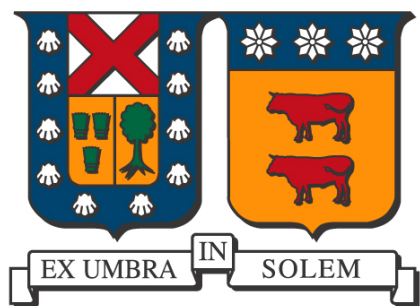


UNIVERSIDAD TECNICA FEDERICO SANTA MARIA
DEPARTAMENTO DE INFORMATICA
VALPARAISO - CHILE



OPEN EDX: EVALUANDO UN SISTEMA MOOC PARA EL
DEPARTAMENTO DE INFORMATICA USM

PABLO INZUNZA FUENTES

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL
INFORMATICO

PROFESOR GUIA	:	DR. HORST VON BRAND
PROFESOR CORREFERENTE	:	MS. LUIS HEVIA

MARZO - 2017

Resumen

Este trabajo consiste en determinar la viabilidad de *Open edX*, software destinado a servir cursos en línea, como base para implementar el formato de cursos *MOOC* en el Departamento de Informática de la UTFSM. El objetivo principal busca caracterizar *Open edX* en tres contextos diferentes: efectividad como un sistema de gestión de aprendizaje, gestión de cuentas de usuarios y compatibilidad con métodos de rendimiento. Los resultados son determinados mediante uso de *rúbricas de evaluación* y se usan para comparar *Open edX* con software similar, en particular el sistema actual utilizado en el Departamento de Informática llamado *Moodle*. Al integrar todos los contextos se determina que *Open edX* es una solución viable, sin embargo no es capaz de implementar en forma ideal todas las funcionalidades requeridas por el Departamento de Informática, indicando que la principal razón es el poco tiempo de desarrollo que la plataforma tiene. Se concluye que, en el caso de que el *DI USM* desee implementar cursos bajo el formato *MOOC*, *Open edX* puede ser adaptado correctamente para cumplir esta función pero se requiere modificar las funciones problemáticas mediante programación, o bien esperar una nueva versión de *Open edX* que incluya las mejoras pertinentes.

Palabras clave: *Sistemas MOOC, rúbricas evaluación, e-learning.*

Abstract

The following work consist in determine *Open edX* viability as software for host *Massive Online Open Courses* for the UTFSM (departamento informatica). The main objective is to characterize *Open edX* in three different contexts: Effectiveness as a learning management system, Management of user accounts and compatibility with performance technics. Results are determinated by evaluation rubrics and used to compare *Open edX* with similar software and, in particular, with *DI USM* current system called *Moodle*. After combining results for the three contexts *Open edX* appears as a viable but not complete solution lacking in some functionatities required by *DI USM* because of the short development time the project has. It is concluded that, if *DI USM* want to implement *MOOC* courses, *Open edX* must be modified, by programming, to match required functions or wait for a new release of *Open edX* software that include the necessary improvements.

Glosario

- **Cluster.** Conjunto de computadores interconectados que realizan tareas en conjunto. Los equipos de un *Cluster* se conectan mediante redes de alta velocidad y se pueden considerar, en varios contextos, como una sola unidad de alto rendimiento.
- **Feedback.** También referido como retroalimentación, ocurre cuando la salida de un sistema es dirigida como entrada formando un circuito. En el contexto de este trabajo, se refiere a captar las reacciones y opiniones de usuarios respecto al funcionamiento de cierto *software* con el fin de evaluar su desempeño y/o modificarlo.
- **LMS** Sigla en inglés para *Learning Management System*. *Software* del tipo *LMS* corresponde a sistemas con funcionalidades orientadas a la administración y entrega de cursos educacionales o programas de entrenamiento. Dentro de las capacidades esperadas para este tipo de sistema destacan la gestión de material para clases, administración de tareas, seguimiento del progreso de estudiantes, entre otras.
- **Metadatos** Corresponden a datos que proveen información sobre otros datos. Se usa comúnmente para describir archivos multimedia como fotos y videos, mensajes e información utilizada en telecomunicaciones, páginas web, entre otras cosas.
- **MOOC** Sigla en inglés para *Massive Online Open Courses*. Corresponde a un formato para realizar cursos educacionales a distancia donde no se aplican más restricciones para matricularse que una cuenta de usuario gratuita y un correo electrónico. Este tipo de cursos destaca por la gran cantidad de estudiantes que pueden inscribirse alrededor del mundo mediante el uso de Internet.
- **Open Source** Término que hace referencia a *software* que cumple con todas las normas definidas en la especificación *Open Source Definition* publicada en el sitio oficial de la institución *Open Source Initiative*.
- **Parche** Corresponde a *software* diseñado para modificar un programa computacional ya existente o sus datos relacionados, con el objetivo de arreglar fallos o mejorar su rendimiento.

- **Plugin** Corresponde a un componente de *software* diseñado para agregar una funcionalidad particular a un programa computacional ya existente.
- **Release** Término utilizado para referirse a la distribución de un programa computacional, incluyendo documentación y material de apoyo, al público. Esta distribución se puede publicar de diversas maneras, por ejemplo mediante discos compactos o a través de Internet.
- **Roadmap** Corresponde a una planificación para alcanzar ciertas metas a corto y largo plazo. En el contexto de *software* las metas pueden ser nuevas funcionalidades o la resolución de problemas presentes en el programa.
- **Sandbox** En desarrollo de *software*, el término se refiere a un ambiente de prueba que permite aislar experimentos y códigos no probados del sistema o repositorio de producción.
- **SSO** Sigla en inglés para *Single Sign On*. Corresponde a una propiedad de control de acceso para sistemas formados por varios programas computacionales independientes. Esta propiedad permite a usuarios el acceso a todos los programas independientes que conforman el sistema utilizando una única identificación y contraseña evitando la necesidad de crear múltiples cuentas para cada programa.
- **Software** Corresponde a la parte de un sistema computacional compuesta por datos en forma de instrucciones computacionales, en contraste con *hardware* que corresponde a las piezas con las cuales el sistema computacional es construido. En informática se llama *software* computacional a toda la información procesada por sistemas computacionales. Ejemplo de *software* computacional son programas computacionales, bibliotecas y datos no ejecutables como archivos de documentación y medios digitales.
- **Wiki** Corresponde a un sitio web cuyo contenido y estructura se puede modificar en forma colaborativa a través de un navegador web.

Índice general

Índice de cuadros

Índice de figuras

1	Introducción	1
2	Definición del Problema	3
2.1.	Problema	5
2.2.	Alternativas de Solución	5
2.3.	Objetivos	8
3	Estado del Arte	10
3.1.	Introducción	10
3.2.	Sistemas LMS	13
3.3.	Clasificación Becta	15
3.4.	Sistemas MOOC	19
4	Moodle como LMS	24
4.1.	Características	25
4.2.	Moodle en el DI USM	28
4.3.	Rúbrica para un Software	33
5	Open edX	39
5.1.	Arquitectura	41
5.2.	Desarrollo Actual	44

6 Evaluación	47
6.1. Funciones de LMS	47
6.2. Sistemas de Cuentas y Autenticación	58
6.3. Rendimiento y Escalabilidad	59
6.4. Viabilidad y Administración	61
7 Conclusiones y Recomendaciones	64
7.1. Objetivos	64
7.2. Recomendaciones y Trabajo Futuro	67
Bibliografía	71
A Moodle Alto Rendimiento	73
A.1. Moodle Cluster	73
A.2. Sistemas Cloud	74
B Estándares digitales	76
B.1. Open Source	76
B.2. LTI	78
B.3. SCORM	79
B.4. Mozilla Open Badges	81
B.5. OLX	81
C Plataformas Alternativas	83
C.1. Sakai	84
C.2. ATutor	85
C.3. Dokeos	86
D Herramientas Gestión Código	87
D.1. Github	87
D.2. JIRA	88
D.3. Confluence	88
D.4. Django Admin	89
D.5. Tar	89

Índice de cuadros

2.1. Distintos formatos de implementación y despliegue para instancias de <i>Moodle</i> con muchos usuarios.	4
2.2. Resumen de alternativas solución.	6
2.3. <i>OpenMOOC</i> , información oficial.	7
2.4. <i>P2PU</i> , información oficial.	7
2.5. <i>Open edX</i> , información oficial.	8
3.1. Preferencias de instituciones sobre los <i>LMS</i>	14
3.2. Resumen proveedores de cursos <i>MOOC</i> populares a la fecha.	22
4.1. Perfil sistema <i>Moodle</i>	24
4.2. Rúbrica tipo <i>RECS</i> para evaluar concordancia con los ramos elegidos representativos. Los criterios son evaluados como <i>cumple (C)</i> y <i>no cumple (NC)</i>	36
4.3. Ejemplo de definición de intervalos exigente.	37
4.4. Ejemplo de definición de intervalos menos exigente.	37
5.1. Perfil sistema <i>Open edX</i>	40
6.1. Rúbrica de evaluación para la categoría <i>Gestión de contenido</i> de la clasificación <i>Becta</i>	49
6.2. Rúbrica de evaluación para la categoría <i>Planificación y clasificación curricular</i> de la clasificación <i>Becta</i>	50
6.3. Rúbrica de evaluación para la categoría <i>Compromiso y administración de estudiantes</i> de la clasificación <i>Becta</i>	51
6.4. Rúbrica de evaluación para la categoría <i>Herramientas y servicios de comunicación y colaboración</i> de la clasificación <i>Becta</i>	52
6.5. Resumen de resultados para cada categoría y evaluación final.	52
6.6. Evaluación utilizando clasificación <i>DI USM</i> de requerimientos funcionales.	54
6.7. Resumen ponderaciones para las categorías <i>Becta</i> según requerimientos relevantes para el <i>DI USM</i>	55
6.8. Resumen resultados de rúbricas para la plataforma <i>open edX</i>	56
6.9. Resumen de puntaje obtenido por alternativas a <i>Open edX</i>	57

6.10. Resumen de promedios de puntaje obtenidos por alternativas a <i>Open edX</i>	57
6.11. Principales métodos de autenticación disponibles para <i>open edX</i>	59
A.1. Tópicos importantes para implementar una instancia <i>Moodle cluster</i>	74
B.1. Recursos especificaciones <i>SCORM</i>	81
C.1. Perfil sistema <i>Sakai LMS</i>	84
C.2. Perfil sistema <i>ATutor LMS</i>	85
C.3. Perfil sistema <i>Dokeos LMS</i>	86

Índice de figuras

3.1. Línea temporal desarrollo proveedores <i>MOOC</i>	12
4.1. <i>Moodle landing page</i>	29
4.2. Búsqueda de ramos <i>Aula USM</i>	32
5.1. Arquitectura de componentes de <i>open edX Eucalyptus.2</i>	41
5.2. Sistemas componentes para la persistencia de datos en <i>Open edX</i> . .	43
5.3. Gráfico de contribuciones hasta 2017.	45
B.1. Arquitectura de una conexión especificada por <i>LTI 2.0</i>	79

Capítulo 1

Introducción

Este trabajo se desarrolla en el contexto de los cursos en formato *MOOC*, sigla en inglés para *Massive Online Open Courses*, utilizados en el área de la *educación a distancia*. En particular se estudian sistemas de *software* utilizados para ofrecer estos cursos a la comunidad global a través de *Internet*. El avance de la tecnología, principalmente en el área de las telecomunicaciones, sirve como puente entre las personas y una gran cantidad de servicios donde destacan el entretenimiento, utilidades de productividad y educación, entre muchos otros servicios presentes en la actualidad, que afectan considerablemente el estilo de vida y las oportunidades de cada persona. Uno de los factores determinantes en el futuro de una persona es el acceso a educación y material de estudio, que actualmente es facilitado por los sitios y bases de datos disponibles en *Internet*, así como aplicaciones pedagógicas con fines tan variados como enseñar determinados idiomas o comunicar conocimiento matemático. Sin embargo el conocimiento disponible en *Internet* no siempre es confiable debido a que las personas o instituciones que ofrecen el material usualmente no enfrentan restricciones para asegurar calidad en sus publicaciones, por lo que se ha generado demanda por contenidos educacionales publicados por instituciones reconocidas. Es así como aparecen en escena sistemas que sin imponer restricciones ofrecen versiones a distancia de cursos dictados en varias de las universidades más prestigiosas del mundo, como *Stanford*, *MIT* o *Harvard*, dando lugar a cursos que deben adaptar el contenido presentado para un público más general y a sistemas que puedan manejar la considerable demanda asociada a la no imposición de restricciones para matricularse.

La situación descrita es trabajada por sistemas que se agrupan en dos corrientes con enfoques distintos. La primera corriente persigue la adaptación de sistemas creados antes del escenario descrito, mientras que la otra corriente se enfoca en crear sistemas nuevos que aprovechen al máximo las cualidades modernas que ofrece la ingeniería de software y herramientas actuales. Es así como surgen sistemas nuevos como *Open edX* y otros sistemas ya existentes, como *Moodle* o *ATutor*, son actualizados con nuevas interfaces compatibles con los protocolos de comunicación especificados en los últimos años, como *JSON*

o *REST*. Algunos sistemas antiguos destacan por sus diseños visionarios que han facilitado la integración con las tecnologías actuales, por ejemplo *Moodle* integra un motor de extensiones que permite agregar funciones y soporte adicional. No obstante el principal obstáculo que enfrentan las instituciones para ofrecer los cursos en formato *MOOC* es el alto costo inicial, que comprende el desarrollo o adquisición del sistema, el *hardware* necesario para mantener la disponibilidad ante alta demanda de usuarios y los honorarios de las personas encargadas de crear y mantener los cursos, *software* asociado y equipos.

Los sistemas *open source* o de código abierto surgen como una alternativa atractiva para las instituciones ya que permiten reducir los altos costos iniciales. Estos sistemas tienen condiciones en su licencia que permiten su distribución sin la necesidad de pagar, y en algunos casos son compatibles con actividades y negocios que tengan fines de lucro. En el caso de una institución educacional que imparta carreras en el área de informática estos sistemas tienen como valor agregado que sirven a estudiantes y practicantes que, al tener acceso a los códigos fuente de los sistemas, pueden experimentar en forma directa la relación entre el código de una función programada y su uso por parte de usuarios.

Este trabajo fija su atención en una de las más recientes plataformas para servir cursos *MOOC*, llamada *Open edX*. Este sistema es desarrollado en conjunto por dos universidades prestigiosas de Estados Unidos, *Harvard* y *MIT*, las cuales decidieron liberar el código de esta plataforma y publicarlo bajo el nombre de *Open edX*. El objetivo de este trabajo consiste en describir y contrastar esta plataforma con otras alternativas *open source* disponibles en *Internet*, principalmente *Moodle* que corresponde a la actual plataforma de educación a distancia utilizada por el Departamento de Informática de la Universidad Santa María, institución en donde se realiza este trabajo. El capítulo 2 trata sobre los desafíos que debe enfrentar una institución educacional, específicamente el *DI USM*, para implementar cursos en formato *MOOC* y las características principales que debe tener una solución al problema planteado por estos desafíos, lo que se describe en la definición formal del problema a estudiar y los objetivos perseguidos por este trabajo. El capítulo 3 consiste en la exposición de conceptos relacionados con *e-learning* que conforman el contexto base para entender el formato de cursos *MOOC*. Adicionalmente se tratan sistemas de *software* creados para dicho contexto, como los sistemas de gestión del aprendizaje o *LMS* y los sistemas *MOOC*, cuyas funcionalidades en la gestión de cursos son agrupadas en categorías específicas por la institución británica *Becta*. El capítulo 4 describe la plataforma *Moodle* y define un tipo de rúbrica para evaluar la calidad con la que un sistema cumple con requerimientos funcionales. El capítulo 5 describe la plataforma *Open edX*, destacando su arquitectura por componentes y la distribución de sus sistemas de persistencia de datos. Finalmente, los capítulos 6 y 7 tratan la evaluación de ambas plataformas junto a otros sistemas alternativos respecto de los objetivos definidos en el capítulo 2, indicando los resultados y conclusiones derivadas. Destaca la formulación de algunos temas relacionados que pueden servir de base para definir temas a tratar por memorias de titulación futuras.

Capítulo 2

Definición del Problema

El problema a tratar por este trabajo se centra en el modelo de educación a distancia llamado *Massive Online Open Courses*, traducido como *Cursos de Acceso libre Masivos en Línea*. Este modelo, llamado de ahora en adelante *MOOC* por sus siglas en inglés, permite a cualquier persona acceder a cursos dictados por una institución educacional mediante una plataforma web sin la restricción de estar matriculado en algún curso o carrera formal de la institución en cuestión.

Existen varias plataformas nuevas que implementan este modelo y otras anteriores que pueden adaptarse para servir este tipo de cursos. En el caso del Departamento de Informática de la Universidad Técnica Federico Santa María, que de ahora en adelante se le referirá como *DI USM* o simplemente *DI*, se utiliza una plataforma para educación a distancia llamada *Moodle* donde los alumnos matriculados en alguna carrera asociada pueden acceder utilizando la cuenta institucional creada para ellos. Esta cuenta es la misma para acceder a la mayoría de los servicios que el *DI* presta a todos sus alumnos matriculados. Entre estos se encuentra el servicio de correo electrónico, el nombre de usuario para ingresar al laboratorio de computación y cualquier otro servicio implementado por las personas del laboratorio para ayudar a realizar las actividades académicas, como cuentas de bases de datos o sistemas de control de versiones. La plataforma *Moodle* implementada en el *DI USM* es una buena alternativa para ofrecer cursos en formato *MOOC* debido a que cuenta con una amplia gama de funcionalidades para crear y gestionar cursos y material educativo. Adicionalmente cuenta con un diseño de software por capas, completamente modular, con varias alternativas de despliegue para gran cantidad de usuarios. El cuadro 2.1 indica distintos formatos de *Moodle* diseñados para manejar gran cantidad de usuarios y alta concurrencia de conexiones. Cada uno de los formatos mencionados en el cuadro se explican en detalle en el anexo A.

Cuadro 2.1: Distintos formatos de implementación y despliegue para instancias de *Moodle* con muchos usuarios.

Implementaciones de <i>Moodle</i> para alto rendimiento
<i>Moodle</i> Cluster
<i>Moodle</i> Cloud
<i>Moodle</i> Amazon Web Services

Fuente: Elaboración propia. Los datos y referencias pertinentes se detallan en el anexo A.

El *DI USM* cuenta con una cantidad de usuarios bastante grande que incluye las cuentas de los profesores y docentes, personal administrativo y auxiliares, alumnos de cada una de las promociones considerando al menos nueve años hacia atrás. La principal diferencia en el uso de *Moodle* que el *DI USM* presenta en relación al formato *MOOC* corresponde a la cantidad de alumnos matriculados en cada curso que es determinado en conjunto por profesores y directivos. Por ejemplo, Un ramo típico dictado por el *DI USM* usualmente tiene menos de 100 alumnos inscritos por curso y en el caso de tener más alumnos se crean varios cursos coordinados. En un proveedor de cursos *MOOC* los alumnos son manejados en un mismo curso sin separarlos por lo que no hay temas de coordinación entre cursos paralelos. Considerando los datos publicados por la *UTFSM* en sus informes de gestión en [18], el total de alumnos matriculados de pregrado en la universidad no supera los 20,000, alcanzando los 6,094 estudiantes en su casa central. Tomando este valor como cota superior para los estudiantes que el *DI USM* tendría que servir con el sistema *Moodle* y considerando que el número de profesores y alumnos de postgrado es menor que los matriculados en pregrado, el uso que el *DI USM* le da a su plataforma *Moodle* no es masivo si se compara con un proveedor de cursos *MOOC* famoso como *Coursera* el cual, según el recuento realizado por la página *class central* sobre proveedores *MOOC* durante el 2015 tenía al 50 % del total de alumnos inscritos en cursos *MOOC* que eran alrededor de 35 millones de estudiantes. El reporte se encuentra en [3]. Adicional a ser el sistema actual en el *DI USM*, *Moodle* es un ejemplo de plataforma no diseñada en un principio para cursos en formato *MOOC* que posee alto potencial para adaptarse a ese propósito. La razón tras este enunciado consiste en el diseño modular que posee la plataforma y que, al igual como sucede con la otra plataforma a evaluar *Open edX*, es altamente compatible con técnicas de escalabilidad basadas en replicación de instancias de *software*. *Moodle* junto con otros tres sistemas potenciales, descritos en el anexo C, son evaluados posteriormente en conjunto a *Open edX*. La pregunta sobre si *Moodle* y otros sistemas potenciales han evolucionado lo suficiente para servir cursos *MOOC* resulta interesante, lo que motiva a incluir estos sistemas en una evaluación comparativa posterior. Se regresará a esta pregunta durante la exposición de las conclusiones en el capítulo 7.

2.1. Problema

Si el *DI USM* busca implementar cursos bajo el formato *MOOC* hay que considerar la situación expuesta anteriormente. Los cursos *MOOC* no cuentan con restricción en el número de alumnos matriculados por curso y, adicionalmente, estos alumnos no necesitan estar matriculados en la universidad o en alguna carrera formal del *DI USM* por lo que se debe realizar una distinción en el rol que los estudiantes de los cursos *MOOC* tienen respecto de los estudiantes oficiales de la institución. Además del posiblemente mayor número de matrículas de los cursos *MOOC* respecto a los cursos tradicionales, es importante tener en cuenta que el *DI USM* maneja datos sobre el límite de alumnos que ingresan todos los años junto con el total de profesores, administrativos y otros funcionarios por lo que poseen una buena estimación del total de usuarios que *Moodle* debe manejar. En el caso de los cursos *MOOC* este total de usuarios es mucho más difícil de estimar ya que depende del interés de los inscritos y el compromiso con el que tomen los ramos dado que es posible que los cursos experimenten deserción académica en el caso que los estudiantes de los cursos *MOOC* cambien sus prioridades. Esta situación se puede resumir en la siguiente definición de problema:

Problema El *DI USM* debe resolver dos desafíos importantes si es que busca implementar cursos en el formato *MOOC*, los cuales son:

- Manejar una cantidad variable y potencialmente grande de estudiantes inscritos en el sistema.
- Utilizar un mecanismo para separar los roles de las cuentas de estudiantes formales de aquellos inscritos en los cursos *MOOC*

Como se indicó anteriormente, *Moodle* posee lo necesario para resolver estos desafíos de varias formas, sin embargo resulta interesante buscar otras alternativas que puedan cumplir la misma función y resolver el problema. En particular buscar software que esté diseñado pensando en servir cursos *MOOC* como idea fundamental.

2.2. Alternativas de Solución

Cumplir con las condiciones impuestas por el problema descrito se puede realizar de varias formas. En primer lugar, se puede modificar la instancia actual de *Moodle* instalada en el *DI USM* para que considere el rol distinto de los nuevos estudiantes de los cursos *MOOC* y asignar más recursos computacionales para que se puedan manejar las posibles ráfagas variables de usuarios inscritos en los nuevos cursos abiertos. *Moodle* por omisión no impone un límite de alumnos matriculados por curso y soporta varios métodos para matricular, destacando en el contexto de los cursos *MOOC* el uso de archivos con valores separados por coma (o *CSV*) que son fáciles de generar mediante *software*.

Otra alternativa consiste en crear una nueva instancia de *Moodle* dedicada exclusivamente a servir los cursos *MOOC* con el fin de simplificar el trabajo de configuración. Se puede, por ejemplo, copiar la instancia actual y enlazarla a una base de datos nueva para guardar la información de los profesores y estudiantes inscritos en los nuevos cursos en forma independiente. Por último, la alternativa final es implementar software diferente a *Moodle* para manejar los cursos *MOOC*. El cuadro 2.2 resume las alternativas de solución para el problema planteado.

Cuadro 2.2: Resumen de alternativas solución.

SI	Modificar y asignar nuevos recursos computacionales a <i>Moodle</i> .
SII	Crear una nueva instancia de <i>Moodle</i> dedicada a los <i>MOOC</i> .
SIII	Implementar una plataforma alternativa independiente

Fuente: Elaboración propia.

Si bien el curso de acción lógico es seguir con la plataforma *Moodle* como se indica en las alternativas de solución **SI** y **SII**, existe una situación en particular al respecto. El desarrollo de *Moodle* comenzó antes de que los cursos *MOOC* pasaran a tener la popularidad actual por lo que el desarrollo de cursos de este tipo no era el objetivo principal por el cuál la plataforma fue desarrollada. En particular *Moodle* se define como un sistema de gestión del aprendizaje o *learning management system* (*LMS* por sus siglas en inglés). Por esta situación resulta interesante buscar software actual cuya concepción y desarrollo sea dirigida en su totalidad a servir cursos en el formato *MOOC*. Sin embargo se espera que un sistema de estas características sea capaz de, al menos, incorporar herramientas comparables a las que ofrece *Moodle* u otro sistema *LMS* para gestionar cursos incluyendo usuarios inscritos, creación y gestión de contenidos educacionales, evaluación y trabajo colaborativo, entre otras cosas.

Entre las plataformas de este tipo, es decir enfocadas en servir cursos *MOOC*, se encuentran tres alternativas *open source*: *OpenMOOC*, *P2PU* y *Open edX*.

