seguimiento a las acciones de los usuarios. La implementación se encarga de registrar las creaciones y modificaciones de datos de los modelos que utilicen la interfaz *Loggable*

Registrar estos cambios en los datos le permite a Ósmosis:

- Presentar a cada usuario los cambios ocurridos recientemente de modo que pueda mantenerse informado.
- Permitir a los administradores hacer una aproximación de qué herramientas y secciones de la aplicación son de mayor uso.
- Cumplir con la condición de no-repudio (?) de la seguridad de datos.

En su estado actual, Ósmosis no provee una herramienta para hacer cálculo de estadísticas complejas sobre los datos recabados, de modo que el segundo punto es una potencialidad que ha de ser implementada.

Como se indicó, el seguimiento que hace Ósmosis es estrictamente sobre los cambios a los datos. Extender esta funcionalidad para hacer un seguimiento de cada solicitud es posible, sin embargo se consideró como excesivo e innecesario.

5.4. Internacionalización

Con la finalidad facilitar la implantación de la plataforma en múltiples lenguajes, se aprovecharon las capacidades del framework utilizado al transformar todas las cadenas de caracteres, visibles para los usuarios, mediante una función de traducción. Dicha función evalúa las preferencias locales de la máquina del usuario, zona desde donde solicita la página, preferencias de la aplicación y otros aspectos para elegir el idioma de traducción.

El idioma predeterminado utilizado en la plataforma es el inglés, y las reglas de transformación para cada idioma deben definirse en archivos específicos que siguen las reglas de la librería GNU gettext y que consisten en la creación de pares clave-valor, para cada uno de los textos que se deseen traducir. Dichos archivos son luego compilados e interpretados por la librería antes referida para aumentar el rendimiento al momento de la traducción.

Capítulo 6

Conclusiones y recomendaciones

El sistema de Gestión del Aprendizaje Ósmosis2 fue diseñado e implementado considerando dos aspectos muy importantes. A nivel técnico se entrega un software fácil de mantener con una separación clara y organizada entre sus capas; y a nivel pedagógico presenta un concepto innovador en el área de *e-learning* a través de la integración del conectivismo y los principios de la Web 2.0 con los modelos clásicos de aprendizaje.

Para ello fue necesario realizar una investigación sobre las teorías del aprendizaje vigentes y bajo las cuales se han desarrollado otros LMS que están actualmente en el mercado. Así mismo, y con el propósito de seleccionar aquellas herramientas que se adaptaran a los conceptos bajo los cuáles se construiría Ósmosis2, se realizó un análisis comparativo de los diferentes LMS, incluyendo la versión 1.5 de Ósmosis, destacando aquellas características comunes y que son esenciales en todo sistema de aprendizaje.

En función de las características antes señaladas se seleccionaron aquellas herramientas que se consideraban necesarias para la nueva implementación y se trató de vincularlas con enfoque pedagógicos específicos, que en conjunto, permitieran crear una nueva visión del sistema. De la misma manera se realizó un estudio sobre los principios de la Web 2.0 y de cómo podían adaptarse a la nueva filosofía que se quería incorporar.

Para reforzar la concepción sobre la cuál se levantó el sistema (una simbiosis de los modelos clásicos de aprendizaje con el conectivismo) se decidió que cada una de las herramientas sería un

plugin separado, dándole libertad al profesor para seleccionar aquellas que se adapten a sus objetivos.

Por otra parte, como resultado de la investigación realizada durante el presente proyecto de grado y con el objetivo de complementar las funcionalidades implementadas, se proponen las siguientes recomendaciones y mejoras al sistema:

Nuevas funcionalidades

- Complementar el diseño actual de los cursos con plantillas elaboradas en función de los distintos enfoques pedagógicos presentados. Estas plantillas utilizarían como herramientas principales aquellas que se puedan asociar directamente con el enfoque que representan, además podrían ser “personalizadas” incorporando otros recursos. De esta manera los profesores tienen la posibilidad de escoger el modelo de curso que se adapte mejor a sus necesidades y al enfoque con el cuál se identifican.
- Incorporar a la aplicación la capacidad de crear, modificar o eliminar grupos de trabajo, así como como extender las herramientas implementadas para aceptar este nuevo nivel de organización.
- Implementar un herramienta que le permita al profesor mantener un acta de calificaciones en la cual pueda llevar un control del desempeño del estudiante en las actividades del curso.
- Suministrar a los usuarios una libreta de contactos que permita mantener los datos de otros usuarios conocidos.
- Habilitar la opción de exportar los contenidos de las herramientas para poder ser consultados de sin conexión.
- Desarrollar recursos como la Pizarra, que permitan hacer uso de medios audiovisuales para llevar a cabo, de forma interactiva, la enseñanza a distancia.
- Agregar una herramienta que permita visualizar las estadísticas del uso del sistema: por curso, por herramienta y seguimiento de usuarios.

- Permitir a cada usuario mantener un portafolio o currículum de las actividades académicas, laborales y extra-académicas en las que ha participado.
- Permitir a Ósmosis exponer sus recursos en forma de Learning Objects.
- Crear una herramienta que le permita a los usuarios registrar enlaces o fuentes de noticias (RSS) con material relacionado a un curso.
- Dar soporte, de manera global, para el almacenamiento de borradores al redactar mensajes de modo que los usuarios tengan la posibilidad de reanudar su trabajo posteriormente.
- Permitir a los dueños del curso exportar e importar los contenidos del curso.

Extensión a las herramientas existentes

Blog

- Mejorar la interconexión de los blogs por medio de Trackback y Pingback.
- Permitir a los usuarios importar o sincronizar su blog en Ósmosis con uno externo.
- Agregar la posibilidad de publicaciones multimedia: Screencasts y Podcast.

Pruebas

- Permitir al profesor preparar y programar la aplicación de pruebas.
- Completar la implementación del estándar *QTI* para la definición de pruebas en línea. Esto permitiría a Ósmosis interoperar con otras aplicaciones y LMS existentes.

Chat

- Agregar la posibilidad de registrar, de manera automatizada, las preguntas interesantes que se generen en el chat.
- Implementar la sala de chat para el curso completo con la posibilidad de expulsión de un usuario y el control de quienes pueden escribir.

Foro

- Incorporar la capacidad de calificar los mensajes de modo que se destaquen las respuestas inteligentes.
- Permitir al profesor programar la fecha de cierre de las discusiones.

Casillero

- Incluir de manera nativa el control de versiones sobre los archivos permitiendo ver los cambios ocurridos entre las distintas versiones.
- Mejorar la seguridad por medio de la búsqueda de virus a los archivos.

Lecciones

- Diseñar contenidos cuyos elementos se adapten a la versión del estándar que fue implementada (2004 3ra. edición).
- Completar la implementación de SCORM incorporando el `sequencingCollection` según se define en el libro *SCORM Content Aggregation Model (CAM) (?)*.

Apéndice A

Stakeholders y Usuarios

A.1. Stakeholders

En la tabla que se muestra a continuación, se ofrece un breve resumen de los stakeholders que participan en el desarrollo de este proyecto.

	Nombre	Descripción	Responsabilidades
DSM		Moderador técnico. Es la unidad encargada de dar apoyo, servicio y asesoría en el uso de recursos multimedia para programas de docencia, investigación y extensión de la USB	<ul style="list-style-type: none">■ Garantizar el mantenimiento del sistema■ Hacer seguimiento al desarrollo del sistema■ Garantizar la disponibilidad de los recursos necesarios para el funcionamiento del sistema

Nombre	Descripción	Responsabilidades
Fidel Gil Hermes Rodríguez	Mentor y Asesor, respectivamente. Prestan asesoría en cuanto a la determinación de los requerimientos del sistema y de las características de su comportamiento.	<ul style="list-style-type: none"> ■ Prestar asesoría en la determinación de los requerimientos y características del sistema. ■ Monitoreo del progreso del sistema
Ana Gabriela Díaz José Lorenzo Rodríguez Joaquín Windmüller	Desarrollador. Encargado del diseño e implementación del Sistema	<ul style="list-style-type: none"> ■ Desarrollo de la aplicación. ■ Aporte de ideas para mejorar la funcionalidad de la aplicación.

Cuadro A.1: Resumen de los Stakeholders

A.2. Usuarios

En la tabla que se muestra a continuación, se ofrece un breve resumen de los usuarios que maneja el sistema.

Nombre Descripción Responsabilidades		
Nombre Descripción Responsabilidades		
Administrador	Usuario del sistema que posee todos los permisos	<ul style="list-style-type: none"> ■ Realizar el mantenimiento del sistema ■ Incorporar nuevas funcionalidades al sistema, de ser necesario. ■ Mejorar las herramientas disponibles.
Profesor	Persona con habilidades pedagógicas encargada de impartir la educación. Usuario principal o clave del sistema que tiene la posibilidad de usarlo como herramienta para impartir un curso	<ul style="list-style-type: none"> ■ Crear, actualizar, modificar o gestionar la información pedagógica que se quiera utilizar como ayuda de la metodología seleccionada para impartir el curso. ■ Hacer seguimiento de las actividades planificadas ■ Evaluar el desempeño del estudiante.
Facilitador	Ayudante del profesor que contribuye con el proceso de enseñanza	<ul style="list-style-type: none"> ■ Se le atribuyen responsabilidades similares a las del profesor.

Nombre	Descripción	Responsabilidades
Estudiante	Persona que cursa estudios	<ul style="list-style-type: none">■ Contribuir en el desarrollo de las actividades del curso.■ Hacer uso de las herramientas del sistema propuestas para complementar el proceso de aprendizaje.■ Evaluar o dar realimentación sobre el sistema y sus características.

Cuadro A.2: Resumen de los Usuarios

Apéndice B

Características del Sistema

B.1. Justificación de funcionalidades

B.1.1. Núcleo de Ósmosis

Está formado por los componentes básicos de Ósmosis, sobre los cuáles se construirán e instalarán las herramientas que servirán de soporte a las actividades de los cursos. Dichas herramientas contendrán características que contribuyen a la orientación del curso en cualquiera de los enfoques pedagógicos planteados: constructivismo, cognitivismo, conectivismo o una combinación de ellos.

Funcionalidades básicas ofrecidas

- **Manejo de usuarios**
- **Manejo de cursos**
- **Registro y recuperación de estadísticas**
- **Editor de texto enriquecido con guardado automático**
- **Manejo de palabras claves o etiquetas por usuario**

B.1.2. Foros

La herramienta de foros es en sí misma agnóstica en cuanto a la metodología pedagógica que apoya. Pueden ser usados desde un enfoque eminentemente cognitivista, por ejemplo, una sesión

de preguntas y respuestas sobre un tema, entre el profesor y el estudiante, puede tener como medio de comunicación e intercambio de ideas un foro. Desde este enfoque hay un máximo control sobre lo que los alumnos pueden hacer y la forma en que pueden participar de la actividad.

Así mismo, los foros pueden ser utilizados bajo la óptica constructivista, según la cual el docente puede indicarle a sus alumnos que discutan libremente sobre un tema o bajo lineamientos. Así mismo el docente actúa como un moderador de la actividad y es quien establecerá las conclusiones finales a partir de lo que se haya expuesto.

Una nueva visión puede ayudar a entender lo que sucede actualmente con los foros en la red, la teoría del conectivismo. Con esta herramienta el comportamiento de sus participantes puede parecer caótico, pero usualmente se encamina por medio de la interacción de los que allí escriben. Esto permite una libertad mucho mayor que la que se logra con el constructivismo, lo cual fomenta la creatividad y el flujo de la información.

Justificación de funcionalidades propias

- **Granularizar permisos de acceso:** Apoyado por el cognitivismo y el constructivismo puesto que al limitar o ampliar la capacidad de participación sobre un elemento del curso, el docente tendrá la capacidad de guiar el mismo y enfocar su enseñanza en actividades con significado. Además permite la creación foros pro subgrupos dentro de la clase y el manejo de los mismos por figuras como las de un “preparador”.
- **RSS de los posts:** Se apoya en uno de los principios del conectivismo que es la de mantener la información actualizada e interconectada entre los nodos que la generan. Proveer una manera de syndicar lo que se escribe en los foros permite mantener actualizado al estudiante y al docente sin que ellos estén obligados a visitar la aplicación para leer los mensajes que le interesen. Simplemente necesitaría su lector preferido de RSS.
- **Puntuaciones a los posts:** Se apoya en el principio del conectivismo que afirma que la intención del proceso es la información precisa. Una manera de asignar relevancia a una información dentro de una red es dándole una puntuación de valor. Las entradas del foro que tengan más relevancia con el tema o aporten algo realmente valioso serán mejor puntuadas por los participantes del mismo. Se puede proveer un mecanismo mediante el cual se diferencie o se le asigne **más peso a la puntuación** de un profesor que a la de un estudiante

- **Tipos de Foro:** Los foros, según el alcance que tengan pueden ser privados para los participantes de un curso, públicos, compartidos entre varios cursos o privados para un grupo dentro de un curso. Los foros privados sirven como herramienta de comunicación alumno-profesor bajo una óptica cognitiva/constructiva, mientras que los foros privados entre grupos tienen como finalidad la comunicación entre los estudiantes y favorece las actividades en las que la creatividad o trabajo propio del estudiante es requerido. Los foros de tipo pregunta-respuesta caen también en la categoría del constructivismo, dado que es necesaria la creatividad del estudiante para responder la pregunta o aportar soluciones a un problema. No obstante, los foros públicos y compartidos entre curso también pueden ser enfocados bajo metodologías pedagógicas tradicionales, facilitan el intercambio de la información, la apertura de la misma además de la libertad de creación, valores todos asociados con el conectivismo.

B.1.3. Intercambio de Archivos

El intercambio de archivos es una herramienta que favorece cualquier enfoque pedagógico. En el caso cognitivista puro, sirve para la entrega de material de apoyo y estudio por parte del instructor al alumno, así como para la entrega de asignaciones. Desde el enfoque constructivista y constructivista social, las razones para el intercambio de archivos no son muy distintas a las usadas en la visión anterior. No obstante vista desde las perspectiva conectivista, encontramos que el intercambio de archivos favorece en buena medida el intercambio entre los nodos de la red de aprendizaje y aumenta la cantidad de conocimiento que ya se posee.

El ejemplo más típico de como el intercambio de archivos puede favorecer la interconexión entre los nodos de la red se da cuando un alumno publica una guía de estudio o un conjunto de ejercicios resueltos para un curso, estos se hacen inmediatamente disponibles para todos en la red, de modo que eventualmente alguien más, que los encuentre útiles, los reseñará y utilizará para su propio estudio. Finalmente, otra persona en busca de algo parecido encontrará la dicha reseña y entenderá que si para él fueron útiles, probablemente valga la pena utilizarlos y reseñarlos igualmente.

