



Facultad 4

Integración de un mundo virtual diseñado en OpenSim con la plataforma educativa Moodle.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores

José Rafael Aguilera Zaldivar

Eilder Jorge García

Tutor

MSc. Noralbis De Armas Rodriguez

Co-Tutor

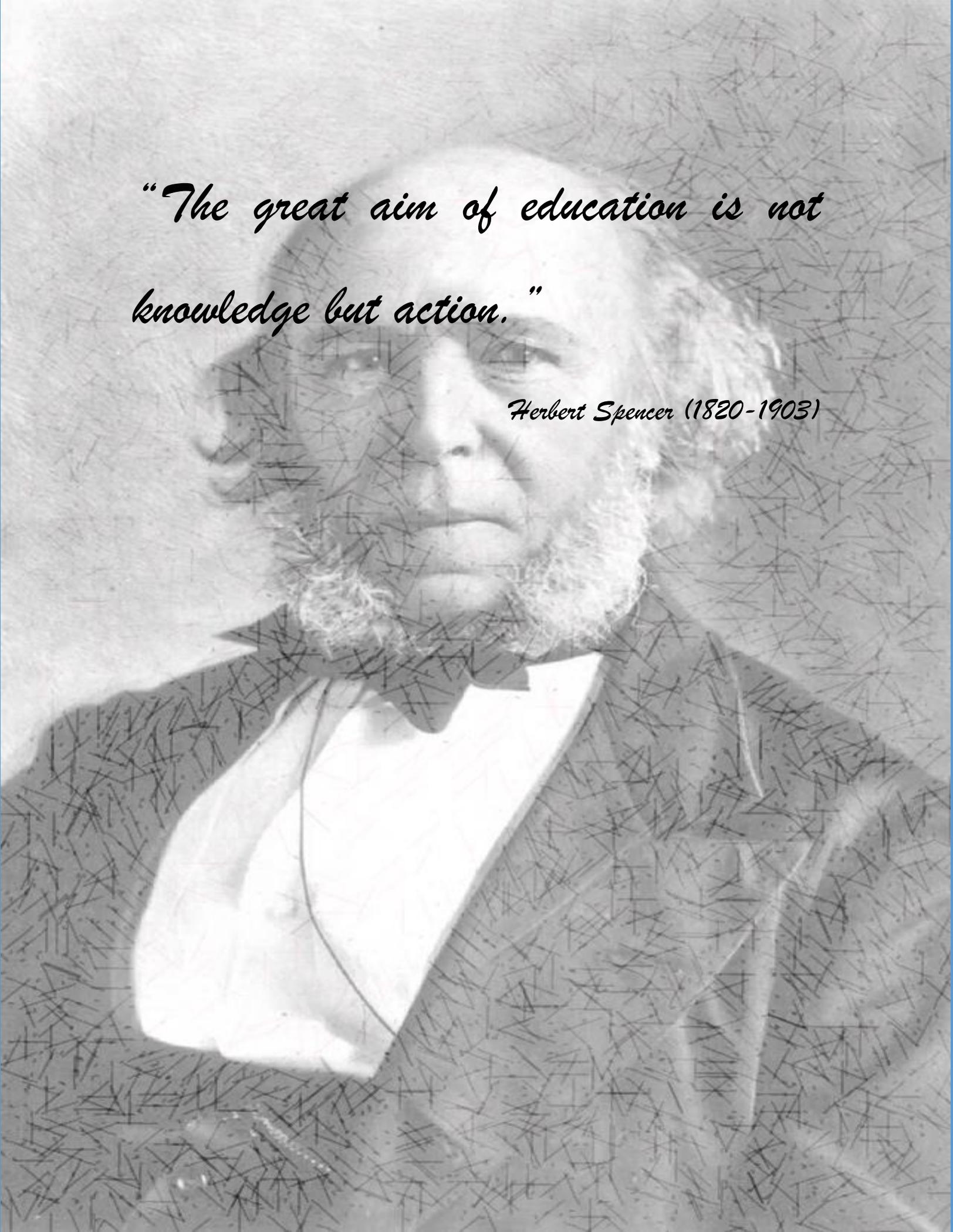
Ing. Enelis Blanca Cuba Rondón

La Habana. Cuba

2014 – 2015

*"The great aim of education is not
knowledge but action."*

Herbert Spencer (1820-1903)



DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año _____.

Eilder Jorge García

José Rafael Aguilera Zaldivar

Firma del autor

Firma del autor

MSc. Noralbis De Armas Rodriguez

Firma del tutor

Dedicatoria:

A esas personas importantes en nuestras vidas, que siempre estuvieron listas para brindarnos toda su ayuda, ahora nos toca regresar un poquito de todo lo inmenso que nos han otorgado. Con todo nuestro cariño esta tesis se las dedicamos a ustedes:

Nuestros padres, familia y amigos.

Agradecimientos

Rafael

Gracias a la vida por darme la oportunidad de contar con las personas que de una forma u otra me han apoyado y hecho una mejor persona a lo largo de estos 25 años y permitirme tener un montón de personas bellas a las cuales agradecerles por haber llegado hasta donde estoy y siendo quien soy.

- ✓ A mi mamá y mi papá por ser dos de las personas más importantes en mi vida, por ser mi apoyo, por tantos años de sacrificio y esfuerzo, por haber dado lo mejor de sí. A ellos les debo quien soy, este trabajo también es de ustedes.
- ✓ A mi esposa Yaque por haberme dado la bebé tan hermosa que tenemos y por permitirme tener una vida junto a ellas, juntos en los buenos y malos momentos.
- ✓ A mi hermana Any y a su esposo que también han aportado muchas experiencias a todos estos años, y a mis sobrinos Aaron y Allison por hacerme reír tanto.
- ✓ A mi abuelito por querernos tanto a todos en la familia.
- ✓ A mis primos(as), tíos(as) y toda mi familia en general por estar siempre presente en todos los momentos de mi vida.
- ✓ A mi compañero de tesis, Eilder, porque sin el trabajo en equipo no habiéramos llegado al logro de la meta, ser ingenieros.
- ✓ A mis amigos(as) de siempre, a todos los que seguimos juntos desde el círculo infantil y a todos aquellos que han tenido que irse por caminos diferentes.
- ✓ A todos los amigos(as) que hice en el grupo 4101: Nunca los olvidaré.
- ✓ A todos los amigos(as) que hice en el nuevo grupo que tuve desde 3ro: Nunca los olvidaré.
- ✓ A todos los demás amigos(as) que hice en todos estos años de universidad.
- ✓ A mis tutores Maikel y Noralbis por tanta preocupación, por ayudarnos y guiarnos.
- ✓ A los profesores que pusieron su grano de arena para que sea quien soy hoy.
- ✓ A todas las personas que de una forma u otra contribuyeron a que hiciera realidad el ser ingeniero y un mejor hombre.

Muchas gracias.

Eilder

Eilder Jorge García: Agradezco primeramente a mi mamá y a mi papá por traerme a este mundo y darme todo su apoyo, dándome la posibilidad de continuar con mis estudios hasta el día de hoy. Agradezco enormemente a mi padrastro por su paciencia y condescendencia, has sido más allá de excelente. También quiero agradecer a mi abuela por su cariño y sus cuidados, te extraño un montón, abuela Mirtha. A mis tíos y mis primos, a pesar de que rara vez los veo, los quiero en el alma. A mi mejor y más viejo amigo, Raúl, por siempre inspirarme a superarme a mí mismo y llegar a cimas cada vez más altas. A todos mis amigos y compañeros, estos años de estudio no hubieran sido lo mismo sin ustedes, agradezco su apoyo y paciencia. Y finalmente, pero no por ello menos importante, agradezco a Rafael, mi compañero de tesis, por darme la paciencia y voluntad para proceder con esta investigación en sus momentos más duros. Hay muchos otros que debería agradecer, pero el espacio es corto, a todos los restantes, les agradezco su apoyo y espero poder algún día hacer por ustedes lo que han hecho por mí.

Muchas gracias.

Resumen

El uso de los mundos virtuales en la educación es ampliamente aceptado a nivel mundial, reconociendo en la misma inmensas capacidades que potencian el aprendizaje como la sensación de inmersión y presencia dentro del mundo tridimensional sumado a la posibilidad de experimentar, explorar y realizar simulaciones en tiempo real de forma conjunta con otros estudiantes compartiendo recursos online como videos, audio, textos o imágenes. La plataforma Moodle está dedicada a la educación basando su funcionamiento en la gestión de cursos creados por los profesores, ayudándolos a concebir comunidades de aprendizaje en línea, por lo que es muy usada actualmente en la Universidad de las Ciencias Informáticas y a nivel mundial. El Departamento de Ciencias Sociales y Humanidades, y el Centro Vertex de la Facultad 5, se encuentra desarrollando un mundo virtual para el idioma Inglés diseñado en OpenSim, nombrado “*Virtual English*”, el cual ha presentado dificultades al interactuar con muchos de los recursos de Moodle. Es por ello que el presente trabajo consiste en lograr la interacción de un mundo virtual con la plataforma Moodle.

Palabras clave: 3D, Metaverso, Moodle, Mundo virtual y Sloodle.

Abstract

*The use of virtual worlds in education is widely accepted worldwide, recognizing the capacities that improves learning such as the feeling of immersion and the presence inside the tridimensional world alongside the possibility of experimenting, exploration and the realization of real time simulations together with other students sharing online resources such as videos, audio, text or images. The Moodle platform is dedicated to education and its core capacity is the management of courses created by teachers, helping them create online learning communities; this is the reason why it's very used nowadays in the Universidad de las Ciencias Informáticas and worldwide. The Department of Social Science and Humanities along with Vertex, a center of work in the Faculty 5 is developing a virtual world for the learning of the English language designed in OpenSim and named “*Virtual English*”; this world has presented several difficulties interacting with many of the resources offered by Moodle. It is because of all these advantages that the following paperwork consist on the integration a virtual world with the Moodle platform.*

Keywords: 3D, Metaverse, Moodle, Virtual World and Sloodle.