OpenMOOC Es una plataforma web para servir cursos en formato *MOOC*. El cuadro 2.3 muestra información de referencia sobre el sistema y su sitio web oficial, entre otros. El sistema en sí cuenta con un repositorio online en *GitHub*¹ y ha sido mencionado en numerosas conferencias, sin embargo el repositorio no ha visto actividad desde el año 2015 por lo que es válido asumir que el desarrollo de la plataforma está en un receso. Por último, la comunidad de desarrolladores parece ser pequeña lo que se ve reflejado en el sitio del repositorio donde se indica que el proyecto ha tenido menos de 20 contribuyentes. Estos hechos

¹*GitHub* es un proveedor de repositorios especializados en código fuente y proyectos de programación. Su página oficial es github.com (revisada al 29 de marzo de 2017).

hacen que la plataforma *OpenMOOC* sea menos atractiva para considerarla en este trabajo dado su estado actual (a la fecha del 29 de marzo de 2017).

Cuadro 2.3: *OpenMOOC*, información oficial.

<i>OpenMOOC</i>
Página oficial: openmooc.org
Tipo de plataforma: Sistema web, <i>Open Source</i>
Repositorio de distribución: github.com/OpenMOOC

Fuente: Recopilado del sitio oficial.

P2PU Esta plataforma web tiene un objetivo distinto a las otras alternativas ya que consiste en un sitio de coordinación, para que profesores y alumnos se junten a clases fuera del campus en lo que su comunidad define como *círculo de aprendizaje*. El equipo de desarrollo posee varios proyectos laterales alrededor de esta plataforma y tienen todo el código publicado en *GitHub*, sin embargo no cuentan con un sistema de gestión de aprendizaje para cursos dictados totalmente a distancia ya que requiere que los usuarios se reúnan en un lugar físico para realizar las actividades del curso (no necesariamente en una institución educativa). El cuadro 2.4 indica la información oficial del proyecto. A diferencia del proyecto anterior, la comunidad de *P2PU* es muy activa, lo que se refleja en la gran participación y aportes registrados en su página de *GitHub*, sin embargo la falta de un sistema unificado de funciones hace que el utilizar este sistema no sea tan atractivo además de que el objetivo de este proyecto no es ofrecer cursos a distancia sino reunir gente y practicar enseñanza presencial.

Cuadro 2.4: *P2PU*, información oficial.

<i>P2PU</i>
Página oficial: www.p2pu.org/en
Tipo de plataforma: Sistema web, <i>Open Source</i>
Repositorio de distribución: github.com/p2pu

Fuente: Recopilado del sitio oficial.

Open edX Este sistema para servir cursos *MOOC* es desarrollado por equipos de la Universidad de Harvard y el MIT. Actualmente cuenta con una comunidad de instituciones de educación superior y empresas alrededor del mundo². El cuadro 2.5 muestra la información oficial en forma resumida de la plataforma. *Open edX* es el objeto de este estudio por las siguientes razones:

² Referido como *XConsortium*, los miembros actuales se pueden encontrar en el sitio oficial www.edx.org/schools-partners (revisado al 29 de marzo de 2017)

- Como sistema se desarrolló pensando en servir cursos *MOOC* como idea principal, a diferencia de sistemas como *Moodle* que pueden ser adaptados para la tarea pero que en un inicio se pensaban sólo como sistemas de gestión de aprendizaje.
- El proyecto se encuentra activo y cuenta con una comunidad internacional de apoyo incluyendo varias universidades prestigiosas alrededor del mundo, a diferencia de otros proyectos como *OpenMOOC* que se encuentran sin desarrollo aparente o estancados, lo que influye en la calidad del código y la documentación disponible para resolver posibles problemas de la plataforma.

Cuadro 2.5: *Open edX*, información oficial.

<i>Open edX</i>
Página oficial: open.edx.org
Tipo de plataforma: Sistema web, <i>Open Source</i>
Repositorio de distribución: github.com/edx/edx-platform

Fuente: Recopilado del sitio oficial.

Desarrollo en Moodle la plataforma *Moodle* lleva más tiempo de desarrollo que los sistemas anteriormente señalados, además cuenta con una comunidad de desarrolladores mucho más activa lo que se ve reflejado en la alta actividad que presenta el repositorio oficial de *Moodle* en *GitHub* ubicado en github.com/moodle/moodle donde, a la fecha del 29 de marzo de 2017, se registran más de 83000 propuestas de modificaciones creadas por más de 300 contribuyentes. De las tres alternativas mencionadas, *Open edX* y *P2PU* están cerca del nivel de *Moodle* en términos de actividad en sus grupos de desarrollo.

2.3. Objetivos

Considerando lo expuesto anteriormente, incluyendo las razones por las cuales evaluar la plataforma *Open edX*, el objetivo principal de este trabajo es determinar si dicha plataforma corresponde a una solución viable al problema planteado en la sección 2.1, es decir si corresponde a una solución del tipo **SIII**. Para ello es necesario determinar las condiciones necesarias que debe cumplir la plataforma. En primer lugar hay que determinar si *Open edX* puede cumplir con las funciones necesarias para gestionar cursos y material educativo como el sistema actual del *DI USM*, *Moodle*. En segundo lugar se debe determinar si *Open edX* permite la gestión de cuentas de usuario necesaria para implementar los cursos *MOOC*, en particular la matrícula a cursos sin un límite máximo y la gestión de cantidades altas de usuarios concurrentes. Por último es necesario determinar las capacidades de *Open edX* para asegurar la disponibilidad del

servicio ante ráfagas de usuarios, por lo que es necesario determinar si la plataforma es compatible con métodos de mejora de rendimiento y escalabilidad.

Los objetivos a perseguir en relación a la plataforma *open edX* se describen a continuación:

- **Objetivo general.** Evaluar si cumple con los requisitos de una solución **SIII**.
- **Específico I.** Determinar si cumple con las funciones necesarias para servir como sistema de gestión del aprendizaje.
- **Específico II.** Determinar si posee un sistema de gestión de usuarios capaz de manejar roles y permitir matriculación en cursos sin un límite máximo, así como varios mecanismos de autenticación.
- **Específico III.** Comprobar si admite métodos para mantener y/o mejorar el rendimiento en caso de tener ráfagas de demanda.

El cumplir con los objetivos servirá para tener una idea clara sobre la viabilidad de la plataforma *Open edX* como solución viable para implementar el modelo de cursos *MOOC* en el *DI USM*, sin embargo es necesario realizar una evaluación económica independiente acorde a la realidad de la institución para tomar una decisión final al respecto de si utilizar dicha plataforma.

Capítulo 3

Estado del Arte

El creciente aumento en la conectividad y acceso a Internet alrededor del mundo ha traído consigo muchas oportunidades de desarrollo para las instituciones educativas. En lo que sigue se describirán en forma general conceptos relacionados al modelo de educación a distancia y en particular al formato de cursos *MOOC*.

3.1. Introducción

En la actualidad las instituciones educativas usan el *e-learning* como motor para facilitar el acceso a la educación y potenciar sus imágenes en el plano internacional como instituciones modernas. Sin embargo el término en sí posee varias definiciones, influenciadas por los momentos históricos en que se plantearon. Oficialmente se considera que el término *e-learning* fue acuñado por Jay Cross en 1998. Cross describe el contexto en el cual planteó el término *e-learning* en su trabajo recopilatorio [9], donde se entrega un resumen histórico del concepto datando su origen en 1997 bajo el nombre de *online learning*, concepto que fue mencionado por Elliot Masie en el contexto de usar la tecnología de redes para mejorar el proceso de aprendizaje. Una de las primeras definiciones y usos del concepto es planteada por Clark Aldrich en [2], donde se define *e-learning* como sigue:

Definición 1 (e-learning, Aldrich). *Amplia combinación de procesos, contenido e infraestructura que usa computadores y redes para escalar y/o mejorar una o más partes significativas de la cadena del valor del aprendizaje, incluyendo gestión y distribución.*

Al definir el concepto de *e-learning* se toma en consideración la tecnología utilizada para implementarlo. Por ejemplo Jay Cross, al describir la historia del e-learning en [9], indica que antes de 1980, las corporaciones utilizaban equipos de instructores para educar a sus trabajadores pero todo esto cambió a mediados de los años 80 con la aparición de los discos compactos (*CD*) educativos. Actualmente se relaciona el *e-learning* al uso de la tecnología de redes e

internet para realizar cursos. Como estos formatos de aprendizaje usualmente mantienen separado al profesor de los alumnos, es necesario contar con un sistema que posea características que mitiguen los problemas de no realizar una comunicación presencial, por ejemplo:

- Formas de gestionar conocimiento.
- Foros y mecanismos de discusión.
- Sistemas de evaluación.
- Mecanismos de *feedback* para profesores, estudiantes y administradores.

Actualmente, un sistema que implementa las funcionalidades mínimas para que el proceso de enseñanza a distancia sea efectivo es llamado *sistema de gestión de aprendizaje* o, en sus siglas en inglés, *LMS*¹. En sus inicios también fueron conocidos como *sistemas de gestión de cursos* o, en sus siglas en inglés, *CMS*². En [10], John Meerts durante el 2003 realizó una guía para la organización *EDUCAUSE* sobre criterios a considerar por instituciones que busquen elegir sistemas *LMS* para uso particular, definiendo estos sistemas como:

Definición 2 (LMS, Meerts). *Software que provee al instructor un conjunto de herramientas y un marco de trabajo para la creación de contenido online, y permite la posterior gestión y enseñanza de estos cursos mediante varias interacciones distintas con los estudiantes inscritos.*

En la actualidad existen varios sistemas de este tipo clasificados en tres grupos:

- Sistemas *open source*.
- Sistemas propietarios.
- Sistemas basados en tecnología *CLOUD*³

Para una lista detallada sobre los sistemas *LMS* más importantes en la actualidad, consultar el trabajo recopilatorio realizado por Don McIntosh en [4]. Uno de los avances actuales más famosos en *e-learning* surge con la aparición del modelo para cursos masivos de libre acceso llamado *massive online open courses* o, desde ahora por sus siglas en inglés, *MOOC*. Este modelo está pensado para permitir a cualquier persona el acceso a cursos dictados por instituciones de educación alrededor del mundo. Sistemas que implementan este modelo fueron creados recientemente, datando su origen en el año 2011 gracias a los esfuerzos de dos profesores pertenecientes a la Universidad de Stanford llamados Andrew Ng y Daphne Koller para dar acceso online al curso de *Machine Learning*

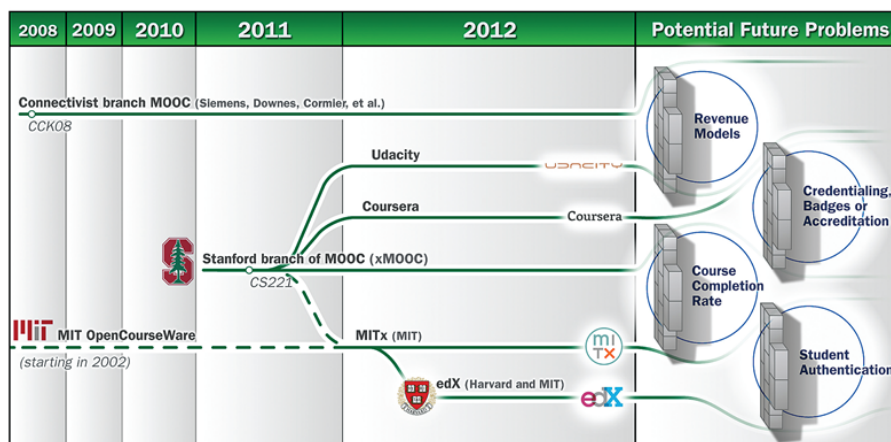
¹*Learning Management System.*

²*Course Management System.*

³ Servicios computacionales donde los recursos son provistos bajo demanda a través de Internet.

CS221 dictado en la Universidad de Stanford. La historia y desarrollo de éste y otros sistemas *MOOC* es captada en el trabajo recopilatorio desarrollado por Michael Gaebel para la Asociación Europea de Universidades en [11] La figura 3.1 muestra gráficamente en forma de línea temporal el desarrollo de las plataformas *MOOC*. En esta figura se exponen dos clases de sistemas *MOOC*, los sistemas conectivistas (*cMOOC*) y los sistemas tradicionales (*xMOOC*).

Figura 3.1: Línea temporal desarrollo proveedores *MOOC*.



Fuente: trabajo publicado por Phil Hill para la institución *EDUCAUSE* en [14]

El enfoque conectivista data de mucho antes que los proveedores actualmente conocidos como *Coursera*, *Udacity* o *edX* fueran creados, sin embargo las ideas tras los sistemas conectivistas son muy distintas a las propuestas por los sistemas tradicionales.

Definición 3 (*cMOOC*, Gaebel). *Sistema que ofrece cursos a distancia con un enfoque que privilegia la generación de conocimiento por medio de actividades basadas en el aprendizaje social, creatividad y autonomía.*

Los sistemas *cMOOC* están basados en la *filosofía conectivista* y el trabajo realizado por Ivan Illich quién en 1970 propuso la creación de *Redes de Aprendizaje* con la nueva tecnología que, en su tiempo, él esperaba que surgiera durante los años futuros. La filosofía conectivista en educación realiza una metáfora entre el proceso de aprendizaje y una red con nodos y conexiones en la cuál el conocimiento es distribuido. Bajo esta premisa, la filosofía conectivista realiza la analogía entre aprender y navegar esta red. En un curso tradicional, los nodos de esta red los forman los actores: profesores, estudiantes y ayudantes. Usualmente, visto desde el contexto descrito, el proceso de enseñanza normal de estudiantes escuchando al profesor hablar ignora la comunicación entre los

misimos estudiantes dedicando poco o nada de tiempo a este propósito, lo que se traduce en una navegación parcial de la red. En términos de la filosofía conectivista se está realizando un aprendizaje parcial que se podría mejorar permitiendo a los estudiantes discutir sobre el conocimiento visto en clases. Tecnología actual como foros de discusión, mensajería y videoconferencias son clave para un sistema *cMOOC* ya que permiten realizar discusiones del conocimiento entre los actores sin restricciones físicas como la reunión presencial de todos en un lugar. Más información sobre el conectivismo se puede encontrar en el material publicado por uno de sus mayores exponentes Stephen Downes. El artículo digital publicado por él en [17] es un buen punto de partida.

Definición 4 (xMOOC, Gaebel). *Sistemas que ofrecen cursos a distancia enfocados a la educación tradicional, es decir privilegiando el aprendizaje por medio de exposición (en formato de videos y contenido audiovisual) y pruebas. El énfasis de sus actividades es la duplicación del conocimiento.*

3.2. Sistemas LMS

Los sistemas de gestión de aprendizaje son una especialización de los más generales sistemas de gestión de contenido. Este tipo de sistemas se ha utilizado ampliamente a lo largo de los años para automatizar el almacenamiento y acceso a información de distintos tipos en variados campos de aplicación. Los campos de aplicación más importantes de estos sistemas son las instituciones académicas, que se dedican en su totalidad a la formación de personas, y las empresas, que necesitan capacitar periódicamente a sus trabajadores para permanecer competitivas en el tiempo. El autor Ryann K. Ellis entrega en [16] una visión de los *LMS* centrada en el contexto de *e-learning* para empresas que se describe a continuación. Los sistemas de gestión del aprendizaje o *LMS* se pueden describir de múltiples maneras siendo la más simple como software que automatiza la administración, seguimiento y reportes de un evento formativo (o curso en el contexto académico), sin embargo dicha descripción no es suficiente para abarcar todo lo que comprende un sistema de este tipo, por eso se exige que un *LMS* robusto sea capaz de cumplir con al menos las siguientes características:

- Automatizar en forma centralizada la administración de los eventos.
- Provee de servicios bien guiados o auto-explicativos.
- Reúne y entrega contenido en forma rápida.
- Permite la consolidación del servicio entregado mediante una plataforma web escalable.
- Contiene soporte para estándares de contenido y es portable.
- Permite la personalización y reutilización del contenido.

Desde el punto de vista de los usuarios, Ellis rescata las opiniones de personas del contexto empresarial en una encuesta realizada el año 2009 por el equipo de la publicación *Learning Circuits* de la asociación para el desarrollo del talento *ASTD* en la cual se destacan las preferencias de las instituciones sobre las cualidades que buscan en un sistema de gestión del aprendizaje. La información se resumen en el cuadro 3.1.

Cuadro 3.1: Preferencias de instituciones sobre los *LMS*.

Características más populares / solicitadas en un <i>LMS</i> :
★ Generación de reportes. ★ Seguimiento de resultados, rendimiento de estudiantes. ★ Gestión de tareas, trabajos y pruebas. ★ Gestor de material y contenido. ★ Catálogo de cursos disponibles.
Guías en la implementación de un <i>LMS</i> en la organización:
★ Gestión centralizada de las actividades formativas. ★ Medición del uso real de los cursos. ★ Seguimiento en el cumplimiento de normas y regulaciones. ★ Gestión logística para cursos con instructor. ★ Medir niveles de satisfacción relativo a los cursos.
Desafíos principales al implementar un <i>LMS</i> :
★ Personalización de la plataforma. ★ Generación e integración de contenido. ★ Integración de empleados y usuarios al uso de la plataforma. ★ Mantenimiento y administración de la plataforma. ★ Integración con sistemas legado (previamente existentes). ★ Rendimiento de la plataforma.

Fuente: Encuesta realizada por *Learning Circuits Team* y publicada en [16].

En el caso de las instituciones académicas, las prioridades en torno a los sistemas de gestión de aprendizaje están dadas por los proyectos educativos de cada institución, sin embargo existen características comunes en el funcionamiento de estos sistemas para que su uso sea efectivo. Según McIntosh en [4], las instituciones educativas se diferencian de las organizaciones privadas en la disponibilidad de tiempo de los instructores. A diferencia de una universidad o instituto, que cuenta con una plana docente permanente o de planta, las empresas no pueden contar con la presencia de los instructores todo el tiempo. Por esta razón, los sistemas orientados a instituciones educativas son diseñados bajo el supuesto que el profesor está disponible para generar contenidos y material pedagógico en conjunto con el alumno, por lo que el enfoque del sistema *LMS* se centra en la creación de espacio de discusión y trabajo colaborativo

entre los estudiantes y profesores adicional a las horas de clase.

3.3. Clasificación Becta

Para una descripción más objetiva sobre los requerimientos que una plataforma de gestión de aprendizaje debe cumplir en el ámbito académico, se seguirá el trabajo publicado por la agencia británica *Becta* durante el 2008 [1], antes de su disolución el 2011 debido a recortes de presupuesto por el gobierno británico. En este trabajo se clasifican los requerimientos funcionales que una plataforma tipo *LMS* debe cumplir en cuatro categorías descritas a continuación.

CI Gestión de contenido.

CII Planificación y clasificación curricular.

CIII Compromiso y administración de estudiantes.

CIV Herramientas y servicios de comunicación y colaboración.

Gestión de contenido Categoría referida a la capacidad de la plataforma para interactuar con recursos digitales. Los sistemas *LMS* deben abrir, guardar y ejecutar los recursos digitales necesarios para facilitar a los profesores y estudiantes las labores pedagógicas. Estos recursos suelen ser de diversa naturaleza y pueden variar desde pruebas de selección múltiple a través de la página web, hasta tareas de varias entregas con material audiovisual variado. Adicionalmente, la plataforma debe ser capaz de integrar recursos y material digital ya existente. Esta categoría comprende los siguientes requerimientos específicos:

- **GCI:** *Localizar, cargar y mostrar recursos digitales.* La plataforma permite a los usuarios navegar y buscar recursos mostrando los archivos elegidos. También permite que los usuarios carguen recursos, como tareas o archivos multimedia en un área accesible por medio de la interfaz de la plataforma.
- **GCI:** *Abrir e interactuar con recursos digitales.* La plataforma permite a los usuarios interactuar con los recursos digitales antes mencionados por medio de herramientas incluidas dentro de la plataforma, como por ejemplo un visor de documentos *PDF*.
- **GCI:** *Marcar recursos.* Los usuarios pueden clasificar recursos digitales utilizando marcas (*tags*) relacionadas a los *metadatos* de los recursos.
- **GCI:** *Mostrar e importar metadatos.* La plataforma permite cargar y guardar registros de *metadatos* además de mostrar información derivada de estos a los usuarios. En el caso de que existan políticas de restricción para esta información, la plataforma otorga acceso restringido conforme a las políticas establecidas.

- **GCV:** *Creación de recursos.* La plataforma incluye herramientas para crear y guardar contenido, incluyendo recursos colaborativos como discusiones en foros o encuestas. Adicionalmente, estos recursos son accesibles por medio de la interfaz de la plataforma y se pueden exportar utilizando un procedimiento estandarizado.

Planificación y clasificación curricular Categoría referida a la habilidad del sistema *LMS* para relacionar los contenidos educativos a sus respectivas mallas curriculares y cursos. Es usual que se clasifiquen los recursos utilizando etiquetas en forma de *metadatos*, por lo que el sistema *LMS* debe permitir utilizar estas etiquetas para realizar operaciones como búsquedas contextuales sobre los contenidos que se encuentran dentro de la plataforma. Los requerimientos en esta categoría apoyan aspectos relacionados a la personalización del contenido educativo con el fin de entregar material adecuado o modificar la interfaz de la plataforma. De esta manera los estudiantes pueden crear y compartir material en forma más simple. La categoría comprende los siguientes requerimientos específicos:

- **PCCI:** *Accesibilidad.* El sistema maneja métodos de accesibilidad que permitan a personas con discapacidades o necesidades educativas especiales para ingresar a la interfaz, herramientas y otros servicios que el sistema provee.
- **PCCII:** *Tareas y evaluaciones.* La plataforma permite que los estudiantes puedan realizar tareas y diagnósticos que apoyen el aprendizaje. Es necesario que la plataforma entregue al menos un tipo de tarea, como preguntas de selección múltiple, entrega de ensayos o pruebas de preguntas y respuestas, además debe permitir a los profesores solicitar reportes respecto a los resultados de estas pruebas. Adicionalmente, la plataforma permite autoevaluación y evaluación entre pares.
- **PCCIII:** *Interfaz personalizable.* La interfaz de usuario permite su personalización con el objetivo de adaptarla a las necesidades de accesibilidad de los distintos usuarios. Entre otras cosas, el usuario puede cambiar la fuente y los colores mostrados en la pantalla.
- **PCCIV:** *Planificación de clases.* Es posible para los profesores crear y gestionar planes para las clases, donde la plataforma entrega las facilidades para crear, guardar y editar dichos planes.
- **PCCV:** *Navegación y búsqueda.* Los usuarios pueden navegar a través de los recursos en la plataforma utilizando información sobre los cursos, los cuales son presentados a los usuarios de acuerdo a estructuras curriculares previamente establecidas.
- **PCCVI:** *Personalización.* La plataforma permite designar recursos a los estudiantes de acuerdo a sus necesidades específicas y los profesores pueden acceder a recursos en base a sus perfiles de usuario.

- **PCCVII:** *Secuencias de recursos o actividades* Usuarios pueden establecer secuencias de recursos o actividades controladas por contenido, profesores o alumnos, acorde a un conjunto de reglas.

Compromiso y administración de estudiantes Las plataformas educativas apoyan el proceso educativo por medio del concepto conocido como *anywhere anytime access* que consiste en el acceso a recursos educativos sin las restricciones impuestas por los horarios de los profesores y las ubicaciones físicas de las salas de clase. Esta categoría cubre los requerimientos involucrados en el acceso remoto a la plataforma educativa, es decir acceso fuera del campus universitario, y los requerimientos para la gestión de la información personal de los estudiantes. Entre la información del estudiante dentro del alcance de estos requerimientos se encuentran registros de actividad e ingreso a cursos, así como interacciones con otros sistemas de información que puedan colaborar con la plataforma, como por ejemplo servicios de autenticación externos.

- **CAEI:** *Acceso remoto.* Los usuarios pueden ingresar a la plataforma fuera de las dependencias de la institución y en horarios distintos a los horarios de atención establecidos por la misma.
- **CAEII:** *Autenticación.* Todos los usuarios deben tener un indentificador único llamado *username*⁴ el cual la plataforma puede verificar. Con el fin de facilitar la integración de la plataforma con otros sistemas existentes, el sistema *LMS* debe integrar capacidades *single sign-on* que permitan al usuario identificarse una vez y no tener que repetir la autenticación cuando quiera usar el servicio de otra plataforma durante la misma sesión de trabajo. Se espera que los proveedores del servicio tomen las medidas necesarias para evitar accesos no autorizados a la plataforma.
- **CAEIII:** *Información consistente sobre estudiantes.* La información relativa a los estudiantes debe ser consistente dentro de la plataforma, por lo que los cambios en dicha información se deben sincronizar en cada componente distinto del sistema. Se espera mínima replicación de datos con el fin de facilitar la mantención de la consistencia en los datos por medio de procesos automatizados.
- **CAEIV:** *Protección de datos.* La información y datos almacenados en la plataforma deben estar asegurados y se deben manejar de acuerdo a la legislación vigente.
- **CAEV:** *Grupos y roles.* La plataforma permite la definición de distintos grupos y roles a los cuales los usuarios pueden ser asignados. estos grupos y roles definen cómo los usuarios interactúan con las distintas interfaces del sistema mediante perfiles y permisos. Ejemplos de roles y perfiles son: *administrador, profesor, ayudante, etc.*

⁴Nombre de usuario en inglés.