Justificación de funcionalidades propias

La idea detrás del intercambio de archivos dentro de Ósmosis2, es diferente a la acostumbrada en un correo electrónico, que es la de adjuntar archivos a un mensaje dirigido a una o más personas.

En lugar de enviar archivos, se trata de que las personas “recojan” los mismos en el “casillero” de quién los generó.

En el caso de entregar una asignación a un profesor, la metáfora más correcta no es que el profesor la busque en los casilleros de los alumnos, sino que los alumnos la entregue en el casillero del instructor. En este caso si podemos hablar de una entrega de archivos.

- **Casillero personal publico:** Permite al dueño del casillero dejar archivos para que sea vistos por cualquier persona en la red.
- **Casillero de entregas para el profesor:** Permite a los integrantes de un curso entregar asignaciones o trabajos al instructor del mismo, siendo visibles estos archivos solamente por el profesor luego de su entrega.
- **Casillero personal privado:** Aunque parezca deseable el tener un casillero privado, se debe entender que una plataforma de aprendizaje concebida con una filosofía centrada en el intercambio, desde sus inicios, no es compatible con esta idea. Uno de los usos más comunes que se le ha dado a esta funcionalidad es la de ocultar contenidos de materias que no están siendo dictadas, pero que lo estarán nuevamente en un futuro. No obstante, este uso, visto desde la perspectiva conectivista, significa que se está sustrayendo conocimiento de la red, o lo que es lo mismo, cuando una persona quiera hacer referencia a un archivo que consideraba existente, encontrará que el mismo ya no está disponible.
- **Descripción de archivos y comentarios:** El usuario tendrá la posibilidad de agregar una descripción o comentario al archivo que coloca en cualquiera de los casilleros donde tenga permiso, y también habrá la posibilidad de que el resto de la comunidad (según sea el permiso de lectura del archivo) pueda comentar sobre la calidad o contenido del mismo.

B.1.4. Mensajería Interna

La mensajería interna permitirá recibir los mensajes a aquellos usuarios más sensibles con su privacidad o que no desean recibir correo electrónico de la plataforma. La existencia de una mensajería interna fomenta, además, un uso más frecuente del sistema, y por ende mayor participación de los usuarios en los cursos que integran.

Justificación de funcionalidades propias

- **Envío de mensajes a correo externo:** Para aquellos que no deseen revelar su dirección de correo, pero aún así prefieren leer sus mensajes desde su buzón preferido, se les da la oportunidad de reenviar automáticamente los mensajes que reciban en su correo interno directamente a una dirección de correo electrónico externa.
- **Libreta de direcciones en microformato hCard:** El uso de microformatos para el manejo de conocidos ayudará al explorador a reconocer la herramienta favorita del usuario para comunicarse con ellos o ubicarlos en el mapa.

B.1.5. Blogs

Los Blogs son uno de los pilares fundamentales de Ósmosis2. Esta herramienta permite que los usuarios expresen sus opiniones sobre los cursos, expongan sus ideas y muestren al resto de la comunidad cualquier tema que pueda ser de interés general. Igual que otras funcionalidades, esta no está atada a ningún enfoque pedagógico en especial, no obstante poco tiene que ver con el cognitismo puro. Un blog puede ser utilizado por el docente como una herramienta constructivista, mediante la cual puede comentar y dirigir las ideas de sus estudiantes con respecto a un tema específico. Bajo el enfoque conectivista, el blog, es una fuente generadora de nodos de información, que al interconectarse con otros nodos, genera conocimiento.

La idea general para los blogs dentro de Ósmosis 2 es que cada usuario tenga su blog personal. Por lo tanto, no existe un blog para cada curso, puesto que este tipo de herramienta puede considerarse, realmente, personal y no sería útil bajo un esquema dentro de una asignatura.

Este esquema favorece la interconexión de la información y la libertad de creación, que son principios del conectivismo y el constructivismo.

Justificación de funcionalidades propias

- **Agregador de feeds RSS/Atom externos:** Permite que aquellos usuarios que ya posean un blog y no deseen usar el suministrado por la plataforma Ósmosis2, puedan seguir aportando ideas de la misma manera que lo harían aquellos que sí utilizan la herramienta interna, mediante la agregación de los RSS que genere su aplicación.

B.1.6. Wiki

Al igual que los Blogs, el wiki es una herramienta de gran utilidad para los enfoques constructivista y conectivista. Los wikis favorecen el principio del conectivismo que afirma que la capacidad de generar conocimiento es más importante que el conocimiento que ya se posee.

En Ósmosis2, y a diferencia de los Blogs, los wikis son una posesión grupal y no individual. Es por esto que cada curso tendrá la posibilidad de tener un Wiki propio, cuyos contenidos se irán agregando automáticamente al Wiki global de la aplicación.

Justificación de funcionalidades propias

- **Alcance del Wiki:** El wiki pueden ser configurado por el administrador de modo que sean accesibles sólo para los participantes del curso; en lugar de la configuración predeterminada que es la de wikis públicos.

B.1.7. Chat

La principal función del chat es contribuir a la aplicación de los enfoques pedagógicos conectivista y cognitivista. El enfoque cognitivista se ve reflejado en las actividades que se lleven a cabo con la participación del profesor, como impartir lecciones, fomentar sesiones de preguntas y respuestas, entre otras. El enfoque conectivista se podría evidenciar en las discusiones o conversaciones entre alumnos, en tiempo real, en las cuales se compartan ideas y conocimientos.

Justificación de funcionalidades propias

- **Suspensión (por parte del profesor) de estudiantes:** Permite a los profesores suprimir, por un período definido o permanentemente, el derecho de participación de los estudiantes que hayan incumplido alguna de las normas establecidas para el chat o cualquier otra funcionalidad de interés.
- **Historial de conversaciones:** Ofrece a los usuarios del chat la posibilidad de guardar las conversaciones en las cuales han participado. De esta forma es posible conservar datos, preguntas o respuestas que sean consideradas útiles para el estudio o la evaluación del contenido del curso.

B.1.8. Pizarra

La pizarra es, al igual que el chat una herramienta que favorece el enfoque cognitivo del aprendizaje. El uso que se le puede dar no es muy diferente a lo que sucede en un aula de clase tradicional. El profesor se dirige a sus estudiantes mediante el uso de elementos audiovisuales y ellos atienden a la clase participando eventualmente.

B.1.9. Lecciones

Las lecciones favorecen la implantación de cursos a distancia en los cuales el facilitador o docente puede controlar completamente la forma en que debe aprender el estudiante, lo cual es un enfoque plenamente cognitivista. Con el uso de esta herramienta es posible que el profesor indique que recursos debe utilizar el alumno y en qué orden, además de tener la posibilidad de medir progresivamente el avance del mismo a través uso de pruebas que son requisitos para continuar a la siguiente lección.

B.1.10. Enlaces

Los enlaces sirven como herramienta de los enfoques pedagógicos cognitivista, ya que a partir de ellos se podrá extraer la información que permitirá generar el conocimiento sobre uno o varios contenidos del curso, y conectivista porque es posible compartarlos con otros usuarios o hacerlos públicos de manera tal que todos tengan acceso a la información.

Justificación de funcionalidades propias

- **Guardar enlaces privados/públicos:** Los usuarios podrán guardar la ruta o URL de aquellos sitios que consideren de interés. Los enlaces pueden ser guardados como privados cuando se desea cierta confidencialidad de la información que contiene el enlace mostrado. Pueden ser públicos cuando se quiere que todos los usuarios utilicen dichos enlaces como recurso común para la extracción de información.

Nota: Los enlaces privados están representados por las referencias utilizadas por los profesores o las empleadas por los estudiantes en sus grupos de estudio o trabajo

B.1.11. Calendario/Agenda

La función principal tanto del calendario del curso como de la agenda es permitir la planificación y desarrollo, de forma ordenada, de las actividades.

El calendario y la agenda presentan la misma funcionalidad, su diferencia radica en el tipo de usuario al cual están destinados. El calendario está diseñado, principalmente, para el profesor mientras que la agenda está destinada tanto a profesores como a estudiantes, siendo estos últimos los usuarios frecuentes.

Justificación de funcionalidades propias

- **Profesores pueden guardar eventos en el calendario:** Los profesores podrán colocar en el calendario las actividades programadas para el período de clases en curso, de esta forma se facilita el seguimiento de las actividades pautadas.
- **Estudiantes tienen calendario privado para programación personal. Dicha programación es compatible:** Los estudiantes pueden crear un calendario personal para la planificación de sus actividades. En caso de realizar planificaciones para trabajos en grupo se tiene la posibilidad de compartir el calendario con el resto de los integrantes del mismo.
- **Los eventos se pueden enlazar a archivos, actividades, enlaces, etc. del curso:** Los usuarios pueden relacionar los eventos plasmados en el calendario o la agenda con archivos que describan de forma detallada la actividad, enlaces que faciliten la búsqueda de información o cualquier otro recurso que permita llevar a cabo la tarea indicada.
- **Publicación de eventos usando microformatos (hCalendar) y XML, iCal:** Estas opciones permiten que el usuario que así lo prefiera use la aplicación que más le agrade para visualizar los eventos del calendario, como pueden ser Google Calendar, iCal, entre otros.

B.1.12. Capacidades de búsqueda

La capacidad de búsqueda es una ayuda que se les brinda a los usuarios que no pueden ubicar de una manera rápida las funcionalidades o recursos que hay dentro de un curso. Se espera que cada módulo implemente capacidades de búsqueda sobre los elementos que sea relevante recuperar para llevar a cabo la actividad deseada.

B.1.13. Agregador de Noticias

Su función es aprovechar los contenidos publicados por otros sitios de interés y publicar, dentro del sistema, dichas noticias o artículos.

B.1.14. Grupos de Trabajo

Esta herramienta apoya los enfoques pedagógicos del constructivismo y el constructivismo social, ya que genera el intercambio de información y conocimiento entre los miembros de un grupo. Los participantes actúan como fuente de conocimientos y entre todos se crea una red que contribuye a la adquisición de nuevos conceptos o ideas.

Justificación de funcionalidades propias

- **Asignación de usuarios a grupos de trabajo (Manual y aleatoria):** los integrantes de los grupos de trabajo pueden ser asignados directamente por el profesor o el sistema puede generar un grupo de trabajo a partir de la lista de estudiantes inscritos en el curso.
- **Los módulos manejan la noción de grupos de trabajo y clases enteras:** Cada módulo posee herramientas que permiten realizar las actividades del curso, ya sea a través de grupos de trabajo o de forma individual para todos los estudiantes de una clase. De igual manera se presentan funcionalidades específicas para actividades en grupo.

B.1.15. Ayuda del Sistema

Poseer un entorno uniforme de ayuda para el sistema y sus módulos mejorará la experiencia de los usuarios.

Justificación de funcionalidades propias

- **Tour virtual de cada módulo:** Proporciona a los estudiantes y profesores información general sobre el funcionamiento y las características de cada módulo del sistema.
- **Ayuda contextual dentro de cada módulo:** Contiene información específica dentro del contexto de cada herramienta de manera tal que el usuario pueda obtener la ayuda necesaria fácilmente y sin necesidad de buscar en toda la documentación del usuario.

B.1.16. Comunidad

La creación de una comunidad alrededor de temas educativos es importante para mantener la vigencia y frescura de los contenidos. De esta manera se apoya no solo el conectivismo sino que la disponibilidad de contenidos diversos puede ayudar a los profesores que decidan hacer uso de una metodología cognitivista, o a los estudiantes que trabajan en una actividad constructivista.

Justificación de funcionalidades propias

- **Los usuarios se pueden registrar en los cursos que deseen:** Permite a los estudiantes inscribirse, como oyentes, en cursos de su interés.

B.1.17. Portafolios

Permite crear un repositorio digital que mantenga un registro de las actividades realizadas, los títulos académicos, premios o certificaciones obtenidas, referencias personales, entre otros componentes que proporcionen información sobre las competencias adquiridas por el usuario.

Justificación de funcionalidades propias

- **El programa de los cursos completados y aprobados se agregan automáticamente al portafolio personal:** Una vez finalizado y aprobado un curso, el sistema lo agrega al historial de actividades realizadas por el usuario. De esta manera quedan reflejados los conocimientos adquiridos en dicho curso.
- **Publicación del portafolios con microformatos (hResume):** Brinda la posibilidad de utilizar HTML semántico para exportar los datos y el portafolios.
- **El portafolios se puede exportar para ser impreso o enviado digitalmente:** Le brinda al propietario del portafolio la posibilidad de convertir el currículum virtual a un formato que le facilite la impresión, para entregas en físico, o el envío por correo electrónico.

B.1.18. Registro e integración

- **Api para obtener información de los usuarios de un curso:** Esta funcionalidad permitiría a otras instancias de la institución educativa a utilizar la aplicación como una base de datos

para su utilización en otros sistemas.

B.1.19. Administración automatizada de pruebas

Las pruebas son herramientas disponibles para el profesor, cuya finalidad es medir el desempeño del estudiante en el curso. El profesor puede aplicarlas siguiendo el enfoque pedagógico de su preferencia. Algunos de los tipos de prueba son:

- Selección múltiple
- Verdadero- falso
- Matching
- Ordenamiento
- Preguntas abiertas
- Preguntas cerradas
- Completar el espacio en blanco
- Encuestas
- Ensayos
- Preguntas que contengan imágenes, video o audio
- Definidas por el usuario

Justificación de funcionalidades propias

- **Generar preguntas y respuestas de forma aleatoria:** Brinda la posibilidad de que el sistema seleccione, dadas varias opciones de preguntas, aquellas que serán colocadas en la prueba.
- **El sistema debe soportar un editor que permita la inclusión de fórmulas:** El sistema suministra un editor de texto que permita construir fórmulas para aquellas preguntas o respuestas que así lo requieran.

- **Los profesores pueden establecer el tiempo límite de las pruebas:** Las pruebas presentadas de forma virtual pueden tener, al igual que las pruebas presenciales, un límite de tiempo en el cual deben ser completadas. El profesor puede establecer el tiempo estimado para la realización de las pruebas.
- **Los profesores pueden mostrar las respuestas correctas como feedback:** Una vez que las pruebas hayan sido completadas por todos los estudiantes el profesor puede publicar las respuestas correctas de las preguntas contenidas en la prueba.
- **El sistema debe proveer seguridad sobre las pruebas:** El sistema debe garantizar que cada prueba pertenezca y pueda ser modificada sólo por un estudiante y por el profesor encargado del curso.
- **Pruebas restringidas por dirección IP:** Cada prueba será asociada con una dirección IP, como una de los mecanismos de seguridad aplicados a las pruebas.