ÍNDICE

Introducción	1
Capítulo: 1 Fundamentación teórica	7
1.1 Introducción	7
1.2 Metaversos y los mundos virtuales.....	7
1.3 Mundos Virtuales	8
1.3.1 Mundos virtuales de Second Life	10
1.3.2 Mundos Virtuales de OpenSim	10
1.4 LMS	11
1.4.1 Plataforma educativa Moodle	12
1.5 Integración entre Moodle y Mundo Virtual	13
1.6 Biblioteca Sloodle.....	14
1.7 Análisis de soluciones existentes	14
1.7.1 <i>Virtual Laboratory for Algorithms:</i>	15
1.7.2 Universidad de St. Andrews:	16
1.7.3 Facultad de Ingeniería:.....	17
1.7.4 <i>LEARN-IN-3D:</i>	18
1.7.5 UCIGRID:.....	18
1.7.6 <i>Virtual English:</i>	18
1.8 Metodologías de desarrollo de <i>software</i>	19
1.8.1 Proceso unificado ágil (AUP).....	19
1.8.2 XP (eXtreme Programming)	21
1.8.3 SCRUM.....	22
1.8.4 SXP (SCRUM y eXtreme Programming)	23
1.8.5 Valoración de selección.....	24
1.9 Conclusiones Parciales	24

Capítulo: 2	Exploración, Planificación y Diseño de la solución propuesta	25
2.1	Introducción	25
2.2	Herramientas de Ingeniería de <i>Software</i> Asistida por Computadora (CASE).....	25
2.2.1	Visual Paradigm	25
2.3	OpenSim	26
2.4	Lenguajes y tecnologías de desarrollo	27
2.4.1	Lenguaje de programación C#	28
2.4.2	<i>Linden Scripting Language</i>	28
2.4.3	Lenguaje de programación PHP	29
2.4.4	Sloodle.....	29
2.5	Servidores Web.....	31
2.6	Cliente - Visor.....	32
2.7	Propuesta de solución.....	34
2.7.1	Personal relacionado con el mundo virtual	34
2.8	Proceso de captura de funcionalidades del sistema	35
2.8.1	Lista de Reservas del Producto.....	35
2.8.2	Características del sistema	36
2.9	Exploración	37
2.9.1	Historias de usuarios.....	37
2.10	Planificación.....	39
2.10.1	Estimación de esfuerzos por historia de usuario	40
2.10.2	Iteraciones	40
2.10.3	Plan de Entregas.....	41
2.11	Diseño.....	42
2.11.1	Mundo Virtual	42
2.12	Conclusiones parciales	44

Capítulo: 3 Implementación y prueba de la solución propuesta	45
3.1 Introducción	45
3.2 Descripción de la Arquitectura.....	45
3.2.1 Moodle	45
3.2.2 OpenSim.....	46
3.3 Implementación.....	47
3.3.1 Primera Iteración.....	47
3.3.2 Segunda Iteración	49
3.3.3 Tercera Iteración	50
3.4 Patrones de diseño	50
3.4.1 Patrones GRASP	51
3.4.2 Patrones GoF.....	52
3.5 Diagrama de despliegue	57
3.6 Pruebas.....	59
3.6.1 Pruebas de aceptación.....	60
3.7 Conclusiones parciales	61
Conclusiones generales.....	63
Recomendaciones	64
Glosario de Términos.....	65
Bibliografía.....	67
Anexos	71

Introducción

En la actualidad las personas a nivel mundial viven en constante interacción con los mundos virtuales surgidos de la “Web 2.0”, donde la participación y la colaboración son la base de generación del conocimiento colectivo. En este contexto una de las tecnologías emergentes en los últimos años son los metaversos, mundos virtuales tridimensionales e inmersivos que permiten la interacción en tiempo real, la telepresencia y la creación de contenidos por parte de los usuarios.

La invención de los metaversos, la cual tributa a la educación e investigación en el siglo XXI, surge a partir del año 2003, pero el término posee sus orígenes en la novela Snow Crash publicada en 1992 por Neal Stephenson. Los humanos interactúan dentro de los metaversos social y económicamente como íconos (avatares¹) a través de un soporte lógico en un ciberespacio² que actúa como una metáfora del mundo real (Carlos, 2010).

Los mundos virtuales tienen muchas aplicaciones en diferentes esferas como: el entretenimiento donde apareció por primera vez, la industria militar, el sector de la salud, la educación, entre otras. En cuanto al entretenimiento está enfocado a los juegos de participación colectiva, roles diversos y estrategias. En la industria militar se utiliza para desarrollar la toma de decisiones, práctica de diferentes armamentos, artillería naval y en la fuerza aérea. En el sector de la salud posee también grandes aplicaciones como el estudio de la anatomía virtualmente, prácticas médicas en situaciones controladas y la rehabilitación. Además en los últimos años sus aplicaciones en la educación ha aumentado considerablemente ya que la educación a distancia ha tenido gran auge, además muchas universidades han tomado esta iniciativa para que sus propios estudiantes tomen un nivel superior de motivación para el aprendizaje (Rodríguez, 2011).

El uso de los mundos virtuales en la educación es ampliamente aceptado a nivel mundial (Según: Horizon Report³, SimTeach⁴, New Media Consortium⁵ en el año 2011), reconociendo en la misma amplias capacidades que potencian el aprendizaje como la sensación de inmersión y presencia dentro del mundo tridimensional sumado a la posibilidad de experimentar, explorar y realizar simulaciones en tiempo real de forma

¹ Personas virtuales que representa a un individuo real en el metaverso.

² Ámbito artificial creado por medios informáticos.

³ <http://net.educause.edu/ir/library/pdf/HR2011.pdf>

⁴ <http://www.simteach.com>

⁵ <http://www.nmc.org/initiatives/immersive-learning/more>

Introducción

conjunta con otros estudiantes compartiendo recursos *online* como videos, audio, textos o imágenes. Debido a esto la experiencia de aprendizaje dentro de los mundos virtuales es una forma muy efectiva de enseñar cualquier tipo de proceso o procedimiento específico mediante simulaciones y juegos serios combinados con elementos más tradicionales como la lectura de textos, videos o conferencias sobre determinados temas en tiempo real y de forma inmersiva.

Los mundos virtuales brindan la oportunidad de explorar nuevos métodos para la formación y el aprendizaje, proponiendo una nueva dimensión a la experiencia de alumnos y docentes en el universo del aprendizaje, ya que el diseño del propio mundo resulta motivador al sumar el aspecto lúdico al objeto mismo del conocimiento fomentando a la vez la confianza y el apoyo mutuo entre sus usuarios:

- a) Mediante diferentes tipos de juegos y competiciones, así como otras actividades de aprendizaje social.
- b) Mediante la organización de eventos y actividades que estimulen a los alumnos a integrarse con el entorno de aprendizaje, que posibiliten la interacción y el aprender unos de otros.

Como consecuencia a todas las posibilidades que brindan, los mundos virtuales para la educación, aumentan cada vez más las instituciones educativas que abogan por esta solución a nivel mundial. España es uno de los países pioneros en la inclusión de los mundos virtuales para la formación y aprendizaje, la compañía Vértice⁶ que radica en este mismo país ha creado una metodología basada en el *e-learning* 3D llamada ONROOM⁷ cuyo eslogan es “*Estás en casa pero también estás en clases*”, el cual define dos de los principios de su metodología, disponibilidad e interacción.

En Cuba, país donde los autores de esta tesis realizan la investigación, también se ha ido avanzando en el desarrollo de mundos virtuales enfocados a la educación. Es por ello que la Universidad de las Ciencias Informáticas (UCI) tiene como estrategia para darle cumplimiento a sus objetivos en el proceso de formación proveer herramientas y medios a los estudiantes para que alcancen el máximo desarrollo en sus estudios y estén en condiciones de ofrecer servicios diversos a la sociedad.

Una de las alternativas con las que cuenta el Centro de Informática Industrial (CEDIN) de

⁶ <http://www.vertice.org>

⁷ <http://www.vertice.org/metodologia.html>

Introducción

la UCI para los procesos formativos y productivos, está relacionada con la simulación de ambientes reales, que pueden ser utilizados en el proceso de aprendizaje.

Por su parte el Departamento de Ciencias Sociales y Humanidades, y el Centro Vertex – Entornos Interactivos 3D de la Facultad 5, se encuentra desarrollando un entorno virtual para el idioma Inglés diseñado en OpenSim⁸, nombrado “*Virtual English*”, como vía alternativa para el aprendizaje y auto superación de los interesados en este idioma.

Es en consecuencia a estos adelantos e iniciativas para mejorar el aprendizaje que algunas tendencias y tecnologías como los LMS (del inglés *Learning Management Systems*) se han inclinado en el uso de estos tipos de mundos virtuales para fomentar el aprendizaje de los contenidos interactivos que en ellos se exponen. Los LMS permiten planificar el aprendizaje de acuerdo a las necesidades de los usuarios, sean estos estudiantes o trabajadores, mejorar sus competencias y su intercomunicación. Es posible adaptar la formación a los requisitos de cada uno y al propio desarrollo profesional; permite la distribución de cursos, recursos, noticias y contenidos relacionados con la formación en general (Mayor, 2012).

Dentro de los LMS se encuentra Moodle (del inglés, *Modular Object-Oriented Dynamic Learning Environment* / Entorno Modular de Aprendizaje Dinámico Orientado a Objetos), el cual permite crear cursos que contienen actividades y recursos, existen cerca de 20 tipos de actividades: foros, glosarios, wikis, tareas, cuestionarios, encuestas, reproductores SCORM⁹ (del inglés *Sharable Content Object Reference Model*), bases de datos, entre otras, y cada una de ellas puede ser adaptada. La potencia de este modelo basado en actividades viene dada al combinar estas en secuencias y grupos, lo que permite guiar a los participantes a través de caminos de aprendizaje; por ende cada actividad puede apoyarse en los resultados de la anterior. Moodle es uno de los sistemas que ha potencializado la interacción con los mundos virtuales en 3D, mediante la implementación de módulos o funcionalidades que le permitan integrar sus materiales didácticos con los mundos virtuales.

Ante las inmensas posibilidades que presentan poder interactuar con los recursos de Moodle desde un mundo virtual diseñado en OpenSim, se hace necesario investigar y desarrollar herramientas y contenidos más adecuados para llevar a cabo de manera más

⁸ Servidor 3D de código abierto que permite crear ambientes virtuales.

⁹ SCORM es un conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados.

Introducción

eficaz los espacios para la educación universitaria. Es por ello que se han realizado investigaciones para integrar estos dos entornos pero no se ha logrado completamente. Además algunos de los avances realizados en el tema se han hecho bajo versiones de Moodle muy antiguas. Por todo lo expresado anteriormente se decidió buscar la interacción del Moodle 2.8, la versión más actual estable en el momento de la investigación, con un mundo virtual diseñado en OpenSim. Así se le dará solución a los problemas que actualmente presentan los proyectos de la Universidad relacionados con el desarrollo de mundos virtuales que necesitan la interacción para un mejor funcionamiento de estos mundos.

Debido a la situación problemática antes planteada se propone como **problema a resolver**: ¿Cómo interactuar desde un mundo virtual diseñado en OpenSim con Moodle en su versión 2.8.x en la UCI?

Se tiene como **objeto de estudio**: los mundos virtuales diseñados en OpenSim. Se define como **objetivo general**: desarrollar la integración entre Moodle en su versión 2.8.x con un mundo virtual diseñado en OpenSim en la UCI y como **campo de acción**: integración de un mundos virtual diseñado en OpenSim con Moodle en su versión 2.8.x en la UCI.

Preguntas Científicas:

- ¿Cómo sistematizar los fundamentos teóricos relacionados con los mundos virtuales?
- ¿Cómo desarrollar la integración de un mundo virtual diseñado en OpenSim con Moodle en su versión 2.8.x?
- ¿Cómo implementar la integración de un mundo virtual diseñado en OpenSim con Moodle en su versión 2.8.x?
- ¿Cómo probar la integración de un mundo virtual diseñado en OpenSim con Moodle en su versión 2.8.x?

Tareas a resolver:

- Revisión bibliográfica para generar el marco teórico conceptual sobre los mundos virtuales.
- Estudio de las soluciones similares.
- Estudio de las herramientas y tecnologías para el desarrollo de la solución propuesta.
- Levantamiento de funcionalidades y características de la propuesta a desarrollar.
- Implementación de la integración de un mundo virtual diseñado en OpenSim con Moodle

en su versión 2.8.x.

- Realización de pruebas para el funcionamiento de la integración de un mundo virtual diseñado en OpenSim con Moodle en su versión 2.8.x.

Para dar cumplimiento a los objetivos trazados en la investigación se emplearán los métodos científicos que se describen a continuación:

Métodos Teóricos:

- **Analítico-sintético:** para el análisis de las tendencias históricas de desarrollo de entornos virtuales 3D para el aprendizaje de contenidos en línea.
- **Histórico-lógico:** para el estudio de la evolución de mundos virtuales en función del aprendizaje, desde sus orígenes hasta la actualidad, así como factibilidad de su utilización en la universidad.
- **Sistémico-estructural:** para crear manuales siguiendo las normas internacionales previamente establecidas.
- **Revisión documental:** para fundamentar el propósito de la investigación y permitir el desarrollo del marco teórico y/o conceptual de la misma.