- **CAEVI:** *Acceso a la información.* El acceso a la información depende de los permisos asignados a los roles del usuario que consulta. Esta dependencia puede ser extendida a la gestión de información perteneciente a sistemas externos integrados a la plataforma.
- **CAEVII:** *Exportación e importación de información sobre estudiantes.* Es importante que la plataforma permita exportar e importar los datos e información relativa a los estudiantes con el fin de facilitar la transferencia de los registros de alumnos entre instituciones colaboradoras.
- **CAEVIII:** *Portafolios.* La plataforma permite a los usuarios la creación de colecciones de contenido o *portafolios*, con el fin de compartir contenido y apoyar el desarrollo personal de los usuarios. Es necesario que los portafolios contengan información relativa al usuario propietario y deben ser portables. La plataforma debe proveer acceso y almacenamiento para los portafolios.
- **CAEIX:** *Planificación.* La plataforma debe permitir la definición de reglas para controlar el acceso a los recursos. El acceso a recursos puede estar sujeto a restricciones de tiempo, por ejemplo en una tarea con varias entregas se puede limitar el acceso a los archivos para que se permita acceso solo al material asociado a la entrega actual que se está trabajando. De esta forma es necesario terminar una entrega para acceder al material de la siguiente.
- **CAEX:** *Seguimiento.* El sistema debe entregar facilidades para seguir el progreso y rendimiento de los estudiantes. Dentro de las facilidades entregadas se deben incluir reportes cuando los estudiantes accedan a unidades de contenido o tareas particulares.
- **CAEXI:** *Datos sobre uso de la plataforma.* Se debe presentar datos sobre el uso de los recursos en la plataforma, tanto datos individuales como grupales. Los reportes deben resumir cómo y cuándo los recursos son utilizados.

Herramientas y servicios de comunicación y colaboración Los sistemas *LMS* permiten tanto a estudiantes, profesores y padres (en el caso de educación secundaria y preparatoria) colaborar y comunicarse en formas más efectivas y focalizadas respecto a la misma situación sin el apoyo de una de estas plataformas. Las herramientas y servicios prestados por la plataforma para este fin incluyen foros, espacios de trabajo colaborativos, sistemas de mensajería, entre otros.

- **HSCI:** *Foros de discusión.* Usuarios pueden tomar parte en foros de discusión leyendo y publicando mensajes. La plataforma debe permitir que las discusiones en los foros se puedan moderar y que el acceso a dichas discusiones sea controlado, por ejemplo para grupos de estudiantes o periodos académicos particulares.

- **HSCII:** *Gestión de derechos de autor.* La plataforma debe permitir agregar derechos de autor a los contenidos creados utilizando el sistema, por ejemplo etiquetando los contenidos creados por los usuarios. Las etiquetas pueden incluir licencias tipo *Creative Commons*⁵, por ejemplo.
- **HSCIII:** *Servicios web.* La plataforma debe ser capaz de interactuar con servicios web utilizando protocolos estándar. Este soporte para servicios web debe ser independiente de la naturaleza del servicio mismo y debe centrarse en cumplir con los protocolos requeridos para la conexión.

3.4. Sistemas MOOC

MOOC corresponde a una categoría especial de curso o entrenamiento con cualidades específicas. En [11] Michael Gaebel responde a la pregunta sobre qué es un curso *MOOC* indicando que debe cumplir las siguientes características:

- El curso debe dictarse bajo la modalidad *online* a través de internet.
- No debe tener requerimientos formales para matricularse.
- No debe existir límite de participación en el curso.
- El curso debe ser libre de costos para los estudiantes.
- No se otorgan créditos universitarios por completar el curso.

Los sistemas *MOOC* son plataformas web diseñadas para la realización de este tipo de cursos. Crear una plataforma de este estilo involucra varios desafíos a la institución involucrada debido a que el sistema debe disponer de amplios recursos computacionales para permitir la conexión simultánea de, posiblemente, cientos o miles de usuarios. Dado que los sistemas de este tipo deben prestar los servicios necesarios para que los usuarios puedan realizar las funciones de un estudiante dentro de la plataforma, es común pensar en estos sistemas como un *LMS* con soporte adicional para manejar muchas conexiones, sin embargo las características propias de los cursos de este tipo requieren de características específicas que faciliten a los usuarios la operación dentro del sistema.

Atracción de participantes Como los cursos en la plataforma no deben presentar restricciones de matrícula, muchos de los estudiantes serán externos a la Universidad que ofrece los cursos. Esto implica que no necesariamente los cursos se van a llenar de estudiantes como es el caso de las instituciones de educación que van creando cursos paralelos de acuerdo a la cantidad de estudiantes que conocen a priori. Dado que ofrecer un curso utiliza los recursos

⁵Tipo de licencia para derechos de autor creada por la organización *Creative Commons Corporation*. Más información en el sitio web oficial <https://creativecommons.org> (revisado al 29 de marzo de 2017).

de la plataforma, no es eficiente que el curso tenga muy pocos estudiantes, por lo que es necesario utilizar algún medio de publicidad que usualmente es parte del sitio web de ingreso de la plataforma misma mediante el uso de una *landing page*⁶ apropiada.

Recomendación de cursos Es posible que no todos los futuros usuarios tengan un curso en específico en mente al momento de ingresar al sitio de la plataforma. Personas pueden llegar por diversos motivos como recomendaciones de amigos y familiares, o simplemente por visitar sitios web al azar, sin embargo es necesario que la plataforma cuente con un catálogo que muestre a los visitantes las alternativas disponibles en cuanto a cursos ofrecidos. El catálogo es necesario para promover los cursos y despertar el interés de los usuarios indecisos.

Cursos y planificación Los cursos publicados en la plataforma no tienen que seguir necesariamente los semestres académicos. Un curso puede partir y terminar varias veces en un periodo académico si es suficientemente breve, o bien puede durar más de un semestre en el caso de cursos complejos. De esta forma es necesario que el sistema publique información sobre el inicio y fin de los cursos. Idealmente, cada curso nuevo debería contar con una página de perfil que contenga la descripción del curso, su planificación y posible material introductorio, así como contenidos recomendados a saber.

Comunidad Importante es permitir a los usuarios ver opiniones de otros usuarios sobre los cursos ofrecidos en la plataforma. Incluir soporte para foros de discusión, valoración y encuestas permite a los usuarios más información a la hora de elegir inscribirse en un curso de la plataforma, además de generar una noción de comunidad activa que suele mejorar la motivación de los usuarios por participar.

Existen varios desafíos al momento de implementar un sistema de este tipo. En particular se tienen desafíos técnico-económicos, como el costo monetario asociado a montar y mantener la plataforma, y desafíos pedagógicos asociados a los estudiantes en sí. En términos técnico-económicos hay cuatro puntos clave que requieren consideración:

- Tipo de plataforma y costos asociados a adquirirla.
- Personal para montar y mantener la plataforma.
- Hardware necesario para que la plataforma permanezca operativa.
- Personal para crear cursos y dar soporte a usuarios.

⁶ En marketing digital se le llama *landing page* a la primera página mostrada al seguir el enlace del sitio web.

Estos puntos son determinantes al momento de calcular los costos económicos involucrados y las decisiones que se tomen en cada uno de ellos van a repercutir en la viabilidad de este tipo de proyectos en el mediano y largo plazo. Para ilustrar estos puntos es útil tomar un caso real. Los profesores Dr. Paul Nissenson y Dra. Angela Shih publicaron en [13] su experiencia montando una plataforma *MOOC* para la Universidad Politécnica del Estado de California. En este trabajo se indica que la principal barrera para instalar este tipo de sistemas es la gran inversión inicial que, según los autores puede superar los cien mil dólares. Formas para mitigar los costos son descritas a continuación.

Plataforma Dependiendo del tipo de software, los costos de adquirir una plataforma *MOOC* pueden variar de manera considerable. Generalmente existen tres opciones para la elección de una plataforma web: Comprar una plataforma existente, utilizar software *open source* o crear una plataforma propia desde cero. Cada uno de estos cursos de acción tiene beneficios y costos, por ejemplo comprar una plataforma existente trae el beneficio de contar con el soporte y las garantías de la empresa que ofrece el sistema a expensas del costo económico que significa adquirir la licencia. En el caso de una plataforma *open source*, los costos de adquisición son mínimos debido a que el código fuente del sistema es accesible pero no hay soporte ni garantías por lo que la institución debe contratar personas con experiencia en la plataforma. La alternativa de hacer una plataforma desde cero es costosa y requiere mucho tiempo pero puede salvar costos a largo plazo debido a que se tiene control total sobre todo el código lo que permite modificar la plataforma a gusto cuando se requiera. Adicionalmente se tienen cursos de acción intermedios que tratan de mitigar los aspectos negativos de uno con los positivos del otro como, por ejemplo, un sistema mixto donde usar componentes *open source* mitigue el gasto de adquirir otros componentes de software, o que los componentes comprados sean fáciles de operar para evitar contratar programadores y así disminuir costos de personal.

Personal necesario Para montar y mantener una plataforma web es necesario disponer personal que maneje distintas especialidades. En primer lugar se necesitan administradores de sistemas para instalar, asegurar y mantener software de la plataforma. También es necesario contar con programadores que puedan modificar la parte web del software y acomodarlo a las necesidades e imagen corporativa de la institución. Finalmente hay que contar con un equipo de profesores/ayudantes que se preocupen de crear y mantener los cursos ofrecidos por la plataforma. Las instituciones educativas que ofrecen carreras de informática a menudo poseen el capital humano necesario para realizar estas tareas. En el caso del *DI USM* existen varias instancias de formación para programadores y administradores de sistemas por lo que resulta viable utilizar ayudantes o alumnos en práctica para las labores requeridas por la plataforma lo que podría ayudar a reducir costos asociados al personal.

Hardware Todo sistema web debe ejecutarse sobre la infraestructura adecuada. En el caso de un sistema *MOOC* que espera recibir gran número de conexiones debido a la naturaleza del contenido ofrecido es necesario contar con equipos rápidos y compatibles con técnicas de escalabilidad, como virtualización bajo demanda.

Cursos y contenido Usualmente los cursos que la plataforma publica como *MOOCs* son dictados previamente en la institución por lo que mucho del contenido para presentar está listo, sin embargo hay que adaptarlo para los usuarios del sistema. La razón principal tras esta tarea es que el perfil del estudiante de una plataforma *MOOC* no necesariamente es igual al perfil del estudiante usual que se encuentra en la institución, que tiene que cumplir con los prerrequisitos impuestos para cursar cada clase. De esta forma el material debe editarse para que sea lo más fácil de entender posible. Todo este trabajo no puede ser llevado a cabo por el profesor en solitario por lo que es necesario que cada curso a publicarse en la plataforma cuente con un equipo de personas que ayude en la tarea de generar y adaptar contenido para los estudiantes de la plataforma web.

Actualmente, existen varias plataformas y proveedores de cursos en el formato *MOOC* separados según si la institución es con o sin fines de lucro. muchos de estos proveedores están relacionados directamente con alguna Universidad conocida y otros se enfocan al ámbito comercial entregando acceso a cursos de capacitación para empresas. El cuadro 3.2 muestra algunos de los proveedores más populares indicando el país donde se localiza la institución, fecha de fundación y tipo de proveedor, donde **C** se refiere a un sitio comercial y **NC**, a un sitio no comercial o sin fines de lucro.

Cuadro 3.2: Resumen proveedores de cursos *MOOC* populares a la fecha.

Proveedor	País	Fund.	Tipo	Sitio web oficial
Stanford Online	USA	2006	NC	online.stanford.edu
Coursera	USA	2012	C	www.coursera.org
Khan Academy	USA	2006	NC	www.khanacademy.org
NPTEL	India	2006	NC	npTEL.ac.in
Eliademy	Finlandia	2012	C	eliademy.com
Udacity	USA	2012	C	www.udacity.com
OpenClassrooms	Francia	2007	C	openclassrooms.com
OpenLearning	Australia	2012	C	www.openlearning.com
edX	USA	2012	NC	www.edx.org

Fuente: Elaboración propia. Enlaces revisados al 29 de marzo de 2017.

Regresando al tema de las plataformas, muchos de los proveedores listados anteriormente manejan sus plataformas de maneras distintas dependiendo de la orientación comercial, acceso a recursos o políticas institucionales. Por ejemplo *Coursera* utiliza una plataforma privada propia de la institución, mientras que el proveedor Finandés *Eliademy* utiliza como base de su plataforma a *Moodle* y, en el otro extremo *edX* creó y publicó el código de su plataforma bajo el formato *open source* el cual está disponible desde el sitio web oficial open.edx.org.

Capítulo 4

Moodle como LMS

La situación actual del *DI USM* está dada por el uso de *Moodle* como su plataforma de *e-learning* institucional. El cuadro 4.1 entrega un perfil resumen sobre el sistema *Moodle* según lo publicado en su página oficial a la fecha del 19 de noviembre del 2016.

Cuadro 4.1: Perfil sistema *Moodle*.

<i>Moodle</i>
★ Fundador: Martin Dougiamas (actual director). ★ Institución: Moodle Pty Ltd (Australia). ★ Sitio oficial: moodle.org ★ Distribución: <i>GNU General Public License</i>.
★ Tipo de plataforma : Sistema web, <i>Open source</i>. ★ Lenguaje base: <i>PHP</i> versión 5.X y 7 (limitaciones)
★ Base de datos: <i>PostgreSQL, MySQL, MariaDB, MS SQL Server, Oracle Database</i>. ★ Soporte webservice: <i>REST (XML), REST (JSON), XML-RPC (sin soporte)</i>.
★ Certificaciones y estándares: <i>Open Source Definition</i> (Código y documentación). <i>IMS LTI™ Certified</i> (<i>LTI tool provider</i> desde versión 3.1)¹ <i>SCORM-ADL</i> (especificación <i>SCORM 1.2</i>) <i>Mozilla Open Badges</i>
★ Extensión: Sistema de plugins propio. (Catálogo disponible en moodle.org/plugins).

Fuente: Elaboración propia. Descripción e información adicional respecto a los estándares cumplidos en anexo B.

4.1. Características

Moodle cuenta con una comunidad activa que está constantemente creando y modificando funcionalidades en el sistema. Para cada una de las nuevas versiones o *releases*, la institución *Moodle Pty Ltd* define una hoja de ruta donde se indican las funcionalidades y áreas de desarrollo sugeridas para que los involucrados en el proyecto concentren sus esfuerzos. Las características principales que ofrece la plataforma están agrupadas en tres grupos: Características generales, administrativas y de gestión de cursos.

Generales Corresponden a herramientas y funciones para los usuarios del sistema y se relacionan fuertemente con la usabilidad de la página web.

- Interfaz responsiva que ajusta su tamaño según el dispositivo utilizado para acceder.
- Página de usuario inicial con tablero de navegación personalizable.
- Actividades colaborativas que incluyen foros, wikis, entre otros.
- Herramienta de calendario integrada a toda la plataforma.
- Gestor de archivos para acceder y almacenar contenido multimedia.
- Editor de texto incorporado.
- Sistema de notificaciones automáticas.
- Herramienta de monitoreo de progreso personalizable para profesores y estudiantes.

La idea tras estas funciones es que los usuarios puedan desarrollar todas las actividades necesarias para el acceso y gestión de cursos directamente desde la página web desde una interfaz común para todos los usuarios, donde las funciones específicas para profesores, estudiantes y administradores sean mostradas en la interfaz según el rol del usuario.

Administración Las herramientas administrativas son clave para el correcto funcionamiento de la plataforma. En el caso de *Moodle*, el sistema cuenta con una amplia gama de herramientas que se pueden utilizar directamente desde el sitio web minimizando la interacción con el sistema operativo base lo que ayuda a evitar que se altere por error la configuración o parte del sistema base lo que podría causar que toda la plataforma deje de funcionar. Entre las herramientas más importantes están:

¹ *LTI de Learning Tool Interoperability* es un estándar especificado por *IMS Global Learning Consortium* para la integración de aplicaciones de aprendizaje con plataformas y servicios web.

- Personalización del sitio mediante *temas*. Se pueden crear o ajustar otros existentes.
- Autenticación e inscripción en cursos con múltiples opciones²
- Soporte para múltiples idiomas.
- Creación y gestión de cursos por lotes.
- Sistema de gestión para permisos y roles de usuario.
- Soporte de estándares (ver en el perfil, cuadro 4.1).
- Soporte para interoperación con sistemas externos mediante plugins.
- Sistema de plugins y complementos incorporado.
- Actualizaciones regulares de seguridad para proteger el sistema de vulnerabilidades conocidas y nuevas.
- Reportes y bitácora para monitorear participación y actividad de usuarios.

Gestión de cursos Las herramientas de esta categoría incluyen el soporte tradicional de un *LMS* para la realización de cursos online junto con el soporte de interoperabilidad y estándares definidos por las certificaciones que posee la plataforma. Es importante destacar que hay muchas otras funciones disponibles en formato de complementos o *plugins* que no son incluídas aquí y se pueden consultar en el catálogo moodle.org/plugins (revisado al 29 de marzo de 2017).

- Definición de rutas de aprendizaje mediante un sistema de prerrequisitos.
- Actividades colaborativas como foros, tareas en grupo y sobre bases de datos³.
- Compatibilidad con recursos externos y material multimedia de otros sitios.
- Soporte multimedia para presentar archivos de audio y video en los cursos.
- Gestión de estudiantes por grupos.
- Sistema de calificación de tareas que se puede distribuir entre varios ayudantes.

² Lista completa de métodos: docs.moodle.org/19/en/Authentication (enlace revisado al 29 de marzo de 2017).

³ Actividad donde un grupo de estudiantes trabaja sobre una colección de datos común subida por el profesor.

- Calificaciones en línea y retroalimentación mediante anotaciones en los documentos (*pdf*).
- Evaluación de pares y auto evaluación.
- Compatibilidad con el sistema de insignias *Open Badges* de *Mozilla*⁴.
- Métodos avanzados de calificación por medio de competencias y rúbricas.
- Asignación de espacios privados para grupos de aprendizaje.

Facilidades de uso *Moodle* cuenta con facilidades importantes a la hora de gestionar cursos en forma masiva con el fin de reducir la carga de trabajo de profesores y ayudantes. El primer elemento destacable corresponde a la interfaz para agregar, modificar y eliminar elementos en los cursos, que consiste en un modelo del tipo *drag and drop* en el cual se selecciona el componente que se desea agregar y una vez puesto en el sitio del curso se puede mover con libertad a otra posición por medio del uso del *mouse*. La dificultad presente para profesores y ayudantes corresponde a conocer las propiedades de cada actividad que se desea agregar, las cuales tienen a ser numerosas debido a que los desarrolladores buscan crear herramientas generales para cubrir la mayor parte de los casos de uso que podrían requerir los administradores de los cursos como profesores y ayudantes, sin embargo las actividades cuentan con configuraciones predefinidas que simplifican la configuración cuando se necesite. En el caso de los estudiantes, el uso de la plataforma es simplificado mediante el uso de la página del curso, donde se colocan todas las actividades necesarias para completar el mismo. Las pruebas, tareas y otras evaluaciones son accesibles por medio de un click donde se indica mediante una interfaz simple las acciones necesarias para que el estudiante lleve a cabo la actividad. Por ejemplo en una encuesta, se requiere que el profesor ingrese la descripción de la actividad y la plataforma presta la interfaz para responder la encuesta y reportar los resultados. Ejemplo de una encuesta predefinida, con configuración e imagen de cómo se ve en la plataforma se encuentra en la documentación oficial de *Moodle* en docs.moodle.org/all/es/Encuesta_predefinida (revisado al 29 de marzo de 2017). Todas las actividades se operan en forma similar.

Cursos El manejo de cursos creados en la plataforma permite copiar cursos ya cursados manteniendo los contenidos subidos al mismo, además de crear y borrar múltiples cursos simultáneamente. Para ello *Moodle* provee la herramienta *upload* y *backup*, las cuales guardan y cargan cursos utilizando archivos en formato de valores separados por coma o *CSV*. Por ejemplo, si un curso hecho en *Moodle* manualmente quiere copiarse varias veces como base para otros cursos se puede utilizar la herramienta *backup* para guardar la información de curso para luego elaborar el correspondiente archivo *CSV* que declare múltiples veces el curso con los distintos nombres correspondientes. De esta manera,

⁴ Sistema creado por *Mozilla* para compartir logros. Más información en openbadges.org

al entregar el archivo a la herramienta *upload*, la plataforma creará todos los cursos declarados en el archivo con las configuraciones y material declarado en el curso original que se incluyan en el archivo utilizado. el documento sobre la función *upload* en la documentación oficial de *Moodle* detalla totalmente su aplicación en crear múltiples cursos, entregando un archivo *CSV* de ejemplo e indicando todas las propiedades disponibles para la personalización de los cursos a través de este mecanismo mediante archivo. La documentación oficial se encuentra en docs.moodle.org/32/en/Upload_courses (revisado al 29 de marzo de 2017). En el caso de copiar cursos, la documentación oficial presenta el uso de la herramienta *backup*, la que permite exportar la información de los cursos y otros tipos de contenidos pedagógicos dentro de la plataforma en un formato estándar empaquetado en un archivo *ZIP* llamado archivo *mbz* de *Moodle Backup ZIP*. El enlace docs.moodle.org/21/en/copy_course muestra como utilizar la herramienta *backup* para copiar cursos completos sin los datos de estudiantes. Información detallada de la herramienta *backup* incluyendo enlaces a casos de uso específicos se pueden encontrar en la documentación oficial en docs.moodle.org/30/en/Course_backup.

4.2. Moodle en el DI USM

A todos los cursos dictados por el *DI USM* se les asigna una sección de curso en *Moodle* con dos objetivos. El primero es utilizar la plataforma para actividades del curso, como publicación y entrega de tareas, publicación de notas, etc. El segundo es utilizar *Moodle* como herramienta de respaldo para los cursos con el fin de acceder al material publicado en ocasiones futuras. La plataforma utilizada por el *DI USM* se encuentra en el dominio moodle.inf.utfsm.cl y es utilizada por estudiantes, profesores y profesionales en cursos de capacitación. La figura 4.1 muestra la *landing page* del sistema.

Figura 4.1: Moodle landing page

Home » Log in to the site You are not logged in. English (en)

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

LIDER EN INGENIERIA, CIENCIA Y TECNOLOGIA

Departamento de Informática Universidad Técnica Federico Santa María

LOG IN

Username

Password

☐ Remember username

[Forgotten your username or password?](#)

Cookies must be enabled in your browser

You are not logged in.

Fuente: sitio moodle.inf.utfsm.cl revisado al 29 de marzo de 2017.

Es común que cada institución que utiliza un sistema *LMS* considere ciertas funcionalidades del sistema por sobre otras, según las necesidades que la institución presenta. En el caso del *DI USM*, la mejor forma de determinar las funcionalidades de la plataforma que son más relevantes para la institución es considerar el uso que le dan los profesores y alumnos a la plataforma en los ramos que son dictados por el *DI USM*. Para este caso se considerarán dos ramos en particular que, a juicio del autor, cumplen con ciertos requisitos para ser considerados relevantes en este estudio. A continuación se describen los ramos y se exponen las razones de su elección.

Seminario de Memoria Corresponde al ramo de sigla *ICI-309* de la *USM* y es dictado por el profesor Ms. Luis Hevia Rodriguez. El ramo corresponde al primer semestre del último año de la malla curricular de la carrera *Ingeniería civil Informática* y se considera el prerrequisito para cursar la Memoria de Titulación. Lo que hace a este ramo relevante para este trabajo es que el profesor encargado utiliza ampliamente la plataforma *Moodle* en todos los trabajos y entregas del ramo, colocando especial atención en el uso del calendario incorporado en el sistema para planificar el ramo desde el comienzo por lo que todos los estudiantes saben desde el primer día cuáles serán las tareas y evaluaciones que deberán entregar junto con las fechas tentativas de cada una de estas. Adicionalmente, el profesor realiza encuestas y utiliza el sistema de evaluación incluido en *Moodle* para publicar todas las calificaciones y comentarios pertinentes. Este es el ejemplo perfecto de un ramo que realiza casi todas las labores de evaluación y calificación a través de un sistema *LMS*, dejando las horas de cátedra para cumplir con objetivos transversales al ramo, por ejemplo

con cursos y exposiciones para reforzar habilidades blandas útiles a la hora de presentar, relatos y experiencias de ex-alumnos e invitados, entre otras actividades. La única parte de la evaluación del curso que no se realiza por *Moodle* es la calificación de asistencia a clases que el profesor realiza personalmente durante las horas de cátedra. En resumen:

- Ramo importante preparatorio para realizar la Memoria de Titulación.
- Uso intensivo de varias funcionalidades de *Moodle* como calendario, encuestas, reportes de calificaciones, entre otros.
- Ejemplo en que utilizar el sistema *LMS* deja el tiempo de cátedra para realizar actividades formativas variadas fomentando el aprendizaje y la participación de los estudiantes.