B.1.20. Manejo de Puntuaciones

Justificación de funcionalidades propias

- **Se puede puntuar cada estudiante en todas las preguntas o cada pregunta de todos los estudiantes:** El profesor tiene la libertad de corregir toda la prueba, para cada estudiante, o corregir las pruebas de todo el curso por pregunta (para todos los estudiantes se corrige la misma pregunta y una vez finalizada la corrección de ésta se avanza a la siguiente. Este proceso se repite hasta completar la corrección de toda la prueba)

B.1.21. Control de calificaciones del curso (gradebooks)

Justificación de funcionalidades propias

- **El profesor puede agregar calificaciones para actividades off-line:** Permite incorporar a la tabla de calificaciones aquellas actividades que no son realizadas de forma virtual y que deben ser evaluadas.
- **Se pueden exportar las calificaciones contenidas en la tabla:** Brinda la posibilidad de convertir el tabla de calificaciones presentada en formato digital a un formato que le facilite la

impresión o el envío por correo electrónico.

- **Se puede crear una escala de evaluación que contemple porcentajes, status de aprobado/reprobado, calificación alfabética:** El profesor o preparador puede establecer la escala de puntuación de su preferencia o que se adapte a sus necesidades.

B.1.22. Seguimiento de usuarios

Esta herramienta permite medir el desempeño o el progreso del estudiante bajo cualquiera de los enfoques pedagógicos propuestos.

Justificación de funcionalidades propias

- **Estadísticas de lo que más hace un estudiante/ todos los estudiantes:** Permite llevar un registro de aquellas actividades o herramientas que son más frecuentadas dentro de un curso. Esto podría suministrar la información para que el profesor, si así lo desea, tome en cuenta como evaluación la participación de cada estudiante dentro del sistema.
- **Historial de navegación de cada estudiante:** Ofrece la posibilidad de mantener un informe detallado de las herramientas del sistema utilizadas por los estudiantes inscritos en un curso.

B.1.23. Accesibilidad

Justificación de funcionalidades propias

- **La aplicación entera debe satisfacer los estándares de accesibilidad, tanto para la visualización como para la edición de cursos:**
- **La edición de los campos de texto debe tener un checker de accesibilidad:** La implementación de dicha revisión debe hacerse de tal manera que cualquier usuario pueda corregir los errores.

B.1.24. Compartir/Reusar

El sistema brinda la posibilidad de compartir la información, de acuerdo a la relevancia o potencial utilidad que tenga para los estudiantes. Esto puede o no contribuir con el enfoque pedagógico

conectivista de acuerdo a la apertura que le de el autor al material pedagógico.

Justificación de funcionalidades propias

- **Repositorio de learning objects:** la existencia de un repositorio de learning objects le permite a los profesores guardar y, opcionalmente, publicar de manera estructurada uno o varios recursos pedagógicos así como encontrar otros recursos que puedan ser de su interés.
- **Conexión a repositorios externos el acceso a otros repositorios:** permitirá una mayor diversidad en los enfoques y en el nivel de calidad de los recursos.
- **Palabras claves o etiquetas para los learning objects:** De manera de facilitar la búsqueda de los recursos almacenados por el sistema, se permitirá a los usuarios crear una glosario por medio del etiquetado de los recursos.
- **Capacidad de importar y exportar el curso en SCORM, IMS, AICC:** Provee al profesor la oportunidad de exportar el contenido de un curso de manera que pueda ser utilizado por los estudiantes o aquellas personas interesadas, aún si no están conectados a internet.

B.1.25. Otras funcionalidades

- **Los cursos pueden ser cerrador o eliminados:** Los cursos pueden ser cerrados, una vez finalizado el período de clases en curso, o eliminados si el curso no está considerado para ser abierto nuevamente.
- **Los contenidos de los cursos eliminados no se pueden recuperar:** Si el curso es eliminado todas la herramientas y contenidos relacionados con él son eliminados. Por esta razón no existe la posibilidad de recuperarlos.
- **Los contenidos de los cursos cerrados se restauran una vez que el curso se reabre:** Si el curso es cerrado su contenido y herramientas permanecen inalteradas hasta el momento en que se reabra el mismo.
- **Al reabrir un curso, se da la opción para restaurar los profesores y estudiantes o no:** al momento de reabrir el curso se permitirá seleccionar a los usuarios que se desea mantener inscritos en el curso.

- **Utilizar learning objects para el almacenamiento de pruebas anteriores o modelos de prueba que puedan ser utilizados como material de estudio** El profesor, al registrar una prueba podrá almacenarla como learning object. Opcionalmente se podrá almacenar junto a las respuestas correctas.
- **Exportar a:** El sistema y sus módulos permitirán exportar en PDF o imprimir los contenidos.
- **Opción para la herramienta de manejo de archivos:** manejo de versiones para archivos, detección de virus al subir y descargar.

Apéndice C

Información Complementaria

C.1. Learning Objects

Es un término amplio que se refiere a:

- “Cualquier entidad, digital o no, que puede ser usado para el aprendizaje, educación o entrenamiento.”
- “Cualquier recurso que puede ser reusado para dar soporte al aprendizaje.”
- “Una entidad digital que puede ser usada, reusada o referenciada durante el aprendizaje apoyado por la tecnología.”

Estas entidades educativas son descritas por un modelo de datos estandarizado, codificado con XML. El propósito de este modelo de datos es facilitar el descubrimiento de los contenidos, usualmente dentro del contexto de un LMS.

Las principales características de un Learning Object son:

- **Auto-contenido y reusable**

Cada learning object es independiente y puede ser compartido y reusado, en distintas ocasiones y para distintos fines, como una entidad indivisible.

- **Entidad de aprendizaje reducida**

Al contrario de una clase completa, los learning object contienen recursos más breves de aprendizaje de unos 2 a 15 minutos.

- **Pueden ser agregados**

Los learning objects pueden ser agrupados en colecciones más grandes, incluso en estructuras tradicionales de educación.

- **Rastreados con metadatos**

Cada learning object contiene información descriptiva sobre la que se pueden realizar búsquedas.

Los principales usos de este estándar son para permitir...

- A a los estudiantes e instructores, buscar, evaluar, adquirir y utilizar objetos de aprendizaje.
- El intercambio de Learning Objects entre LMS que soporten la tecnología.
- A organizaciones educativas, expresar los contenidos educativos en una manera estandarizada e independiente del contenido en sí.
- La creación de objetos de aprendizaje en unidades que puedan ser combinadas en formas significativas.

(?)

C.2. Pautas de Accesibilidad Web

Las pautas dictadas por la W3C (?) contienen información técnica que se ha intentado resumir y explicar en la continuación:

1. **Proporcione alternativas equivalentes para el contenido visual y auditivo**

Con la intención de permitir el acceso a contenidos no textuales a personas discapacitadas.

2. **No se base sólo en el color**

Los colores no deben ser usados como guías visuales, o al menos deben proveerse alternativas de tal modo que las personas con discapacidades visuales reciban la misma información. Se debe asegurar el contraste entre el fondo y el primer plano en el texto y en las imágenes.

3. **Utilice marcadores y hojas de estilo y hágalo apropiadamente.**

El uso de las etiquetas de marcaje definidas en html deben ser usadas de acuerdo a su semántica asociada. Se destacan los siguientes usos inapropiados:

- El uso de la etiqueta BLOCKQUOTE para generar sangrías cuando el uso adecuado es para citar textos.
- El uso de etiquetas de títulos (H1, H2, etc.) para manipular el tamaño de la letra.

4. **Identifique el idioma usado**

Identificar el idioma predominante del contenido y de las distintas secciones (en caso de que cambie) permite a los navegadores especializados cambiar el idioma y adecuarse para reconocer el cambio de idioma. Así mismo se debe expandir el significado de los acrónimos (ACRONYM) y abreviaciones (ABBR) la primera vez que aparecen en el documento.

5. **Cree tablas que se transformen correctamente**

Las tablas deben ser usadas únicamente para información tabular (tablas de datos) y no como elementos para hacer la maquetación de los contenidos. Se debe hacer uso del marcaje para definir los encabezados, resumen, agrupamiento de celdas y demás dentro de la tabla.

6. **Asegúrese de que las páginas que incorporan nuevas tecnologías se transformen correctamente.**

Los contenidos deben ser legibles y comprensibles cuando las hojas de estilos son desactivadas, cuando las capacidades de scripting del lado del usuario (Javascript) estén deshabilitadas,

y que los contenidos accesibles por medios dinámicos sean accesibles sin dichas capacidades. Debe hacerse uso de Javascript no obstructivo (?).

7. Asegure al usuario el control sobre los cambios de los contenidos tempo-dependientes

Debe permitirse a los usuarios detener cualquier contenido que presente algún tipo de movimiento o parpadeo.

8. Asegure la accesibilidad directa de las interfaces de usuario incrustadas

Los objetos incrustados (applets y otros) deben asegurar los principios de un diseño accesible: funcionalidad de acceso independiente del dispositivo, teclado operable, voz automática, etc.

9. Diseñe para la independencia del dispositivo

Permita la activación de los elementos independientemente de los dispositivos de entrada. Definir el orden en que se recorren los elementos de la interfaz con el tabulador (tabindex) y métodos abreviados mejoran la accesibilidad (sin embargo, estos últimos deberán ser definidos por el usuario a fin de evitar conflictos con métodos abreviados del sistema operativo).

10. Utilice soluciones provisionales

Utilice soluciones de accesibilidad provisionales de forma que las ayudas técnicas y los antiguos navegadores operen correctamente. (Estas indicaciones perdieron vigencia debido a los avances desde la creación del documento - en 1999)

11. Utilice las tecnologías y pautas W3C

El uso de tecnologías no desarrolladas por la W3C como shockwave y PDF deben ser vistos como plugins y evitados en todo lo posible debido a que no ofrecen las características accesibles ofrecidas por la tecnología de la W3C. En la realidad, los avances en accesibilidad se pueden constatar en la publicación las guías de accesibilidad para PDF y para Flash publicadas por Adobe en <http://www.adobe.com/accessibility/>. Sin embargo, se recomienda disponer una página accesible como alternativa.

12. Proporcione información de contexto y orientación

Algunos elementos como listas de items pueden ser difíciles de manejar así como las etiquetas asociadas a los elementos de un formulario, se recomienda separarlos por medio del uso de OPTGROUP en el primer caso y usar la etiqueta LABEL en el segundo.

13. Proporcione mecanismos claros de navegación

Se debe identificar el “destino” o funcionalidad de un enlace, agregar metadatos, tablas de

contenidos o mapas del sitio, agrupar los vínculos relacionados (barras de navegación), si se proporcionas funcionalidades de búsqueda se deben permitir distintos niveles de habilidad, localizar la información relevante al principio del párrafo y hacer los títulos relevantes y entendibles fuera de contexto (ayuda al hojear el contenido), si un documento se extiende por varias página proporcione dicha información con la etiqueta LINK y sus atributos rel y rev (<http://www.seoconsultants.com/meta-tags/link-relationship.asp>) y permita maneras de saltar elementos o grupos de elementos que sean comúnmente ignorados cuando se busca leer el contenido (navegación, búsqueda, etc.)

14. Asegúrese de que los documentos sean claros y simples

Se debe asegurar que las imágenes tienen textos equivalentes (atributo alt) de modo que los discapacitados visuales, o quienes no pueden o han elegido no ver los gráficos tengan acceso a la información. Así mismo, la utilización de un lenguaje claro y simple promueve una comunicación efectiva.

Apéndice D

Revisión de otros LMS

Para realizar la selección de funcionalidades que se podrían incorporar a Ósmosis2 se estudiaron varios sistemas de gestión del aprendizaje. A continuación se muestran las características de cada LMS y las semejanzas entre ellos (?).

D.1. Sistemas revisados

D.1.1. Sakai 2.3

Enfoque Pedagógico: Constructivismo, cognitivismo.

Características:

■ **Foros:**

- Permite habilitar/deshabilitar el recibir los posts como e-mails.
- Herramienta de revisión ortográfica.
- Los foros pueden hacerse visibles o bloquearse (ver y no participar) en una fecha específica
- Los estudiantes pueden ser moderadores de foros
- Se pueden marcar las discusiones como “leídas”

■ **Diario online/notas:**

- Los estudiantes pueden agregar material (aprobado por el profesor) de utilidad en la herramienta “Recursos”.
- Se muestran los usuarios conectados que están usando el sistema.

■ **Portafolios:**

- OSP (Open Source Portfolio), herramienta provisional que ofrece las funcionalidades de un portafolio.

■ **Pizarra:**

- Se puede incorporar Elluminate, Breeze, entre otros.

■ **Intercambio de Archivos:**

- Entrega de tareas a través de “drop boxes” o casilleros del profesor.

■ **Correo Interno:**

- Envío de mensajes a toda la clase

■ **Chat:**

- El sistema crea archivos de registro para todas las salas de chat.

■ **Calendario/Agenda:**

- Los profesores y los estudiantes pueden añadir eventos al el calendario del curso.
- El estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
- El estudiante puede ver la calificación de las actividades realizadas, el total de puntos posibles, calificación del curso y comparar sus calificaciones con las del salón.

■ **Comunidad:**

- Sakai Wiki Tool: Permite a los usuarios crear, compartir y administrar contenido en un ambiente de Wiki. Se pueden compartir conceptos con otros Wikis open source como Wikipedia, TWiki, phpWiki, etc.