Método Empírico:

- **Observación:** Para observar y verificar los aportes de un entorno virtual 3D en la universidad y determinar los problemas de la comunidad que se puedan resolver.

Estructura del contenido: el documento consta de tres capítulos en los que se abordan diferentes temáticas. Estos están estructurados de la siguiente forma:

Capítulo 1 Fundamentación teórica

Este capítulo expone algunos elementos teóricos necesarios para realizar el trabajo. Se muestran los conceptos relacionados con el dominio y estado del arte. Se analizarán los conceptos: mundo virtual, metaversos, la integración entre el mundo virtual y Moodle, la plataforma Moodle y la biblioteca Sloodle. Se realiza un análisis de las soluciones similares y se escoge la metodología a utilizar en el desarrollo del sistema.

Capítulo 2 Exploración, Planificación y Diseño de la solución propuesta

Se realiza un análisis de las herramientas a utilizar para el desarrollo de la solución y se especifican los requisitos que debe cumplir la aplicación así como la descripción de las

Introducción

funcionalidades utilizando las Historias de usuarios. También se muestra la Planificación para el correcto funcionamiento del desarrollo así como el Diseño del Mundo Virtual.

Capítulo 3 Implementación y prueba de la solución propuesta

Se abordan los aspectos relacionados con la construcción de la solución propuesta, las Descripciones de la Arquitectura de Moodle y OpenSim, además se definen los Casos de Pruebas de Aceptación para poder realizar las pruebas al *software*, para así verificar la integridad del mismo y el cumplimiento de los requisitos definidos por el cliente. Finalmente se realiza la validación de la solución propuesta a través del método de caja negra.

Capítulo: 1 Fundamentación teórica

1.1 Introducción

Investigadores como Livingstone¹⁰ enfatizan en la importancia de integrar los recursos de aprendizaje con las tecnologías, incluidas los mundos virtuales, como facilitadores de los procesos de enseñanza y aprendizaje (Rubén González Crespo, 2012). Se define como tema fundamental de la investigación lograr la interacción entre la plataforma educativa Moodle y un mundo virtual diseñado en OpenSim. Para poder determinar correctamente los elementos que deben interactuar es necesario conocer en profundidad los elementos que componen a los mundos virtuales, Moodle y, las características únicas de los mundos virtuales diseñados en OpenSim. Se analizarán las raíces de las palabras integración y su correspondencia con la situación, probando que es la solución al problema propuesto. Se realizará un estudio de las soluciones similares y de las metodologías escogiendo la idónea para el desarrollo de la investigación.

1.2 Metaversos y los mundos virtuales

Durante la investigación se encontró con un término recurrente que sirve de sinónimo a mundo virtual y que es muy utilizado por expertos y por otros conocedores del tema: “metaverso”, a efectos prácticos es un sinónimo de mundo virtual pero se consideró necesario investigar acerca de los orígenes del término, su relación con los mundos virtuales y la realidad virtual.

El término tiene su origen en la novela Snow Crash publicada en 1992 por Neal Stephenson, y se usa frecuentemente para describir una visión de trabajo en espacios 3D. Los metaversos son entornos donde los humanos interactúan social y económicamente como iconos, a través de un soporte lógico en un ciberespacio que actúa como una metáfora del mundo real (Cruz, 2010).

En palabras de Neal Stephenson, el metaverso es “mi idea cuando me encontré con que algunas palabras existentes tales como realidad virtual eran simplemente demasiado torpes para utilizarlas” (Stephenson, 1992).

¹⁰ Es una de las mentes detrás del desarrollo de la biblioteca Sloodle, Líder del Programa en el Estudio de Diseño Digital. Con experiencia docente en Informática, Gráficos, Desarrollo de Juegos y Mundos Virtuales.

Ubicándonos en la fecha en la que fue realizada la novela (1992) es posible que ya no sea factible comparar metaverso con realidad virtual, sino que el mundo que el describía como metaverso sea lo que hoy se conoce como mundo virtual. El término avatar, acuñado a lo virtual y a lo cibernético también tiene su origen en esta novela, ya que él se basa en el significado de avatar según la RAE (Real Academia Española):

Avatar.

(Del fr. *avatar*, y este del sánscr. *avatâra* ‘descenso o encarnación de un dios’) (Real Academia Española, 2012). Y de su tercer significado común: Reencarnación, transformación; para establecer a los avatares como reencarnaciones de las personas en los metaversos.

1.3 Mundos Virtuales

Es necesario para la investigación ubicar el concepto de mundo virtual que se utilizará ya que en la sociedad moderna existe un auge de las tecnologías virtuales y existen muchas definiciones acerca de lo que es un mundo virtual. El ISEA (Innovación en Servicios Empresariales Avanzados, España) establece que un *mundo virtual* es:

“...una simulación por ordenador de un espacio, que normalmente es una representación en tres dimensiones de accidentes geográficos, ciudades y representaciones digitales de entornos reales. Dentro de un mundo virtual, se puede desplazar e interactuar con el resto de los usuarios mediante una representación virtual del estudiante, denominado avatar”. (ISEA, 2008)

En la revista de investigación de Mundos Virtuales se realizó una búsqueda orientada a encontrar una definición formal de mundo virtual. Esta investigación estableció la inexistencia de definiciones formales y tras realizar un análisis de varias investigaciones informales se llegó a la conclusión de que un *mundo virtual* es:

Una red persistente y síncrona de personas, representadas mediante avatares, facilitada por computadoras interconectadas (Virtual Worlds Research: Past, Present & Future, 2008).

Es fácil ver en esta definición la importancia de las personas y de su comunicación cuando

se mencionan los mundos virtuales; sin embargo, Mark J. W. Lee¹¹ establece en la conferencia del 2010 realizada en Sydney y que hoy forma parte de la revista australiana de la tecnología educativa (AJET, del inglés Australian Journal of Educative Technology) que un *mundo virtual* es:

Un ambiente simulado en una computadora en el cual los usuarios son capaces de sumergirse y dentro de los cuales son capaces de, mediante sus avatares (representaciones computarizadas de ellos o de personas alternativas), experimentar, manipular, interactuar con y/o crear objetos virtuales y lugares que están gráficamente mostrados en tres dimensiones. Los objetos y lugares dentro de un mundo virtual pueden ser modelados acorde a aquellos en el mundo real o pueden estar basados en una fantasía. La mayoría de las aplicaciones de los mundos virtuales permiten múltiples usuarios e incluyen facultades que permiten a los usuarios comunicarse e interactuar unos con los otros dentro del ambiente virtual (Mark J. W. Lee, 2010).

Los autores de esta investigación consideran que la definición de Lee es la que mejor describe al concepto de mundo virtual debido al desarrollo acelerado que ha experimentado la tecnología ya que si bien antes el concepto ubicado en la revista de Investigación de Mundos Virtuales era el más ideal; en fechas actuales es inconcebible la existencia de un mundo virtual que no se encuentre graficado en tres dimensiones y con el que se pueda interactuar libremente. Sin embargo, los autores quisieran resaltar dos elementos de esta definición que Lee no establece y que son sumamente importantes para la existencia de un mundo virtual:

- ✓ Sincronía: los elementos del mundo virtual deben existir simultáneamente en el mismo espacio tiempo para todos los usuarios.
- ✓ Persistencia: el mundo virtual debe existir y continuar funcionando de manera síncrona y constante incluso si todos los usuarios se retiran del mismo.

Para continuar la investigación acerca de los mundos virtuales se analizarán las características únicas de los mundos virtuales creados en OpenSimulator, el cual es un

¹¹ Catedrático adjunto de la Facultad de Educación en la Universidad Charles Sturt y un investigador de honor de la Facultad de Ciencias, Tecnologías de la Información e Ingeniería de la Federación Universitaria en Australia. Ver <http://olc.onlinelearningconsortium.org/node/210566>

servidor multiusuario 3D de código abierto basado en la plataforma Second Life.

1.3.1 Mundos virtuales de Second Life

En 1999, Philip Rosedale concibió Linden Lab, empresa que desarrolló un programa que permitía a sus usuarios la inmersión en un mundo virtual. Ya en el año 2003 nace Second Life, mundo virtual en el cual sus usuarios, conocidos como residentes, pueden explorar el mundo virtual e interactuar con otros residentes (Linden_Lab, 2015).

Una de las características más interesantes de Second Life es la posibilidad de construir objetos. Aunque existe la posibilidad de crear objetos en otros programas y exportar el resultado a Second Life, el resultado suele ser pobre. Es preferible construir desde el principio, recoger material gratuito o usar objetos de la biblioteca y remodelarlo o reformarlo.

Ya que Second Life es un mundo virtual existente administrado por la compañía del mismo nombre le es imposible a instituciones educativas personalizar completamente el mundo virtual ni es posible tampoco crear otros mundos virtuales especializados con las funcionalidades que brinda Second Life.

Para concluir se puede establecer que Second Life no es una herramienta totalmente propicia para desarrollar actividades de aprendizaje con metas de crecimiento a largo plazo. El uso de este trae marcadas desventajas, entre las que se encuentra el modelo de negocio basado en la compra de regiones o islas, lo que obliga a las instituciones a comprar terrenos o islas y a pagar una renta anual o semestral.

Hoy existe un gran portafolio de servicios y productos tomando como base a mundos virtuales orientados a la educación y una de las más grandes oportunidades se encuentra en la plataforma de creación y manejo de mundos virtuales de código abierto llamado OpenSimulator que si bien no fue concebido para educación este si permite crear un ambiente propicio para tal fin.

1.3.2 Mundos Virtuales de OpenSim

OpenSimulator, también llamado OpenSim (OS), es un sistema 3D de código abierto, multi-plataforma y multi-usuario. El proyecto fue fundado en 2007 por Darren Guard. En enero de 2007 SecondLife (SL) lanzó un nuevo cliente de código abierto, además de una biblioteca, para que otros desarrolladores pudiesen crear nuevos clientes gráficos que se conecten a su servidor. Gracias a ello nace la idea de OpenSim como un servicio 3D al que

se pueda conectar un cliente gráfico de SL. Está programado en C# y por lo tanto es compatible con los sistemas operativos *Windows*, Framework .NET de *Microsoft*, y sobre el Framework Mono para sistemas operativos Linux (Luzcando, 2012).

Una de las ventajas que posee es que se encuentra bajo la licencia BSD¹² (OpenSim, 2012). En este servidor se pueden llevar a cabo casi todas las actividades que se pueden realizar en Second Life, existiendo problemas fundamentales en solo dos áreas, voz y búsqueda de caminos. Existen varios soportes para la voz, pero hay muchas dificultades y problemas existentes con el mismo, dependiendo principalmente de módulos de terceros. La búsqueda de caminos es un elemento fundamental en la Inteligencia Artificial para determinar rutas óptimas entre dos puntos y en la actualidad aún no se encuentra implementada en OpenSim.

Los autores consideran que una ventaja fundamental de OpenSim con respecto a Second Life es la capacidad de manejar múltiples mundos virtuales e incluso conectarlos libremente mediante un sistema de rejillas (*grid*, en inglés). El soporte y administración del servidor es muy simple y puede ser realizado por poco personal, siendo posible destinar más recursos a la personalización de los mundos virtuales.

El uso de la tecnología en la educación ha sido explorado continuamente y entre las alternativas disponibles para mejorar la enseñanza y el aprendizaje tanto en el aula como en las actividades en la educación a distancia se encuentra el uso de los LMS (Sistemas de Gestión de Aprendizaje).

1.4 LMS

En la tesis doctoral de la maestra mexicana Clara Lopez Guzmán¹³ se define a un LMS como:

“...conjunto de herramientas empleadas en la administración, distribución y control de materiales y actividades didácticas y formativas, gestión de cursos, gestión de clases, herramientas para la comunicación, herramientas para los discentes, gestión del contenido, herramientas de evaluación y gestión de organismos y entidades educativas.” (Lopez

¹² Berkeley Software Distribution, **licencia de software libre** que nos permite ver el código y modificarlo pero también cerrar el sistema o la aplicación.