Fundamentos de Informática Corresponde a los ramos de sigla *ILI-152* e *ILI-153* de la *USM*, donde ambos son dictados por el profesor Dr. Horst von Brand. Actualmente, al año 2016, el ramo fue reemplazado por el ramo *Estructuras Discretas* debido a un cambio de malla en la carrera de *Ingeniería Civil Informática* con el fin de acortar la duración de la carrera, sin embargo el ramo nuevo incluye contenidos seleccionados de ambos predecesores y mantiene la misma metodología de trabajo y uso de la plataforma *Moodle*. Esta secuencia de ramos es relevante para el estudio debido a que el autor trabajó como ayudante durante 5 años, por lo que se conoce de primera mano los casos de uso de la plataforma *Moodle* tanto para estudiantes como para profesor y ayudantes. Adicionalmente, esta secuencia de ramos, tanto para la malla nueva como para la anterior, es dictada a la gran cantidad de alumnos que cursan segundo año de carrera o se cambian a la carrera y deben utilizar *Moodle* por primera vez creando un ambiente con gran variedad de usuarios del sistema en distintos roles. Los usos principales que se dan a la plataforma en estos ramos son los siguientes:

- Medio oficial de comunicación entre profesores, estudiantes y ayudantes por medio del foro de discusión. Se publica liberación de tareas, fechas y cambios de fecha de evaluaciones, listados de alumnos para interrogación, dudas y consultas de alumnos, avisos comunitarios como publicación de notas, entre otros.
- Subida y publicación de material como apuntes de clase, enunciados de tareas y actividades extra, publicación de pautas solución y material de apoyo como códigos fuente de programas auxiliares.
- Subida y almacenamiento de aportes creados por los estudiantes como programas auxiliares, documentos y ensayos, entre otros.
- Recepción de tareas y trabajos dados a los estudiantes. En este ramo no se suele utilizar la función de publicación para calificaciones de *Moodle* ya que el profesor titular utiliza una planilla propia para realizar y publicar evaluaciones la cual se sube a la plataforma como un archivo externo.

Los ramos de *Fundamentos de Informática* consideran muy importante el trabajo que los estudiantes realizan en la elaboración de las tareas que suele incluir ejercicios prototipo para los certámenes, por esta razón es prioritario que todos los estudiantes estén inscritos en el respectivo curso de *Moodle* al comenzar el curso. La herramienta de creación e inscripción automática de cursos de *Moodle* resulta vital para este propósito y permite que los alumnos inscritos por medio del sistema de gestión de la *USM*, llamado *SIGA*⁵, sean inscritos automáticamente en *Moodle* al comienzo del semestre. Adicionalmente, *Moodle* provee al profesor la posibilidad de inscribir alumnos por su cuenta lo que permite cubrir casos de inscripciones atrasadas o gente en situaciones especiales.

Los ramos descritos cubren las funciones más importantes de un sistema *LMS* para la operación de ramos en el *DI USM*. Estas funciones se pueden clasificar en dos categorías, lo que se llamará en adelante la *clasificación DI USM*:

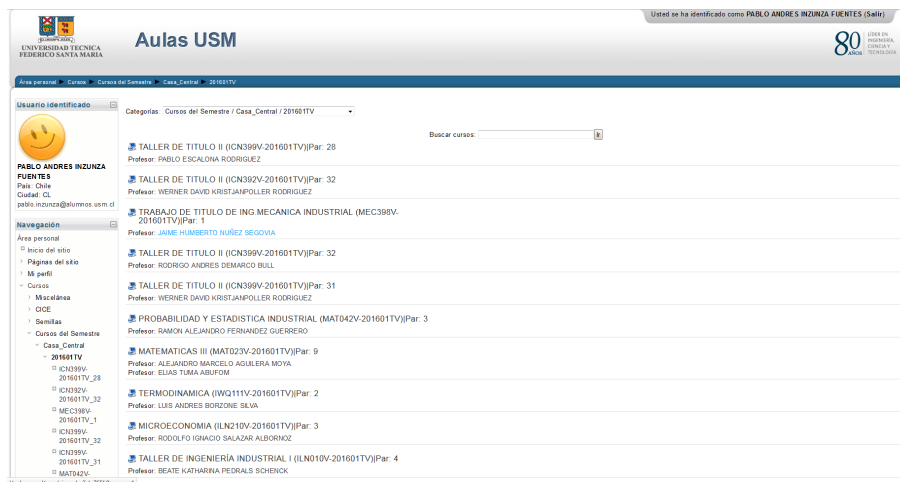
CI Funciones operacionales.

CII Funciones transversales.

Otros ramos La *UTFSM* como institución incluyó su propia instancia de *Moodle* como plataforma *LMS* institucional llamada *Aula*, la cual se puede acceder por medio de la cuenta de usuario creada para los servicios que entrega la universidad al momento de la matriculación. El servicio que presta *Aula* está automatizado para que se creen los cursos correspondientes con los debidos alumnos inscritos en sincronía con la información oficial del sistema de gestión académica de la universidad llamado *SIGA*. Los departamentos son libres de utilizar o no el espacio provisto por la plataforma y, en el caso del *DI USM*, no suele utilizarse porque este estamento ya cuenta con su propia instancia de *Moodle*. Otros estamentos de la misma universidad, como el *Departamento de Matemáticas* y el *Departamento de Física* hacen extenso uso de los cursos provistos en *Aula* para implementar los ramos comunes que dictan a todos los estudiantes matriculados como los primeros cursos de matemáticas y física. El sitio *Aula* es un ejemplo de cómo utilizar *Moodle* para automatizar labores administrativas para muchos alumnos, cualidad necesaria para cualquier sistema que quiera implementar cursos en formato *MOOC*, sin embargo no cuenta con una página o interfaz inicial atractiva para descubrir cursos ya que asume que los alumnos ya saben a que cursos entrar gracias al proceso previo de inscripción de ramos en el sistema de gestión de la universidad. La figura 4.2 muestra la interfaz de *Aula* para la búsqueda de ramos.

⁵Es la plataforma web que la Universidad utiliza para inscripción de ramos, generación de certificados, entre otros.

Figura 4.2: Búsqueda de ramos *Aula USM*.



Fuente: Página oficial aula.usm.cl (revisado al 29 de marzo de 2017).

En general, las funcionalidades más utilizadas por los ramos en *Aula* son las siguientes:

- Publicar información sobre fechas de tareas, pruebas, controles.
- Realizar anuncios oficiales a través del foro de noticias de los cursos.
- Publicar planillas con las notas. Estas planillas suelen ser archivos externos y no se utiliza a menudo el sistema de evaluación de *Moodle*.
- Subir guías y archivos de material relevante, usualmente tareas y resúmenes de las lecciones en formato *PDF*.

En la experiencia del autor, todos los ramos cursados utilizaban diversos medios digitales, como *blogs* personales y correos electrónicos pero se limitaban a las funcionalidades anteriormente descritas que forman parte de las incluidas en

Funciones operacionales Corresponden a funcionalidades de la plataforma que influyen directamente en la evaluación o nota final de los estudiantes y están relacionadas a entrega de tareas, material para certámenes y pruebas, entre otros. Adicionalmente se incluyen las funciones relacionadas a la participación del estudiante en el curso (que es necesario para poder evaluarlo) como inscripción y administración de roles en el sistema, funciones de comunicación como foros de noticias y calendarios, y funciones de calificación como la herramienta de publicación de notas de *Moodle* utilizada en el ramo *Seminario de Memoria*.

Funciones transversales Corresponden a funcionalidades de la plataforma que no influyen directamente en la evaluación o nota final de los estudiantes, sin embargo son relevantes para el cumplimiento de los objetivos transversales que se planteen en el ramo. Ejemplo de esto son las funciones que permiten a los estudiantes generar y subir contenido creado por ellos como aporte, foros de discusión y colaboración, encuestas y métodos de evaluación docente, gestión de material multimedia y audiovisual, entre otros.

Aplicación Con el fin de utilizar esta información para la evaluación de la plataforma *Open EdX* se pueden utilizar las funciones descritas para crear una rúbrica que permita determinar qué tan compatible es una plataforma *LMS* con las necesidades del *DI USM*. Si bien las rúbricas de evaluación son un concepto desarrollado en el ámbito académico pedagógico y se pueden enfocar de varias maneras, son útiles para medir el cumplimiento de objetivos. En su trabajo del 2015 el autor Phillip Dawson [15] entrega un resumen sobre los distintos usos y descripciones del término *rúbrica de evaluación*, el cual es mostrado como un concepto *opaco* utilizado y definido de diversas maneras generando una gran cantidad de significados y usos distintos a lo largo de la historia. Para esclarecer el significado del concepto, Dawson cita al autor James Popham y su trabajo [8] realizado en 1997, donde se expone que una rúbrica de evaluación debe cumplir con tres condiciones relevantes.

Definición 5 (Rubrica de evaluación). *Herramienta de evaluación que, independiente de como se defina, debe poseer las siguientes tres cualidades:*

RI *Criterio(s) de evaluación totalmente definidos.*

RII *Definiciones de calidad para cada uno de los criterios en sus distintos niveles.*

RIII *Estrategia de valoración (puntaje).*

A pesar que el mismo Dawson menciona que muchos de los usos que se les dan a las rúbricas no cumplen en su totalidad con las condiciones dadas por Popham, las rúbricas presentadas en este trabajo se entenderán bajo estas condiciones con el fin de asegurar que estos instrumentos de evaluación pertenezcan a una misma clase.

4.3. Rúbrica para un Software

La principal razón por la que utilizar una rúbrica para evaluar un software está dada por el contexto de este trabajo. Para este caso se tiene una situación particular que es la implementación del modelo de educación a distancia *MOOC* por medio de una plataforma. Esta situación requiere que el software cumpla una condición muy objetiva que es la compatibilidad con el modelo *MOOC*, y otra no tan objetiva referida a satisfacer las expectativas que tiene el *DI USM* para una plataforma de este tipo. Esta combinación de objetivos requiere de una

evaluación que permita estrategias de valoración diferentes pero que se puedan integrar. En el caso de la evaluación del software *open edX*, la primera condición tiene como criterio que la plataforma cumpla cada una de las funciones definidas en las cuatro categorías de la clasificación *Becta* (ver sección 3.2), mientras que la segunda condición considera las distintas categorías de manera distinta dependiendo de la importancia que le otorga el *DI USM* a cada una de estas funciones, lo que se refleja en el uso que le dan los usuarios durante el curso de los ramos.

A continuación se define un tipo de rúbrica para evaluar un sistema computacional bajo el contexto de este trabajo, alineada con las cualidades definidas por James Popham.

Definición 6 (RECS). *Una rúbrica para la evaluación de competencia de un software (RECS) consiste en una herramienta para determinar lo competente de un sistema de software respecto de un conjunto de objetivos a alcanzar definidos en términos de requerimientos funcionales y no funcionales. La herramienta RECS cuenta con las siguientes tres cualidades:*

- RI** *Criterio de evaluación. Dado un conjunto de objetivos, definidos según requerimientos funcionales y no funcionales, se procede a revisar si las funciones del software satisfacen los requerimientos. El criterio de evaluación consiste en decidir si el sistema cumple los objetivos determinando si presenta las funciones necesarias para cumplir con los requisitos pedidos.*
- RII** *Calidad. Dependiendo de factores como tecnología o plataforma que el software usa, ciertos requerimientos pueden ser total o parcialmente cumplidos. Incluso si dos softwares poseen las mismas funcionalidades, estas pueden variar debido a diferencias en la implementación de cada una, por lo que se debe definir una escala de valores para verificar el nivel de satisfacción que el sistema otorga al requerimiento. La escala de calidad debe ser del tipo $[0, \dots, -n]$, donde 0 significa “totalmente satisfecho” y $-n$ significa “no satisfecho”. El valor de n dependerá de la cantidad total de posibilidades intermedias que presente el sistema. Estas calificaciones representan descuentos o penalizaciones que son descontados al total de puntos inicial asignado a cada criterio de evaluación. La cantidad de puntos inicial es igual a n .*
- RIII** *Estrategia. Se considerarán dos tipos siendo la primera el uso de una medida de localización estadística sobre los puntajes de cada criterio obtenidos luego de aplicar las calificaciones (descuentos) con el fin de obtener una evaluación dentro de la misma escala de calidad. La naturaleza de la medida de localización dependerá del contexto en el cual se realiza la evaluación. La segunda estrategia, utilizada en el ámbito educacional, consiste en determinar el porcentaje de puntos obtenidos en la evaluación respecto del total posibles considerando todos los criterios. Este porcentaje es interpretado mediante el uso de intervalos definidos según el contexto de la evaluación.*

La evaluación entrega dos resultados, un promedio de puntos y un porcentaje de aprobación, donde la influencia de cada uno de los criterios es determinada por el contexto de la evaluación, el tipo de estadística a utilizar y los intervalos de porcentaje elegidos por el evaluador.

Resulta importante recalcar que, a la luz de esta definición es perfectamente posible que dos evaluaciones de los mismos objetivos entreguen resultados completamente distintos. Ejemplo de esto sería evaluar el mismo sistema (*Open edX* por ejemplo) en dos Universidades distintas. Considere el caso de dos Universidades llamadas A y B, respectivamente. Ambas quieren evaluar la adquisición del software S, el cual es famoso por ofrecer avanzadas funciones de evaluación y reporte automático de notas pero no está a la vanguardia en otros campos como la inclusión de medios cooperativos como foros de discusión y mensajería. La Universidad A es muy conocida y posee gran cantidad de alumnos que han sido atraídos por la imagen de la institución como centro tecnológico, en cambio la Universidad B, que maneja menos estudiantes, tiene una plana docente que prefiere métodos más convencionales para enseñar y ven los sistemas de software como una herramienta de comunicación. En este caso, la Universidad A va a evaluar muy bien al software S debido a que tiene muchos alumnos y sus profesores valoran las herramientas de evaluación automática que ofrece la plataforma, sin embargo esta fortaleza del software S no es tan bien recibida en la Universidad B donde la carga de aprender a utilizar las herramientas automáticas es mayor que seguir realizando las evaluaciones y reportes a mano. Además el software S no tiene funciones avanzadas para trabajo colaborativo fallando en el área de comunicación que tanto le importa a la Universidad B, por lo que se tiene una evaluación con iguales objetivos y resultados distintos debido a las preferencias y situaciones particulares de cada usuario de la herramienta de evaluación. Estas diferencias se reflejan en la *estrategia de evaluación*, por ejemplo si en el caso ambas Universidades evaluaran utilizando un promedio ponderado (aún con los mismos resultados parciales), la Universidad A tendría más ponderación a las funciones relacionadas con automatización logrando una nota más alta y la Universidad B habría asignado la mayoría de la ponderación a los métodos de comunicación y cooperación que el software S tiene con bajo nivel de calidad, resultando en una nota inferior.

Rúbrica de ejemplo Considerando todo lo anterior, se pueden utilizar los requerimientos de los ramos representativos para elaborar una rúbrica tipo *RECS* para evaluar si un software puede ser utilizado por los ramos del departamento. Un ejemplo de rúbrica de este tipo se expone en el cuadro 4.2. En esta rúbrica se ha utilizado una escala de valoración tipo $[0, -1]$ lo que refleja que para la evaluación lo que importa es si hay o no una función en el sistema que cumpla con el requisito sin admitir valores de satisfacción intermedios. Este es un ejemplo de *valoración cerrada*, donde se busca determinar si el sistema cumple en su totalidad o no, siendo rechazado si es que falla en alguno de los criterios. Sin embargo se puede utilizar una medida de localización estadística como estrategia de valoración que de más relevancia a, por ejemplo, las fun-

ciones operacionales debido a que son vitales para la realización del curso por medio de una plataforma computacional.

Cuadro 4.2: Rúbrica tipo *RECS* para evaluar concordancia con los ramos elegidos representativos. Los criterios son evaluados como *cumple* (**C**) y *no cumple* (**NC**).

Criterio	C (0 pts.)	NC (−1) pts.
Clasificación: Funciones operacionales.		
Creación de usuarios.		
Definición de roles.		
Creación de cursos.		
Inscripción de estudiantes por lotes.		
Inscripción individual.		
Importación contenido cursos previos.		
Calendario y planificación en sistema.		
Sistema noticias y anuncios.		
Publicación material educativo.		
Publicación tareas.		
Integración evaluaciones con calendario.		
Entrega tareas.		
Sistema recuperación tareas entregadas.		
Publicación de notas.		
Sistema archivos para aportes de usuarios.		
Sistema de notificaciones.		
Total:		
Clasificación: Funciones transversales.		
Sistema de encuestas.		
Foros de discusión.		
Retroalimentación y comentarios de usuarios.		
Sistema de archivos material audiovisual.		
Herramientas de visualización en sistema.		
Total:		
Total Evaluación:		

Fuente: Elaboración propia.

Ejemplo de evaluación Para exponer el uso de este tipo de rúbrica considérese la misma expuesta en el cuadro 4.2 pero con una escala de cinco valores. Esto implica que la evaluación posee las siguientes características:

- Se tienen 16 criterios de evaluación para funciones operacionales con un total de 64 puntos, y 5 criterios de evaluación para funciones transversales

con un total de 20 puntos.

- Cada calificación puede restar entre 0 y 4 puntos al total de puntajes de la evaluación dependiendo de la calidad con que la plataforma satisface el criterio de evaluación correspondiente, por lo que cada criterio de evaluación tiene un puntaje de calidad que va desde 0 (no satisface) hasta 4 (satisface totalmente).
- según la definición 6, para obtener una evaluación de calidad dentro de la misma escala se puede utilizar una medida de localización estadística, por ejemplo un promedio aritmético, entre los puntajes de cada criterio.
- Otra evaluación la provee el porcentaje de puntos obtenidos luego de evaluar, respecto del total alcanzado. Para determinar la calidad que representa el porcentaje obtenido es necesario crear intervalos que representen los distintos niveles de aceptación que se le exige a la plataforma.

Ponderaciones e intervalos El contexto de la evaluación queda plasmado en los pesos con que se considera cada criterio de evaluación y los intervalos utilizados para clasificar el porcentaje de satisfacción. En el caso de un promedio aritmético se está utilizando igual peso a cada uno de los criterios. Los intervalos representan la exigencia que se le coloca a la evaluación, por ejemplo la selección en el cuadro 4.3 representa una evaluación exigente respecto a la presente en el cuadro 4.4 ya que el segundo intervalo utilizado para indicar satisfacción es más amplio.

Cuadro 4.3: Ejemplo de definición de intervalos exigente.

Satisfecho	[91 %, 100 %]
Parcialmente satisfecho	[31 %, 90 %]
No satisfecho	[0 %, 30 %]

Fuente: Elaboración propia. La exigencia es relativa al cuadro 4.4.

Cuadro 4.4: Ejemplo de definición de intervalos menos exigente.

Satisfecho	[71 %, 100 %]
Parcialmente satisfecho	[31 %, 70 %]
No satisfecho	[0 %, 30 %]

Fuente: Elaboración propia. La exigencia es relativa al cuadro 4.3.

Este formato de rúbrica se utiliza como base para las evaluaciones presentadas posteriormente sobre la plataforma *open edX* como plataforma *MOOC*.

Adicionalmente, los requisitos enunciados en la rúbrica son utilizados para definir una ponderación tentativa en la estrategia de evaluación del sistema *openedX* respecto de las necesidades del *DI USM*, según lo determinado por los ramos representativos elegidos anteriormente la que no representa necesariamente la realidad de la institución pero sirve de base para realizar evaluaciones posteriores más rigurosas o precisas.

Capítulo 5

Open edX

Al igual que el sistema *Moodle*, la plataforma *Open edX* cuenta con un sistema de gestión de aprendizaje incorporado y posee funciones especializadas en el manejo y gestión de cursos en el formato *MOOC*. El cuadro 5.1 entrega un perfil resumen de *Open edX* según la información de documentación publicada en el sitio oficial al 29 de marzo de 2017. Existen varias diferencias entre *Open edX* y *Moodle* partiendo por el lenguaje base, en donde el primer sistema es programado usando un *web framework* más actual como *Django* y el segundo da preferencia a un esquema más tradicional utilizando *PHP* con el fin de crear una plataforma propia sin forzar los conceptos de diseño que vienen al utilizar cualquier *web framework*. La otra diferencia fundamental entre ambos sistemas es la definición de sus arquitecturas de software y la importancia que dan a este concepto en ambos casos. *Open edX* considera fundamental mantener una arquitectura de software bien definida y estable a lo largo de sus distintas versiones. El modelo de la plataforma y sus componentes es parte integral de su documentación y se pide expresamente que los contribuyentes ayuden a mantener esta arquitectura creando componentes independientes que se comuniquen con las funciones principales del sistema por medio de sus interfaces en vez de conectar o modificar estas clases. El uso de componentes de software es una decisión fuerte en *open edX* que separa su arquitectura en cuatro capas bien definidas que contienen aplicaciones independientes programadas para trabajar utilizando interfaces *REST*.

REST La sigla *REST* corresponde al término inglés “*Representational state transfer*”. Corresponde a un protocolo de comunicación entre servicios web cuyas operaciones cumplen con la característica de ser *stateless* (sin estado). Que un protocolo maneje operaciones sin estado quiere decir que cada transacción realizada es independiente de las demás y, por consiguiente, la operación consiste en pares de *consulta* y *respuesta* sin ninguna relación con otros pares previos. Como consecuencia de usar un esquema de este tipo, el servidor no necesita guardar información relativa a la *sesión* o *status* del usuario del servicio.

Cuadro 5.1: Perfil sistema Open edX.

<i>Open edX</i>
★ Fundador: Universidad de Harvard y MIT (proyecto conjunto). ★ Institución: edX Incorporated, USA. ★ Sitio oficial: www.edx.org ★ Sitio oficial código: open.edx.org ★ Distribución: <i>GNU Affero General Public License.</i> <i>Apache License.</i>
★ Tipo de plataforma : Sistema web, <i>Open source, Free Software</i> ¹. ★ Lenguaje base: <i>Python</i> versión 2.7.10 ★ Web Framework: <i>Django</i> versión 1.8.7 ★ Release más reciente: - Open edX Release Eucalyptus.2 (2 de Septiembre de 2016).
★ Front End: (sitios revisados al 29 de marzo de 2017) - Mako template generator. www.makotemplates.org - JavaScript. www.javascript.com - CoffeeScript. coffeescript.org - Backbone.js Framework. backbonejs.org - Sass CSS Framework. sass-lang.com - Bourbon library for Sass. bourbon.io
★ Base de datos Principales: - MongoDB (cursos y foros). - MySQL (Datos de usuarios y otros). ★ Soporte webservices: <i>REST (JSON)</i>
★ Certificaciones y estándares: <i>Open Source Initiative</i> (Código y documentación). <i>IMS LTI™ Certified (LTI tool provider)</i> <i>SCORM-ADL</i> (especificación <i>SCORM 1.2</i>)² <i>Mozilla Open Badges</i>
★ Extensión: Sistema de componentes <i>XBlock</i>. openedx.atlassian.net/wiki/display/COMM/XBlocks+Directory

Fuente: Elaboración propia. Descripción e información adicional respecto a los estándares cumplidos en anexo B.

¹ Según lo especificado por la *Free Software Foundation* en su sitio oficial www.fsf.org (revisado al 29 de marzo de 2017).

² El servicio para la especificación *SCORM 1.2* es provisto por un componente de software aparte definido en el formato de extensión de la plataforma llamado *XBlock*. El código a la fecha del 29 de marzo de 2017 se puede encontrar en github.com/appsembler/edx_xblock_.

Al adoptar este diseño de componentes que se comunican por medio de protocolos sin estado, *open edX* apunta a crear una plataforma cuyos componentes sean fáciles de replicar lo que permite satisfacer crecidas de demanda replicando las instancias de componentes que estén ocasionando cuellos de botella para así atender a más usuarios simultáneamente, mientras existan recursos de hardware disponibles. Información sobre el origen y aplicación del concepto *REST* se encuentra en el trabajo realizado por Roy Fielding, el autor del término, en el año 2000 en [5] correspondiente a su tesis doctoral.

5.1. Arquitectura

El desarrollo de *Open edX* ha sido orientado a una arquitectura por capas donde cada uno de sus componentes pertenecen a una capa específica. La última versión liberada durante el desarrollo de este trabajo (indicada en el perfil) posee cuatro capas definidas independientes, las cuales son:

CI Herramientas y clientes.

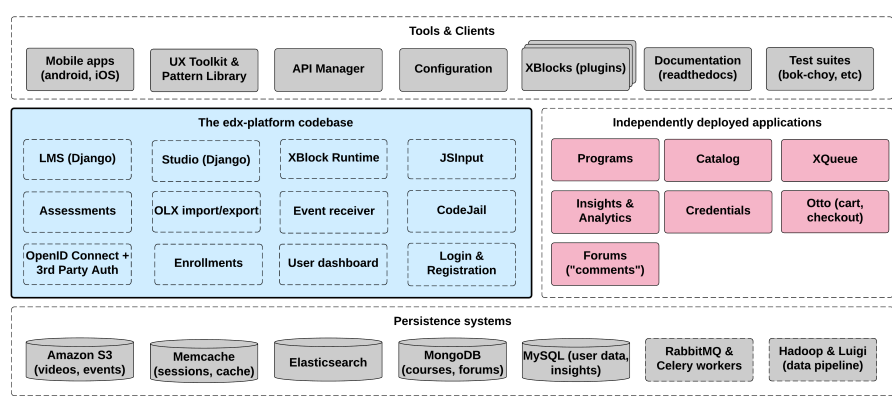
CII Base de código plataforma.