- News/RSS - para visualizar el contenido de fuentes online (RSS feeds)
- **Capacidades de búsqueda:**
 - El estudiante puede realizar búsquedas en todo el cursos.
 - El estudiante puede buscar en todos los hilos de discusión.–
- **Ayuda y Orientación del curso:**
 - El sistema incluye tutoriales en línea para ayudar al estudiante a aprender el uso del sistema
 - Los estudiantes pueden acceder a un menú de ayuda para cualquiera de las herramientas del sistema.
- **Grupos de Trabajo:**
 - El profesor puede asignar estudiantes a grupos.
 - A cada grupo se le pueden asignar tareas o actividades específicas.
 - Los grupos pueden ser privados o monitoreados por el profesor.
- **Tipos de pruebas:**
 - Tipos de pruebas: selección múltiple, respuestas múltiples, completar el espacio en blanco, preguntas cortas, encuestas, ensayos, matching, calculadas, preguntas que contengan imagenes, videos , audio,etc.
 - Manejo automatizado de Pruebas
 - El sistema puede aleatorizar las preguntas y respuestas.
 - El profesor puede crear exámenes de prueba.
 - El profesor puede definir tiempos límites para las evaluaciones.
 - El profesor puede permitir múltiples intentos.
 - Los estudiantes puede revisar intentos anteriores de una prueba.
 - Los profesores definen si se debe mostrar las respuestas correctas.
 - Soporte para pruebas automatizadas

- Los profesores pueden crear un banco personal de pruebas.
 - El sistema provee análisis de los datos de las pruebas.
 - Se pueden importar preguntas desde bancos de pruebas que soporten QTI.
 - Herramientas de puntuación online
 - Se puede puntuar cada estudiante en todas las preguntas o cada pregunta de todos los estudiantes.
 - Control de calificaciones del curso (gradebooks):
 - Cuando el profesor agrega una actividad al curso, ésta se agrega automáticamente a la tabla de actividades a evaluar.
 - Se puede crear una escala de evaluación que contemple porcentajes, estatus de aprobado/reprobado, calificación alfabética.
 - El profesor puede agregar calificaciones para actividades off-line.
 - Se pueden exportar las calificaciones contenidas en la tabla.
 - Los profesores pueden evaluar las respuestas de los estudiantes de forma anónima.
- **Seguimiento del estudiante:**
- Site Statistics tool: agrupa y muestra datos estadísticos acerca de la clase, como o un todo o por cada estudiante, lo que han hecho y los sitios que han accedido.
- **Manejo del Curso:**
- El instructor puede programar la aparición de eventos, tareas o pruebas, anuncios en el sistema de acuerdo a una fecha de inicio y una fecha de fin.
- **Plantillas del curso:**
- El sistema permite a los administradores usar un curso existente o una plantilla predefinida como base para un nuevo curso.
 - El contenido del curso puede ser cargado a través de WebDAV.
- **Personalización de la interfaz:**

- El sistema proporciona plantillas (por defecto) para la interfaz de los cursos.
- Los profesores pueden modificar los iconos y colores de un curso.

■ **Herramientas de diseño de cursos:**

- Los profesores pueden organizar los learning objects, las herramientas del curso y el contenido del material que puede ser reusable.
- Reutilización de los cursos como plantilla para otros cursos en los cuales se maneje la misma información.

■ **Compartir/Reusar contenido:**

- El material puede ser importado o copiado del sitio de un curso a otro.

■ **Requerimientos de base de datos:**

- El sistema soporta MySQL.
- El sistema soporta Oracle.
- La aplicación requiere sólo una base de datos y puede coexistir con tables de otras aplicaciones.

■ **Comunidad de Usuarios**

D.1.2. Claroline 1.8.1

Enfoque Pedagógico: Cognitivista, constructivista.

Características:

■ **Foros:**

- Permite habilitar/deshabilitar el recibir los posts como e-mails.
- Herramienta de revisión ortográfica

■ **Manejo de discusiones:**

- Los profesores pueden permitir a los estudiantes la creación de grupos de discusión.
- **Intercambio de Archivos:**
 - Entrega de tareas a través de “drop boxes” o casilleros del profesor.
 - Posibilidad de compartir archivos
 - Notificación de cambios en el material a través de RSS.
- **Correo Interno:**
 - Envío de mensajes a toda la clase.
- **Chat:**
 - Soporte ilimitado de grupos de discusión simultáneos.
 - El sistema crea archivos de registro para todas las salas de chat.
- **Calendario/Agenda:**
 - El estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
- **Capacidades de búsqueda:**
 - El estudiante puede buscar en todos los hilos de discusión.
- **Grupos de Trabajo:**
 - El profesor puede asignar estudiantes a grupos.
 - El sistema puede crear grupos, de forma aleatoria y con un cierto tamaño.
 - A cada grupo se le pueden asignar tareas o actividades específicas.
 - Los grupos pueden ser privados o monitoreados por el profesor.
 - Los estudiantes escoger o seleccionar grupos.
- **Tipos de pruebas:**

- Tipos de pruebas: selección múltiple, respuestas múltiples, completar el espacio en blanco, preguntas que contengan imágenes, videos , audio,etc.
- Manejo automatizado de Pruebas
- El sistema puede aleatorizar las preguntas y respuestas.
- El profesor puede crear exámenes de prueba.
- El profesor puede definir tiempos límites para las evaluaciones.
- El profesor puede permitir múltiples intentos.
- Los estudiantes puede revisar intentos anteriores de una prueba.
- Los profesores definen si se debe mostrar las respuestas correctas.
- Soporte para pruebas automatizadas
- Los profesores pueden crear un banco personal de pruebas.
- El sistema provee análisis de los datos de las pruebas.
- Se pueden importar preguntas desde bancos de pruebas que soporten QTI.

■ **Seguimiento del estudiante:**

- Historial de navegación de cada estudiante.
- Los profesores pueden obtener reportes del número de veces, tiempo, fecha, frecuencia y dirección IP de cada estudiante que ha accedido al contenido de un curso, foro, tareas, etc.
- Estadísticas de lo que más hace un estudiante/ todos los estudiantes.

■ **Personalización de la interfaz:**

- Las instituciones pueden crear su propio estilo y plantillas para todo el sistema, incluyendo logo, encabezados y pie de páginas.

■ **Requerimientos de base de datos:**

- El sistema soporta MySQL.
- La aplicación requiere sólo una base de datos y puede coexistir con bases de otras aplicaciones.

■ **Comunidad de Usuarios:**

(?)

- 93 países, entre los cuáles se encuentran: Argentina, Australia, Brasil, Canadá, Colombia, Francia, Alemania, Israel, Mexico, entre otros.
- 1184 organizaciones.

D.1.3. ATutor 1.5.4

Enfoque Pedagógico: Cognitivista, constructivista.

Características:

■ **Foros:**

- Permite habilitar/deshabilitar el recibir los posts como e-mails.
- Los estudiantes pueden ser asignados como administradores de foros.

■ **Manejo de discusiones:**

- Las discusiones pueden ser compartidas entre cursos, departamentos y cualquier otra unidad de la institución.

■ **Intercambio de Archivos:**

- Entrega de tareas a través de “drop boxes” o casilleros del profesor
- Administradores pueden definir el tamaño de espacio en disco para cada usuario
- Los estudiantes tienen una carpeta privada en la cual pueden almacenar sus archivos.

■ **Correo Interno:**

- Envío de mensajes a toda la clase
- Libreta de direcciones
- Capacidad de hacer forward a correo externo.

■ **Chat:**

- Soporte ilimitado de grupos de discusión simultáneos.
- El sistema crea archivos de registro para todas las salas de chat.
- Los estudiantes pueden crear nuevas salas de chat.
- **Pizarra:**
 - La pizarra soporta imágenes y archivos PowerPoint.
- **Calendario/Agenda:**
 - Los profesores y los estudiantes pueden añadir eventos al calendario del curso.
 - El estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
- **Capacidades de búsqueda:**
 - El estudiante puede realizar búsquedas en todo el curso.
- **Trabajo desconectado:**
 - Los estudiantes pueden “compilar” y descargar todo el contenido de un curso para ser impreso o guardado localmente.
- **Ayuda y Orientación del curso:**
 - Los estudiantes pueden acceder a un menú de ayuda para cualquiera de las herramientas del sistema.
 - Los estudiantes y profesores pueden contribuir con el material de ayuda
 - El material de ayuda es buscable.
- **Grupos de Trabajo:**
 - El profesor puede asignar estudiantes a grupos.
 - El sistema puede crear grupos, de forma aleatoria y con un cierto tamaño.
 - Cada grupo puede tener su propio foro de discusión.
 - A cada grupo se le pueden asignar tareas o actividades específicas.

- Los grupos pueden ser privados o monitoreados por el profesor.
- **Comunidad:**
 - Los estudiantes, de distintos cursos, pueden participar en todas las salas de chat o foros del sistema.
 - Se pueden asignar tests/quizzes a los grupos.
- **Tipos de pruebas:**
 - Tipos de pruebas: selección múltiple, respuestas múltiples, completar el espacio en blanco, preguntas cortas, encuestas, ensayos, preguntas que contengan imágenes, videos , audio,etc.
 - Manejo automatizado de Pruebas
 - El sistema puede aleatorizar las preguntas y respuestas.
 - El profesor puede crear exámenes de prueba.
 - El profesor puede definir tiempos límites para las evaluaciones.
 - El profesor puede permitir múltiples intentos.
 - Los estudiantes puede revisar intentos anteriores de una prueba.
 - Los profesores definen si se debe mostrar las respuestas correctas.
- **Soporte para pruebas automatizadas:**
 - Los profesores pueden crear un banco personal de pruebas.
 - El sistema provee análisis de los datos de las pruebas.
 - Se pueden importar preguntas desde bancos de pruebas que soporten QTI.
- **Herramientas de puntuación online:**
 - Se puede puntuar cada estudiante en todas las preguntas o cada pregunta de todos los estudiantes.
 - Las puntuaciones pueden ser parciales para respuestas parcialmente correctas
- **Manejo automatizado de pruebas**

- Los resultados de las pruebas pueden permanecer ocultos hasta que todas las pruebas hayan sido recibidas.
- Los resultados de la prueba pueden ser liberados en el momento en que un estudiante entrega la prueba.
- El sistema puede mostrar resultados parciales de las preguntas de selección y verdadero/falso.

■ **Manejo del Curso:**

- El instructor puede programar la aparición de eventos, tareas o pruebas, anuncios en el sistema de acuerdo a una fecha de inicio y una fecha de fin.
- Los profesores pueden enlazar las discusiones a fechas específicas o eventos del curso.

■ **Seguimiento del estudiante:**

- Historial de navegacion de cada estudiante.

■ **Compartir/Reusar contenido:**

- Los profesores pueden compartir contenidos con otros profesores o estudiantes a través de un repositorio central de learning objects.

■ **Plantillas del curso:**

- El sistema permite a los administradores usar un curso existente o una plantilla predefinida como base para un nuevo curso.

■ **Personalización de la interfaz:**

- El profesor puede cambiar el orden y los nombres de los items ubicados en el menú del curso.
- El sistema puede soportar múltiples instituciones, departamentos, escuelas u otras organizaciones en una sola instalación donde cada unidad puede tener su estilo y sus plantillas así como imágenes, encabezados u pie de páginas.

■ **Herramientas de diseño de cursos:**

- Reutilización de los cursos como plantilla para otros cursos en los cuales se maneje la misma información.

■ **Requerimientos de base de datos:**

- El sistema soporta MySQL.
- La aplicación requiere sólo una base de datos y puede coexistir con tables de otras aplicaciones.

■ **Comunidad de Usuarios**

D.1.4. Moodle 1.8

Enfoque Pedagógico: Constructivismo social (Se puede adaptar para el uso cognitivista puro).

Características:

■ **Foros:**

- Permite habilitar/deshabilitar el recibir los posts como e-mails.
- Herramienta de revisión ortográfica.
- Posibilidad de que los estudiantes sean administradores de foros individuales del curso
- RSS de los posts

■ **Manejo de discusiones:**

- Los profesores pueden permitir a los estudiantes la creación de grupos de discusión.
- Las discusiones pueden ser compartidas entre cursos, departamentos y cualquier otra unidad de la institución.
- Se debe opinar en el foro como requisito para ver los posts del resto de los estudiantes que participan.

■ **Intercambio de Archivos:**

- Entrega de tareas a través de “drop boxes” o casilleros del profesor.

■ **Correo Interno:**

- Envío de mensajes a toda la clase.
- Libreta de direcciones.
- Capacidad de hacer forward a correo externo.

■ **Chat:**

- Soporte ilimitado de grupos de discusión simultáneos.
- El sistema crea archivos de registro para todas las salas de chat.
- Los estudiantes pueden crear nuevas salas de chat.

■ **Calendario/Agenda:**

- Los profesores y los estudiantes pueden añadir eventos al el calendario del curso.
- El estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
- El estudiante puede ver la calificación de las actividades realizadas, el total de puntos posibles, calificación del curso y comparar sus calificaciones con las del salón.

■ **Pizarra:**

- Se puede incorporar Elluminate, DimDim, etc.

■ **Capacidades de búsqueda:**

- El estudiante puede buscar en todos los hilos de discusión.

■ **Ayuda y Orientación del curso:**

- El sistema incluye tutoriales en línea para ayudar al estudiante a aprender el uso del sistema
- Los estudiantes pueden acceder a un menú de ayuda para cualquiera de las herramientas del sistema.

■ **Grupos de Trabajo:**

- El profesor puede asignar estudiantes a grupos.
- Cada grupo puede tener su propio foro de discusión.
- Cada grupo puede tener chat o pizarra.
- Los estudiantes, de distintos cursos, pueden participar en todas las salas de chat o foros del sistema.

■ **Portafolios:**

- Los estudiantes pueden crear una página principal para cada curso.

■ **Tipos de prueba**

- Tipos de pruebas: selección múltiple, respuestas múltiples, completar el espacio en blanco, matching, ordenamientos, calculadas, encuestas, ensayos, ordenar una frase. Se pueden personalizar el tipo de preguntas a realizar, preguntas que contengan imágenes, videos , audio,etc..

■ **Manejo automatizado de pruebas**

- El sistema puede aleatorizar las preguntas y respuestas.
- El profesor puede crear exámenes de prueba.
- El profesor puede definir tiempos límites para las evaluaciones.
- El profesor puede permitir múltiples intentos.
- Los estudiantes puede revisar intentos anteriores de una prueba.
- Los profesores definen si se debe mostrar las respuestas correctas.
- El sistema soporta el Protocolo de Quices Remotos, que permite que las preguntas se muestren y sean evaluadas externamente por medio de servicios web.

■ **Soporte para pruebas automatizadas:**

- Los profesores pueden crear un banco personal de pruebas.
- El sistema provee análisis de los datos de las pruebas.
- Se pueden importar preguntas desde bancos de pruebas que soporten QTI.

■ Herramientas de puntuación online:

- Se puede puntuar cada estudiante en todas las preguntas o cada pregunta de todos los estudiantes.

■ Control de calificaciones del curso (gradebooks):

- Cuando el profesor agrega una actividad al curso, ésta se agrega automáticamente a la tabla de actividades a evaluar.
- Se puede crear una escala de evaluación que contemple porcentajes, estatus de aprobado/reprobado, calificación alfabética.
- El profesor puede agregar calificaciones para actividades off-line.
- Se pueden exportar las calificaciones contenidas en la tabla.