¹³ Clara López Guzmán tiene el grado de Maestra en Tecnologías de Información y Administración por el Instituto Tecnológico Autónomo de México y Maestra en Sistemas de Información para las Empresas, por la Escuela Superior de Telecomunicaciones de Rennes, Francia.

Guzmán, 2005).

Generalmente estos LMS atienden a una estructura modular que permite administrar y configurar diversidad de procesos tales como: la gestión de alumnos y profesores, la creación y calificación de exámenes, la estructuración de cursos, la asignación de los mismos a los estudiantes matriculados y mejoran el aprendizaje del conocimiento y posibilitan que el mismo pueda ser adquirido de una manera centralizada o distribuida.

En el Centro de Comunicación y Pedagogía de España establecen que:

“Un LMS es un sistema de gestión de aprendizaje online, que permite administrar, distribuir, monitorear, evaluar y apoyar las diferentes actividades previamente diseñadas y programadas dentro de un proceso de formación completamente virtual (*eLearning*), o de formación semi-presencial (*Blended Learning*).” (Mayor, 2012).

Al analizar estas definiciones se puede determinar que Moodle, con sus características, es un LMS y cuyos elementos fundamentales están relacionados con las necesidades que resuelven los mismos, y debe, resaltando la definición anterior, administrar, distribuir, monitorear y evaluar.

Moodle en su versión 2.8.x es capaz de realizar estas tareas mediante un grupo de módulos integrados en la aplicación y ofrece incluso más posibilidades a los estudiantes y profesores, creando entornos especializados para el aprendizaje.

1.4.1 Plataforma educativa Moodle

El próximo paso en el análisis del estado del arte es la búsqueda de las características fundamentales que posee Moodle que garantiza la posibilidad y efectividad de la integración propuesta; pero primero hay que responder: ¿Qué es Moodle?

Según el sitio *web* dedicado a esta herramienta, Moodle.org, Moodle es una plataforma de aprendizaje diseñada para proporcionarles a educadores, administradores y estudiantes un **sistema integrado único, robusto y seguro** para crear ambientes de aprendizaje personalizados (Moodle, 2014).

En la escuela politécnica del ejército de Ecuador se define el concepto de Moodle:

Moodle (del inglés: *Modular Object-Oriented Dynamic Learning Environment* / Entorno Modular de Aprendizaje Dinámico Orientado a Objetos) es un ambiente educativo virtual

que ayuda a los educadores a crear comunidades de aprendizaje en línea. Su diseño está basado en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante, en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo, esto proporciona aspectos pedagógicos perdidos en muchas otras plataformas de aprendizaje virtual (Diana Guerra, 2010). Es una herramienta de distribución libre usada en más de 50000 sitios en todo el mundo y con más de 100 millones de usuarios enrolados en sus cursos (Moodle, 2015).

Con esto se demuestra que Moodle está muy enfocado al conocimiento y a cómo proporcionarlo; y no solamente a la gestión de los mismos, por lo que es calificado como un LMS (del inglés: *Learning Management Systems* / Sistemas de Administración del Aprendizaje).

1.5 Integración entre Moodle y Mundo Virtual

Es necesario que la interacción entre Moodle y un mundo virtual desarrollado en OpenSim sea capaz de administrar, distribuir, monitorear y evaluar los contenidos de Moodle desde el mundo virtual y viceversa. No obstante, los autores consideran necesario establecer la definición de interacción y de integración, para así mostrar cómo se puede resolver el problema propuesto al lograr la integración entre ambos sistemas.

Primero se analizará la definición de interacción según la RAE (Real Academia Española):

Interacción.

1. f. Acción que se ejerce recíprocamente entre dos o más objetos, agentes, fuerzas, funciones, etc (Real Academia Española, 2012).

Y a continuación las definiciones de integración según la propia RAE que se acercan al ámbito de la investigación:

Integrar.

(Del lat. *integrāre*).

1. tr. Dicho de las partes: Constituir un todo.
2. tr. Completar un todo con las partes que faltaban.
3. tr. Hacer que alguien o algo pase a formar parte de un todo. U. t. c. prnl (Real Academia Española, 2012).

Rápidamente se puede observar que integrar dos elementos es unirlos en un todo funcional, y que la definición de interactuar se refiere a la capacidad de realizar acciones entre dos o más objetos o agentes.

Observando esta definición de integrar podemos determinar que al unir dos o más objetos en un todo entonces es posible usar las capacidades y características de uno en el otro y viceversa, además de las capacidades únicas que surgen por la unión, lo cual hace necesaria una interacción constante entre los elementos de ambos agentes.

Uno de los caminos para integrar las tecnologías mencionadas, Moodle y mundos virtuales, es usando la herramienta Sloodle. De acuerdo a Guomin, Sloodle permite traer a un mundo virtual un entorno educacional tal como Moodle, mejorando la interacción y la percepción de los usuarios en la educación *online* (Rubén González Crespo, 2012).

1.6 Biblioteca Sloodle

Entre los elementos que existen mundialmente que podrían facilitar la integración entre un mundo virtual diseñado en OpenSim con Moodle existe una biblioteca llamada Sloodle la cual, según el sitio web dedicado es:

Sloodle (del inglés, *Simulation Linked Object-Oriented Dynamic Learning Environmet*) es un proyecto de código abierto que integra el entorno virtual multiusuario de SecondLife/OpenSim con el sistema de gestión de aprendizaje Moodle. Proporciona una gama de herramientas para apoyar el aprendizaje y la enseñanza en el mundo virtual. Con Sloodle se pueden combinar las posibilidades que ofrece Moodle con la interfaz de SecondLife en un método de aprendizaje envolvente que para casos como la enseñanza a distancia podría romper el aislamiento de los alumnos creándoles una verdadera impresión de estar en presencia de sus compañeros (Sloodle, 2012).

Esta herramienta ofrece facilidades para la comunicación entre Moodle y OpenSim, los autores consideran que se debe analizar más a fondo esta herramienta y ver qué beneficios aporta en aras de lograr la integración entre Moodle y un mundo virtual diseñado en OpenSim. Teniendo en cuenta estos elementos se decide realizar un análisis de mundos virtuales que usan esta biblioteca.

1.7 Análisis de soluciones existentes

Para el desarrollo de la investigación es necesario analizar soluciones existentes

tanto a nivel internacional como nacional que utilicen la biblioteca Sloodle para la interacción de los mundos virtuales con la plataforma Moodle. El avance de las nuevas tecnologías ha permitido facilitar el desarrollo de mundos virtuales con diferentes fines.

Existen gran cantidad de mundos virtuales dedicados a la educación, sin embargo debido a la complejidad de la construcción de estos, muchos han sido diseñados por especialistas y profesionales del 3D. Debido a esta dificultad de creación de mundos virtuales educativos se realizará un análisis de varias aplicaciones ya existentes usando herramientas comunes en la construcción de mundos virtuales educativos.

1.7.1 *Virtual Laboratory for Algorithms:*

Los mundos virtuales pueden ser utilizados para mejorar la educación de casi cualquier rama, como demuestra la revista TISE (Conferencia Internacional de Informática Educativa) del 2013 con su mundo virtual para el aprendizaje de algoritmos y programación. Este mundo virtual fue desarrollado usando Moodle 2.5 y la versión 2.1 de Sloodle para facilitar el trabajo de los educadores en el mundo virtual. Según la bibliografía consultada se encontraron con varias dificultades en el uso de Sloodle ya que este solo es completamente compatible con la versión 2.1 de Moodle. Además de Sloodle utilizaron una herramienta llamada IDEOne¹⁴ para el desarrollo de algoritmos de diferentes lenguajes integrado al mundo virtual. Este mundo es de acceso privado y no se encuentra publicado y accesible desde internet (Integrating Virtual Worlds and Virtual Learning Environments through Sloodle: from theory to practice in a case of study for teaching of algorithms, 2013). La siguiente imagen, figura 1, muestra el mundo virtual del cual se habla anteriormente.

¹⁴ Herramienta que permite compilar y depurar código fuente en más de 60 lenguajes de programación como Java, SQL, Objective-C, C, Ruby, Python, PHP o C++, y ejecutarlos online desde la propia plataforma.



Figura 1. Mundo virtual Virtual Laboratory for Algorithms

1.7.2 Universidad de St. Andrews:

En vista de las facilidades que brindan estos mundos virtuales muchas universidades han creado mundos virtuales propios, ya que estos enfocan el aprendizaje hacia el estudiante y no hacia el contenido.

Ejemplos destacados de estos trabajos son los realizados por la universidad de St. Andrews, la cual posee múltiples mundos virtuales orientados a diferentes especialidades. Entre sus proyectos destacados se encuentra la *Laconia Acropolis Virtual Archaeology (LAVA)*, un proyecto que permite a los estudiantes participar en una excavación arqueológica simulada y luego explorar una recreación del sitio; también existe *Wireless Island*, un proyecto que ayuda al aprendizaje colaborativo y la exploración de tráfico inalámbrico a través de simulaciones y de multimedia activa. Existen muchos otros desarrollados por esta universidad, pero todos son realizados por especialistas en el diseño de mundos virtuales 3D y responden a necesidades muy específicas de la universidad (EdITLib, 2012).

En la figura 2 se muestran los diferentes mundos virtuales desarrollados en esta universidad para el desarrollo del aprendizaje en diversas especialidades.

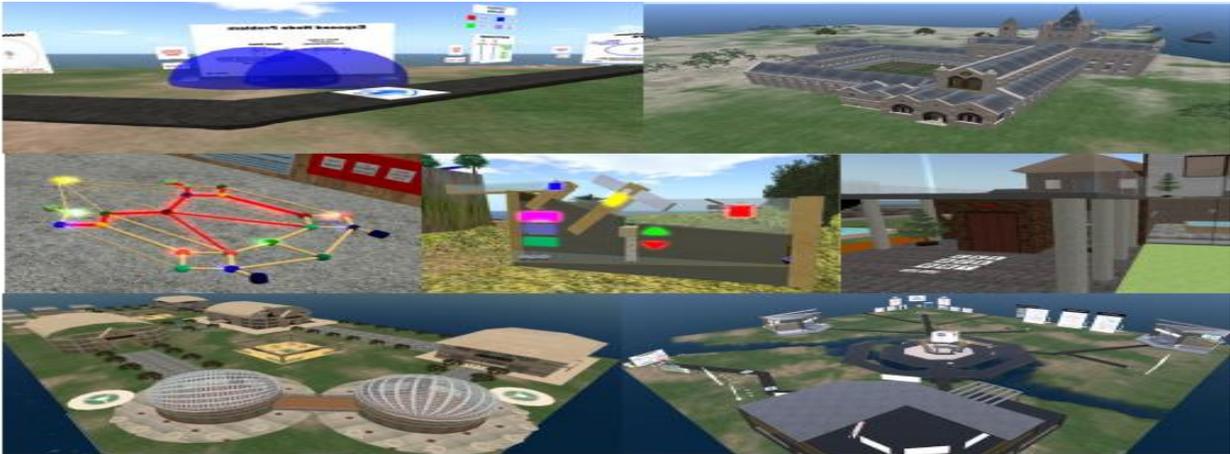


Figura 2. Mundos virtuales, St. Andrews

1.7.3 Facultad de Ingeniería:

El último caso a escala mundial que se quiere analizar es el mundo virtual basado en la facultad de ingeniería de la universidad de distrito “Francisco José de Caldas”. Este mundo virtual fue construido usando OpenSim 0.7.4 y se utilizó Sloodle 2.1 para vincularlo con Moodle 2.3. La idea de este mundo virtual es reconstruir la facultad de ingeniería y facilitar el acceso al contenido de la misma desde dentro del mundo virtual. Para lograr esto se usó Sloodle y se crearon muchos objetos referenciando a los elementos de la facultad poseyendo enlaces a los diferentes sitios web de la misma (Rubén González Crespo, 2012).