CIII Aplicaciones desplegadas independientemente.

CIV Sistemas de persistencia.

La figura 5.1 muestra las distintas capas y los componentes de software que integran cada capa.

Figura 5.1: Arquitectura de componentes de *open edX Eucalyptus.2*



Fuente: Documentación oficial en edx.readthedocs.io/projects/edx-developer-guide/en/latest/architecture.html (revisado al 29 de marzo de 2017).

scorm.

Herramientas y clientes Corresponde a los componentes auxiliares de la plataforma utilizados para brindar funciones adicionales que no son necesarias para el funcionamiento básico del sistema. Entre los componentes están las bibliotecas y herramientas para la creación de interfaces de usuario, el sistema de extensiones *XBlock*, aplicaciones para la gestión de archivos de configuración y documentación, aplicaciones móviles, entre otras.

Base de código Corresponde a los componentes que implementan las funcionalidades principales del sistema y que permiten el funcionamiento de los cursos. Está compuesto por tres componentes principales y el resto son componentes auxiliares. Los componentes principales son el *LMS* o sistema de gestión de aprendizaje, *Studio* que es un sistema de gestión de contenido para los profesores, y el *XBlock Runtime* correspondiente al motor para ejecutar las extensiones del sistema. Debido a la arquitectura por componentes de la plataforma, el *LMS* no incluye las funcionalidades administrativas, las cuales son manejadas por los componentes de software auxiliares como *Enrollments* (inscripciones), *Login and registration* (cuentas de usuario) y *User dashboard* (funciones por roles). Adicionalmente, *Open edX* creó su propio estándar para crear y compartir contenido educativo llamado *Open Learning XML* el cual es utilizado en toda la plataforma y se puede exportar a otros estándares como *IMS LTI* o *SCORM* por medio de extensiones. Este estándar es manejado por el componente *OLX import/export*. Una de las características propias de la plataforma es que incluye en su base de código y no como extensión un ambiente para ejecutar código fuente en forma segura. Este ambiente manejado por el componente *CodeJail* permite aislar los programas subidos por los estudiantes mitigando posibles vulnerabilidades de la plataforma pero impone restricciones en el sistema operativo a instalar debido a que el componente requiere del software *AppArmor* para funcionar correctamente el cual está disponible para algunos sistemas basados en *Linux* pero no en todas las distribuciones. Ejemplo es el sistema *Red Hat Enterprise Linux* utilizado en el *DI USM* que no considera a *AppArmor* como paquete estándar de la distribución. Para información detallada sobre la disponibilidad de *AppArmor* consultar su wiki oficial en wiki.apparmor.net/index.php/Main_Page (revisado al 29 de marzo de 2017).

Aplicaciones independientes La plataforma considera otras funciones auxiliares como aplicaciones independientes de la plataforma con el fin de reducir la complejidad de la base de código simplificando la curva de aprendizaje para los desarrolladores y contribuyentes que se integran recientemente al desarrollo del sistema. Los componentes principales desplegados en forma independiente corresponden al sistema de análisis llamado *Insights and Analytics* que recolecta información transversal de la plataforma con fines de investigación, el componente *Catalog* que gestiona la presentación y descubrimiento de los cursos y el componente *XQueue* que permite al *LMS* de la base de código conectarse con evaluadores externos. Otras aplicaciones corresponden al sistema de credenciales y el manejo de foros de discusión externos a los ramos.

Sistemas de persistencia *Open edX* maneja varios sistemas de persistencia distintos en forma simultánea para distintos tipos de dato, además de incluir código listo para integrarse a sistemas de persistencia Corporativos como *Amazon Web Services*. La información más sensible correspondiente a cuentas de usuario y datos personales es almacenada en una base de datos local donde la tecnología de preferencia es *MySQL*. Los datos gestionados por el *LMS* relativos a los cursos y foros es almacenado en una instancia de *MongoDB*. Adicionalmente, la arquitectura incluye componentes especializados como un módulo basado en *memcached* para gestionar la información y datos de sesión, *ElasticSearch* como motor de búsqueda y módulos de trabajo distribuido o gestión y transporte de datos, como *Celery Workers*, *Hadoop* and *Luigi* y *RabbitMQ*. La figura 5.2 lista los sistemas mencionados que componen el esquema de persistencia de datos de *Open edX*. En cada recuadro se indica el nombre del sistema junto al conjunto de datos que debe gestionar, el tipo de sistema e información sobre su sitio web oficial.

Figura 5.2: Sistemas componentes para la persistencia de datos en *Open edX*.

Persistencia

Datos Usuarios	MySQL		Datos Sesión	memcached	
Tipo	Base de Datos Relacional.		Tipo	Sist. Caché Distribuido de Datos	
Web Oficial	www.mysql.com		Web Oficial	memcached.org	

Foros y Cursos	MongoDB	NoSQL	Búsqueda	ElasticSearch	
Tipo	Base de Datos Orientada a Documentos		Tipo	Motor Búsqueda Distribuido	
Web Oficial	www.mongodb.com		Web Oficial	www.elastic.co	

Otros

Celery	Cola de Tareas / Trabajos Asincronizados	RabbitMQ	Sistema Comunicación por Paso de Mensajes
Web Oficial	celeryproject.org	Web Oficial	www.rabbitmq.com

Hadoop	Biblioteca para Computación Distribuida	Luigi	Paquete Python para Trabajos por Lotes
Web Oficial	hadoop.apache.org	Web Oficial	github.com/spotify/luigi

Fuente: Elaboración propia. Las páginas oficiales se han revisado al 29 de marzo de 2017.

Presentación *Open edX* trae consigo interfaces web simples para acceder al listado de cursos y otras configuraciones en la plataforma permitiendo su uso inmediato luego de instalar y desplegar el sistema. La distribución de *Open edX* no incluye una *home page* o página inicial para recibir a los usuarios lo que implica que las instituciones que usen *Open edX* deben programar sus propias

páginas para presentar su instancia de la plataforma. Ejemplo de esto es el sitio oficial de *edX* el cual corresponde a una página independiente basada en *Drupal*³ que presenta información y noticias a los usuarios redirigiendo a la instancia de *Open edX* cuando es necesario, la cual tiene un dominio independiente. La página usada por *edX* para presentar cursos a los usuarios no forma parte de la plataforma y también es un sitio independiente basado en *Drupal*, sin embargo los contenidos (cursos) presentados por dicha página son comunicados a la misma mediante la herramienta *course discovery* incluida en *open edX*.

En general, *Open edX* adopta un enfoque de flexibilidad contra simplicidad recurrente en toda su arquitectura, entregando una solución inicial de amplias posibilidades que requiere de programación para refinar y ajustar a las necesidades particulares de cada institución aumentando los costos necesarios de implementación. Este enfoque contrasta con *Moodle* y sistemas similares que utilizan personalización dentro de la plataforma, como temas y otras configuraciones, para editar la presentación de la instancia sin recurrir a programación.

Elementos MOOC El elemento principal que distingue a *Open edX* como plataforma *MOOC* es su motor de *course discovery* o descubrimiento de cursos. Como se indica en la sección 3.4, la recomendación de cursos es un elemento fundamental para la implementación de cursos *MOOC* debido a la naturaleza de los futuros estudiantes, los cuales no necesariamente tienen decidido que tipo de cursos buscan. Actualmente este servicio se encuentra en un repositorio independiente y se define como *data aggregator software*, es decir un sistema que recopila información en bases de datos y las agrupa en conjuntos para su posterior procesamiento. El motor posee el nombre de *edX Course Catalog Service* y cuenta con una interfaz para comunicarse con programas independientes a *Open edX* y tiene como objetivo prestar utilidad a estudios de mercadotecnia, publicidad de los cursos y generación de catálogos de datos para posterior procesamiento. El repositorio oficial revisado al 29 de marzo de 2017 se encuentra en github.com/edx/course-discovery. En cuanto al elemento de comunidad, *Open edX* cuenta con un sistema de foros de discusión para los cursos durante su desarrollo pero otras funcionalidades más generales, como un sistema de foros general, se encuentran proyectados como funciones futuras a implementar como es indicado en la hoja de ruta del proyecto donde, por ejemplo en open.edx.org/features/discussion-forums (revisado al 29 de marzo de 2017), se planea incorporar un sistema de foros de discusión para la versión móvil de *Open edX*.

5.2. Desarrollo Actual

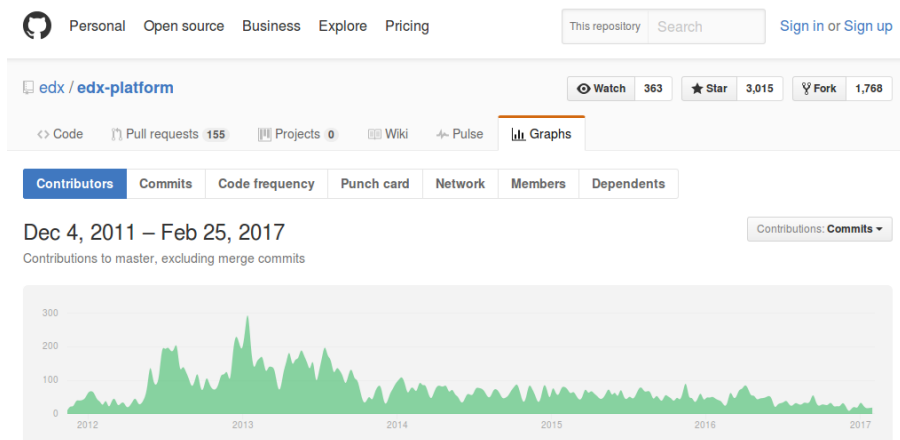
Open edX es un proyecto en desarrollo que al utilizar el formato *open source* publica los códigos fuente de la plataforma para que usuarios alrededor del mundo tengan acceso a sus propias instancias de prueba y puedan modificarlas,

³ Sistema de gestión de contenido *open source*. Página oficial en www.drupal.org (revisado al 29 de marzo de 2017).

participando en el desarrollo de nuevas funcionalidades para las futuras entregas del proyecto. El desarrollo del proyecto es guiado principalmente por una hoja de ruta que está publicada en el sitio web oficial del proyecto y tiene como objetivo ayudar a los desarrolladores voluntarios a decidir dónde concentrar sus esfuerzos. Adicionalmente existen grupos de desarrollo para las extensiones creadas y futuras de extensiones (llamadas por el proyecto *XBlocks*) que poseen sus propios foros y repositorios.

Desarrollo principal El proyecto *Open edX* utiliza como herramienta de comunicación oficial el sistema de listas de correo y foro de *Google* llamado *Google Groups*⁴. A la fecha de este trabajo, el foro principal de trabajo de la comunidad corresponde a la lista `edx-code@googlegroups.com` donde se entregan enlaces a otros foros más especializados. Este foro cuenta con alta actividad y diversos temas de discusión relacionados a la situación actual y planes futuros de desarrollo para *Open edX*. El repositorio oficial donde se encuentra el código fuente de la plataforma *Open edX* es `github.com/edx/edx-platform` y a la fecha revisada (29 de marzo de 2017) cuenta con una comunidad de contribuyentes de 334 personas. En términos de actividad, la plataforma *GitHub* recoge la actividad en el repositorio mediante gráficos de contribución a la rama principal. La figura 5.3 muestra este gráfico donde se aprecia que la actividad de los desarrolladores y contribuyentes se mantiene a pesar de no ser tan alta como en los primeros años de desarrollo.

Figura 5.3: Gráfico de contribuciones hasta 2017.



Fuente: *GitHub*, repositorio oficial de *Open edX* en `github.com/edx/edx-platform/graphs/contributors?from=2011-12-04&to=2017-02-25&type=c` (revisado al 29 de marzo de 2017).

⁴Sitio web oficial `groups.google.com` (revisado al 29 de marzo de 2017).

Desarrollos adicionales La comunidad también participa activamente en el desarrollo de extensiones para la plataforma, las que inicialmente se manejaban a través de una lista de correos en *Google Groups* y que actualmente funcionan en forma independiente en el sitio del directorio de extensiones *XBlock Directory* en xblocks.org (revisado al 29 de marzo de 2017). Adicionalmente, *Open edX* mantiene un directorio en *Atlassian Confluence* para informar sobre el estado de desarrollo de cada una de las extensiones desarrolladas e inscritas en el sitio oficial. Este directorio puede consultarse en openedx.atlassian.net/wiki/display/COMM/XBlocks+Directory junto con varias secciones de documentación relevante que incluye recursos para desarrolladores, estrategias de programación y buenas prácticas, documentación sobre la arquitectura y definición de desarrollos futuros en forma de *roadmap*, entre otros recursos. A la fecha de este trabajo las extensiones programadas para *Open edX* en formato de *XBlock* registran los siguientes datos:

- En el sitio web oficial xblocks.org se registran y ofrecen 28 extensiones.
- En el sitio de *Xblock Directory* hay 55 extensiones registradas.
- 25 de ellas cuentan con mantención activa y son compatibles con la versión de *Open edX* más reciente.
- 8 de estas extensiones han cesado de dar mantención o desarrollo y sus repositorios se encuentran sin actividad.

La gran cantidad de extensiones mantenidas da cuenta del interés y compromiso que la comunidad de *Open edX* ha mantenido desde el 2012, cuando la plataforma fue liberada, por lo que es de esperar que el trabajo en torno a *Open edX* se mantenga al menos por unos años más.

Capítulo 6

Evaluación

Con el fin de determinar si la plataforma *open edX* corresponde a una solución clase **SIII** se requiere de la realización de tres evaluaciones, cada una asociada a uno de los objetivos de la evaluación definidos en la sección 2.3. La evaluación sobre las funciones del *LMS* se realiza por medio de una rúbrica tipo *RECS*. Las otras evaluaciones se realizan verificando directamente la documentación oficial de la plataforma, correspondiente a gestión de cuentas de usuario para el objetivo **Específico II**, y arquitectura de la plataforma para el objetivo **Específico III**. En cada una de las tres secciones se agrega una tabla comparativa resumida con otras plataformas *open source* alternativas

6.1. Funciones de LMS

Como se describió en el capítulo 5 las funciones relativas a un *LMS* definidas por las categorías *Becta* se encuentran repartidas entre componentes de software dentro de las capas de la arquitectura de *Open edX*. Con el fin de simplificar la evaluación, se considerará que una función es cumplida si existe algún componente en alguna de las capas de la plataforma que satisface el requerimiento funcional *Becta* respectivo. La razón por la que se utilizará la clasificación de requerimientos funcionales *Becta* para la evaluación por medio de rúbricas es que las categorías descritas incluyen todas las funciones utilizadas por el *DI USM* en *Moodle*, según

lo encontrado en los ramos representativos elegidos por el autor. Es perfectamente posible que algunos ramos actuales o fuera del conocimiento del autor utilicen alguna funcionalidad nueva o no representada por las categorías *Becta*, sin embargo basta con agregar el criterio al conjunto definido en la rúbrica y ajustar la medida de localización estadística según la relevancia de ese criterio. Incluso si la evaluación resultante al agregar este nuevo criterio fuera muy diferente, se puede utilizar el resultado expuesto en este trabajo como punto de comparación para determinar la importancia de esta funcionalidad.

A continuación se define la rúbrica a utilizar y se presentan los resultados según los criterios de evaluación expuestos en la definición.

Definiendo la rúbrica

Siguiendo las categorías de la clasificación *Becta* para requerimientos funcionales de un *LMS* se presentará una rúbrica segmentada en cuatro partes. El objetivo de esta evaluación es determinar si el sistema *open edX* cumple con los requerimientos funcionales descritos por la clasificación sin dar preferencia a alguno en específico. Las condiciones de Popham de la evaluación son las siguientes.

Criterio de evaluación Se utilizan los requerimientos funcionales descritos en las cuatro categorías de requerimientos *Becta* para sistemas *LMS*. El criterio a utilizar corresponde a si el sistema *Open edX* cumple con funciones para satisfacer cada requerimiento. En este caso, debido a que *Open edX* es una plataforma relativamente reciente, se le exigirá una escala de calidad con cuatro niveles dependiendo de la flexibilidad que presente en el cumplimiento de cada requerimiento.

Calidad Se utiliza una escala de calidad de cuatro valores con escala de descuentos tipo $[0, -1, -2, -3]$ donde 0 significa *totalmente satisfecho* (**S**), -1 significa *parcialmente satisfecho* (**PS**), -2 significa *insuficientemente satisfecho* (**IS**) y -3 significa *no satisfecho* (**NS**).

Estrategia Se utiliza como estrategia de valoración para esta evaluación las dos descritas para el formato de rúbrica *RECS* descrito en 6. La primera estrategia utiliza una medida de localización estadística (promedio aritmético) entre los puntos obtenidos de cada criterio luego de aplicar los descuentos determinados en la evaluación para determinar una nota de calidad dentro de la escala definida de cuatro valores. La segunda estrategia es calcular el total de puntos obtenidos respecto del total de puntos posibles para estimar un porcentaje de cumplimiento de la plataforma respecto de la evaluación. Ambas medidas de indican en cada una de las rúbricas de evaluación realizadas para cada categoría de criterios.

A continuación se presentan las rúbricas para cada categoría indicando en forma de resumen la justificación de las evaluaciones para los requerimientos funcionales solicitados por la categoría *Becta* correspondiente.

Evaluación: Gestión de contenido El sistema reparte las funciones de gestión relacionadas al contenido en múltiples componentes a lo largo de las capas que forman la arquitectura. Los requerimientos **GCI** y **GCII** están ampliamente soportados por la componente *Studio* que permite a los profesores crear, subir y gestionar material educacional durante la creación y gestión de cursos. Sin embargo, en los roles de estudiantes y ayudantes el soporte es limitado a los menús de entrega de tareas y foros de discusión. La plataforma entrega las funcionalidades completas para los requerimientos mencionados a los estudiantes y ayudantes por medio de su sistema de extensión. El requerimiento **GCIII**

relacionado con marcas de recursos está actualmente en debate. Desde el 2014 que grupos de desarrolladores están discutiendo sobre la implementación del sistema de marcas. El origen de la discusión se puede revisar en un archivo del *Google Group* groups.google.com/forum/#!topic/edx-code/XXb3PIPdopU (revisado al 29 de marzo de 2017). La gestión de los metadatos está bien soportada en lo relativo a los cursos y contenido creado con el estándar propio *OLX*, sin embargo los usuarios (salvo los profesores) solo pueden utilizar metadatos en el motor de búsqueda. La creación de recursos está soportada solo para los profesores en el componente *Studio*. El cuadro 6.1 muestra la rúbrica de evaluación para esta categoría.

Cuadro 6.1: Rúbrica de evaluación para la categoría *Gestión de contenido* de la clasificación *Becta*.

Criterio	S (0 pts.)	PS (−1 pts.)	IS (−2 pts.)	NS (−3 pts.)
GCI	✓			
GCII	✓			
GCIII		✓		
GCIV		✓		
GCV	✓			
Total:	13/15 puntos			
Promedio:	2,6 puntos por criterio			
Satisfacción:	86,7 %			

Fuente: Elaboración propia.

Evaluación: Planificación y clasificación curricular La plataforma *open edX* implementa las interfaces de usuario, tanto en su plataforma web como en sus aplicaciones móviles, conforme al estándar de accesibilidad para personas con discapacidad propuesto por la organización *W3C* en [19]. El módulo *Assessments* incluye tareas y evaluaciones de varios tipos. Se cubre la evaluación entre estudiantes, trabajos en grupo y diagnósticos. La interfaz por sí sola no dispone de herramientas para personalizarla y los cambios en el tema se deben realizar del lado del servidor. Los cambios en la interfaz se deben programar utilizando herramientas de *Front end*. El componente *Studio* permite a los profesores planificar sus cursos, material, tareas y actividades con total antelación. Los usuarios pueden ver la planificación antes de inscribirse en un curso. Adicionalmente se cuenta con un motor de búsqueda independiente que se utiliza para realizar búsquedas de acuerdo a etiquetas de los cursos y otros parámetros, sin embargo el sistema no trae la interfaz para el descubrimiento de cursos implementada en su totalidad. La personalización del sistema es total ya que se puede modificar toda la plataforma utilizando código y extensiones, sin embargo no hay muchas herramientas para personalizar desde la misma interfaz de usuario. La plataforma es capaz de sugerir secuencias de cursos y actividades agrupando los cursos en *planes de estudio*. En cuanto al sistema

de evaluación, *Open edX* provee un servicio de notas base para todos los cursos que se configura definiendo políticas que consideran el tipo de tarea, peso respecto a la nota final, entre otros aspectos. Con respecto a las evaluaciones, el equipo de desarrollo de *Open edX* publica en una de sus listas de correos¹ que se está terminando una funcionalidad llamada *librería de contenido* cuyo objetivo es la creación de *bancos de preguntas* para usar como base en la creación de evaluaciones para los distintos cursos. La plataforma cuenta con una interfaz para autoevaluación y evaluación entre pares. El cuadro 6.2 muestra la rúbrica de evaluación para esta categoría.

Cuadro 6.2: Rúbrica de evaluación para la categoría *Planificación y clasificación curricular* de la clasificación *Becta*.

Criterio	S (0 pts.)	PS (-1 pts.)	IS (-2 pts.)	NS (-3 pts.)
PCCI	✓			
PCCII	✓			
PCCIII			✓	
PCCIV	✓			
PCCV		✓		
PCCVI	✓			
PCCVII	✓			
Total:	18/21 puntos			
Promedio:	2,6 puntos por criterio			
Satisfacción:	85,7 %			

Fuente: Elaboración propia.

Evaluación: Compromiso y administración de estudiantes El sistema *Open edX* está pensado para brindar servicio *e-learning* completo, por lo que los requisitos en esta categoría están bien soportados. En términos de acceso remoto, la plataforma cuenta con una interfaz que utiliza el protocolo de conexión *open SSL*² el cual cifra los datos de la conexión. Al incorporar el protocolo por medio del sistema de paquetes del sistema operativo base, la plataforma se beneficia de cualquier actualización de seguridad que se le aplique al protocolo. *Open edX* incluye una interfaz de autenticación compatible con sistemas de autenticación externos y sistemas basados en *Single Sign On* por medio de *SAML*. El sistema cumple con el estándar de autenticación *OAuth 2.0* descrito en [6]. El resto de los requerimientos como *protección de datos* y *Acceso a la información* son tratados utilizando las herramientas del *web framework Django*. Los portafolios de contenidos, la planificación de cursos e información de

¹Dirección de la lista en groups.google.com/forum/#!topic/edx-code/wv1TdmnLNMY, revisado al 29 de marzo de 2017.

²Biblioteca de software para seguridad en capa de transporte (*TLS*) y capa de *sockets* (*SSL*). Sitio web oficial en www.openssl.org (revisado al 29 de marzo de 2017).

estudiantes están incorporados en los componentes de la capa *base de código*. El seguimiento y datos de uso de la plataforma son tratados por la aplicación de despliegue independiente *Insights and Analytics*. El cuadro 6.3 muestra la rúbrica de evaluación para esta categoría.

Cuadro 6.3: Rúbrica de evaluación para la categoría *Compromiso y administración de estudiantes* de la clasificación *Becta*.

Criterio	S (0 pts.)	PS (−1 pts.)	IS (−2 pts.)	NS (−3 pts.)
CAEI	✓			
CAEII	✓			
CAEIII	✓			
CAEIV	✓			
CAEV	✓			
CAEVI	✓			
CAEVII	✓			
CAEVIII	✓			
CAEIX	✓			
CAEX	✓			
CAEXI	✓			
Total:	33/33 puntos			
Promedio:	3 puntos por criterio			
Satisfacción:	100 %			

Fuente: Elaboración propia.

Evaluación: Herramientas y servicios de comunicación y colaboración El punto fuerte de la plataforma en esta categoría es el manejo de foros de discusión y colaboración. El sistema permite a profesores que suban material bajo derechos de autor pero fuerza a que cualquier contenido que no sea explícitamente declarado como dominio público quede bajo las leyes de derechos de autor definidas en los Estados Unidos. En términos de servicio web, la plataforma presenta una interfaz *REST* específica que, aunque moderna y rápida, fuerza a los servicios basados en otros estándares (como *SOAP* y *XML-RPC*) a adaptar sus consultas. El cuadro 6.4 muestra la rúbrica de evaluación para esta categoría.

Cuadro 6.4: Rúbrica de evaluación para la categoría *Herramientas y servicios de comunicación y colaboración* de la clasificación *Becta*.

Criterio	S (0 pts.)	PS (−1 pts.)	IS (−2 pts.)	NS (−3 pts.)
HSCI	✓			
HSCII		✓		
HSCIII		✓		
Total:	7/9 puntos			
Promedio:	2,3 puntos por criterio			
Satisfacción:	77,8 %			

Fuente: Elaboración propia.