■ Seguimiento del estudiante:

- Historial de navegación de cada estudiante.
- Los profesores pueden obtener reportes del número de veces, tiempo, fecha, frecuencia y dirección IP de cada estudiante que ha accedido al contenido de un curso, foro, tareas, etc.
- Estadísticas de lo que más hace un estudiante/ todos los estudiantes.

■ Plantillas del curso:

- El sistema permite a los administradores usar un curso existente o una plantilla predefinida como base para un nuevo curso.
- El contenido del curso puede ser cargado a través de WebDAV.

■ Personalización de la interfaz:

- El sistema proporciona plantillas (por defecto) para la interfaz de los cursos.
- Los profesores pueden modificar los iconos y colores de un curso.
- El profesor puede cambiar el orden y los nombres de los items ubicados en el menú del curso.

- El sistema puede soportar múltiples instituciones, departamentos, escuelas u otras organizaciones en una sola instalación donde cada unidad puede tener su estilo y sus plantillas así como imágenes, encabezados u pie de páginas.

■ **Herramientas de diseño de cursos:**

- Los profesores pueden organizar los learning objects, las herramientas del curso y el contenido del material que puede ser reusable.
- Reutilización de los cursos como plantilla para otros cursos en los cuales se maneje la misma información.

■ **Compartir/Reusar contenido:**

- Los profesores pueden hacer una copia completa de todo el curso y/o items individuales del curso y compartirlos con otros profesores o cargarlos en uno de los sistemas eCMS con integración en Moodle (Hive, Odalis, etc.)

■ **Requerimientos de base de datos:**

- El sistema soporta MySQL.
- El sistema soporta Oracle.
- El sistema soporta MS SQL Server
- La aplicación requiere sólo una base de datos y puede coexistir con bases de otras aplicaciones.
- El sistema soporta PostgreSQL.

■ **Comunidad de Usuarios:**

(?)

199 países entre los cuales se encuentran: Argentina, Australia, Brasil, Canadá, Chile, China, Colombia, Ecuador, El Salvador, España, Estados Unidos, Japón, India, Italia, Irlanda, Israel, Mexico, Panamá, Paraguay, Perú, Venezuela, entre otros.

D.1.5. Blackboard Learning System Vista 4.1 Enterprise License

Enfoque Pedagógico: Cognitivismo, constructivismo.

Características:

■ **Foros:**

- Herramienta de revisión ortográfica.
- Asociar una discusión al contenido de un curso.

■ **Intercambio de Archivos:**

- Entrega de tareas a través de “drop boxes” o casilleros del profesor
- Administradores pueden definir el tamaño de espacio en disco para cada usuario
- Los estudiantes tienen una carpeta privada en la cual pueden almacenar sus archivos.–

■ **Correo Interno:**

- Envío de mensajes a toda la clase.
- Libreta de direcciones.
- Capacidad de hacer forward a correo externo.
- Los estudiantes pueden colocar notas a cualquier página.
- Posibilidad de que los estudiantes agreguen al material del curso sus apuntes para crear una guía de estudio.–

■ **Chat:**

- Soporte ilimitado de grupos de discusión simultáneos.
- El sistema crea archivos de registro para todas las salas de chat.
- Los estudiantes pueden ver quien más está conectado en el sistema e invitarlo a la sala de chats.
- El sistema NO limita el número de salas simultáneas.

■ **Pizarra:**

- La pizarra soporta imagenes y archivos PowerPoint.

■ **Calendario/Agenda:**

- Los profesores y los estudiantes pueden añadir eventos al calendario del curso.
- El estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
- El estudiante puede ver la calificación de las actividades realizadas, el total de puntos posibles, calificación del curso y comparar sus calificaciones con las del salón.

■ **Capacidades de búsqueda:**

- El estudiante puede realizar búsquedas en todo el curso.
- El estudiante puede buscar en todos los hilos de discusión.
- Los estudiantes pueden restringir la búsqueda a través de filtros.

■ **Trabajo desconectado:**

- Los estudiantes pueden “compilar” y descargar todo el contenido de un curso para ser impreso o guardado localmente.

■ **Ayuda y Orientación del curso:**

- El sistema incluye tutoriales en línea para ayudar al estudiante a aprender el uso del sistema.
- Los estudiantes pueden acceder a un menú de ayuda para cualquiera de las herramientas del sistema.

■ **Grupos de Trabajo:**

- El profesor puede asignar estudiantes a grupos.
- El sistema puede crear grupos, de forma aleatoria y con un cierto tamaño.
- Cada grupo puede tener su propio foro de discusión.
- Cada grupo puede tener chat o pizarra.
- A cada grupo se le pueden asignar tareas o actividades específicas.
- Los grupos pueden ser privados o monitoreados por el profesor.
- Los estudiantes escoger o seleccionar grupos.

■ Portafolios:

- Los estudiantes pueden crear una página principal para cada curso.

■ Tipos de pruebas:

- Tipos de pruebas: selección múltiple, respuestas múltiples, completar el espacio en blanco, matching, calculadas, encuestas, ensayos, ordenar una frase, preguntas que contengan imágenes, videos , audio,etc.

■ Manejo automatizado de Pruebas:

- El sistema soporta el editor MathML para la inclusión de fórmulas matemáticas en las preguntas.
- El acceso a las pruebas puede ser restringido por la dirección IP
- Los profesores pueden crear conjuntos de preguntas y organizarlas por dificultad o habilidad.
- Los profesores pueden importar y exportar pruebas, quizzes, encuestas y autoevaluaciones para compartirlas con otros cursos.
- El sistema puede aleatorizar las preguntas y respuestas.
- El profesor puede crear exámenes de prueba.
- El profesor puede definir tiempos límites para las evaluaciones.
- El profesor puede permitir múltiples intentos.
- Los estudiantes puede revisar intentos anteriores de una prueba.
- Los profesores definen si se debe mostrar las respuestas correctas.

■ Soporte para pruebas automatizadas:

- Los profesores pueden crear un banco personal de pruebas.
- El sistema provee análisis de los datos de las pruebas.
- Se pueden importar preguntas desde bancos de pruebas que soporten QTI.
- Herramientas de puntuación online.

- Se puede puntuar cada estudiante en todas las preguntas o cada pregunta de todos los estudiantes.
- **Herramientas de puntuación online:**
 - Los profesores pueden evaluar las respuestas de los estudiantes de forma anónima.
- **Control de calificaciones del curso (gradebooks):**
 - Cuando el profesor agrega una actividad al curso, ésta se agrega automáticamente a la tabla de actividades a evaluar.
 - Se puede crear una escala de evaluación que contemple porcentajes, estatus de aprobado/reprobado, calificación alfabética.
 - El profesor puede agregar calificaciones para actividades off-line.
 - Se pueden exportar las calificaciones contenidas en la tabla.
- **Manejo del Curso:**
 - El instructor puede programar la aparición de eventos, tareas o pruebas, anuncios en el sistema de acuerdo a una fecha de inicio y una fecha de fin.
 - Los profesores pueden personalizar el acceso a material específico del curso basado en el desempeño del estudiante.
 - Los profesores pueden enlazar las discusiones a fechas específicas o eventos del curso.
- **Seguimiento del estudiante:**
 - Historial de navegacion de cada estudiante.
 - Los profesores pueden obtener reportes del número de veces, tiempo, fecha, frecuencia y dirección IP de cada estudiante que ha accedido al contenido de un curso, foro, tareas, etc.
 - Los profesores pueden exportar toda la data de seguimiento del estudiante así como compartirla con los mismos.
- **Compartir/Reusar contenido:**

- Los estudiantes pueden exportar su página principal personal.
- Los profesores pueden compartir contenidos con otros profesores o estudiantes a través de un repositorio central de learning objects.

■ **Plantillas del curso:**

- El sistema permite a los administradores usar un curso existente o una plantilla predefinida como base para un nuevo curso.
- El contenido del curso puede ser cargado a través de WebDAV.
- Personalización de la interfaz:
- El sistema proporciona plantillas (por defecto) para la interfaz de los cursos.
- Los profesores pueden modificar los iconos y colores de un curso.
- El profesor puede cambiar el orden y los nombres de los items ubicados en el menú del curso.
- El sistema puede soportar múltiples instituciones, departamentos, escuelas u otras organizaciones en una sola instalación donde cada unidad puede tener su estilo y sus plantillas así como imágenes, encabezados u pie de páginas.

■ **Herramientas de diseño de cursos:**

- Los profesores pueden organizar los learning objects, las herramientas del curso y el contenido del material que puede ser reusable.
- Reutilización de los cursos como plantilla para otros cursos en los cuales se maneje la misma información.

■ **Requerimientos de base de datos:**

- El sistema soporta Oracle.
- El sistema soporta MS SQL Server

■ **Comunidad de Usuarios:**

(?)

Algunas universidades que usan esta plataforma son:

- Universidad Baylor de Medicina, Texas
- Universidad de Carolina del Este
- Colegio Mount Sinai de Medicina, New York
- Universidad de Ciencias de la Salud Oregon, Oregon
- Escuela Superior de Medicina Osteopática de Filadelfia, Filadelfia
- Universidad Central del Caribe, Puerto Rico
- Universidad de Chicago-Escuela de Enfermeras, Illinois
- Universidad de Colorado Centrado en ciencias de la Salud, Colorado
- Universidad de Biología de Maryland Chesapeake, Maryland
- Universidad de Tennessee-Memphis, Tennessee
- Universidad de Texas Houston especializada en Odontología, Texas entre otras Universidades.

D.1.6. Desire2Learn 8.2

Enfoque Pedagógico: Cognitivismo, constructivismo.

Características:

■ **Foros:**

- Herramienta de revisión ortográfica.
- Asociar una discusión al contenido de un curso
- Los posts pueden incluir archivos media, ecuaciones, archivos adjuntos o direcciones URL.

■ **Manejo de discusiones:**

- Los profesores pueden permitir a los estudiantes la creación de grupos de discusión.
- Las discusiones pueden ser compartidas entre cursos, departamentos y cualquier otra unidad de la institución.

■ **Intercambio de Archivos:**

- Entrega de tareas a través de “drop boxes” o casilleros del profesor.
- Los administradores pueden definir el tamaño de espacio en disco para cada usuario.
- Los estudiantes tienen una carpeta privada en la cual pueden almacenar sus archivos.
- Detección de virus en el proceso de upload/download de archivos.
- Posibilidad de compartir archivos.

■ **Correo Interno:**

- Envío de mensajes a toda la clase.
- Libreta de direcciones.
- Capacidad de hacer forward a correo externo.
- Los estudiantes pueden colocar notas a cualquier página.
- Posibilidad de que los estudiantes agreguen al material del curso sus apuntes para crear una guía de estudio.

■ **Chat:**

- Soporte ilimitado de grupos de discusión simultáneos.
- El sistema crea archivos de registro para todas las salas de chat.
- Los estudiantes pueden crear nuevas salas de chat.
- Los profesores pueden moderar el chat y suspender estudiantes de las salas de chat.
- Soporta un número limitado de salas simultáneas.

■ **Pizarra:**

- La pizarra soporta imágenes y archivos PowerPoint.

■ **Calendario/Agenda:**

- Los profesores y los estudiantes pueden añadir eventos al el calendario del curso.

- El estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
- El estudiante puede ver la calificación de las actividades realizadas, el total de puntos posibles, calificación del curso y comparar sus calificaciones con las del salón.

■ **Capacidades de búsqueda:**

- El estudiante puede realizar búsquedas en todo el curso.
- El estudiante puede buscar en todos los hilos de discusión.

■ **Trabajo desconectado:**

- Los estudiantes pueden “compilar” y descargar todo el contenido de un curso para ser impreso o guardado localmente.

■ **Ayuda y Orientación del curso:**

- El sistema incluye tutoriales en línea para ayudar al estudiante a aprender el uso del sistema.
- Los estudiantes pueden acceder a un menú de ayuda para cualquiera de las herramientas del sistema.

■ **Grupos de Trabajo:**

- El profesor puede asignar estudiantes a grupos.
- El sistema puede crear grupos, de forma aleatoria y con un cierto tamaño.
- Cada grupo puede tener su propio foro de discusión.
- Cada grupo puede tener chat o pizarra.
- A cada grupo se le pueden asignar tareas o actividades específicas.
- Los grupos pueden ser privados o monitoreados por el profesor.
- Los estudiantes escoger o seleccionar grupos.
- Cada grupo puede tener su carpeta de presentaciones, lista de emails, encuestas, actividades, compartir calendario de eventos, intercambiar archivos y designar un líder de grupo.

■ Comunidad:

- Los estudiantes, de distintos cursos, pueden participar en todas las salas de chat o foros del sistema.

■ Portafolios:

- Los estudiantes pueden crear una página principal para cada curso.

■ Tipos de pruebas:

- Tipos de pruebas: selección múltiple, respuestas múltiples, completar el espacio en blanco, matching, ordenamientos, calculadas, encuestas, ensayos, preguntas que contengan imágenes, videos , audio,etc. Se pueden personalizar el tipo de preguntas a realizar.

■ Manejo automatizado de pruebas:

- El sistema puede aleatorizar las preguntas y respuestas.
- El profesor puede crear exámenes de prueba.
- El profesor puede definir tiempos límites para las evaluaciones.
- El profesor puede permitir múltiples intentos.
- Los estudiantes puede revisar intentos anteriores de una prueba.
- Los profesores definen si se debe mostrar las respuestas correctas.
- El sistema soporta la presentación de ecuaciones matemáticas con LaTeX o MathML.
- El acceso a las pruebas puede ser restringido por dirección IP.
- Los profesores pueden hacer conjuntos de preguntas, organizándolos por dificultad y habilidad.
- Se puede importar y exportar pruebas, encuestas y auto evaluaciones para compartirlas con otros cursos.

■ Soporte para pruebas automatizadas:

- Los profesores pueden crear un banco personal de pruebas.