La figura 3 muestra en el lado izquierdo el edificio real de la facultad de ingeniería de la universidad de Francisco José de Caldas y del lado derecho muestra el que se diseñó en OpenSim



Figura 3. Mundo virtual de la Facultad de ingeniería

1.7.4 *LEARN-IN-3D*:

Es un mundo virtual orientado a mejorar la expresión oral de estudiantes del idioma inglés desarrollada en la Universidad de las Ciencias Informáticas. Este es muy sencillo y contiene todos los elementos necesarios para complementar lo aprendido de inglés en el aula. A pesar de plantear interesantes perspectivas aún sigue en desarrollo y pruebas de aceptación debido a un grupo de problemas con los servidores de voz utilizados para evaluar la expresión oral de los estudiantes, problemas de conexión con Moodle y uso incorrecto de algunas herramientas de Sloodle. Cuenta además con un aula virtual, varios laboratorios y un salón de presentaciones como se muestra en la figura 4.



Figura 4. Mundo virtual LEARN-IN-3D

1.7.5 *UCIGRID*:

Un mundo virtual de carácter expositivo de la Universidad de las Ciencias Informáticas que pretende permitir una visita virtual de la misma y ver a los proyectos, estudiantes y estilo de vida, además de mostrar los premios artísticos que han obtenido las diferentes facultades. Este mundo virtual aún no se encuentra en uso en la universidad.

1.7.6 *Virtual English*:

Es un mundo virtual desarrollado en el centro Vertex de la Facultad 5 en la Universidad de las Ciencias Informáticas orientado al aprendizaje del idioma inglés a distancia. Cuenta con muchas áreas y elementos desarrollados dedicados a varias especialidades. Se encuentra actualmente en desarrollo pero su principal limitante es que no se ha logrado conectar con el Moodle de la universidad y por ende el contenido está limitado a un repositorio local. En la figura 5 se muestra el mundo virtual de forma general.



Figura 5. Metaverso Virtual English

Según la bibliografía consultada hasta el momento, no existe la integración que se necesita entre los mundos virtuales y la plataforma Moodle. Por tanto es necesario desarrollar un mundo virtual mediante el cual se pueda interactuar con los recursos que brinda la plataforma Moodle en su versión 2.8.x. Para desarrollar la solución que se propone es necesario guiarse por una metodología de desarrollo de *software*, para esto a continuación se presenta un análisis de las mismas tomando las metodologías ágiles como referencia por las características del equipo de desarrollo y de la solución en sí.

1.8 Metodologías de desarrollo de *software*

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos (Hermes Romero, 2012). Imponen un proceso disciplinado sobre el desarrollo de *software* con el fin de hacerlo más predecible y eficiente. Desarrollan un proceso detallado con un fuerte énfasis en planificar las actividades por las que debe pasar un proceso de desarrollo de *software*. Estas se dividen en dos grandes grupos: metodologías robustas y ágiles. Se va a enfocar la investigación hacia las metodologías ágiles ya que se adaptan al equipo de desarrollo de dos personas y al tiempo relativamente corto. Estas permiten el trabajo conjunto entre el cliente y el equipo de desarrollo, el cual es muy importante en la situación actual.

1.8.1 Proceso unificado ágil (AUP)

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process (AUP)* en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (*test driven development - TDD*),

Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.

El proceso unificado (*Unified Process* o UP) es un marco de desarrollo *software* iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto *software*. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (*Rational Unified Process*) de IBM.

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la fase de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un modelo más simple que el que aparece en RUP debido a que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- ✓ *Inception* (Concepción): El objetivo de esta fase es obtener una comprensión común entre el cliente y el equipo de desarrollo del alcance del sistema y definir una o varias arquitecturas candidatas para el mismo.
- ✓ Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- ✓ Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- ✓ Transición: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Las disciplinas se llevan a cabo de manera sistemática, de acuerdo a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software de trabajo que responda a las necesidades de sus interlocutores (Lic. Ervin Flores, 2015).

1.8.2 XP (eXtreme Programming)

Según Kent Beck, es una metodología ligera, eficiente y flexible, para desarrollar software. Se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Tiene como objetivo la disminución del costo, la satisfacción del cliente y potenciar al máximo el trabajo en grupo (Ingeniería de Software). Además, contiene cuatro fases:

- ✓ Planificación
- ✓ Diseño
- ✓ Desarrollo
- ✓ Pruebas

XP es uno de los procesos ágiles más destacados que pone más énfasis en la adaptabilidad que en la previsibilidad. Los jefes de proyecto, los desarrolladores y el cliente son parte del equipo de desarrollo. Los desarrolladores deben conocer cada cambio que se realice durante el desarrollo del proyecto. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

Característica:

- ✓ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- ✓ Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- ✓ Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto, para lograr una mayor calidad del código fuente escrito.

- ✓ Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- ✓ Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- ✓ Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- ✓ Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- ✓ Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo (Álvarez, Solís; Díaz, Figueroa, 2007).

1.8.3 SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Es una metodología de trabajo (no de análisis y diseño) usada para la planificación de los proyectos, basada en gestionar proyectos de *software*. Esta es especialmente indicada para proyectos con un rápido cambio de requisitos. Tiene como característica el desarrollo de *software*, el cual se realiza mediante iteraciones, denominadas “corridas” o sprints. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Exige reuniones a lo largo del proyecto para coordinación e integración. Esta metodología durante su ciclo de vida pasa por determinadas etapas, estas son:

- ✓ Planeamiento: Se encarga de establecer la visión, definir expectativas y aseguramiento financiero.

- ✓ Montaje: Tiene como objetivo identificar otros requerimientos y priorizar las tareas para la primera iteración.
- ✓ Desarrollo: Encargada de implementar un sistema listo para entrega en una serie de iteraciones de treinta días (*sprints*).
- ✓ Liberación: El propósito es el despliegue operacional.

SCRUM es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas lo que permite obtener el mejor resultado posible de un proyecto. Permite realizar entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto (SCRUM, 2014).

1.8.4 SXP (SCRUM y eXtreme Programming)

Es una metodología ágil creada en Cuba, basada en las metodologías SCRUM y XP que permite actualizar los procesos de desarrollo de *software*, lo cual repercute en el mejoramiento de las actividades creativas. XP fue seleccionada para guiar el proceso ingenieril, puesto que facilita una documentación más discreta y mayor dinamismo para el desarrollo. SCRUM es entonces la metodología ideal para toda la gestión de proyectos. Sirve de soporte para acelerar el dinamismo que identificó XP, al posibilitar una planificación y organización del proyecto, centrada en los principales aspectos del sistema. Esto evita la pérdida de tiempo en tareas secundarias (SXP, Metodología ágil para el Desarrollo de Software, 2010).

SXP permite el aumento del nivel de preocupación y responsabilidad de los miembros del equipo y ayuda al líder del proyecto a tener un mejor control del equipo de trabajo. Esta metodología cuenta con 4 fases principales:

- ✓ Planificación-Definición: Se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- ✓ Desarrollo: Se realiza la implementación del sistema hasta que esté listo para ser entregado.
- ✓ Entrega: Se entrega al cliente y se pone en marcha.
- ✓ Mantenimiento: Se realiza el soporte para el cliente.

1.8.5 Valoración de selección

Después de realizar el estudio de las metodologías, se selecciona XP debido a que entre sus pilares fundamentales engloba las siguientes características:

- ✓ Grupos pequeños y trabajando en el mismo sitio.
- ✓ El cliente o el usuario forma parte del equipo de desarrollo.
- ✓ Existe una comunicación efectiva entre el equipo de desarrollo y el cliente.
- ✓ La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de *software*.
- ✓ Capacidad adaptativa al cambio.
- ✓ Los diseños del *software* son sencillos y libres de complejidad.
- ✓ Los cambios se implementan rápidamente tal y como fueron sugeridos.
- ✓ El manejo del cambio se convierte en parte sustantiva del proceso.
- ✓ La estructura de roles es adaptable al equipo de desarrollo.

Es necesario tener en cuenta las herramientas de ingeniería de *software* asistidas por computadora ya que estas brindan una mejor organización y versatilidad en cuanto a los modelos y diagramas necesarios para el desarrollo de un sistema.

1.9 Conclusiones Parciales

Es primordial destacar que no puede existir un mundo virtual sin interacción síncrona de usuarios y que es fundamental la persistencia de este mundo, independientemente de los agentes que interactúen con él. Es notable mencionar que Moodle es un LMS y que sus elementos fundamentales están orientados a administrar, monitorear, distribuir y evaluar contenidos para el aprendizaje.

Se estudiaron las raíces de varias palabras técnicas asociadas a la investigación, mostrando la similitud de metaverso y de mundo virtual, y de cómo una integración provee una interacción entre los componentes de la misma. Destacar que la herramienta Sloodle puede resultar imprescindible para realizar una rápida y efectiva integración entre Moodle y un mundo virtual diseñado en OpenSim. Ninguna de las soluciones similares analizadas brinda las funcionalidades necesarias para darle solución a la problemática planteada.

Capítulo: 2 Exploración, Planificación y Diseño de la solución propuesta

2.1 Introducción

El presente capítulo centra sus objetivos en realizar un estudio para seleccionar las herramientas a utilizar para la solución propuesta y el marco de trabajo dirigido principalmente por la metodología, abordando las fases de Exploración, Planificación de la Entrega, Iteraciones y Pruebas en las cuales se tocaran temas acerca de las necesidades del *software* y la planificación del proyecto. Se muestran las Historias de Usuarios, construyéndose también el plan de entrega con la cantidad de iteraciones del proceso de desarrollo.

2.2 Herramientas de Ingeniería de *Software* Asistida por Computadora (CASE)

CASE es una filosofía que se orienta a la mejor comprensión de los modelos de empresa, sus actividades y el desarrollo de los sistemas de información. Esta filosofía involucra además el uso de programas que permiten (SOMMERVILLE, 2005):

- ✓ Construir los modelos que describen la empresa.
- ✓ Describir el medio en el que se realizan las actividades.
- ✓ Llevar a cabo la planificación.

En el desarrollo del proyecto se hace necesario crear diagramas que apoyen a la descripción de componentes y clases. Desde el curso 2008-2009 la Universidad de las Ciencias Informáticas ha orientado el uso de la herramienta Visual Paradigm para el proceso docente y productivo con la justificación de ser un producto basado en *software* libre aunque se pagó la licencia que se establece para su uso (CALIDAD, 2008). A continuación se describen algunas características.

2.2.1 Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El *software* de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y

proyectos UML. Tiene soporte en todos los sistemas operativos y aunque precisa de licencia, la universidad tiene acceso a la misma. Permite diseñar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. Algunas características de esta herramienta son (Company Headquarters, 2012):

- ✓ Posibilita la creación de diagramas de flujo de datos.
- ✓ Posee distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- ✓ Ingeniería inversa - Código a modelo, código a diagrama.
- ✓ Ingeniería inversa Java, C++, Esquemas XML, XML, .NET exe/dll, CORBA IDL.
- ✓ Generación de código - Modelo a código, diagrama a código.
- ✓ Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- ✓ Generador de informes para generación de documentación.
- ✓ Importación y exportación de ficheros XMI.

En la Universidad de las Ciencias Informáticas se han realizado diversos mundos virtuales desarrollados en el servidor OpenSim. Por esto se realiza el análisis siguiente del mismo.