Evaluación total El cuadro 6.5 combina los resultados obtenidos de evaluar cada categoría por separado generando una nota general basada en la estrategia de evaluación definida dentro de la especificación de la rúbrica. En este caso corresponde al promedio entre las notas de cada categoría considerando cada una con la misma ponderación que las demás. El valor promedio obtenido es de $-0,38$ que se encuentra en el subintervalo $[0, -1]$ de la escala de medición definida, del cual se interpreta que la plataforma *open edX* satisface en su mayoría los requerimientos funcionales solicitados por la clasificación *Becta* dado que la nota se encuentra entre el rango de "satisfecho" (**S**) y "parcialmente satisfecho" (**PS**).

Cuadro 6.5: Resumen de resultados para cada categoría y evaluación final.

Categoría	Tot.	Prom.
Gestión de contenido	13/15	2,6
Planificación y clasificación curricular	18/21	2,6
Compromiso y administración de estudiantes	33/33	3,0
Herramientas y servicios de comunicación y colaboración	7/9	2,3
Total:	71/78	
Promedio:		2,63
Satisfacción:		91,0 %

Fuente: Elaboración propia.

Evaluación DI USM

La evaluación anterior no toma en cuenta las preferencias y requerimientos particulares de la institución en la que se desarrolla este trabajo dado que asigna la misma relevancia a cada requerimiento por igual sin considerar cuáles de estos son más importantes para el *DI USM*. Con el fin de lograr una evaluación

más realista es necesario considerar estas preferencias dentro de las rúbricas. Para conseguirlo se puede proceder de varias formas. Uno de estos caminos es evaluar la plataforma *Open edX* bajo la rúbrica definida en el cuadro 4.2 la cual utiliza como requerimientos los definidos en la clasificación *DI USM* basada en los ramos tipo. Esta evaluación tiene el problema de su resultado no es comparable con el obtenido por medio de las rúbricas que siguen la clasificación *Becta* ya que utiliza una escala de calidad con solo dos valores. El cuadro 6.6 muestra la aplicación de esta rúbrica según la información sobre la plataforma *Open edX* descrita anteriormente. La principal debilidad de *Open edX* se aprecia en la rúbrica y consiste en que no posee un calendario que se pueda sincronizar con las evaluaciones y otros eventos, sino que se limita a mostrar las fechas de evaluaciones inminentes en la página de inicio de los estudiantes. Actualmente se puede integrar al sistema el servicio de calendarios de *Google* por medio de una extensión³ que se debe activar aparte. El mismo caso sucede con los aportes y material generado por estudiantes. El resultado final de esta evaluación va a depender de la estrategia de valoración a utilizar. Por ejemplo, si cada uno de los requisitos se considera mandatorio, entonces *open edX* no cumpliría con los requerimientos debido a que falla en dos de estos. Si se utiliza una estrategia de valoración basada en una medida de localización estadística, entonces la evaluación va a depender de cuál categoría es más importante para el *DI USM* entre "*funciones operacionales*" y "*funciones transversales*".

³La herramienta viene en el formato interno de la plataforma (*OLX*) y permite incrustar el calendario de *Google* dentro de las páginas del curso en modo similar al calendario que provee *Moodle*. más información en edx.readthedocs.io/projects/open-edx-building-and-running-a-course/en/latest/exercises.tools/google_calendar.html (revisado al 29 de marzo de 2017).

Cuadro 6.6: Evaluación utilizando clasificación *DI USM* de requerimientos funcionales.

Criterio	C (0 pts.)	NC (−1) pts.
Clasificación: Funciones operacionales.		
Creación de usuarios.	✓	
Definición de roles.	✓	
Creación de cursos.	✓	
Inscripción de estudiantes por lotes.	✓	
Inscripción individual.	✓	
Importación contenido cursos previos.	✓	
Calendario y planificación en sistema.	✓	
Sistema noticias y anuncios.	✓	
Publicación material educativo.	✓	
Publicación tareas.	✓	
Integración evaluaciones con calendario.		✓
Entrega tareas.	✓	
Sistema recuperación tareas entregadas.	✓	
Publicación de notas.	✓	
Sistema archivos para aportes de usuarios.		✓
Sistema de notificaciones.	✓	
Total:	14/16	
Promedio:	0,85	
Satisfacción:	85 %	
Clasificación: Funciones transversales.		
Sistema de encuestas.	✓	
Foros de discusión.	✓	
Retroalimentación y comentarios de usuarios.	✓	
Sistema de archivos material audiovisual.	✓	
Herramientas de visualización en sistema.	✓	
Total:	5/5	
Promedio:	1	
Satisfacción:	100 %	
Total Evaluación:	19/21	
Satisfacción Total:	90,5 %	

Fuente: Elaboración propia. La medida de *satisfacción total* se calculó considerando el total de puntos obtenidos en la evaluación (19) respecto del total máximo posible a obtener (21 puntos).

A continuación se presentan las estrategias de ponderación a utilizar para integrar los distintos resultados obtenidos de cada una de las cuatro rúbricas aplicadas a *Open edX*. Cada una de estas estrategias representa un escenario distinto con el fin de ver el efecto que tienen distintas estrategias de evaluación en la evaluación resultante. Las dos primeras ponderaciones se aplican sobre los resultados obtenidos en la rúbrica que utiliza la clasificación *Becta* de requerimientos funcionales, mientras que las restantes se utilizan para ponderar los resultados obtenidos de aplicar la rúbrica basada en los criterios *DI USM* derivados de los ramos tipo. El cuadro 6.8 resume los resultados finales para los distintos casos expuestos según la siguiente clave:

- **Becta equitativo.** Se consideran todas las categorías de igual importancia.
- **Becta DI.** Las ponderaciones se ajustan para cada categoría según la cantidad relativa de requerimientos funcionales de la clasificación *DI USM* que incluyen. Por ejemplo, la categoría *Gestión de contenido* incluye tres de los trece requerimientos operacionales, por lo que su ponderación en este contexto es 23,1 %. El cuadro 6.7 indica las ponderaciones determinadas para cada categoría *Becta*. Por simplicidad se considera que las dos categorías *DI USM*, referidas como *requerimientos operativos* y *transversales* tienen igual ponderación al combinar los resultados.
- **DI cerrado.** Se pide que se cumplan todos los requerimientos en forma mandatoria. La estrategia de evaluación en este caso es el promedio de las evaluaciones para ambas categorías *operacional* y *transversal* compuestas con una función *techo* o *ceil*. De esta formula la rúbrica resultará con una evaluación de 0 (C) sólo en el caso que ambas evaluaciones por separado también lo sean.
- **DI ponderado (60/40 o 40/60).** La evaluación de una de las dos categorías, sea *operativos* o *transversales*, tiene mayor ponderación que la otra (respectivamente). Los porcentajes son de ejemplo por lo que se pueden variar acorde a la realidad del *DI USM* en caso de no considerarse representativos.

Cuadro 6.7: Resumen ponderaciones para las categorías *Becta* según requerimientos relevantes para el *DI USM*.

Categoría	Op.	Trans.
Gestión de contenido	23,1 %	40,0 %
Planificación y clasificación curricular	38,5 %	0,0 %
Compromiso y administración de estudiantes	30,8 %	20,0 %
Herramientas y servicios de comunicación y colaboración	7,6 %	40,0 %

Fuente: Elaboración propia.

Cuadro 6.8: Resumen resultados de rúbricas para la plataforma *open edX*.

Estrategia	Promedio por Criterio	Calidad (Escala)
<i>Becta</i> equitativo	2,63	S ([S , PS , IS , NS])
<i>Becta</i> <i>DI</i>	2,63	S ([S , PS , IS , NS])
<i>DI</i> cerrado	0	NC ([C , NC])
<i>DI</i> ponderado (60/40)	0,91	C ([C , NC])
<i>DI</i> ponderado (40/60)	0,94	C ([C , NC])

Fuente: Elaboración propia. Notar que el resultado *promedio por criterio* es un valor en la misma escala de calidad utilizada, por ejemplo un promedio de 2,9 en la escala de cuatro valores se aproxima a 3 que corresponde a la cualificación **S** de *satisfecho*.

En conclusión, al observar los resultados de las rúbricas es posible concluir que el sistema *open edX* cumple con la mayoría, pero no todos los requerimientos funcionales requeridos para un software tipo *LMS*. Esta conclusión está directamente relacionada con las estrategias de valoración de calidad definidas para cada una de las rúbricas utilizadas por lo que no es una conclusión absoluta y puede cambiar si se redefinen estas estrategias, sin embargo los puntos en contra en varias de las categorías *Becta* impiden que la plataforma en su estado actual satisfaga todos los requerimientos funcionales. Es importante notar que muchos de estos requerimientos se pueden cubrir mediante extensiones de la plataforma, como el caso del calendario utilizando una aplicación de *Google*.

Otras Alternativas

Para complementar la evaluación se aplicó este sistema a otras cuatro alternativas de plataformas *open source* conocidas en el medio educacional. Las plataformas alternativas corresponden a: *Moodle*, *Sakai*, *ATutor* y *Dokeos*. Más información sobre las últimas tres plataformas se indica en el anexo C. A continuación se presenta el resumen de puntos obtenidos por cada alternativa en cada una de las cuatro categorías *Becta* y la satisfacción total alcanzada. El resumen es expuesto en el cuadro 6.9. El cuadro 6.10 muestra el promedio final de puntos de cada alternativa según las estrategias de valoración definidas anteriormente para las rúbricas *Becta*. Es interesante notar que entre los sistemas tipo *LMS*, el promedio de puntos más alto según la valoración *Becta* *DI* corresponde a *Moodle*, la actual plataforma utilizada por la institución. *Open edX* tiene menor promedio de puntos que las alternativas principalmente por el poco tiempo de desarrollo en comparación, además de que las plataformas alternativas poseen alto nivel de especialización en cada una de sus áreas como *ATutor* en el campo de accesibilidad o *Dokeos* en el área empresarial y los criterios usados en las rúbricas se enfocan en evaluar calidad de los sistemas como *LMS* según lo requerido por el objetivo **I**, ignorando cualidades de *Open edX*.

especializadas en cursos *MOOC*. Estas cualidades se tocan en las evaluaciones restantes sobre gestión de usuarios y rendimiento.

Cuadro 6.9: Resumen de puntaje obtenido por alternativas a *Open edX*.

Categoría	<i>Moodle</i>	<i>Sakai</i>	<i>ATutor</i>	<i>Dokeos</i>	<i>Open edX</i>
GC	14/15	14/15	13/15	15/15	13/15
PCC	20/21	19/21	21/21	19/21	18/21
CAE	32/33	33/33	32/33	32/33	33/33
HSC	8/9	9/9	8/9	7/9	7/9
Total:	74/78	75/78	74/78	73/78	71/78
Satis.:	94,87 %	96,15 %	94,87 %	93,59 %	91,0 %
Clave					
GC	<i>Gestión de contenido</i>				
PCC	<i>Planificación y clasificación curricular</i>				
CAE	<i>Compromiso y administración de estudiantes</i>				
HSC	<i>Herramientas y servicios de comunicación y colaboración</i>				

Fuente: Elaboración propia.

Cuadro 6.10: Resumen de promedios de puntaje obtenidos por alternativas a *Open edX*.

Categoría	<i>Moodle</i>	<i>Sakai</i>	<i>ATutor</i>	<i>Dokeos</i>	<i>Open edX</i>
GC	2.8	2.8	2.6	2.8	2.6
PCC	2.9	2.7	3.0	2.7	2.6
CAE	2.9	3.0	2.9	2.9	3.0
HSC	2.7	3.0	2.7	2.3	2.3
<i>Becta eq.</i>	2.83 (S)	2.88 (S)	2.80 (S)	2.68 (S)	2.63 (S)
<i>Becta DI</i>	2.86 (S)	2.83 (S)	2.85 (S)	2.75 (S)	2.63 (S)
Clave					
GC	<i>Gestión de contenido</i>				
PCC	<i>Planificación y clasificación curricular</i>				
CAE	<i>Compromiso y administración de estudiantes</i>				
HSC	<i>Herramientas y servicios de comunicación y colaboración</i>				

Fuente: Elaboración propia.

6.2. Sistemas de Cuentas y Autenticación

Para determinar las capacidades de autenticación y manejo de cuentas que posee el sistema *open edX* es necesario recurrir en primera instancia a la documentación, y luego al código fuente. En la figura 5.1 donde se presenta la arquitectura del *release open edX Eucalyptus.2* se indica que el sistema utiliza una base de datos *MySQL* para gestionar los datos e información de los usuarios por lo que la plataforma toma como primera opción de autenticación este sistema de persistencia. Inicialmente, en base a esta información se puede concluir que *open edX* posee una gestión de cuentas independiente verificando el objetivo específico II, sin embargo es relevante conocer la variedad de sistemas de autenticación que puede manejar con el fin de determinar qué tan flexible es la plataforma.

Autenticación externa El sistema *open edX* maneja los mecanismos de autenticación externa sin una interfaz programable común. La mayoría de las alternativas de autenticación externa se deben realizar directamente en la configuración del servidor web y modificando los archivos de configuración de la plataforma directamente. El primer ejemplo de este problema es expuesto en la página sobre configuración del sistema dentro de la documentación (en formato *wiki* (glosario!)) en el repositorio donde está el código fuente⁴ en el cual se configura la autenticación externa utilizando variables y código de las librerías dentro del *web framework* utilizado para crear el sistema (*Django*). Otro sistema importante de autenticación es *LDAP* el cual es utilizado dentro del *DI USM*. Configurar *open edX* para utilizar *LDAP* requiere modificar parámetros de inicio al desplegar la aplicación por el lado del servidor y, como se indica en varios grupos de discusión en internet (principalmente *Google Groups*) hay varias situaciones sin resolver⁵. Sin embargo la comunidad está desarrollando avances en dirección al tema de autenticación externa. La aplicación *edX OAuth2 Provider* permite utilizar proveedores de autenticación basados en este estándar, principalmente orientada a *OpenID* el cual es un estándar de autenticación definido por la *OpenID Foundation* que consiste en permitir a los usuarios autenticarse en sitios web mediante un servicio externo que actúa como proveedor de identidad. Para más información visitar el sitio web oficial en [12].

El cuadro 6.11 resume los principales métodos de autenticación disponibles para la plataforma. Las cuentas de usuario y datos son almacenados en una instancia de *MySQL* dedicada de la plataforma.

⁴ github.com/edx/configuration/wiki/Setting-Up-External-Authentication

⁵ Ejemplos de problemas y discusiones relacionadas en groups.google.com/forum/#!topic/django-users/N37wJE8rcbE y groups.google.com/forum/#!topic/edx-code/KxUerzU_gPU (revisado al 29 de marzo de 2017)

Cuadro 6.11: Principales métodos de autenticación disponibles para *open edX*.

Protocolo	Tipo
Login directo	Base de datos interna (<i>MySQL</i>)
Certificados <i>SSL</i>	Soporte servidor web
<i>Central Authentication Service (CAS)</i>	Soporte servidor web
<i>Shibboleth</i> Federated Service	Soporte servidor web
<i>LDAP</i>	Librerías web framework
<i>edX-OAuth2-provider</i>	Aplicación de apoyo

En el caso de los sistemas alternativos todos tienen implementados módulos de compatibilidad con los sistemas *LDAP* y el protocolo *OAuth* destacando a estos esquemas de autenticación como alternativas populares al esquema de login directo. Una de las principales razones es la seguridad en instituciones ya que al utilizar un sistema externo de autenticación como *LDAP* permite que se puede controlar en forma más interna que la creación y gestión de los usuarios respecto a un sitio web. La alternativa más débil en términos de autenticación es *ATutor* que no trae soporte directo para sistemas de autenticación federados como *Shibboleth* y el soporte para *LDAP* debe agregarse mediante módulos de extensión a la plataforma.

En términos de gestión de usuarios todas las plataformas alternativas cuentan con formas de configurar la cantidad de usuarios inscritos por curso permitiendo una inscripción arbitraria de estos, lo que favorece su aplicación para implementar cursos *MOOC*. Todas las plataformas mencionadas contienen un esquema de definición de roles para los usuarios con el fin de identificar las opciones que la plataforma les presenta en cada una de sus interfaces. Algunas plataformas tienen un sistema de roles con menos opciones dependiendo si consideran adicionales a los tres básicos que son *estudiante*, *profesor/instructor* y *administrador*. Por ejemplo *Moodle* considera el rol de *ayudante* que puede modificar y agregar contenido a los cursos ayudando al rol de *profesor* en tareas administrativas del sitio.

6.3. Rendimiento y Escalabilidad

El método más común para mejorar el rendimiento de los servicios web consiste en la replicación del sistema. Si la plataforma recibe muchas solicitudes de usuarios, la replicación del servicio permite aliviar la carga al aumentar la cantidad de usuarios manejados al mismo tiempo en forma similar al funcionamiento de las cajas en un banco o supermercado donde se habilitan más cajas conforme aumenta el flujo de usuarios generando atenciones paralelas. Para que un software sea más fácil de replicar debe estar diseñado de tal forma que las transacciones que realizan los usuarios no dependan de información sobre otras transacciones. A este tipo de operaciones se les denomina *stateless* debido a que el servidor no necesita guardar información relativa a sesiones o estado sobre quiénes están realizando las transacciones. Un sistema que maneje sus

servicios mediante un protocolo de este tipo es más fácil de replicar y puede manejar subidas de peticiones mientras el hardware lo permita.

En el capítulo 5 se indica que la arquitectura de *open edX* consiste en componentes de software independientes que se comunican por medio de interfaces propias. Cada una de estas interfaces sigue el protocolo *REST* el cual, según lo expuesto en el mismo capítulo, exige que las transacciones realizadas a través de los servicios sean *stateless*. Esto implica que cada uno de los componentes de software de *open edX* es replicable. Sin embargo hay ciertos componentes que forman cuellos de botella en la arquitectura y limitan el rendimiento. En general estos son sistemas de *persistencia de datos* como bases de datos relacionales (*MySQL* en el caso puntual de *open edX*). Este tipo de software es difícil de replicar debido a que se necesita sincronizar todas las instancias cada vez que la base de datos se modifica para mantener la integridad del sistema. Para una exposición detallada sobre las características de escalabilidad de la plataforma y otros temas importantes (como seguridad), visitar el sitio web oficial open.edx.org/videos/edx-hosting-architecture (revisado al 29 de marzo de 2017) donde el Ingeniero Feanil Patel explica el despliegue de la plataforma *open edX*.

Como resumen, se tiene que el sistema *open edX*:

- Posee un diseño de componentes sin estado que le permiten ser compatible con diseños de *escalamiento horizontal* (replicación) manteniendo el rendimiento mientras exista hardware disponible.
- Los componentes de persistencia (principalmente la base de datos *MySQL*) constituyen un cuello de botella para el sistema ya que la forma más efectiva de incrementar su rendimiento es mediante *escalamiento vertical* (computadores más poderosos) lo que aumenta los costos considerablemente. Los sistemas de persistencia se pueden replicar pero es necesario manejar mecanismos de sincronización para mantener los datos consistentes.

Con excepción de *Moodle* que tiene sus propios formatos de despliegue para alto rendimiento/disponibilidad, las características de diseño de *Open edX* superan a los sistemas alternativos en términos de escalabilidad principalmente por el esquema de componentes independientes sin estado. La razón fundamental tras este enunciado resulta en que las plataformas alternativas se distribuyen como una única aplicación rígida donde la replicación requiere que se levanten instancias completas de cada plataforma en vez de solo los componentes que están con más tráfico. Las alternativas utilizan principalmente diseños de cluster para aumentar su rendimiento siendo *Sakai* un buen ejemplo, ya que utiliza toda la tecnología asociada a *Java* para replicar sus instancias en forma simple gracias a la *Java Virtual Machine* que le otorga portabilidad y *Apache Tomcat* que actúa como coordinador entre las instancias y facilita el balance de carga. Información detallada sobre el despliegue de aplicaciones web en *Java* se puede encontrar en el sitio oficial de *Tomcat* tomcat.apache.org/ revisado al 29 de marzo de 2017.

6.4. Viabilidad y Administración

Las tres secciones anteriores ayudan a determinar si las alternativas son sistemas viables para servir cursos en formato *MOOC*, sin embargo hay dos aspectos adicionales a considerar para completar este análisis correspondientes a:

- Nivel de madurez en el desarrollo de las plataformas considerando actividad en comunidad de desarrollo, canales para reportar errores, calidad de documentación, entre otros.
- Facilidades administrativas para administradores del sistema, profesores/ayudantes y alumnos.

Se procede a describir en mayor detalle estos aspectos para cada una de las plataformas alternativas en base a los repositorios y medios de comunicación oficiales para cada caso. Información sobre las herramientas relacionadas a la gestión de los proyectos, como *GitHub*, *JIRA*, *Confluence*, entre otros se encuentra disponible en D.

Open edX En términos de código fuente *Open edX* corresponde a una aplicación programada en *Python* utilizando las herramientas de programación web incluidas en la distribución del *framework Django* en su versión 1,8,7. El código se encuentra versionado y dividido en 127 repositorios de código accesibles desde el sistema *GitHub* en github.com/edx, la cual incluye una comunidad de 107 personas a cargo de los repositorios. El repositorio principal llamado *edx-platform* incluye el código de la plataforma base y es el más activo contando con más de 700 *releases* en cinco años de actividad registrada. Los grupos de desarrollo se comunican principalmente mediante listas de correo por medio de *Google Groups*. El sistema de documentación y reporte de errores utilizado por *Open edX* es una combinación de los sistemas *Confluence* y *JIRA*, aplicaciones creadas por *Atlassian* para la gestión de proyectos informáticos. Estos sistemas requieren poseer una cuenta en ellos para reportar errores y se debe seguir un formato específico impuesto por estas plataformas para los reportes con el fin de entregar uniformidad a las listas de problemas. Ambos sistemas ofrecen un tipo de cuenta sin costo para que personas externas al proyecto puedan reportar errores. Las labores administrativas de la plataforma *Open edX* son gestionadas a través de la herramienta *Django Admin* incorporada en el *framework*, diseñada para gestionar el contenido del sitio mediante la recolección de *metadatos* directo desde los archivos fuente del sistema. Su punto débil es que la interfaz muestra información sin omitir detalles de implementación, agregando complejidad a la operación del sitio. Las funciones administrativas para estudiantes y profesores se incluyen en la interfaz web del sitio respectivo. Para los estudiantes, la plataforma ofrece funciones como gestión de la interfaz, configurar cuenta de usuario, navegación de cursos, entre otras, mediante un menú de fácil acceso. Para los profesores se utiliza el sitio asociado a la com-

ponente *Studio* de *Open edX* para acceder al material y opciones de creación y gestión de los cursos, mediante menús apropiados ofrecidos por la interfaz.

Moodle Esta plataforma utiliza *PHP* como lenguaje base. El repositorio oficial accesible desde *GitHub* en github.com/moodle tiene actividad registrada desde el año 2005 y a partir del año 2009 presenta una gran cantidad de actividad que se mantiene hasta el presente año. La plataforma ha registrado en el repositorio 257 *releases* con más de 300 personas registradas con contribuciones al código fuente. La organización *Moodle Pty* proporciona a la comunidad *Moodle* sitios en su página oficial en formato de cursos para discusión, documentación e información oficial. La página oficial también proporciona un sitio para el reporte de errores conectado a una cuenta en *JIRA* que lleva el registro y estado de los reportes. La interfaz de *Moodle* se caracteriza por ofrecer todo tipo de funciones administrativas directamente a través de menús de fácil navegación. Se puede configurar el comportamiento del sitio, como usuarios y roles, políticas de respaldo e interacciones con *software* complementario mediante opciones en el menú disminuyendo las labores de configuración fuera de la plataforma. La interfaz ofrece para cada rol de usuario una vista principal que incluye los distintos menús para labores administrativas. Es destacable el sistema de gestión de contenido para los cursos, donde los distintos elementos son agregados y editados mediante el método *drag and drop*, es decir mover los distintos elementos a través de la interfaz utilizando el *mouse*. Esta característica permite que gestionar la presentación y material de los cursos sea más fácil.

Sakai *Sakai Project* cuenta con menos contribuyentes registrados en su repositorio oficial de *GitHub* github.com/sakaiproject, no obstante registra una actividad alta y regular en el tiempo desde el año 2005 indicando que existe alto compromiso de la comunidad con esta plataforma. Actualmente el repositorio cuenta con 154 contribuyentes y se han liberado 6 *releases* correspondientes a la versión 11 de *sakai*. La plataforma utiliza una cuenta en *JIRA* para el reporte y gestión de errores. *Sakai* utiliza un esquema de propiedades para su configuración. Estas propiedades se pueden editar en el archivo de configuración respectivo o modificarse mediante una sección especial de la interfaz del sitio web. Opciones y funciones administrativas para estudiantes y profesores son manejadas en forma íntegra mediante menús definidos. Una característica notable de *Sakai* es que la interfaz busca imitar a una red social utilizando el concepto del *mural*, en que se muestran actividades e información en orden de aparición. Función notable es la herramienta de la interfaz llamada *Roster* la cual muestra en este formato de *mural* los cursos disponibles e información asociada a ellos, como integrantes del curso, profesores y asistentes, resumen y contenidos, entre otros. Esta característica hace de *Sakai* una opción atractiva para servir cursos *MOOC*.