- El sistema provee análisis de los datos de las pruebas.
- Se pueden importar preguntas desde bancos de pruebas que soporten QTI.
- **Herramientas de puntuación online:**
 - **Instructors can enable students to rate and comment on submissions of other students.**
- **Control de calificaciones del curso (gradebooks):**
 - Se puede crear una escala de evaluación que contemple porcentajes, estatus de aprobado/reprobado, calificación alfabética.
 - El profesor puede agregar calificaciones para actividades off-line.
 - Se pueden exportar las calificaciones contenidas en la tabla.
 - El software, automáticamente, calcula la nota mínima, máxima y promedio en cualquier item a evaluar incluyendo tareas y quices.
- **Manejo del Curso:**
 - El instructor puede programar la aparición de eventos, tareas o pruebas, anuncios en el sistema de acuerdo a una fecha de inicio y una fecha de fin.
 - Los profesores pueden enlazar las discusiones a fechas específicas o eventos del curso.
 - Los profesores pueden personalizar el acceso a material específico del curso basado en el desempeño del estudiante.
- **Seguimiento del estudiante:**
 - Historial de navegacion de cada estudiante.
 - Los profesores pueden obtener reportes del número de veces, tiempo, fecha, frecuencia y dirección IP de cada estudiante que ha accedido al contenido de un curso, foro, tareas, etc.
- **Compartir/Reusar contenido:**
 - Los profesores pueden compartir contenidos con otros profesores o estudiantes a través de un repositorio central de learning objects.

■ Plantillas del curso:

- El sistema permite a los administradores usar un curso existente o una plantilla predefinida como base para un nuevo curso.
- El contenido del curso puede ser cargado a través de WebDAV.

■ Personalización de la interfaz:

- El sistema proporciona plantillas (por defecto) para la interfaz de los cursos.
- Los profesores pueden modificar los iconos y colores de un curso.
- El profesor puede cambiar el orden y los nombres de los items ubicados en el menú del curso.
- El sistema puede soportar múltiples instituciones, departamentos, escuelas u otras organizaciones en una sola instalación donde cada unidad puede tener su estilo y sus plantillas así como imágenes, encabezados u pie de páginas.
- Los usuarios pueden seleccionar cualquier página como página principal de Contenidos.

■ Herramientas de diseño de cursos:

- Los profesores pueden organizar los learning objects, las herramientas del curso y el contenido del material que puede ser reusable.
- Reutilización de los cursos como plantilla para otros cursos en los cuales se maneje la misma información.

■ Requerimientos de base de datos:

- El sistema soporta MS SQL Server

D.2. Resumen

Entre los LMS estudiados se observan algunas herramientas con características comunes, entre ellas se encuentran:

- **Foro:** herramienta de revisión ortográfica, posibilidad de habilitar/deshabilitar el recibir los posts como e-mails.

- **Correo Interno:** envío de mensajes a toda la clase, capacidad de hacer forward a correo externo.
- **Chat:** el sistema crea archivos de registro para todas las salas de chat.
- **Manejo de discusiones:** los profesores pueden permitir a los estudiantes la creación de grupos de discusión.
- **Calendario/Agenda:**
 - el estudiante posee una página de inicio personal con los cursos en los cuales participa, todos los eventos o actividades de su calendario personal.
 - El estudiante puede ver la calificación de las actividades realizadas, el total de puntos posibles, calificación del curso y comparar sus calificaciones con las del salón.
- **Intercambio de Archivos:** Entrega de tareas a través de “drop boxes” o casilleros del profesor.
- **Capacidades de búsqueda:** El estudiante puede buscar en todos los hilos de discusión.
- **Ayuda y Orientación del curso:** Los estudiantes pueden acceder a un menú de ayuda para cualquiera de las herramientas del sistema.
- **Comunidad:** Los estudiantes, de distintos cursos, pueden participar en todas las salas de chat o foros del sistema.

Apéndice E

Manual del administrador

E.1. Manual de implantación

E.1.1. Requerimientos mínimos

Servidor de aplicación

- Servidor HTTP: apache 2, lighthttpd, etc.
- PHP 5.0 (5.2.3 recomendado) o posterior, con los módulo Rewrite (deseable) y Zip (necesario en caso de utilizar módulo “Lecciones”).
- Línea de comandos de PHP (php_cli).
- Acceso por shell.
- Subversion Client. Recomendado para la descarga de Ósmosis y CakePHP. En caso de no estar disponible, tener el *tarball* de Ósmosis.

Servidor de base de datos

- Cualquier servidor de base de datos relacional soportado por CakePHP: Oracle, PostgreSQL, MySQL, SQLite, etc.

Cliente

- Navegador web con buen soporte para CSS2.1 y Javascript.

E.1.2. Código de terceras personas

El siguiente es un listado de librerías que utiliza y se distribuyen junto a Ósmosis2.

HTMLPurifier

- Creado por: Edward Z. Yang
- Fecha de creación: 2006-2007
- Estado: en constante desarrollo.
- Sitio oficial: <http://htmlpurifier.org/>
- Restricciones de uso: ninguna.
- Licencia: GNU Lesser General Public
- Referencias de uso en Ósmosis: plugin wiki, plugin blog.
- Descripción: tiene doble función, en primer lugar corrige código HTML incorrecto para hacerlo cumplir con los estándares definidos por la W3C; y en el proceso elimina código de ataques XSS (cross-site scripting).

JQuery

- Creado por: John Resig
- Fecha de creación: 2008
- Estado: en constante desarrollo.
- Sitio oficial: <http://jquery.com/>
- Restricciones de uso: ninguna.

- Licencia: dual GPL/MTI
- Referencias de uso en Ósmosis: visor de archivos SCORM, selector múltiple, campos auto-completados, chat.
- Descripción: librería JavaScript que simplifica el manejo dinámicos de los elementos del HTML, manejo de eventos, animación y Ajax.
- Plugins de JQuery utilizados:
 - Autocomplete (<http://bassistance.de/jquery-plugins/jquery-plugin-autocomplete/>)
 - bgIframe (<http://brandonaaron.net>)
 - BlockUI (<http://malsup.com/jquery/block/>)
 - fieldSelection (<http://blog.0xab.cd>)
 - FlyDOM (<http://flydom.socianet.com/>)
 - Metadata (<http://plugins.jquery.com/project/metadata>)
 - OsmosisSelector
 - Star Rating (<http://www.fyneworks.com/jquery/star-rating/>)
 - ScrollTo (<http://flesler.blogspot.com/2007/10/jqueryscrollto.html>)

TinyMCE

- Creado por: Moxiecode Systems AB
- Fecha de creación: no registrada
- Estado: en constante desarrollo.
- Sitio oficial: <http://tinymce.moxiecode.com/>
- Restricciones de uso: ninguna.
- Licencia: GNU Lesser General Public
- Referencias de uso en Ósmosis: plugin wiki.

- Descripción: editor Javascript WYSIWYG, basado en la web, para HTML. Tiene la capacidad de convertir los textarea en editores de texto enriquecido.

E.1.3. Estructura del sistema

Ósmosis2 mantiene la estructura básica de una aplicación desarrollada con CakePHP (?):

- app (se referirá a este directorio como APP)
- cake (se referirá a este directorio como CAKE_DIR)
- docs
- index.php
- vendors

Sin embargo, Ósmosis2 no se distribuye junto con las librerías que conforman el core de CakePHP. Sólo se distribuye el equivalente al contenido del directorio *app*.

E.1.4. Instalación

El proceso de instalación de Ósmosis se puede dividir en cuatro secciones:

1. Instalación de CakePHP
2. Instalación de Ósmosis
3. Configuración de Ósmosis
4. Instalación y habilitación de Plugins

Instalación de CakePHP

Existen múltiples maneras de instalar CakePHP (?), el definir cuál es la más adecuada dependerá de las capacidades administrativas que se tengan sobre el servidor.

Instalación de Ósmosis

Una vez verificada la correcta instalación de CakePHP, es necesario sustituir el directorio *app* que se distribuye con Cake por Ósmosis.

```
rm -rf app
svn co http://tools.assembla.com/svn/osmosis/trunk/osmosis app
```

En caso de no poseer el cliente de subversion:

```
rm -rf app
tar -xzf osmosis.tar.gz
```

Con cualquiera de las dos opciones el directorio *app* deberá contener los archivos de Ósmosis. También se debe asegurar que la carpeta *tmp* dentro del directorio de Ósmosis tenga permisos de escritura por el usuario que corre el servidor. En linux esto podría ser como sigue:

```
chown -R www-data app/tmp
```

Configuración de Ósmosis

La configuración inicial de Ósmosis consiste en la modificación del **archivo APP/config/database.php.default**, la cual tiene instrucciones claras para la configuración.

Una vez modificado este archivo, debe guardarse como **database.php**

```
class DATABASE_CONFIG {
    var $default = array(
        'driver'      => 'mysql',
        // Alguno de los drivers soportados: mysql, postgres, oracle ...
        'persistent' => false,
        // Utilizar o no conexiones persistentes
        'host'        => 'localhost',
        // Host del servidor de bases de datos
        'login'       => 'user',
        // Nombre de usuario de la base de datos
        'password'    => 'password',
        // Clave del usuario de la base de datos
        'database'    => 'name',
```

```
// Nombre de la base de datos
'prefix'      => ''
// Prefijo de las tablas (opcional)
);
}
```

Una vez configurada la conexión a la base de datos, es necesario cargar las tablas. Para ello Ósmosis hace uso de los esquemas de base de datos de CakePHP.

```
alias cake=CAKE_DIR/console/cake
cake schema run create
```

Siguiendo las instrucciones disponibles en la consola se cargarán las tablas en la base de datos configurada.

El último paso del proceso de configuración de Ósmosis consiste en ejecutar la acción `init` del controlador `InitAcl`, la cual configura la base de datos de permisos y crea el primer usuario administrador. Para ejecutar dicha acción apunte el navegador a `www.example.com/initAcl/init`.

Para poder ejecutar dicha acción es necesario modificar temporalmente el archivo **APP/config/core.php** para deshabilitar la autenticación:

```
Configure::write('Auth.disabled', true);
```

No olvide reactivar la autenticación posteriormente:

```
Configure::write('Auth.disabled', false);
```

Por razones de seguridad se recomienda cambiar el valor del *sal* de seguridad en el archivo **APP/config/core.php** por cualquier cadena de caracteres suficientemente aleatoria:

```
Configure::write('Security.salt', '
$hl56MMsdb0qyJfIxfS2guVoUubWwvniR2G0FgaC9mi');
```

Esta cadena de caracteres es usada como semilla de encriptamiento de las claves de usuarios, y para evitar robo de sesiones, entre otras, por lo que si se deja igual a como es distribuido actualmente el paquete de Ósmosis se deja más vulnerable la aplicación.

Instalación y habilitación de plugins

Las herramientas que posee Ósmosis2 están construidas utilizando su arquitectura de plugins. La lista de plugins distribuidos oficialmente es la siguiente:

- Agenda
- Blog
- Chat
- Foro
- Locker
- Quiz
- Scorm
- Wiki

Si desea instalar otro plugin debe colocarlo en el directorio **APP/plugins** junto con los anteriores. Una vez colocados los plugins correctamente, estarán disponibles, para ser instalados, en el área administrativa de plugins (www.example.com/admin/plugins).

E.2. Desarrollo de Plugins

Estructura Inicial

CakePHP ofrece a los desarrolladores una herramienta de línea de comandos, la consola de Cake, que permite agilizar el proceso de desarrollo de aplicaciones. Una de las facilidades que ofrece es la generación de la estructura de archivos de un plugin de CakePHP.

```
cake bake plugin nombre_plugin
```

Esto genera un directorio “nombre_plugin” en el directorio plugins de Ósmosis.

Descripción del Plugin

Una vez creada la estructura básica del plugin, es recomendable crear el archivo **APP/plugin-s/nombre_plugin/config/description.php** en el cual se colocaran los siguientes descriptores del plugin.

```
<?php
Configure::write(
    'NombrePlugin.description',
    'Lorem ipsum dolor sit amet, consectetur adipiscing elit.'
);
Configure::write(
    'NombrePlugin.title',
    'Nombre del Plugin'
);
Configure::write(
    'NombrePlugin.type',
    array('tool')
);
Configure::write(
    'NombrePlugin.author',
    'Osmosis Team'
);
?>
```

Todos los valores son de texto libre, excepto *type* que debe ser un arreglo. El los valores aceptados *type* son 'tool' si desea que aparezca en la barra de herramientas, sino marcarlo como 'other'.

La existencia de este archivo es recomendable, en caso de no existir se tratará el plugin como 'other' y no se mostrará la descripción.

Estructura de la Base de Datos

Ósmosis no impone restricciones explícitas sobre la estructura de la base de datos de un plugin, sin embargo se recomienda:

- Seguir las convenciones de CakePHP.

- Prefijar los nombres de las tablas con el nombre del plugin, de este modo se facilita la identificación de las tablas.

Para la carga de las tablas de los plugins se recomienda usar los schemas de base de datos de CakePHP, ya que son agnósticos al manejador de base de datos y permiten automatizaciones interesantes.

La generación de los esquemas de base de datos se hacen con la consola de Cake, desde el directorio en el cual se quiere crear el schema.

```
cake schema generate
```

El schema creado debe ubicarse en el directorio */config/sql* del plugin.

E.3. Internacionalización de la plataforma

Los mensajes visibles por el usuario en Ósmosis están totalidad en inglés, y se recomienda seguir esta convención para los plugins. Sin embargo, CakePHP permite la internacionalización de sus aplicaciones por medio de archivos de traducción.

E.3.1. Funciones de Internacionalización

Dentro de cualquier sección de una aplicación Cake está disponible un juego de funciones para la internacionalización de las cadenas de caracteres. De dichas funciones, la más utilizada es `__()` la cual recibe dos parámetros:

1. Cadena de caracteres internacionalizable
2. Booleano que determina si la función debe imprimir o devolver la cadena traducida

Un ejemplo del uso de esta función:

```
<?php
    $del = __('Delete', true); //La variable $del contiene la palabra
                             traducida
    __('Delete'); // Se imprime la palabra traducida
?>
```

E.3.2. Extracción de cadenas internacionalizadas

CakePHP ofrece otra herramienta para ayudar a los desarrolladores a generar los archivos para traducir sus aplicaciones. Nuevamente, utilizado la consola de CakePHP.

```
cake i18n
```

Siguiendo las instrucciones se generará el archivo de traducciones. Cada plugin tiene un archivo de traducción, por lo cual, al generarlo, se debe indicar la ruta completa hasta el plugin que se desea traducir. Por ejemplo `/Users/user/Sites/osmosis/agenda`, en donde `agenda` es el plugin seleccionado. Así mismo, para llevar a cabo la traducción se debe contar con la misma estructura de archivos que posee la aplicación principal. El archivo de traducción, **default.pot**, se generará en la carpeta **locale** de cada plugin.