2.3 OpenSim

Se va a utilizar la versión 0.8.0.3, que salió al mercado el 11 de noviembre del 2014. Una de las ventajas que posee es que se encuentra bajo la licencia BSD¹⁵ (OpenSim, 2012). En este servidor se pueden llevar a cabo varias actividades como por ejemplo:

- ✓ Simulaciones.
- ✓ Realizar exhibiciones (esculturas 3D, fotos, pinturas, etc.).
- ✓ Dinámica y juego de roles.
- ✓ Clases online, conferencias, talleres, capacitaciones.
- ✓ Proyectos colaborativos.

Entre las características que posee se encuentra la compatibilidad con mundos virtuales 3D

¹⁵ **Berkeley Software Distribution**, licencia de software libre que nos permite ver el código y modificarlo pero también cerrar el sistema o la aplicación.

multi-usuario en línea tan pequeños como un simulador o tan grandes como miles de simuladores. También soporta espacios virtuales en 3D de tamaño variable dentro de una única instancia. Soporta múltiples protocolos y clientes, los cuales tienen acceso al mismo mundo virtual al mismo tiempo, a través de múltiples protocolos. Soporta en tiempo real Simulación Física, con múltiples opciones de motores. Soporta los clientes que crean contenido en 3D en tiempo real. Soporta el uso de *scripts* dentro del mundo virtual mediante el uso de LSL / OSSL y C #. Proporciona capacidad ilimitada para personalizar las aplicaciones del mundo virtual a través del uso de módulos (OpenSim, 2012).

Por todo lo anteriormente planteado el uso de esta tecnología en cualquier sector aporta facilidades. Es una aplicación que integra varias funcionalidades que le pueden dar solución a varios problemas en un solo espacio. En la UCI existen varios servicios que necesitan de un monitoreo y administración por diferentes personas, se pueden poner de ejemplo: la gestión académica, exposiciones en espacios de esparcimiento y de apoyo al aprendizaje de otros idiomas. Los mundos virtuales permiten que todos esos servicios sean integrados y que una persona con solo acceder con su avatar al mundo pueda realizar cualquiera de esas actividades. Importante destacar que se necesita solo de una persona para administrar el servidor del mundo virtual.

Este servidor ofrece facilidades para configurar un entorno privado (restringido), es atractivo por su parte porque el espacio virtual no implica gasto. Permite la creación de servidores propios y creación de contenido en tiempo real. Incluye herramientas para personalizar avatares, comunicarse con otros usuarios y construir elementos 3D dentro del entorno. Además la plataforma muestra un mundo virtual completo que permite a los profesores y estudiantes de distintos lugares trabajar juntos de manera más eficiente y natural.

Para el desarrollo de la tesis es necesario conocer y estudiar las tecnologías que se necesitan y por ende los lenguajes empleados en ellas.

2.4 Lenguajes y tecnologías de desarrollo

En la actualidad existen una inmensa cantidad de herramientas, tecnologías y lenguajes de programación para brindar soluciones a diversos problemas que se presentan regularmente o no en la vida cotidiana, lo que aumenta el desempeño de grandes y medianas empresas. A continuación se argumentará sobre las herramientas y los lenguajes de programación usados para un correcto desarrollo y obtener los resultados propuestos garantizando la

calidad requerida de la integración. Se hace referencia al lenguaje de programación C#, *Linden Scripting Language* (LSL), PHP y a la herramienta Sloodle que es la encargada de la conexión entre el mundo virtual y los recursos de Moodle.

2.4.1 Lenguaje de programación C#

El lenguaje de programación C# fue creado por el danés Anders Hejlsberg, que diseñó también los lenguajes Turbo Pascal y Delphi. El C# (pronunciado en inglés “C sharp” o en español “C sostenido”) es un lenguaje de programación orientado a objetos. A pesar de que el lenguaje C# forma parte de la plataforma .NET, que es una interfaz de programación de aplicaciones, es un lenguaje independiente que originariamente se creó para producir programas sobre esta plataforma. C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. Además soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo (González Seco, 2011).

Algunas de las características del lenguaje de programación C# son que su código se puede tratar íntegramente como un objeto. Su sintaxis es muy similar a la del lenguaje JAVA. Es un lenguaje orientado a objetos y a componentes. Armoniza la productividad del Visual Basic con el poder y la flexibilidad del C++. Ahorra tiempo en la programación ya que tiene una biblioteca de clases muy completa y bien diseñada.

Otro de los lenguajes que se utilizará en la implementación de la integración es el *Linden Scripting Language* (LSL), utilizado en el servidor OpenSim.

2.4.2 Linden Scripting Language

Linden Scripting Language (LSL) es el lenguaje que se utiliza para crear contenidos interactivos en SecondLife/OpenSim. Se ajusta a la sintaxis de los lenguajes C y Java, LSL es un lenguaje de programación orientado a eventos. Con este lenguaje de programación podremos crear *scripts* para controlar el comportamiento de los objetos dentro del mundo virtual. Los *scripts* están compuesto de uno o varios estados, dentro de cada estado hay uno o varios eventos y dentro de estos una o varias funciones (AIDA, 2011). Pueden adjuntarse varios *scripts* a un mismo objeto, lo que permite tener un conjunto de pequeños *scripts* con funciones simples, que combinados permiten formar nuevos comportamientos más complejos.

En el desarrollo del trabajo se hace necesario el uso pero en menor medida del lenguaje de programación PHP, el cual está dedicado al desarrollo de páginas web.

2.4.3 Lenguaje de programación PHP

PHP (acrónimo recursivo de PHP: *Hypertext Preprocessor*) es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es muy popular ya que un gran número de páginas y portales web están creadas con este lenguaje. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo algunas reglas. Los orígenes de PHP están fechados en torno al año 1995, sin embargo, fue a partir del año 1999 con la publicación de PHP 4 cuando este lenguaje de programación tomó un verdadero auge (Enrique González, 2015).

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Tiene una curva de aprendizaje muy baja, su sintaxis es simple y cumple estándares básicos de la programación orientada a objetos. No son necesarios complejos entornos de desarrollo, que incluso necesitan su propio periodo de aprendizaje. Se puede programar en PHP sin más ayuda que la del bloc de notas, todos los IDEs¹⁶ disponibles son gratuitos y los entornos de desarrollo son de rápida y fácil configuración.

Para lograr la integración del mundo virtual con la plataforma Moodle es necesario utilizar la biblioteca Sloodle, la cual no posee soporte desde el año 2012 por lo que presenta errores que deben ser corregidos, a continuación se realiza un análisis de la misma.

2.4.4 Sloodle

Actualmente Sloodle cuenta con el apoyo de la fundación británica Eduserv y de la Escuela de Ciencias de la Información de la Universidad Estatal de San José (California). Con Sloodle se pueden combinar las posibilidades que ofrece Moodle con la interfaz de SecondLife como muestra la figura 6 en un método de aprendizaje envolvente que para casos como la enseñanza a distancia podría romper el aislamiento de los alumnos

¹⁶ Entorno de desarrollo integrado es una aplicación de software, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de software.

creándoles una verdadera impresión de estar en presencia de sus compañeros (Sloodle, 2012).



Figura 6. Sloodle para la conexión entre Moodle y SecondLife.

En Sloodle los objetos en SL se corresponden con actividades de Moodle. Entre las que se encuentran las mostradas por la figura 7.

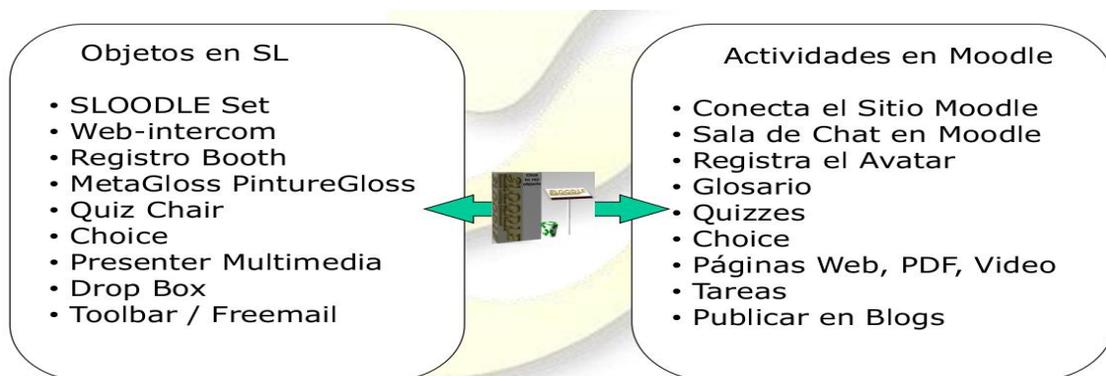


Figura 7. Integración Sloodle.

En la UCI actualmente existen diversos problemas en los mundos virtuales creados y aún no se ha logrado la integración entre Sloodle y Moodle por lo que se necesitará trabajar con el código fuente para lograr la migración de la versión de Sloodle utilizada en la Universidad.

Algunos de los problemas que presenta la herramienta Sloodle son:

- ✓ En el Menú, a la hora de eliminar a los avatares tiene la dirección fija y no relativa, por lo que al redireccionar contra un puerto o similar no funciona.
- ✓ Los módulos Presenter, Password Reset, QuizChair, Quiz Pile-On y el Scoreboard no funcionan como deberían ya que Moodle cambió muchos aspectos después de la versión 2.0.

- ✓ El Loginzone tiene un problema fundamental en su funcionamiento, y es la forma en la que es invocado desde el servidor en Moodle. Loginzone existe desde Sloodle 0.3 y su objetivo es registrar un avatar de un usuario en Sloodle vinculándolo con un usuario en Moodle.
- ✓ El RegEnrol Booth funciona pero a medias, ya que lo hace bien para el usuario administrador pero no para cualquier otro usuario.

Para el correcto funcionamiento de la propuesta de solución es necesario un servidor web el cual se encargará de contestar las peticiones hechas por los usuarios.

2.5 Servidores Web

Es un programa que implementa el protocolo de transferencia de hipertexto (HTTP) o el protocolo HTTPS¹⁷ (la versión cifrada y autenticada). Es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Entre los servidores más conocidos se encuentran Internet Information Services y Apache.

Apache es una plataforma de servidores web de código abierto y gratuito que puede ejecutarse en sistemas operativos como GNU/Linux, Macintosh y Windows. Las capacidades de este servidor pueden ser ampliadas incorporándole nuevos módulos, pues su diseño modular es altamente configurable. Es configurable, robusto y estable, lo que contribuye a que hoy en día sea el servidor web más utilizado en el mundo. Como desventaja no posee interfaz gráfica que facilite su configuración (Apache, 2013).

Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta Apache, optimizando el rendimiento y la rapidez del código. Los módulos de Apache se pueden clasificar en tres categorías:

Módulos Base: Módulo con las funciones básicas de Apache.

Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones.

¹⁷ En el protocolo HTTP las URLs comienzan con "http://" y utilizan por defecto el puerto 80, Las URLs de HTTPS comienzan con "https://" y utilizan el puerto 443

Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Ventajas de Apache:

- ✓ Es personalizable.
- ✓ Corre en gran número de sistemas operativos.
- ✓ Actualmente existen diversos módulos para este tipo de servidor, que pueden ser usados cuando sea necesario.
- ✓ Permiten la implementación de los últimos y más nuevos protocolos.
- ✓ El servidor puede ser administrado vía línea de comandos, lo que hace la administración remota muy conveniente. (Apache, 2013)

Es necesario tener en cuenta que para poder visualizar el mundo virtual creado es imposible hacerlo si no se tiene un visor o cliente adecuado. Se realizará un análisis de las características de los más utilizados a nivel mundial escogiendo el más indicado.