Dokeos Esta plataforma no cuenta con un repositorio oficial para su versión *open source* ya que utilizan como medio de distribución su página oficial donde publican el código fuente para cada *release*. Existen repositorios creados por usuarios donde publican sus propias implementaciones de la plataforma basados en lo publicado dentro de la página oficial pero suelen no presentar mucha actividad en comparación con los repositorios antes mencionados. Actualmente el autor desconoce si existe un sistema de reporte de errores oficial, ya que lo único encontrado son reportes aislados en el sistema de control de errores de *Debian*. En términos de administración, la versión *open source* muestra un enfoque corporativo orientado a ser fácil de instalar y configurable desde la aplicación web. Las labores administrativas para instructores y alumnos son ofrecidas mediante formularios y menús similar al estilo de *Moodle*, pero sin ofrecer el modelo de *drag and drop* para la gestión de contenido y edición del sitio de los cursos. En general, la principal debilidad de este *LMS* es la falta de una comunidad altamente organizada como en los casos anteriores.

ATutor El proyecto canadiense *ATutor* maneja un repositorio oficial en *github.com/atutor* desde el año 2003. El repositorio muestra que la mayor actividad se llevó a cabo durante los periodos 2003 – 2006 y 2011 – 2013. Al presente año el repositorio sigue con actividad menor a los intervalos antes mencionados. El mismo repositorio se utiliza para informe de errores y es constantemente revisado por los administradores del repositorio lo que se puede observar en el poco tiempo que demoran en contestar los reportes publicados. La instalación de *ATutor* se debe realizar manualmente ya que el código se distribuye empaquetado en formato *tarball* y se debe configurar directamente en un archivo de texto dentro de la carpeta de la distribución. Una vez instalado se puede administrar la mayor parte de las propiedades de la distribución desde una cuenta especial en la interfaz web, facilitando las labores de los administradores del sistema. Las funciones de profesores y estudiantes se entregan mediante menús y formularios en la interfaz web pero no cuentan con el nivel de personalización que ofrecen otras plataformas como *Moodle* o *Sakai*. En general la interfaz para usuarios de *ATutor* es rígida con el fin de garantizar que se mantengan las características de accesibilidad para discapacitados que destacan a esta plataforma.

Considerando la información relativa a madurez de código, todas las plataformas salvo *Dokeos* son viables debido a que poseen un mayor soporte y documentación, así como medios para reportar errores en forma eficiente. Estos aspectos son claves para la instalación y mantención de una plataforma ya que la cantidad y calidad de información disponible es vital para resolver dificultades que aparezcan durante el tiempo de vida de la plataforma. Con respecto a servir cursos *MOOC* son de especial interés las plataformas *Open edX* y *Sakai* debido a que traen herramientas incorporadas para descubrimiento de cursos.

Capítulo 7

Conclusiones y Recomendaciones

A continuación se exponen conclusiones y recomendaciones basadas en los resultados de la evaluación a la plataforma *Open edX* y las alternativas presentadas.

7.1. Objetivos

Antes de presentar las conclusiones particulares según los objetivos específicos planteados en la definición del problema es importante considerar que la plataforma *Open edX* es relativamente nueva (fundada el 2012) respecto de la fecha de elaboración de este trabajo, por lo que es muy probable que varias funcionalidades y elementos de diseño sean modificados en el futuro. Con esto en consideración se procede a concluir para cada uno de los objetivos del trabajo.

Objetivo I Con el objeto de verificar si *Open edX* cumple con las funciones de un *LMS* se realizó una evaluación por medio de rúbricas que buscaban determinar el nivel de satisfacción que la plataforma alcanzaba para los requerimientos funcionales solicitados. Se concluye que, aunque la plataforma cumple con la mayoría de los requerimientos solicitados por la agencia *Becta* y los requerimientos identificados como relevantes para el *DI USM* por medio de elección de ramos representativos, *Open edX* no satisface totalmente todos los requerimientos, dejando elementos como calendarios incorporados y gestión de aportes de estudiantes como funciones opcionales manejadas por medio de extensiones y aplicaciones externas. Se concluye que *Open edX* cumple con las funciones necesarias para ser considerado un *Learning Management System*, sin embargo presenta debilidades en algunas de estas funciones que requieren de más trabajo para implementarlas de lo que costaría en otras plataformas *LMS* más desarrolladas como *Moodle*, *Sakai*, *ATutor* o *Dokeos*.

Objetivo II En términos de gestión y autenticación de usuarios *Open edX* maneja varias alternativas muy usadas como *LDAP* y los protocolos *OpenID*,

SSL y *CAS*. Incluso es compatible con protocolos de autenticación más especializados como el sistema *Shibboleth* de autenticación federada que permite a los usuarios utilizar la misma cuenta en instituciones integrantes de la misma federación (por ejemplo una agrupación de universidades). La desventaja que presenta *Open edX* es que la mayoría de las interfaces para utilizar estas herramientas corresponden a las librerías del framework asociado (*Django*) o extensiones del mismo servidor web, lo que añade más trabajo al momento de integrar estos protocolos a la plataforma debido a que hay que configurar servicios externos a *Open edX* contrastando con otras plataformas como *Moodle* que integran los controles de estos servicios a menús accesibles a través de la interfaz web de la plataforma. En conclusión, el sistema *Open edX* ofrece varias alternativas de autenticación independientes lo que verifica que el sistema tiene un manejo de cuentas de usuario adecuado para el modelo *MOOC*, sin embargo debido a que su desarrollo es bastante nuevo la configuración de estos servicios es más difícil que en otras plataformas con más años de desarrollo. Por otra parte, el sistema de gestión de usuarios presente en *Open edX* está a la par con los sistemas presentados por los *LMS* alternativos incluyendo inscripción de estudiantes a cursos sin un límite fijo de matrícula y definición de roles para los usuarios. Estos elementos son vitales para la realización de cursos en formato *MOOC*. Como comentario adicional se destaca la existencia de un motor de descubrimiento de cursos presente en *Open edX* dando cuenta de la especialización que presenta esta plataforma en cursos *MOOC*.

Objetivo III La plataforma posee una arquitectura de componentes independientes y replicables que facilita el rendimiento por medio de la técnica de *escalamiento horizontal*. Adicionalmente, cada componente se puede replicar bajo demanda gracias a que la comunicación entre ellos es realizada por medio de un protocolo *stateless*, añadiendo flexibilidad a la operación del sistema de forma que se pueda concentrar el hardware en los componentes más solicitados en un momento dado. Se concluye que la plataforma es, gracias al diseño de su arquitectura de software, compatible con técnicas de rendimiento y en particular, con el *escalamiento horizontal* o replicación de instancias de software. Este esquema de diseño pensado en escalabilidad es reflejo del hecho que *Open edX* es relativamente reciente respecto de otras plataformas como las alternativas mencionadas y cumple con la tendencia que se ha popularizado en muchas de las plataformas web contemporáneas en varios campos de presentar diseños altamente modulares y con componentes independientes que interactúan mediante *APIs*. Sin embargo, *Open edX* sufre del cuello de botella tradicional presente en sistemas web para muchos usuarios correspondiente al sistema de persistencia. Independientemente que el diseño de componentes sea compatible con esquemas de escalabilidad por replicación, las bases de datos no pueden replicarse sin añadir un sistema de sincronización, ya que si un dato cambia en una instancia replicada, el cambio debe replicarse en todas las otras instancias para mantener el sistema web consistente. La plataforma toma como decisión de diseño no utilizar un único sistema de persistencia y define varios tipos para

conjuntos más reducidos de datos con el fin de separar el tráfico y disminuir el volumen de consultas a cada uno de los sistemas separados. Esta decisión de diseño implica que *Open edX* se beneficia de la capacidad del *hardware* de ejecutar instancias de *software* en paralelo, pero castiga el rendimiento cuando no se tiene dicha capacidad. Para ejemplificar considere el siguiente caso: *Open edX* utiliza una instancia de *MySQL* para manejar la información de los usuarios y otra instancia de *MongoDB* para gestionar la información de cursos y contenido. Compárese con un sistema hipotético **B** que utilice una única instancia de *MySQL* (u otro motor para bases de datos) para ambos conjuntos de información, usuarios más cursos. Si se tiene *hardware*, capaz de ejecutar solo una instancia a la vez, el sistema *B* envía todas las consultas a su motor sin realizar otra operación adicional, pero *Open edX* debe determinar a cuál instancia enviar dependiendo del tipo de información, además de realizar un cambio de contexto entre las instancias respectivas cuando el tipo de información cambie, dando al sistema *B* ventaja de rendimiento ya que realiza menos operaciones para manejar la misma información. Por el contrario, si se cuenta con un *hardware* capaz de ejecutar dos o más instancias en paralelo, el sistema *B* debe determinar si utiliza una de las instancias o replica su motor de base de datos para aprovechar el paralelismo ofrecido. Si utiliza una instancia, las operaciones se realizan en forma secuencial y se pierde tiempo. Si se replica el motor y se asigna a la otra instancia, el motor de base de datos debe determinar si la operación es de lectura o escritura permitiendo que las lecturas se realicen en paralelo y realizando las escrituras en forma secuencial. En el caso de *Open edX* la única operación adicional a realizar es la determinación del tipo de información ya que las bases de datos utilizadas son independientes. Se concluye que el sistema modular de *Open edX*, aplicado también a su esquema de persistencia, se ve beneficiado de las capacidades de paralelismo que posea el *hardware* que ejecute la instancia del sistema pero también es más sensible a otros sistemas cuando esta capacidad de ejecutar instancias paralelas no existe.

Evaluación Adicionalmente se resalta el papel de las herramientas *RECS* para la evaluación de este sistema de software. La razón principal es que, al momento de planificar un proyecto de software se suele definir a partir de los requerimientos funcionales, es decir las tareas que el sistema debe realizar, documentando estos requerimientos por medio de diagramas y contratos. Las rúbricas definidas en este trabajo utilizan estos requerimientos funcionales para evaluar la pertenencia del software a cierta categoría por medio de determinar la calidad con la que se cumplen estos requerimientos, por lo que se puede concluir que un proyecto de software bien documentado y con requerimientos definidos es un buen prototipo para clasificar otros sistemas de software orientados al mismo objetivo. Una evaluación de este tipo no es especialmente difícil de realizar teniendo los requerimientos funcionales claros y puede ayudar a los programadores y desarrolladores a determinar la evolución de su proyecto respecto a la idea inicial que tenían de éste.

Como conclusión general se determina que el sistema *Open edX* es una so-

lución viable del tipo **SIII** al problema inicialmente planteado bajo el contexto de las suposiciones hechas en las evaluaciones, en particular sobre las estrategias de valoración en las rúbricas y la elección de los ramos representativos del *DI USM*. Adicionalmente se determina que para realizar una evaluación más objetiva acorde a la realidad del *DI USM* es necesario definir los requerimientos funcionales que la institución considera relevante en un sistema de este tipo con el fin de establecer estrategias de valoración que se ajusten totalmente a la realidad y contexto actual de la institución. Los requerimientos funcionales definidos pueden usarse para evaluar otras plataformas alternativas a *Open edX* aumentando las opciones para elegir y facilitando a la entidad respectiva del *DI USM* la elección de una plataforma adecuada para servir cursos en formato *MOOC* a futuro.

7.2. Recomendaciones y Trabajo Futuro

El *DI USM* como institución tiene la opción de determinar si va a crear cursos en formato *MOOC* y si va a implementar una plataforma propia para servirlos o utilizará otra plataforma externa. En el caso de que la institución quiera crear este tipo de cursos y desea evaluar el uso de una plataforma propia en contraste a una pagada, se sigue realizando las siguientes acciones previas con el fin de facilitar dicha evaluación:

- Determinar los requerimientos a solicitar en el sistema y su importancia para la institución.
- Estimar la cantidad de usuarios máxima a la que se espera llegar con el fin de tener una cuantificación de costos precisa.

La primera acción tiene como objetivo ayudar a la institución a determinar el tipo de plataforma o sistema web para servir los cursos y la segunda acción busca facilitar la estimación del costo económico de cada alternativa, ya que hay una relación directa entre el total de usuarios a los que se quiere alcanzar y el costo de la plataforma dado que de este número dependerá tanto el *hardware* necesario para implementar la plataforma o el plan a pagar en caso de utilizar un servicio externo.

Recomendaciones Para el caso del *DI USM*, si desean elegir una plataforma para servir cursos *MOOC*, se recomienda que inicialmente se enfoquen en determinar las funcionalidades que más acomodan a los profesores para crear y gestionar los cursos. Este paso es muy importante y si se eligen funciones adecuadas para buscar en los sistemas, entonces los profesores podrán gestionar los cursos sin mayores dificultades disminuyendo la cantidad de capacitación necesaria para que ellos integren sus cursos a la plataforma nueva. A juicio del autor se considera muy relevante buscar plataformas que permitan incorporar material ya existente para que los profesores puedan subir material anteriormente creado con facilidad en el caso de deseen hacerlo. Una segunda recomendación

es probar cada curso con un conjunto de estudiantes de prueba para determinar la dificultad que presenta la interfaz de usuario a los estudiantes inscritos. Esto es importante porque en un curso *MOOC* no hay inscripciones de ingreso para los usuarios por lo que la interfaz que se les presenta debe ser accesible. Ambas recomendaciones se ven facilitadas si se cuenta con instancias de prueba de cada plataforma a evaluar. Si se desea disminuir costos es posible realizar estos cursos de prueba a través de implementaciones *sandbox* de estos sistemas, por ejemplo *Open edX* posee un sitio para probar la creación e inscripción de cursos, al igual que las otras alternativas mencionadas en el anexo C. Algunas de estas páginas de prueba se mencionan a continuación con los enlaces revisados al 29 de marzo de 2017.

- **Open edX.** La página oficial cuenta con una instancia para pruebas de la plataforma que tiene habilitada la inscripción a cursos y el uso de la herramienta *Studio* para crear cursos de prueba. La página se encuentra en el enlace sandbox.edx.org y el componente *Studio* en studio.sandbox.edx.org.
- **Sakai.** El proyecto *Sakai* mantiene plataformas de prueba para evaluar las características del sistema a través de socios comerciales. el enlace trysakai.longsight.com corresponde a una instancia de prueba de la plataforma mantenida por la empresa *Longsight* donde se puede crear un curso y probar las características disponibles luego de registrarse sin costo.
- **ATutor.** EL sitio oficial de este sistema cuenta con una plataforma demo para probar las características principales. El sitio se encuentra en www.atutor.ca/atutor/demo.php.
- **Moodle.** El servicio *Moodle Cloud* ofrece un plan gratuito para probar la plataforma considerando un curso con un número limitado de inscritos. Se puede ingresar a este sitio en moodle.com/cloud/free.
- **Dokeos.** La organización *Softaculous* ofrece un sitio de pruebas para la plataforma *Dokeos* en el enlace www.softaculous.com/demos/Dokeos, donde se puede entrar como administrador.

Como recomendación adicional se sugiere seguir buscando posibles alternativas a las plataformas mencionadas en este trabajo con el fin de ayudar a tomar una decisión informada sobre cuál plataforma utilizar.

Futuro Queda como trabajo futuro el diseñar una metodología adecuada para determinar los requerimientos funcionales relevantes para una institución que permita verificar la utilidad de los sistemas de software en forma más objetiva. También queda pendiente realizar una evaluación de costos implementando una instancia de *Open edX* de prueba con el fin de conocer el tiempo que toma montar la instancia, la cantidad de trabajo necesaria y la cantidad de inversión

necesaria para la capacitación de profesores y ayudantes que deseen participar en el formato de cursos *MOOC*. Además de estas tareas generales el formato de cursos *MOOC* y sus posibles implementaciones son un tema potencial para profundizar en varios aspectos, como desarrollo de *software* o administración de sistemas. Adicionalmente, los datos obtenidos de plataformas de este tipo son utilizados por universidades para desarrollar investigación en el área de pedagogía entrelazando las áreas de informática y educación. A continuación se sugieren algunos posibles temas de memoria basados en el desarrollo de este trabajo.

Funcionalidades MOOC Al igual que el trabajo realizado por *Becta* es necesario definir una colección de requerimientos funcionales necesarios para que una plataforma *web* pueda considerarse apta para servir cursos en formato *MOOC*. Para ayudar en este sentido se podría iniciar mediante el experimento de crear y dictar uno de estos cursos en alguna de las plataformas de prueba mencionadas en este trabajo y determinar los casos de uso más representativos. El objetivo general de este trabajo consistiría en definir el conjunto de requerimientos funcionales para servir un *MOOC* y se sugiere utilizar la información ya recolectada y publicada por sistemas actuales como *edX* o *Coursera* en conjunto con la experiencia obtenida de dictar un curso.

Usuarios e intereses Los cursos *MOOC* dependen mucho de la publicidad y presentación para asegurar suficientes inscritos. Un trabajo necesario es determinar los cursos o contenidos que serían interesantes de presentar en este formato. Una investigación sobre los intereses de los estudiantes del *DI USM* puede ayudar a definir posibles cursos o contenidos populares. También es necesario contrastar estos resultados con los cursos presentes en otros proveedores *MOOC* y consultar con personas fuera del área informática. Adicionalmente se debe definir algún tipo de método para adaptar estos ramos informáticos a todo tipo de estudiantes, sin asumir que poseen todos los conocimientos previos necesarios para cursar y aprobar el curso. El objetivo general consistiría en determinar posibles ramos para dictar en formato *MOOC* y definir un procedimiento que los adapte a una mayor variedad de público.

Implementaciones Como posible memoria de continuación se tiene la implementación y/o modificación de alternativas a *Open edX*. En este camino se encuentra el adaptar un sistema *LMS*, como *Moodle* o *Sakai*, mediante programación o desarrollo de extensiones con el fin de facilitar el desarrollo de cursos *MOOC*. Como experimento se sugiere crear un sitio *web* que sirva de página inicial y que cumpla con los objetivos de publicar cursos y sus descripciones, creación de cuentas de usuario y manejo de certificados de aprobación. El objetivo general consistiría en adaptar una plataforma *LMS* para servir cursos *MOOC*.

Taller Por último se sugiere la creación de un taller o ramo libre cuyo objetivo sea, una vez definido el mecanismo para servir cursos en formato *MOOC*, ayudar mediante un curso introductorio a estudiantes y ayudantes sobre los aspectos fundamentales de los cursos *MOOC*. Temas como la creación de videos y material educacional, métodos de evaluación y planificación de clases pueden potenciar el aporte de ayudantes en la gestión de cursos. Se puede solicitar como proyecto que grupos de estudiantes creen material para un curso de su interés el que puede ser utilizado para crear un curso completo y posteriormente ofrecerlo en la plataforma que se decida utilizar.

Estos temas son algunas de las numerosas aristas relacionadas con el aprendizaje a distancia y los cursos *MOOC*. Es de esperar que los desafíos pedagógicos y tecnológicos planteados por este formato de cursos a distancia sirvan para impulsar la investigación y desarrollo de nuevas tecnologías y métodos para la computación de alto desempeño en el futuro. Por último, se espera que el desarrollo de cursos *MOOC* permita al *DI USM* colaborar con el objetivo de entregar a la sociedad una educación más inclusiva y acorde a los tiempos modernos en que se desenvuelve.

Bibliografía

- [1] British Educational Communications and Technology Agency. Functional Requirements of Learning Platforms, 2008. sitio web oficial (archivado): <http://webarchive.nationalarchives.gov.uk/20110125093552/http://www.becta.org.uk>.
- [2] Clark Aldrich. *Simulations and the Future of Learning*. Pfeiffer, primera edición, 2003.
- [3] Class Central. By The Numbers: MOOCS in 2015. www.class-central.com/report/moocs-2015-stats. revisado al 29 de marzo de 2017.
- [4] Don McIntosh. Vendors of Learning Management and E-learning Products. *Trimeritus eLearning Solutions Inc*, 2015.
- [5] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000. AAI9980887.
- [6] IETF OAuth Working Group. OAuth 2.0 Internet Draft. tools.ietf.org/html/draft-ietf-oauth-v2-31. revisado al 29 de marzo de 2017.
- [7] IMS Global Learning Consortium. Learning Tools Interoperability™. www.imsglobal.org/specs/ltiv2p0. revisado al 29 de marzo de 2017.
- [8] James Popham. What's Wrong and What's Right with Rubrics. *Educational Leadership*, 55(2):72–75, 1997.
- [9] Jay Cross. An Informal History of eLearning. *On the Horizon*, 12(3):103–110, 2004.
- [10] John Meerts. Course Management Systems. *EDUCAUSE Evolving Technologies Committee*, 2003.
- [11] Michael Gaebel. MOOCs – Massive Open Online Courses. *EUA Occasional Papers*, 2013.
- [12] OpenID Foundation. OpenID: The Internet Identity Layer. openid.net/. revisado al 29 de marzo de 2017.

- [13] Angela C. Shih Paul M. Nissenson. MOOC on a budget: Development and Implementation of a low-cost MOOC at a State University. *122a edición Conferencia Anual American Society for Engineering Education*, 14 al 17 de junio del 2015.
- [14] Phil Hill. Online Educational Delivery Models: A Descriptive View, 2012. Publicación web (revisada al 29 de marzo de 2017): <http://er.educause.edu/articles/2012/11/online-educational-delivery-models--a-descriptive-view>.
- [15] Phillip Dawson. Assessment Rubrics: Towards Clearer and more Replicable Design, Research and Practice. *Assessment & Evaluation in Higher Education*, 2015. DOI:10.1080/02602938.2015.1111294.
- [16] Ryann K. Ellis. A Field Guide to Learning Management Systems. *ASTD Learning Circuits*, 2009.
- [17] Stephen Downes. What Connectivism Is . halfanhour.blogspot.cl/2007/02/what-connectivism-is.html. revisado al 29 de marzo de 2017.
- [18] Universidad Técnica Federico Santa María. Rectoría: Informes de Gestión. www.usm.cl/universidad/rectoria/informes-de-gestion. revisado al 29 de marzo de 2017.
- [19] World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.0. www.w3.org/TR/WCAG20. revisado al 29 de marzo de 2017.

Anexo A

Moodle Alto Rendimiento

Usualmente, con el fin de mejorar el rendimiento de una plataforma web, se utilizan configuraciones especiales de servidores que permiten incrementar el número total de usuarios que pueden utilizar la plataforma simultáneamente minimizando pérdidas apreciables en la calidad del servicio prestado. En el caso de *Moodle* existen tres enfoques particulares sobre los cuales implementar la plataforma que permiten extender las capacidades de la plataforma. El primer enfoque considera utilizar un conjunto de servidores propios para armar un *cluster* de servidores. Los otros dos consideran utilizar los servidores de un tercero mediante la contratación de un servicio externo. Cada uno de estos enfoques tiene ventajas y desventajas respecto de los otros por lo que es relevante entender cada uno para elegir la solución que más se ajuste a la realidad de la persona o institución que busca usar *Moodle*.

A.1. Moodle Cluster

Corresponde a las especificaciones dadas por la página oficial de *Moodle* y su equipo principal de desarrolladores para la instalación de la plataforma sobre un *cluster* de servidores. Estas especificaciones son dadas para cada una de las versiones liberadas con las correspondientes actualizaciones, si es que las hay. La información oficial publicada en docs.moodle.org/32/en/Server_cluster entrega una guía paso a paso para instalar y configurar *Moodle* en este formato, sin embargo no se entrega una configuración recomendada o mínima para el *hardware*. Otros tópicos importantes relacionados con el rendimiento de la plataforma son manejados en secciones propias. El cuadro A.1 agrupa esta información y los enlaces correspondientes.

Cuadro A.1: Tópicos importantes para implementar una instancia *Moodle cluster*.

MCM	docs.moodle.org/32/en/Server_cluster
MCH	docs.moodle.org/32/en/Caching
MPR	docs.moodle.org/32/en/Performance_recommendations
MDRP	docs.moodle.org/dev/Server_clustering_improvements_proposal
Clave	
MCM	<i>Moodle cluster</i> página principal.
MCH	Configuración para uso de caché.
MPR	Recomendaciones de rendimiento.
MDRP	Proposiciones de desarrollo para <i>Moodle cluster</i> .

Fuente: Elaboración propia. Enlaces referidos a la documentación oficial, revisados al 29 de marzo de 2017.

La comunidad y los desarrolladores recomiendan el uso de *Linux Virtual Server* como base para montar *Moodle* para mantener la arquitectura del *cluster* transparente. Mediante esta tecnología se puede interactuar con el *cluster* de servidores como si fuera un único servidor de alto rendimiento. Más información sobre *Linux Virtual Server* se encuentra en la página web oficial www.linuxvirtualserver.org/ revisada al 29 de marzo de 2017.