E.3.3. Funcionamiento de los archivos de traducción de cake

Los archivos para manejar la traducción del lenguaje se encuentran en el directorio **locale**. Cada lenguaje es un subdirectorio dentro de dicha carpeta, cuyo nombre consta de tres caracteres correspondiente al nombre del idioma al cual se hace referencia, siguiendo el estándar para los códigos de representación de nombres de idiomas (ISO 639-2)(?). Por ejemplo *eng* para el idioma inglés (english), *ita* para el idioma italiano (italian)

En el subdirectorio del idioma se debe crear un directorio de nombre **LC_MESSAGES** que contendrá el archivo “default.po”. Este archivo consta de cadenas de caracteres con claves.

Los archivos .po deben tener un encoding ISO-8859-1 y los strings debe ser menores a 1014 caracteres, a continuación se muestra un ejemplo de cómo se ven las cadenas de caracteres y las claves de las mismas en el archivo .po:

```
msgid "Hello World"
msgstr "Hola Mundo"
```

El archivo de traducción que se obtiene a partir de esta función toma el primer parámetro como el la clave a la cual se le asignará el mensaje traducido correspondiente.

E.3.4. Cómo traducir

Para generar la traducción completa de la plataforma, es necesario completar cada par `msgid - msgstr`, donde el `msgid` siempre queda fijo por ser la clave usada dentro de la plataforma, y `msgstr`

el mensaje correspondiente a la traducción que se desea construir. En algunos casos es posible encontrar dentro de la cadena msgid subcadenas de reemplazo, que son palabras que comienzan por %, usualmente siempre seguidas de una única letra s. Estas cadenas de reemplazo representan una o varias palabras, números, o cualquier otra cosa que haga sentido dentro de la construcción. Un ejemplo de cadena de reemplazo es el siguiente

```
msgid "Hello %s"  
msgstr "Hola %s"
```

Donde la cadena de reemplazo %s, puede representar el nombre de una persona.

Esto existe para la facilidad del programador, pues delega el trabajo de traducción a varios pares clave-valor de traducción, no obstante recomendamos hacer el menor uso posible de estas construcciones, puesto que aunque es posible que para el idioma en que se haya construido el msgid tenga sentido ese reemplazo, para otros idiomas con gramáticas distintas puede ser muy engorroso saber la estructura gramatical adecuada que se debe usar.

E.3.5. Cómo generar el archivo binario (.mo)

El archivo con extensión **.mo** se genera en el mismo directorio que contiene al archivo de extensión **.po**. Para generarlo se emplea el comando

```
'msgfmt -o target.mo source.po'
```

E.4. Más información

CakePHP tiene a disposición de los desarrolladores el manual del framework en la dirección <http://book.cakephp.org/>. En este manual se explica todo lo relacionado a CakePHP, su instalación, configuración y convenciones así como el desarrollo de aplicaciones bajo este framework.

E.5. Diccionario de datos

E.5.1. Tablas de ACL

Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL	
parent_id	int(11)	Yes	NULL	
model	varchar(255)	Yes		
foreign_key	int(11)	Yes	NULL	
alias	varchar(255)	Yes		
lft	int(11)	Yes	NULL	
right	int(11)	Yes	NULL	

Cuadro E.1: Estructura de la tabla acos. Los Access Control Objects son los objetos cuyo acceso debe restringirse (?)

Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL	
parent_id	int(11)	Yes	NULL	
model	varchar(255)	Yes		
foreign_key	int(11)	Yes	NULL	
alias	varchar(255)	Yes		
lft	int(11)	Yes	NULL	
right	int(11)	Yes	NULL	

Cuadro E.2: Estructura de la tabla aros. Los Access Request Objects son los entes que requieren acceso a los objetos restringidos (?).

Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL	
aro_id	int(11)	Yes	NULL	
aco_id	int(11)	Yes	NULL	

Field	Type	Null	Default	Comments
_create	int(11)	Yes	0	
_read	int(11)	Yes	0	
_update	int(11)	Yes	0	
_delete	int(11)	Yes	0	

Cuadro E.3: Estructura de la tabla aros_acos. Establece los permisos de un ARO sobre cada ACO.

E.5.2. Tablas del Núcleo de Ósmosis

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del miembro
institution_id	varchar(20)	Yes	NULL		Identificador del miembro dentro de la institución (canré)
full_name	varchar(50)	Yes	NULL		Nombre completo del miembro
email	varchar(50)	Yes	NULL		Dirección de e-mail del miembro
phone	varchar(20)	Yes	NULL		Número telefónico del miembro
country	varchar(20)	Yes	NULL		País de nacimiento
city	varchar(50)	Yes	NULL		Ciudad de nacimiento
age	int(2)	Yes	NULL		Edad
sex	varchar(1)	Yes	M		Sexo (M o F)
username	varchar(15)	Yes	NULL		Nombre de usuario (Login)

	Field	Type	Null	Default	Comments
	password	varchar(50)	Yes	NULL	Contraseña

Cuadro E.4: Estructura de la tabla members. Mantiene los datos de los miembros registrados en Ósmosis

	Field	Type	Null	Default	Comments
<i>id</i>	smallint(4)	Yes	NULL		Identificador del plugin
	title	varchar(50)	Yes	NULL	Título del plugin
	active	tinyint(1)	Yes	NULL	Activación global del plugin
	name	varchar(100)	Yes	NULL	Nombre del plugin
	description	varchar(255)	Yes	NULL	Descripción del plugin
	author	varchar(100)	Yes	NULL	Autor
	types	varchar(30)	Yes	tool	Tipo del plugin

Cuadro E.5: Estructura de la tabla plugins. Maneja los plugins habilitados para la plataforma.

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del curso
	department_id	int(4)	Yes	NULL	Identificador del departamento al cual esta asociado el curso
	owner_id	int(11)	Yes	NULL	Identificador del miembro creador del curso
	code	varchar(10)	Yes	NULL	Código del curso

	Field	Type	Null	Default	Comments
	name	varchar(100)	Yes	NULL	Nombre del curso
	description	text	Yes	NULL	Descripción del curso
	created	date	Yes	NULL	Fecha de creación del curso

Cuadro E.6: Estructura de la tabla courses. Cursos registrados en la plataforma

	Field	Type	Null	Default	Comments
id	char(36)	Yes	NULL		Identificador de la relación entre los cursos y los miembros
member_id	int(11)	Yes	NULL		Identificador del miembro
course_id	int(11)	Yes	NULL		Identificador del curso
role_id	int(11)	Yes	NULL		Identificador del rol del miembro en ese curso

Cuadro E.7: Estructura de la tabla courses_members. Mantiene la matrícula (Enrollments) de miembros en los cursos de la plataforma

	Field	Type	Null	Default	Comments
id	int(10)	Yes	NULL		Identificador de la relación entre los cursos y las herramientas
course_id	int(11)	Yes	NULL		Identificador del curso
plugin_id	smallint(4)	Yes	NULL		Identificador del plugin

Cuadro E.8: Estructura de la tabla course_tools. Maneja los plugins (tipo tool) activados por curso.

	Field	Type	Null	Default	Comments
<i>id</i>	int(4)	Yes	NULL		Identificador del departamento
name	varchar(150)	Yes	NULL		Nombre del departamento
description	text	Yes	NULL		Descripción del departamento

Cuadro E.9: Estructura de la tabla departments. Maneja los departamentos que agrupan los cursos.

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del Rol
parent_id	int(11)	Yes	NULL		Identificador del Rol padre
role	varchar(10)	Yes	NULL		Nombre del rol

Cuadro E.10: Estructura de la tabla roles. Mantiene una estructura arborescente de los roles del sistema

E.5.3. Tablas relacionadas al plugin Blog

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del blog
title	varchar(200)	Yes	NULL		Título del blog
description	text	Yes	NULL		Descripción del blog
member_id	varchar(100)	Yes	NULL		Identificador del miembro propietario del blog

Cuadro E.11: Estructura de la tabla blogs. Mantiene el listado de los blogs de los miembros.

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del comentario
comment	text	Yes	NULL		Identificador del comentario
post_id	int(11)	Yes	NULL		Identificador de la entrada al la cual está asociado el comentario
member_id	int(11)	Yes	NULL		Identificador del miembro que realizó el comentario

Cuadro E.12: Estructura de la tabla comments. Almacena los comentarios a las entradas

	Field	Type	Null	Default	Comments
<i>id</i>	int(10)	Yes	NULL		Identificador de la entrada
title	varchar(50)	Yes	NULL		Título de la entrada
body	text	Yes	NULL		Cuerpo o contenido de la entrada
created	datetime	Yes	NULL		Fecha de creación de la entrada
modified	datetime	Yes	NULL		Fecha de modificación de la entrada
blog_id	int(11)	Yes	NULL		Identificador del blog al cual pertenece la entrada
slug	text	Yes	NULL		Versión para URLs del título de la entrada
member_id	int(11)	Yes	NULL		Identificador del usuario que escribió la entrada

Field	Type	Null	Default	Comments
-------	------	------	---------	----------

Cuadro E.13: Estructura de la tabla posts. Almacena las entradas de todos los blogs

E.5.4. Tablas relacionadas al plugin Foro

	Field	Type	Null	Default	Comments
	<i>id</i>	char(36)	Yes	NULL	Identificador de la discusión
	topic_id	int(11)	Yes	NULL	Identificador del tema al cual está relacionada la discusión
	member_id	int(11)	Yes	NULL	Identificador del usuario que inició la discusión
	title	varchar(255)	Yes	NULL	Título de la discusión
	content	text	Yes	NULL	Mensaje
	sticky	tinyint(1)	Yes	NULL	Discusión fija (se mantiene en el tope de la lista)
	status	varchar(20)	Yes	unlocked	Estado de la discusión (bloqueada o no)
	response_count	int(11)	Yes	NULL	Número de respuestas a la discusión
	discussion_visit_count	int(11)	Yes	NULL	Número de visitas a la discusión
	created	datetime	Yes	NULL	Fecha de creación de la discusión
	modified	datetime	Yes	NULL	Fecha de modificación de la discusión

Field	Type	Null	Default	Comments
-------	------	------	---------	----------

Cuadro E.14: Estructura de la tabla discussions. Almacena las discusiones del foro

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la respuesta
	discussion_id	char(36)	Yes	NULL	Identificador de la discusión a la cual pertenece la respuesta
	member_id	int(11)	Yes	NULL	Identificador del usuario que escribió la respuesta
	content	text	Yes	NULL	Contenido de la respuesta
	created	datetime	Yes	NULL	Fecha de creación
	modified	datetime	Yes	NULL	Fecha de modificación

Cuadro E.15: Estructura de la tabla responses. Almacena las respuestas a las discusiones

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del tema
	name	varchar(120)	Yes	NULL	Nombre del tema
	description	varchar(255)	Yes	NULL	Descripción del tema
	forum_id	int(11)	Yes	NULL	Identificador del foro al cuál pertenece el tema

	Field	Type	Null	Default	Comments
	created	datetime	Yes	NULL	Fecha de creación del tema
	status	varchar(20)	Yes	unlocked	Estado del tema (bloqueada o no)

Cuadro E.16: Estructura de la tabla topics. Almacena los temas de los foros

E.5.5. Tablas relacionadas al plugin Casillero

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del documento
name	varchar(100)	Yes	NULL		Nombre del documento
description	text	Yes	NULL		Descripción del documento
file_name	varchar(150)	Yes	NULL		Nombre del archivo en el disco
type	varchar(50)	Yes	application/octet-stream		Tipo MIME del archivo
size	int(10)	Yes	NULL		Tamaño del archivo en bytes
member_id	int(11)	Yes	NULL		Dueño del documento
folder_id	char(36)	Yes	NULL		Carpeta contenedora del documento

Cuadro E.17: Estructura de la tabla documents. Almacena los datos de los documentos y su ubicación dentro del Casillero

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de las carpetas del Casillero
name	varchar(100)	Yes	NULL		Nombre de la carpeta
folder_name	varchar(150)	Yes	NULL		Nombre de la carpeta?
parent_id	char(36)	Yes	NULL		Identificador de la carpeta padre
member_id	int(11)	Yes	NULL		Dueño de la carpeta

Cuadro E.18: Estructura de la tabla folders. Almacena la información de las carpetas de los Casilleros de cada miembro

E.5.6. Tablas relacionadas al plugin Quiz

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la opción
choice_question_id	char(36)	Yes	NULL		Identificador de la pregunta de selección a la cual pertenece la opción
text	text	Yes	NULL		Contenido de la opción
position	tinyint(3)	Yes	0		Posición fija

Cuadro E.19: Estructura de la tabla choice_choices. Almacena las opciones de una pregunta de selección

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la pregunta de selección
body	text	Yes	NULL		Planteamiento de la pregunta
shuffle	tinyint(1)	Yes	NULL		Determina si deben mostrarse las opciones ordenadas de manera aleatoria
max_choices	int(11)	Yes	NULL		Número máximo de selecciones que puede hacerse como respuesta
min_choices	int(11)	Yes	NULL		Número mínimo de selecciones que debe hacerse como respuesta

Cuadro E.20: Estructura de la tabla choice_questions. Almacena las preguntas de selección

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		
choice_question_id	char(36)	Yes	NULL		Identificador de la pregunta de selección
quiz_id	char(36)	Yes	NULL		Identificador del quiz

Cuadro E.21: Estructura de la tabla choice_questions_quizzes. Almacena la asociación entre las preguntas y los quizes

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la opción

	Field	Type	Null	Default	Comments
	text	text	Yes	NULL	Contenido de la opción

Cuadro E.22: Estructura de la tabla `matching_choices`. Almacena las opciones para las preguntas de apareamiento

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		
matching_question_id	char(36)	Yes	NULL		Identificador de la pregunta de apareamiento
source	char(36)	Yes	NULL		Identificador de la opción fuente
target	char(36)	Yes	NULL		Identificador de la opción destino
position	tinyint(3)	Yes	0		Posición fija de la opción fuente

Cuadro E.23: Estructura de la tabla `matching_choices_matching_questions`. Almacena la relación entre un par de opciones (pareja correcta) dentro de una pregunta de apareamiento

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la pregunta de apareamiento
body	text	Yes	NULL		Planteamiento de la pregunta
shuffle	tinyint(1)	Yes	NULL		Determina si deben mostrarse las opciones ordenadas de manera aleatoria

	Field	Type	Null	Default	Comments
max_associations	int(11)	Yes	NULL		Número máximo de emparejamientos que se pueden hacer para una respuesta
min_associations	int(11)	Yes	NULL		Número mínimo de emparejamientos que deben realizarse para una respuesta

Cuadro E.24: Estructura de la tabla matching_questions

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		
matching_question_id	char(36)	Yes	NULL		Identificador de la pregunta de apareamiento
quiz_id	char(36)	Yes	NULL		Identificador del quiz