2.6 Cliente - Visor

Es un *software* de navegación 3D que se necesita para conectarse, explorar y comunicarse en el mundo virtual. Es capaz de mostrar correctamente la información almacenada en un archivo, para el caso específico de Second Life el visor es el medio que te enlaza con los servidores de Linden Lab, a través de una interfaz gráfica y que te permite interactuar con otros usuarios.

RealXtend Viewer: es un navegador basado en el visor Second Life de Linden Labs. Las modificaciones realizadas incluyen procesamiento de OGRE (compatible con mallas 3D de Mundo Virtual) además del procesamiento original y un sistema global de avatar.

Hippo OpenSim Viewer: es un visor de Second Life modificado, dirigida a los usuarios OpenSim. Se permite la construcción hasta una altura de 10.000 metros, prima escala hasta 256x256x256 metros y otras interesantes funciones.

Imprudence/Kokua Viewer: es un proyecto de código abierto bajo los términos de la GNU. Su objetivo es el de mejorar, en gran medida, la utilidad del espectador a través de la participación comunitaria, del diseño inteligente, los métodos modernos de desarrollo, y un ambiente a favor del cambio. Imprudence Viewer tiene todas las características del Hippo-Viewer, en la figura 8 se muestra la interfaz del mismo (Imprudence Viewer, 2013).



Figura 8. Interfaz de Imprudence.

Las acciones más básicas que podemos realizar son:

- ✓ **Moverse:** Para ello hay que utilizar las flechas del teclado. Para correr pulsa dos veces sobre la dirección a la que deseas dirigirte.
- ✓ **Hablar:** En la parte inferior aparece un cuadro de texto destinado al chat, lugar en el que podemos escribir para comunicarnos con otros avatares.
- ✓ **Volar:** Al pulsar en el menú inferior esta opción podremos desplazarnos de forma más rápida por el mundo virtual, para volver al suelo pulsa “Dejar de volar”.
- ✓ **Sentarse:** Para ello hay que hacer clic sobre la silla en la que queremos ocupar asiento. Para levantarse hay que pulsar “Levantarse”.

FireStorm: El Proyecto Phoenix está activo actualmente en el desarrollo de la tormenta de FireStorm Viewer, el sucesor del Visor Phoenix ahora suspendido. El visor FireStorm es de código abierto y se basa en la base de código LGPL v3 Linden Lab y tiene una enorme cantidad de características y opciones de personalización de la interfaz, incluyendo una mirada similar a su predecesor Phoenix Visor. FireStorm está desarrollado para Windows, Linux y sistemas operativos Mac OS X.

Entre sus principales características se encuentran:

- ✓ Posee selector de cuadrícula, permite administrar estas cuadrículas y buen gráfico.
- ✓ Posee soporte de OSSL¹⁸ y MOAP¹⁹.
- ✓ Posibilita la opción de múltiples adjuntos y varias capas de ropa.

¹⁸ OSSL es una extensión de LSL, el lenguaje de scripting en OpenSimulator y en Second Life.

¹⁹ MOAP (Media On A Prim), medios en un prim se implementó en OpenSimulator desde 0.7.1. Esta facilidad le permite visualizar y navegar las páginas web de la superficie / textura de un prim.

Por todas las características anteriormente planteadas se escoge como visor para las aplicaciones 3D el FireStorm ya que es de código abierto y se encuentra entre los más aceptados a nivel mundial por su gran cantidad de características en beneficio al desarrollo y compatibilidad con OpenSim y Second Life. Otra de las características por las que se tomó este visor para el desarrollo de la integración es porque posee soporte para Shared Media utilizado para mostrar videos o imágenes, problema muy importante que se desea resolver.

2.7 Propuesta de solución

Tras haber analizado el flujo actual y la situación problemática de integración entre un mundo virtual diseñado en OpenSim y Moodle se ha llegado a la conclusión de que sería de mucha utilidad e importancia crear un mundo virtual que permita la conexión con los recursos de Moodle en su versión 2.8.x para así darle soporte a la biblioteca Sloodle que no ha tenido retroalimentación desde la versión de Moodle 2.3 en el año 2012 y poder contribuir con un mejor funcionamiento de los proyectos existentes en la Universidad y los proyectos futuros. En el camino de desarrollo se dejarán bien explicados los pasos y métodos a seguir para lograr la comunicación y una buena configuración de las herramientas a utilizar para que se puedan emplear en futuros mundos virtuales en la UCI.

2.7.1 Personal relacionado con el mundo virtual

Cuando se menciona el término de personas relacionadas con el sistema se hace referencia a todo aquel sujeto que de una manera u otra puede acceder a la información: administrador, profesor y estudiante.

Tabla 1. Usuarios del sistema.

Usuarios del sistema	Explicación
Administrador	Es la persona encargada de gestionar todas las funcionalidades dentro del metaverso. Administra las cuentas de usuarios y restringe los permisos de los demás usuarios.

Profesor	Es la persona encargada de crear el objeto y relacionarlo con los cursos de Moodle deseados.
Estudiante	Es la persona que tiene acceso a navegar por el metaverso y acceder a los materiales expuestos en él.

2.8 Proceso de captura de funcionalidades del sistema

Antes de establecer una solución al problema siempre se hace necesario llevar a cabo una reunión con el cliente del *software* o aplicación, para llegar a acuerdos y establecer las funcionalidades que se deseen que posea el proyecto. De ahí es común establecer en los primeros encuentros que se tienen con el cliente, peticiones informales de las funcionalidades de *software* y diseños que a la larga aportarán mucho a la concepción del mismo.

XP propone que se realice un levantamiento de funcionalidades en las primeras reuniones, en las que se tomen todos los aspectos relacionados con las necesidades del sistema a desarrollar. En estos encuentros se pueden llevar a cabo prácticas muy efectivas de captura de funcionalidades como:

- ✓ Entrevistas frecuentes para aclarar dudas que de manera continua puedan aparecer.
- ✓ Observación de los procesos a informatizar para lograr mantener una presencia lo más imperceptible posible para no obstaculizarlos.
- ✓ Juegos de rol donde el cliente simule ser un usuario que interactúa con el sistema, este segundo simulado por el desarrollador.

Luego de realizados los encuentros se realiza la captura de las funcionalidades que se muestra a continuación.

2.8.1 Lista de Reservas del Producto

Según se define en la metodología XP la Lista de Reservas del Producto está compuesta por los aspectos funcionales con los que debe contar el sistema para satisfacer eficientemente las necesidades requeridas por el cliente. A continuación se muestra el

listado de los mismos.

1. Crear el avatar para la administración del servidor.
2. Crear y administrar los objetos de Sloodle.
3. Crear laboratorio virtual e integrarlo al mundo virtual existente.
4. Crear los objetos de interconexión entre OpenSim y Moodle.
5. Permitir la autenticación de los usuarios.
6. Vincular el avatar de OpenSim con la cuenta de Moodle.
7. Enrolar a los estudiantes en los cursos de Moodle desde OpenSim.
8. Vincular la sala de chat.
9. Acceder al glosario de Moodle desde el Mundo Virtual.
10. Vincular las preguntas de selección de los cursos.
11. Permitir autenticación rápida de los usuarios luego de haberse registrado.
12. Realizar exámenes con preguntas de selección múltiples.
13. Realizar exámenes de verdadero y falso.
14. Permitir adquirir objetos mediante la obtención de bonificaciones por respuestas.
15. Realizar tareas de seguimientos del Usuario.
16. Visualizar los contenidos de Moodle desde el mundo virtual.
17. Internacionalizar los componentes de Sloodle.

2.8.2 Características del sistema

Las características del sistema son propiedades o cualidades que el producto debe poseer. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a funcionalidades. A continuación se presentan los definidos para la realización de sistema a desarrollar.

Software

- ✓ Se requiere la biblioteca .NET en su versión 4.0 para el sistema operativo Windows y para las distribuciones de GNU/Linux se requiere Mono.

Hardware

- ✓ Memoria RAM: 512 MB a 1 Gb como mínimo para cualquier sistema operativo.
- ✓ Espacio en disco duro: 256 MB mínimo para cualquier sistema operativo.

Usabilidad

- ✓ El mundo virtual será diseñado de forma tal que integre la mayor cantidad de aspectos posibles relacionados con la usabilidad. Esta característica permitirá que el sistema sea eficiente, un poco más entendible para aquel profesor que conozca poco de mundos virtuales, pero siempre teniendo conocimientos básicos acerca del tema.

Rendimiento

- ✓ El mundo virtual deberá ser capaz de conectarse con la plataforma Moodle y realizar cualquier transacción de datos en un lapso de tiempo aceptable, no más de 10 segundos.

Portabilidad

- ✓ La herramienta deberá funcionar en los sistemas operativos Windows y las distribuciones de GNU/Linux.

Aspectos legales

- ✓ La propuesta de solución una vez desarrollada será de código abierto y licencia GPL.

2.9 Exploración

Como punto de partida en el ciclo de vida de un proyecto, XP propone la Exploración como la primera etapa donde los clientes plantean a grandes rasgos las Historias de Usuario (UH) que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología a utilizar (Hernández Sampieri, 2006).

2.9.1 Historias de usuarios

Las Historias de usuario son un instrumento para el levantamiento de requerimientos para el desarrollo de un *software*, que ha emergido con la aparición de los nuevos marcos de trabajo de desarrollo ágil, son descripciones cortas de una necesidad de un cliente del

software que se esté desarrollando.

Algunas características deseables de las HU son (La Oficina de Proyectos de Informática, 2013):

- ✓ Que sean escritas por el usuario o por un analista de negocio que le represente.
- ✓ Frase corta que encaje en una tarjeta de tres por cinco pulgadas.
- ✓ Debe describir el rol desempeñado por el usuario en el sistema, descrito de forma explícita.
- ✓ Debe describir el beneficio para el área de negocio que representa esta funcionalidad.

A continuación se muestran las descripciones de cada una de las HU propuestas por el cliente.

Tabla 2. HU 1: Administrar el servidor y los objetos de Sloodle.

Historia de Usuario	
No.: 1	Nombre: Administrar el servidor y los objetos de Sloodle.
Usuario: Administrador	
Prioridad en el Negocio: alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 1
Descripción: Debe ser posible acceder a la configuración del servidor y ser creados los objetos de Sloodle.	
Información adicional (Observaciones): Da cumplimiento a las funcionalidades “Crear el avatar para la administración del servidor” y “Crear y administrar los objetos de Sloodle”.	

Tabla 3. HU 2: Crear laboratorio virtual.

Historia de Usuario	
No.: 2	Nombre: Crear laboratorio virtual.

Usuario: Administrador	
Prioridad en el Negocio: alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 1
Descripción: El entorno virtual será un laboratorio que se incluirá en un entorno virtual mucho más grande en existencia y deberá poder comunicarse con los recursos del Moodle.	
Información adicional (Observaciones): Da cumplimiento a la funcionalidad “Crear laboratorio virtual e integrarlo al mundo virtual existente”.	

Las demás historias de usuarios se pueden observar en el Anexo 1.

2.10 Planificación

La fase de Planificación se realiza en pocos días, no siendo menos importante, ya que su objetivo principal es priorizar las HU. Los programadores estiman el esfuerzo de cada una de ellas, se define el cronograma y se acuerda el alcance de la entrega, el cual debe incluir varias iteraciones. El cliente prioriza las HU seleccionando cuales se realizarán en cada iteración, logrando en la primera iteración un sistema con la arquitectura de proyecto.

Otro aspecto a tener en cuenta durante esta fase es la velocidad del proyecto, la cual se estima utilizando el tiempo empleado en el desarrollo de las iteraciones terminadas. Esta medida debe reevaluarse una vez concluida 3 o 4 iteraciones y en caso de no cumplir el tiempo estimado, debe ser negociando con el cliente un nuevo Plan de Entregas (Hernández Sampieri, 2006).