A.2. Sistemas Cloud

Moode Cloud La organización *Moodle Pty Ltd* ofrece instancias de *Moodle* de acceso a través del navegador para personas o instituciones que no dispongan de los medios para acceder a un *cluster* de servidores. Este servicio a través del navegador se llama *MoodleCloud* y tiene la ventaja que la organización se encarga por su cuenta de la implementación y mantención de los servidores que sostienen las intancias de la plataforma web, liberando a los interesados del costo de armar y mantener su propia infraestructura. *MoodleCloud* cuenta con un sistema de cuentas de pago mensual y ofrece una cuenta gratuita pensada para realizar cursos pequeños o probar *Moodle* por primera vez. La página oficial, moodlecloud.com revisada al 29 de marzo de 2017, indica los planes de pago e información de contacto con alternativas para proyectos de mayor envergadura.

Amazon Web Services La empresa *Amazon* ofrece como servicio servidores virtuales para levantar instancias de software que necesitan alta disponibilidad.

En caso de que la máquina virtual se vea sobrepasada por la demanda de los usuarios, *Amazon* ofrece la posibilidad de contratar recursos adicionales para atender la demanda sin que el sistema web ofrecido se tenga que bajar. Adicionalmente se entrega protección ante posibles ataques del tipo denegación de servicios a los servidores virtuales utilizados. más información sobre *Amazon Web Services*, incluyendo programas de pago y máquinas disponibles, en su sitio web oficial aws.amazon.com revisado al 29 de marzo de 2017. Si se necesita implementar rápidamente *Moodle* con alta disponibilidad y se tienen los recursos se puede utilizar a *Amazon* como alternativa a comprar un servidor propio o utilizar *Moodle Cloud*. La ventaja de este enfoque es que en *Amazon* la institución debe instalar *Moodle* desde cero sobre la máquina por lo que se tiene control de todos los aspectos de la plataforma que se va a servir y cómo está configurada, aspecto que no es posible de realizar en *Moodle Cloud* que está limitado a las opciones disponibles en el menú de la plataforma a través de la sesión en el navegador web. Si tampoco se cuenta con los recursos para realizar la instalación y configuración de *Moodle*, la empresa *Bitnami* ofrece una imagen de *Moodle* lista y configurada para servir directamente en *Amazon Web Services*. Más información en el sitio de ventas online de *Amazon* en aws.amazon.com/marketplace/pp/B0062NGVZM, revisado al 29 de marzo de 2017.

Anexo B

Estándares digitales

A continuación se expone información relevante a las certificaciones y estándares mencionados durante la exposición de este trabajo. En cada sección se indica el nombre del estándar, la institución que lo especifica o soporta y el sitio oficial donde se puede encontrar la especificación.

B.1. Open Source

Nombre:	<i>Open Source Definition</i>
Institución:	<i>Open Source Initiative</i>
Sitio web:	opensource.org

Este estándar define lo que se entiende por *open source software*, concepto derivado de la especificación realizada por el proyecto *Debian* en su documento *Contrato Social de Debian*¹. La definición considera que un software es *open source* si cumple las siguientes condiciones:

1. **Redistribución libre.** Sin restricción para entidades de vender u ofrecer el programa como parte de otra distribución de software. La licencia no debe imponer pagos de regalías por realizar dichas transacciones.
2. **Código fuente.** El programa debe incluir el código fuente y debe poder distribuirse en forma de código fuente además de en forma binaria. En caso de que no se incluya el código fuente, deben existir formas bien publicitadas para obtenerlo con un costo de reproducción razonable, como una descarga de internet. Código ofuscado o como salida de un preprocesador u otro programa similar no es permitido.

¹ Publicado en www.debian.org/social-contract#guidelines (revisado al 29 de marzo de 2017)

3. **Trabajos derivados.** La licencia debe permitir modificaciones y derivaciones del programa, además de una distribución de las modificaciones bajo los mismos términos que el original.
4. **Integridad del código fuente del autor.** La licencia del programa puede restringir la distribución del código fuente al ser modificado sólo si la misma permite la distribución de archivos *patche* junto con el código fuente para permitir modificaciones al programa en tiempo de construcción. La licencia debe permitir explícitamente la distribución de software construido a partir de código fuente modificado. La licencia puede plantear el requerimiento de que estos programas derivados posean un nombre o número de versión distinto.
5. **No discriminación en contra de personas o grupos.** La licencia no debe discriminar a personas o grupos de cualquier tipo.
6. **No discriminación en contra de campos de aplicación.** La licencia no debe restringir el uso del programa a algún campo de aplicación en específico, como negocios o ciencia genética, por ejemplo.
7. **Distribución de licencia.** Los derechos adjuntos al programa deben aplicarse a todos aquellos a los cuales se ha redistribuido el programa sin la necesidad de crear una nueva licencia.
8. **Licencia no debe ser específica para un producto.** Los derechos adjuntos al programa no deben depender del hecho que el programa sea parte de otra distribución. Si el programa es extraído de la distribución y usado o nuevamente distribuido, las entidades a quienes se les redistribuyó el programa tendrán los mismos derechos adjudicados por el programa que viene en conjunto con la distribución original.
9. **Licencia no debe restringir otro software.** La licencia no debe imponer restricciones sobre otros programas distribuidos en conjunto al programa licenciado por la misma.
10. **Licencia debe ser tecnológicamente neutral.** Ninguna prestación de la licencia debe ser definida sobre una tecnología o estilo de interfaz particular.

B.2. LTI

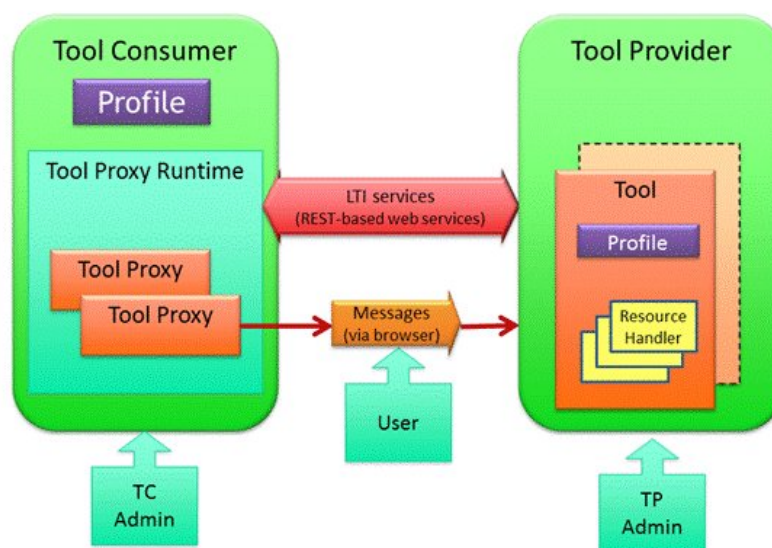
Nombre:	<i>Learning Tools Interoperability</i>
Institución:	<i>IMS Global Learning Consortium</i>
Sitio web:	www.imsglobal.org/activity/learning-tools-interoperability

Esta especificación fue desarrollada para establecer un procedimiento estándar de integración entre aplicaciones de aprendizaje, usualmente servidas remotamente y provistas por desarrolladores de tercera (*third-party*), y plataformas educativas como *LMS*, portales web, repositorios, entre otros. Ambos actores son formalizados como:

- **Herramienta de aprendizaje.** Son las aplicaciones de aprendizaje accesibles por las plataformas.
- **Consumidor de herramientas.** Las diversas plataformas que necesiten los servicios de las aplicaciones.

El objetivo final de *LTI* es permitir a plataformas integrar contenidos de aplicaciones sin la necesidad de mantener interfaces de integración personales para cada plataforma o aplicación. La figura B.1 muestra la arquitectura de una interacción *LTI* tipo. En esta imagen es posible apreciar que la especificación define dos canales de comunicación: el canal de servicio y el de mensajes. El canal de mensaje cubre el caso en que se desea enviar al usuario y otros datos hacia otra aplicación utilizando el navegador, como al utilizar un visor de documentos *pdf*. El canal de servicio permite transacciones utilizando el patrón cliente-servidor. La interacción provista por *LTI* considera la información bajo el formato *JSON*. En términos de seguridad, el proveedor de herramienta delega al consumidor la tarea de autenticar al usuario y espera recibir información sobre el rol y contexto del mismo para la operación. Las comunicaciones de los canales son compatibles con métodos de cifrado como *SSL*. Todos los documentos relacionados al estándar *LTI 2.0* incluyendo documentos dedicados al marco de trabajo, seguridad y otras definiciones se pueden encontrar en [7]

Figura B.1: Arquitectura de una conexión especificada por *LTI 2.0*.



Fuente: Imagen obtenida del documento oficial *Implementación de LTI 2.0*. www.imsglobal.org/specs/ltiv2p0/implementation-guide (revisado al 29 de marzo de 2017).

B.3. SCORM

Nombre:	<i>Sharable Content Object Reference Model</i>
Institución:	<i>Advanced Distributed Learning</i>
Sitio web:	www.adlnet.gov/scorm

La especificación *SCORM* corresponde a un conjunto de otros estándares y especificaciones para la creación de material y contenidos pedagógicos estructurados capaces de ser compartidos entre plataformas educativas diferentes, mientras ambas se adscriban a la especificación. Los requerimientos exigidos por *SCORM* están orientados a enfrentar los siguientes desafíos:

- **Interoperabilidad.** Corresponde a la capacidad de distintos componentes para trabajar en conjunto. Para lograr que plataformas distintas puedan manejar correctamente los contenidos y material pedagógico. La especificación *SCORM Run-time Environment* o *RTE* define modelos de

datos e interfaces de aplicación comunes para contenido relativo al *e-learning*, lo que persigue que la comunicación entre contenidos y componentes de sistema se lleve a cabo en forma estandarizada.

- **Portabilidad.** Uno de los principales desafíos para compartir contenido se presenta cuando las plataformas y sistemas operativos son distintos. La especificación *SCORM Content Aggregation Model* o *CAM* define el procedimiento para empaquetar contenido para su intercambio entre distintos sistemas utilizando un tipo de archivo transferible en formato *ZIP* llamado *Package Interchange Format* o *PIF*. Este tipo de empaquetado permite un mecanismo de portabilidad estandarizada entre ambientes de aprendizaje distintos.
- **Reusabilidad.** Corresponde al desafío de utilizar contenidos y material ya creado para nuevos usos en vez de realizar el trabajo de reproducir nuevamente el material original. La especificación *SCORM Content Aggregation Model* o *CAM* descrita anteriormente también describe los componentes utilizados en la experiencia de aprendizaje y cómo describir estos componentes para permitir su búsqueda y descubrimiento, promoviendo la reutilización del contenido pedagógico a través de los sistemas adscritos a la especificación como *LMS* y repositorios de contenido. Para asegurar que los contenidos mantienen las propiedades señaladas la especificación describe los requerimientos y responsabilidades necesarias para crear y organizar dichos contenidos, como cursos, módulos, tareas, entre otros. Además incluye instrucciones sobre la aplicación de metadatos a los contenidos creados y compartidos. Por el lado del servidor, la especificación exige que se puedan importar contenidos descritos según la especificación.
- **Secuenciación.** Además de proveer los contenidos es necesario determinar las secuencias en que estos se entregan a los estudiantes con el fin de apoyar el proceso pedagógico. La especificación *SCORM Sequencing and Navigation* o *SN* en conjunto con la especificación *CAM* Describen la forma en que contenido bajo la especificación *SCORM* es entregado a estudiantes mediante eventos de navegación que pueden estar determinados por un conjunto estructurado de reglas. Las reglas de secuenciado de *SCORM* definidas en la especificación del 2004 permite a diseñadores de cursos y creadores de contenido especificar el orden en que objetos de contenido compartido o *SCOs* son entregados a los estudiantes. Un *SCO*, del inglés *sharable content objects*, corresponde a la pieza de contenido más pequeña que permite el seguimiento de progreso de un estudiante.

El cuadro B.1 indica las referencias para cada una de las especificaciones mencionadas que conforman *SCORM*. La compañía *iSpring*, creadora de software para creación de cursos y lecciones, realizó una lista con los principales sistemas *LMS* compatibles con *SCORM* para el intercambio de contenido pedagógico. La lista puede encontrarse en www.ispringsolutions.com/supported-lms (revisado al 29 de marzo de 2017).

Cuadro B.1: Recursos especificaciones *SCORM*.

<i>SCORM RTE</i>	www.adlnet.org/wp-content/uploads/2013/09/SCORM_2004_Overview.pdf
<i>SCORM CAM</i>	www.adlnet.gov/wp-content/uploads/2013/09/SCORM_2004_4ED_v1_1_CAM_20090814.pdf
<i>SCORM SN</i>	http://www.adlnet.gov/wp-content/uploads/2013/09/SCORM_2004_4ED_v1_1_SN_20090814.pdf

Fuente: Página oficial de *ADL Net* www.adlnet.gov. Recursos y descripciones de enlaces revisadas al 29 de marzo de 2017.

B.4. Mozilla Open Badges

Nombre:	<i>Open Badges</i>
Institución:	<i>Mozilla & IMS Global Learning Consortium</i>
Sitio web:	openbadgespec.org

Open Badges es una especificación creada originalmente por la fundación *Mozilla* para la comunicación de habilidades y logros mediante el uso de símbolos visuales empaquetados en conjunto con información verificable y evidencia de respaldo que puede ser compartida a través de internet. Actualmente el desarrollo de la especificación es supervisado por la organización *IMS Global Learning Consortium*. La especificación *Open Badges* define para la creación de los paquetes de información tres clases de datos fundamentales: *Assertions*, *BadgeClasses* y *Profiles*. Un conjunto de expresiones de estos tres tipos de datos permite la construcción de una medalla válida la cual es almacenada y distribuida como información mediante el formato *JSON-LD*². Adicionalmente, la especificación incluye definiciones sobre verificación, validación e implementación de este tipo de datos. El proyecto *Badge the World* busca registrar y publicar proyectos e iniciativas notables alrededor del mundo que se adhieren a la especificación de *Open Badges*. Más información sobre sistemas y otros proyectos importantes en la página oficial www.badgetheworld.org.

B.5. OLX

Nombre:	<i>Open Learning XML</i>
Institución:	<i>edX Inc.</i>
Sitio web:	open.edx.org/blog/open-learning-xml-olx-format

²Método de serialización de datos formalizado por el *W3C*. La referencia oficial se encuentra en www.w3.org/TR/json-ld (revisado al 29 de marzo de 2017).

Open Learning XML o *OLX* es un estándar en desarrollo impulsado por la institución *edX Incorporated*. Nace como una herramienta *XML* para definir cursos y contenido para la plataforma *Open edX* con el objetivo de permitir:

- Mover contenido educacional entre instancias distintas de *Open edX*.
- Crear cursos y contenido educacional fuera del componente *Studio* además de convertir contenido existente en otros formatos.
- Mantener el contenido portable y libre de codificación específica o propietaria.

Debido a que el estándar está en desarrollo no cuenta con una especificación formal completa. La información oficial sobre *OLX* se encuentra en el sitio de documentación edx.readthedocs.io/projects/edx-open-learning-xml.

Anexo C

Plataformas Alternativas

A la fecha de este trabajo existe gran variedad en oferta para sistemas del tipo *LMS*. Muchos proyectos *open source* forman empresas y ofrecen versiones pagadas de las mismas plataformas libres que crean incorporando soporte técnico y mayor funcionalidad. Los tres sistemas seleccionados como alternativas son utilizados por gran cantidad de usuarios en varios contextos diferentes, por ejemplo *Sakai* es el sistema institucional de varias universidades alrededor del mundo, *ATutor* es uno de los sistemas precursores en acceso al contenido para discapacitados cumpliendo con las especificaciones de accesibilidad definidas por el *W3C*, y *Dokeos* es un sistema ampliamente utilizado en el mundo empresarial para entrenamiento y capacitación de empleados apoyando a empresas importantes del ámbito tecnológico como *Red Hat*. A continuación se presentan fichas de descripción de software para cada una de las plataformas alternativas.

C.1. Sakai

Cuadro C.1: Perfil sistema *Sakai LMS*.

<i>Sakai</i>
★ Fundador: Universidad de Michigan. Universidad Indiana. Universidad de Stanford. Universidad de California.
★ Institución: Fundación Apereo.
★ Sitio oficial: sakaiproject.org
★ Distribución: <i>Educational Community License - 2.0</i> .
★ Tipo de plataforma : Sistema web, <i>Open source</i> .
★ Lenguaje base: <i>Java</i> versión 1.8
★ Otros: <i>Apache Maven</i> versión 3.2
★ Características notables: Portabilidad plataforma. Apto para alto número de usuarios. Gestión masiva de archivos. Extenso soporte servicios web, <i>REST</i> , <i>SOAP</i> . Soporte <i>SCORM</i> , <i>LTI</i> , material multimedia. Sistema de planilla propio para gestión de notas.

Fuente: Elaboración propia. *Apache Maven* es utilizado para construir el sistema y su documentación. Más información sobre esta herramienta en el sitio oficial maven.apache.org revisado al 29 de marzo de 2017.

El proyecto *Sakai* tiene como repositorio oficial github.com/sakaiproject/sakai revisado al 29 de marzo de 2017. En este repositorio se puede encontrar información sobre el estado actual de las versiones más recientes, así como ver los problemas que se están resolviendo en la plataforma. La documentación oficial del proyecto se encuentra en el sitio confluence.sakaiproject.org revisado al 29 de marzo de 2017.

C.2. ATutor

Cuadro C.2: Perfil sistema *ATutor LMS*.

<i>ATutor</i>
★ Fundador: Inclusive Design Research Centre
★ Institución: Universidad OCAD, Canadá.
★ Sitio oficial: www.atutor.ca
★ Distribución: <i>GNU General Public License</i> .
★ Tipo de plataforma : Sistema web, <i>Open source</i> .
★ Lenguaje base: <i>PHP</i> versión 5.0.2
★ Otros: <i>MySQL</i> versión 4.1.10 o más
★ Características notables: Red social local integrada a la plataforma. Red social permite compartir contenido. Múltiples herramientas navegación. Enfoque en accesibilidad para personas discapacitadas. Instalación y actualización automatizada. Soporte <i>SCORM</i> , <i>LTI</i> , material multimedia. Enfoque seguridad mediante cifrado datos y contenido.

Fuente: Elaboración propia. El sitio web oficial recomienda utilizar *Apache web server* como servidor web de producción.

Al igual que *Sakai*, *ATutor* fue migrado a un repositorio en *GitHub* aunque originalmente era mantenido en un repositorio *SVN* público. El código fuente puede encontrarse en github.com/atutor/ATutor/ junto con la documentación oficial del sistema. El sitio web oficial de *ATutor* cuenta con una instancia web de demostración donde se puede experimentar las funcionalidades ofrecidas por la plataforma. Este sitio de prueba se encuentra en www.atutor.ca/atutor/demo.php y permite probar tanto las funcionalidades académicas, como las administrativas.

C.3. Dokeos

Cuadro C.3: Perfil sistema *Dokeos LMS*.

<i>Sakai</i>
★ Fundador: Thomas De Praetere (bajo nombre original <i>Claroline</i>).
★ Institución: Dokeos SPRL.
★ Sitio oficial: www.dokeos.com .
★ Distribución: <i>GNU General Public License</i> .
★ Tipo de plataforma : Sistema web, <i>Open source</i> .
★ Lenguaje base: <i>PHP</i> versión 5.3+
★ Otros: <i>MySQL</i> versión 5.1+.
★ Otros: <i>Apache web server</i> versión 2.0
★ Características notables: Compatible con software externo mediante archivos <i>CSV</i> . Múltiples herramientas de trabajo colaborativo. Interface visual con sistema de pestañas. Editor de documentos integrado. Sistema de glosario para cursos. Herramienta de diagramas <i>mindmap</i> .

Fuente: Elaboración propia. La versión *open source* de la plataforma se puede descargar de www.dokeos.com/dokeos-community-edition (revisado al 29 de marzo de 2017).

La plataforma *Dokeos* posee un enfoque empresarial fuerte. Incluye la mayoría de las características de los sistemas *LMS* más tradicionales, como *Moodle* o *ATutor*, colocando énfasis en la presentación. La interfaz está pensada para seguir el esquema de uso de un navegador web con pestañas apelando a la familiaridad de los usuarios. Cuenta con herramientas de autoría simplificadas que permiten a los profesores redactar documentos y crear varios tipos de material multimedia dentro del sistema. El problema que tiene la plataforma es que resulta difícil encontrar documentación debido a que está dispersa por varios sitios.

Anexo D

Herramientas Gestión Código

Se procede a describir las herramientas relacionadas a la gestión de código y proyectos informáticos mencionadas en el contexto de la sección 6.4. Estas herramientas cumplen distintos objetivos y no necesariamente tienen relación entre ellas. Caso especial es el de las herramientas *JIRA* y *Confluence* que fueron creadas por la misma organización con el objetivo de trabajar en conjunto. Los enlaces oficiales referidos en este anexo han sido revisados a la fecha de 29 de marzo de 2017.

D.1. Github

Corresponde a un sistema web para realizar almacenamiento y control de versiones de código fuente. Fundada en 2008, la organización del mismo nombre a cargo del sistema web tiene sus oficinas principales en San Francisco, California y es dirigida por Chris Wanstrath, su *CEO*. El sistema consiste en una interfaz para la operación de repositorios *git*, tecnología creada por *Linus Torvalds* para apoyar el desarrollo del *Linux Kernel*. La página oficial de *git* es git-scm.org.

Características La interfaz provista por *GitHub* destaca en la inclusión de varias funciones adicionales a las capacidades de *git*, orientadas a facilitar el trabajo colaborativo. Entre estas funciones destacan:

- Control de acceso a los repositorios.
- Reporte y seguimiento de errores.
- Gestión de tareas.
- Sistema de solicitud para pedir funciones y características.
- Sistema de *Wiki* para los proyectos.

Característica importante de la plataforma es que no impone restricción a los programas alojados en ella para que cumplan las condiciones impuestas por la definición de *open source* (ver anexo B). Se recomienda revisar las condiciones de licencia particulares a cada programa en *GitHub* antes de descargarlo. *GitHub* permite cuentas sin costo y ofrece planes para empresas para proyectos de desarrollo a gran escala. Más información en el sitio web oficial github.com

D.2. JIRA

Es un sistema de seguimiento de incidentes propietario desarrollado por *Atlassian*, empresa australiana fundada el año 2002 y actualmente localizada en Sydney, Australia. *JIRA* es reconocido por sus funciones de seguimiento y reporte de errores, seguimiento de incidentes y gestión de proyectos. *JIRA* es ofrecida en tres paquetes e incluye soporte para desarrollo de *software* bajo metodologías ágiles, integrando herramientas para definir y reportar iteraciones, entre otras cosas. Posee soporte para sistemas de control de versiones, como *CVS*, *git*, *Mercurial* y *Subversion*, además de incluir soporte para interfaces de servicio web en protocolos *REST*, *SOAP* y *XML-RPC*. *Atlassian* ofrece este servicio sin costo a proyectos *open source* que cumplan con criterios específicos definidos en su página oficial. Más información en el sitio oficial de *Atlassian* www.atlassian.com/software/views/open-source-license-request y el sitio oficial de *JIRA* www.atlassian.com/software/jira.

D.3. Confluence

Sistema creado por *Atlassian* para facilitar la colaboración de equipos de trabajo. El primer *release* oficial fue el 2004 en su versión 1,0. *Confluence* utiliza el concepto de *wiki* para presentar y administrar los contenidos incorporados por los usuarios. En la actualidad ofrece gran cantidad de funcionalidades como integración con redes sociales, formato de contenido utilizando *CSS*, integración de contenido multimedia e integración con el sistema de reporte y seguimiento de incidentes de *JIRA*.

El objetivo principal tras el sistema es reunir en un solo sitio las tareas más relevantes de un desarrollo, como reporte de errores, documentación del sistema, tareas y planificación del proyecto, encargados y grupos de trabajo, entre otros.

El sistema es pagado y se ofrece como servicio en formato de nube a través de los servidores de *Atlassian*, aunque también se puede comprar una distribución para instalar en servidores propios. Información adicional en el sitio oficial www.atlassian.com/software/confluence.

D.4. Django Admin

Django es un *web framework* para el lenguaje *Python* y una de las características que lo destacan es que trae incorporado un sistema básico por omisión para gestionar el contenido del sitio web desarrollado. Este sistema se llama *admin* y funciona en base al esquema que define el *framework* para trabajar con los elementos que componen un sistema *web*. La aplicación *admin* consiste en una interfaz *web* configurable donde se presenta información contenida en la base de datos del sitio, por ejemplo información sobre cuentas de usuario. Esta información es obtenida mediante el uso de *metadatos* definidos en los archivos del sitio correspondientes al modelo de datos. Para más información sobre esta herramienta y otras características de *Django*, visitar el sitio oficial de documentación en docs.djangoproject.com.

D.5. Tar

Corresponde a un programa para reunir varios archivos en uno solo para diversos fines, como distribución o respaldo. Actualmente es soportado por una gran cantidad de sistemas operativos y se encuentra estandarizado en la especificación *POSIX* definida por el *IEEE*. Información detallada sobre esta aplicación, como documentación y distribuciones, se encuentra en la página oficial de la implementación creada por *GNU* en www.gnu.org/software/tar.