Cuadro E.25: Estructura de la tabla matching_questions_quizzes. Mantiene las preguntas de apareamiento por quiz

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la opción
ordering_question_id	char(36)	Yes	NULL		Identificador de la pregunta de ordenamiento
text	text	Yes	NULL		Contenido de la opción
position	tinyint(3)	Yes	0		Posición fija de la opción

Field	Type	Null	Default	Comments
-------	------	------	---------	----------

Cuadro E.26: Estructura de la tabla `ordering_choices`. Almacena las opciones para las preguntas de ordenamiento

	Field	Type	Null	Default	Comments
	<i>id</i>	char(36)	Yes	NULL	Identificador de la pregunta de ordenamiento
	body	text	Yes	NULL	Planteamiento de la pregunta
	shuffle	tinyint(1)	Yes	NULL	Determina si deben mostrarse las opciones ordenadas de manera aleatoria
	max_choices	int(11)	Yes	NULL	Número máximo de opciones que pueden ser parte de la respuesta
	min_choices	int(11)	Yes	NULL	Número mínimo de opciones que deben ser parte de la respuesta

Cuadro E.27: Estructura de la tabla `ordering_questions`. Almacena las preguntas de ordenamiento

	Field	Type	Null	Default	Comments
	<i>id</i>	char(36)	Yes	NULL	
	ordering_question_id	char(36)	Yes	NULL	Identificador de la pregunta de ordenamiento
	quiz_id	char(36)	Yes	NULL	Identificador del quiz

Cuadro E.28: Estructura de la tabla `ordering_questions_quizzes`. Mantiene la asociación entre quices y las pregunta de ordenamiento

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador del quiz
name	varchar(255)	Yes	NULL		Nombre o título del Quiz

Cuadro E.29: Estructura de la tabla quizzes. Almacena los quizes registrados

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de la pregunta
title	varchar(255)	Yes	NULL		Título de la pregunta
body	text	Yes	NULL		Planteamiento de la pregunta
format	varchar(5)	Yes	NULL		Formato de la respuesta

Cuadro E.30: Estructura de la tabla text_questions. Almacena las preguntas de desarrollo

	Field	Type	Null	Default	Comments
<i>id</i>	char(36)	Yes	NULL		Identificador de las preguntas de los quizzes
text_question_id	char(36)	Yes	NULL		Identificador de la pregunta de desarrollo
quiz_id	char(36)	Yes	NULL		Identificador del quiz

Cuadro E.31: Estructura de la tabla text_questions_quizzes. Mantiene la asociación entre quizes y las pregunta de desarrollo

E.5.7. Tablas relacionadas al plugin Scorm

	Field	Type	Null	Default	Comments
sco_id	int(11)	Yes	NULL		Identificador del SCO al cual está asociado attendee trackings
student_id	int(11)	Yes	NULL		Identificador del estudiante al cual está asociado attendee trackings
datamodel_element	varchar(255)	Yes	NULL		
value	varchar(255)	Yes	NULL		

Cuadro E.32: Estructura de la tabla scorm_attendee_trackings

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		
sco_id	int(11)	Yes	NULL		Identificador del SCO al cual está asociado choice considerations
preventActivation	varchar(5)	Yes	false		
constrainChoice	varchar(5)	Yes	false		

Cuadro E.33: Estructura de la tabla scorm_choice_considerations

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		
referencedObjective	varchar(255)	Yes	NULL		Corresponde al atributo referencedObjective del elemento ruleCondition de SCORM
measureThreshold	varchar(7)	Yes	NULL		Corresponde al atributo measureThreshold del elemento ruleCondition de SCORM

	Field	Type	Null	Default	Comments
operator	varchar(4)		Yes	noOp	Corresponde al atributo operador del elemento ruleCondition de SCORM
ruleCondition	varchar(27)		Yes	NULL	
rule_id	int(11)		Yes	NULL	Identificador de la regla a la cual está asociada condition

Cuadro E.34: Estructura de la tabla conditions

	Field	Type	Null	Default	Comments
<i>id</i>		int(11)	Yes	NULL	
sco_id		int(11)	Yes	NULL	Identificador del SCO al cual está asociado control modes
choiceExit		varchar(5)	Yes	true	Corresponde al atributo choiceExit del elemento controlMode de SCORM
choice		varchar(5)	Yes	true	Corresponde al atributo choice del elemento controlMode de SCORM
flow		varchar(5)	Yes	false	Corresponde al atributo flow del elemento controlMode de SCORM
forwardOnly		varchar(5)	Yes	false	Corresponde al atributo forwardOnly del elemento controlMode de SCORM

Field	Type	Null	Default	Comments
useCurrentAttemptObjectiveInfo	varchar(5)	Yes	true	Corresponde al atributo useCurrentAttemptObjectiveInfo del elemento controlMode de SCORM
useCurrentAttemptProgressInfo	varchar(5)	Yes	true	Corresponde al atributo useCurrentAttemptProgressInfo del elemento controlMode de SCORM

Cuadro E.35: Estructura de la tabla control_modes

Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL	Identificador de delivery control
sco_id	int(11)	Yes	NULL	Identificador del SCO al cual está asociado delivery controls
tracked	varchar(5)	Yes	true	Corresponde al atributo tracked del elemento deliveryControls de SCORM
completionSetByContent	varchar(5)	Yes	false	Corresponde al atributo completionSetByContent del elemento deliveryControls de SCORM
objectiveSetByContent	varchar(5)	Yes	false	Corresponde al atributo objectiveSetByContent del elemento deliveryControls de SCORM

Cuadro E.36: Estructura de la tabla delivery_controls

Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL	Identificador de map info

	Field	Type	Null	Default	Comments
objective_id		int(11)	Yes	NULL	Identificador del objective al cual está asociado map info
targetObjectiveID		varchar(255)	Yes	NULL	Corresponde al atributo targetObjectiveID del elemento mapInfo de SCORM
readSatisfiedStatus		varchar(5)	Yes	true	Corresponde al atributo readSatisfiedStatus del elemento mapInfo de SCORM
readNormalizedMeasure		varchar(5)	Yes	true	Corresponde al atributo readNormalizedMeasure del elemento mapInfo de SCORM
writeSatisfiedStatus		varchar(5)	Yes	false	Corresponde al atributo writeSatisfiedStatus del elemento mapInfo de SCORM
writeNormalizedMeasure		varchar(5)	Yes	false	Corresponde al atributo writeNormalizedMeasure del elemento mapInfo de SCORM

Cuadro E.37: Estructura de la tabla map_infos

	Field	Type	Null	Default	Comments
<i>id</i>		int(11)	Yes	NULL	Identificador de objectives
sco_id		int(11)	Yes	NULL	Identificador del SCO al cual está asociado objectives
satisfiedByMeasure		varchar(5)	Yes	false	Corresponde al atributo satisfiedByMeasure del elemento objective de SCORM

	Field	Type	Null	Default	Comments
	minNormalizedMeasure	varchar(3)	Yes	1.0	Corresponde al elemento minNormalizedMeasure del elemento objective de SCORM
	objectiveID	varchar(255)	Yes	NULL	Corresponde al atributo objectiveID del elemento objective de SCORM
	primary	tinyint(1)	Yes	0	

Cuadro E.38: Estructura de la tabla objectives

	Field	Type	Null	Default	Comments
	<i>id</i>	int(11)	Yes	NULL	Identificador de randomization
	sco_id	int(11)	Yes	NULL	Identificador del SCO al cual está asociado el randomization
	randomizationTiming	varchar(16)	Yes	never	Corresponde al atributo randomizationTiming del elemento randomizationControls de SCORM
	selectCount	int(11)	Yes	NULL	Corresponde al atributo randomizationTiming del elemento randomizationControls de SCORM
	reorderChildren	varchar(5)	Yes	false	Corresponde al atributo reorderChildren del elemento randomizationControls de SCORM
	selectionTiming	varchar(16)	Yes	never	Corresponde al atributo selectionTiming del elemento randomizationControls de SCORM

Cuadro E.39: Estructura de la tabla randomizations

<i>id</i>	Field	Type	Null	Default	Comments
		int(11)	Yes	NULL	Identificador del rollup
sco_id		int(11)	Yes	NULL	Identificador del SCO al cual está asociado rollup
rollupObjectiveSatisfied		varchar(5)	Yes	true	Corresponde al atributo rollupObjectiveSatisfied del elemento rollupRules de SCORM
rollupProgressCompletion		varchar(5)	Yes	true	Corresponde al atributo rollupProgressCompletion del elemento rollupRules de SCORM
objectiveMeasureWeight		varchar(20)	Yes	1.0000	Corresponde al atributo objectiveMeasureWeight del elemento rollupRules de SCORM

Cuadro E.40: Estructura de la tabla rollups

<i>id</i>	Field	Type	Null	Default	Comments
		int(11)	Yes	NULL	
sco_id		int(11)	Yes	NULL	Identificador del SCO al cual está asociado el rollup consideration
requiredForSatisfied		varchar(15)	Yes	always	Corresponde al atributo requiredForSatisfied del elemento rollupConsiderations de SCORM
requiredForNotSatisfied		varchar(15)	Yes	always	Corresponde al atributo requiredForNotSatisfied del elemento rollupConsiderations de SCORM

	Field	Type	Null	Default	Comments
	requiredForComplete	varchar(15)	Yes	always	Corresponde al atributo required-ForComplete del elemento rollup-Considerations de SCORM
	requiredForIncomplete	varchar(15)	Yes	always	Corresponde al atributo required-ForIncomplete del elemento rollupConsiderations de SCORM
	measureSatisfactionIfActive	varchar(5)	Yes	true	Corresponde al atributo measure-SatisfactionIfActive del elemento rollupConsiderations de SCORM

Cuadro E.41: Estructura de la tabla rollup_considerations

	Field	Type	Null	Default	Comments
	<i>id</i>	int(11)	Yes	NULL	
	sco_id	int(11)	Yes	NULL	Identificador del SCO al cual está asociado rules
	type	varchar(4)	Yes	NULL	
	conditionCombination	varchar(3)	Yes	all	Corresponde al atributo condition-Combination del elemento rule-Conditions de SCORM
	action	varchar(20)	Yes	NULL	
	minimumPercent	varchar(6)	Yes	0.0000	Corresponde al atributo minimum-Percent del elemento rollupRule de SCORM
	minimumCount	varchar(5)	Yes	0	Corresponde al atributo minimum-Count del elemento rollupRule de SCORM
	rollup_id	int(11)	Yes	NULL	

Cuadro E.42: Estructura de la tabla rules

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del SCORM
course_id	int(11)	Yes	NULL		Identificador del curso en el cual se encuentra el SCORM
name	varchar(255)	Yes	NULL		Nombre del recurso SCORM
file_name	varchar(255)	Yes	NULL		Nombre del archivo que contiene el paquete SCORM
description	text	Yes	NULL		Descripción del recurso SCORM
version	varchar(9)	Yes	NULL		Version del SCORM
created	datetime	Yes	NULL		Fecha de creación del SCORM
modified	datetime	Yes	NULL		Fecha de modificación del SCORM
path	text	Yes	NULL		

Cuadro E.43: Estructura de la tabla scorms

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador del SCO
scorm_id	int(11)	Yes	NULL		Identificador del paquete SCORM al cual pertenece el SCO
parent_id	int(11)	Yes	NULL		Identificador del padre del SCO
manifest	varchar(255)	Yes	NULL		
organization	varchar(255)	Yes	NULL		
identifier	varchar(255)	Yes	NULL		Corresponde al atributo identifier del elemento item de SCORM
href	varchar(255)	Yes	NULL		
title	varchar(255)	Yes	NULL		Título del SCO
completionThreshold	varchar(3)	Yes	NULL		Corresponde al elemento completionThreshold de SCORM
parameters	text	Yes	NULL		Corresponde al atributo parameters del elemento item de SCORM

Field	Type	Null	Default	Comments
isvisible	varchar(5)	Yes	true	Corresponde al atributo isvisible del elemento item de SCORM
attemptAbsoluteDurationLimit	varchar(6)	Yes	NULL	Corresponde al atributo attemptAbsoluteDurationLimit del elemento limitConditions de SCORM
dataFromLMS	text	Yes	NULL	Corresponde al elemento dataFromLMS del elemento item de SCORM
attemptLimit	varchar(10)	Yes	NULL	Número máximo de intentos para la actividad
scormType	varchar(6)	Yes	NULL	Tipo del recurso SCORM

Cuadro E.44: Estructura de la tabla scos

Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL	Identificador de la presentación de la actividad
hideKey	varchar(10)	Yes	NULL	
sco_id	int(11)	Yes	NULL	Identificador del SCO al cuál está relacionada la presentación de la actividad

Cuadro E.45: Estructura de la tabla sco_presentations

E.5.8. Tablas relacionadas al plugin Wiki

	Field	Type	Null	Default	Comments
<i>id</i>	int(11)	Yes	NULL		Identificador de la página del wiki
wiki_id	int(11)	Yes	NULL		Identificador del wiki al cual está asociada la página
member_id	int(11)	Yes	NULL		Creador de la página
title	varchar(255)	Yes	NULL		Título de la página
content	text	Yes	NULL		Contenido de la página
revision	int(6)	Yes	1		Número de revisión de la página
created	datetime	Yes	NULL		Fecha de creación de la página
updated	datetime	Yes	NULL		Fecha de actualización de la página
slug	text	Yes	NULL		Versión para URLs del título de la página

Cuadro E.46: Estructura de la tabla entries. Almacena las entradas (páginas) del wiki

	Field	Type	Null	Default	Comments
entry_id	int(11)	Yes	NULL		Identificador de la página
member_id	int(11)	Yes	NULL		Identificador del miembro que realizó el cambio
title	varchar(255)	Yes	NULL		Título de la página
content	text	Yes	NULL		Contenido de la revisión

	Field	Type	Null	Default	Comments
	revision	int(6)	Yes	NULL	Número de la revision
	created	datetime	Yes	NULL	Fecha de creación de la revision

Cuadro E.47: Estructura de la tabla revisions. Almacena las revisiones a las páginas del wiki

	Field	Type	Null	Default	Comments
<i>id</i>	int(10)	Yes	NULL		Identificador del wiki
course_id	int(11)	Yes	NULL		Identificador del curso al cual está asociado el wiki
name	varchar(255)	Yes	NULL		Nombre del wiki
description	text	Yes	NULL		Descripción del wiki

Cuadro E.48: Estructura de la tabla wikis. Listado de wikis registrados