Para la duración de las semanas que se tienen en cuenta en las estimaciones de las historias de usuario anteriores, es necesario aclarar que una “1” semana equivale a los cinco “5” días laborales de la misma. Se considera válida esta aclaración ya que generalmente, los cálculos erróneos de estimación de los tiempos de desarrollo se realizan en base a los siete “7” días de la semana.

2.10.1 Estimación de esfuerzos por historia de usuario

Tabla 4. Plan de estimación de esfuerzo.

Iteraciones	Orden de las Historias de Usuario a implementar	Cantidad de tiempo de trabajo
1	HU1. Administrar el servidor y los objetos de Sloodle.	1 semana
1	HU2. Crear laboratorio virtual.	1 semana
1	HU3. Conexión de OpenSim con Moodle y acceso de los usuarios al Mundo Virtual	1 semana
1	HU4. Vincular el avatar del usuario y enrolar los estudiantes.	1 semana
2	HU5. Vincular los recursos chat y glosario	1 semana
2	HU6. Realizar preguntas de selección y permitir autenticación rápida.	1 semana
2	HU7. Realizar exámenes.	1 semana
2	HU8. Permitir obtención de objetos por bonificación y el seguimiento a los usuarios.	1 semana
3	HU9. Visualizar los contenidos de Moodle desde el mundo virtual.	2 semanas
3	HU10. Internacionalizar los componentes de Sloodle.	2 semana
Total		12 semanas

2.10.2 Iteraciones

XP enfatiza en el carácter iterativo e incremental del desarrollo, la fase de Planificación es la principal en el ciclo de vida de un proyecto. Las iteraciones agrupan un conjunto de HU a implementar en un período de tiempo, generando al final de cada una un entregable funcional. Son relativamente cortas. La entrega con rapidez de módulos al cliente aumenta la retroalimentación y resulta más provechoso en términos de calidad del producto que una entrega a largo plazo.

Las HU no cuentan con suficiente detalle como para permitir su análisis y desarrollo sin la presencia del cliente durante la fase. Por lo que se hace necesario al comienzo de cada iteración realizar las tareas necesarias de análisis en conjunto con el cliente, consolidando los datos necesarios para la implementación. La participación del cliente durante esta fase del ciclo es de vital importancia. Culminar sin errores una iteración constituye un avance notable en el desarrollo del proyecto.

Tabla 5. Plan de iteraciones.

Iteraciones	Historias de Usuario a implementar
Iteración 1	HU1. Administrar el servidor y los objetos de Sloodle.
	HU2. Crear laboratorio virtual.
	HU3. Conexión de OpenSim con Moodle y acceso de los usuarios al Mundo Virtual.
	HU4. Vincular el avatar del usuario y enrolar los estudiantes.
Iteración 2	HU5. Vincular los recursos chat y glosario.
	HU6. Realizar preguntas de selección y permitir autenticación rápida.
	HU7. Realizar exámenes.
	HU8. Permitir obtención de objetos por bonificación y el seguimiento a los usuarios.
Iteración 3	HU9. Visualizar los contenidos de Moodle desde el mundo virtual.
	HU10. Internacionalizar los componentes de Sloodle.

2.10.3 Plan de Entregas

El Plan de Entregas es el resultado final de la fase de Planificación acordado en una reunión entre todos los actores del proyecto, estableciendo cuáles HU serán agrupadas para conformar una entrega. Es un compromiso que establece el grupo de trabajo con el cliente, aspecto de vital importancia, debido a que entregas tardías podrían tener grandes repercusiones en la economía, moral del equipo de trabajo y acarrear insatisfacciones con el cliente.

El cronograma de entregas es realizado en dependencia de las estimaciones de tiempos

de desarrollo realizadas por los programadores. La estimación es uno de los temas más complicados del desarrollo de un proyecto de *software*, es por ello que se recomienda luego de algunas iteraciones realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el Plan de Entregas y ajustarlo en caso de ser necesario. La siguiente tabla muestra el control de versiones que se debe tener al final de cada iteración.

Tabla 6. Plan de entrega.

Entregable	Historias de Usuarios
Entrega 1	Administrar el servidor y los objetos de Sloodle. Crear laboratorio virtual. Conexión de OpenSim con Moodle y acceso de los usuarios al Mundo Virtual. Vincular el avatar del usuario y enrolar los estudiantes.
Entrega 2	Vincular los recursos chat y glosario. Realizar preguntas de selección y permitir autenticación rápida. Realizar exámenes. Permitir obtención de objetos por bonificación y el seguimiento a los usuarios.
Entrega 3	Visualizar los contenidos de Moodle desde el mundo virtual. Internacionalizar los componentes de Sloodle.

2.11 Diseño

La metodología XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un buen sistema robusto y reutilizable tratando de mantener su simplicidad, es decir, crear un diseño evolutivo que se va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente.

2.11.1 Mundo Virtual

En las figura 9, 10 y 11 se muestra el interior del laboratorio virtual con los objetos agregados para permitir la interacción con los recursos de Moodle.

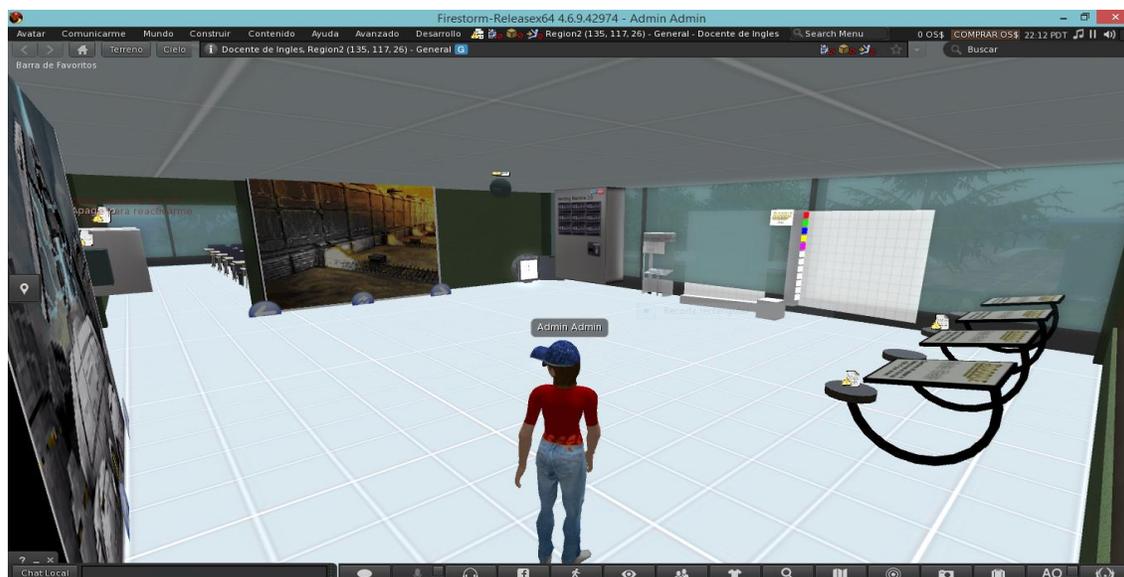


Figura 9. Entrada del laboratorio virtual.

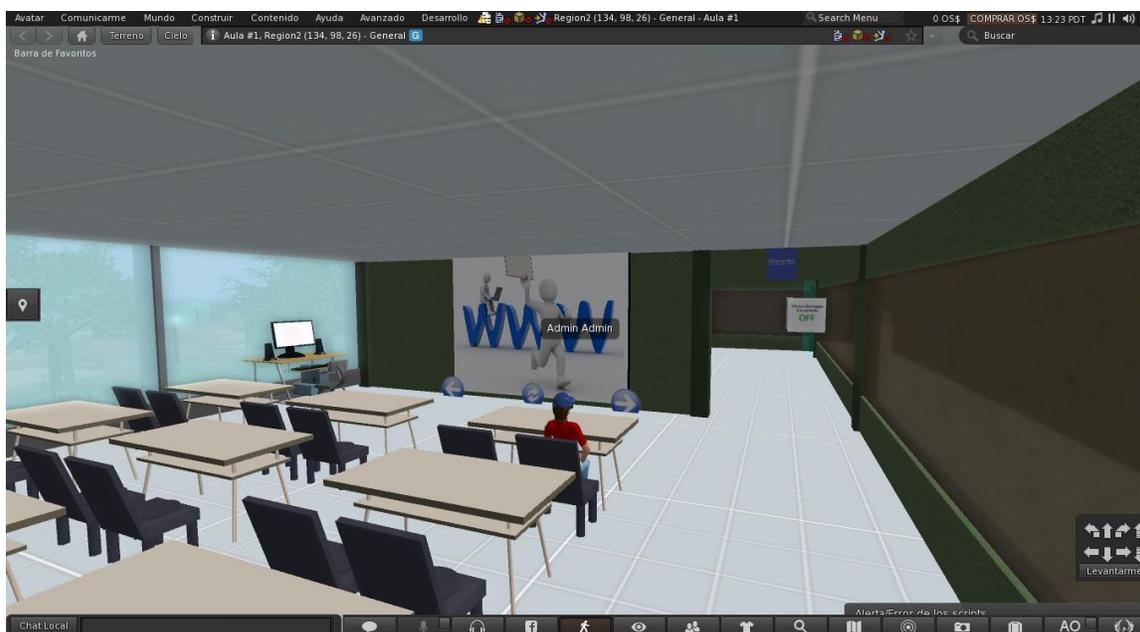


Figura 10. Laboratorio para clases



Figura 11. Avatar en una silla de exámenes

2.12 Conclusiones parciales

Se hizo un estudio de las herramientas necesarias para el desarrollo de la solución propuesta asumiendo Sloodle para la conexión del mundo virtual con la plataforma Moodle, como cliente para visualizar el mundo virtual el FireStorm, el servidor web Apache y los lenguajes de programación C# y *Linden Scripting Language* (LSL). La exploración, planificación y diseño fue orientada a la organización que tendrá luego la solución aportada para la integración. Se obtuvieron los requisitos funcionales y no funcionales de la solución a desarrollar. Además la planificación permitió asignar el tiempo necesario para realizar cada una de las historias de usuarios.

Capítulo: 3 Implementación y prueba de la solución propuesta

3.1 Introducción

La metodología XP no propone concisamente los artefactos a utilizar en la implementación de una solución que utilice la misma. Deja en manos del equipo de desarrollo, dependiendo de sus capacidades de comunicación, la decisión de utilizar tantos tipos de diagramas como crea posible y así, facilitar el proceso de desarrollo. En este capítulo se presenta la descripción de la arquitectura, los diagramas utilizados, las tareas de ingeniería generados por cada historia de usuario y el proceso de pruebas.

3.2 Descripción de la Arquitectura

3.2.1 Moodle

La plataforma Moodle es fácil de instalar, aprender y modificar. Evita usar un diseño orientado a clases, con la finalidad de mantenerlo fácil de entender para los principiantes. La reutilización del código se archiva en bibliotecas con funciones claramente tituladas y con una disposición de los archivos de script consistente. Tiene una serie de características modulares, incluyendo temas, actividades, interfaces de idioma, esquemas de base de datos y formatos de cursos. Esto le permite a cualquier usuario añadir características al código básico principal o incluso distribuirlas por separado. Mantiene todos los archivos para un curso en un único directorio en el servidor. Esto permite que el administrador de un sistema proporcione similares formas de acceso a un nivel de archivo para cada profesor. Todas estas características permiten que Moodle pueda ser usado junto a otros sistemas como OpenSim (Moodle, 2006).

La figura 12 muestra un ejemplo del diagrama de ejecución de un entorno de Moodle.

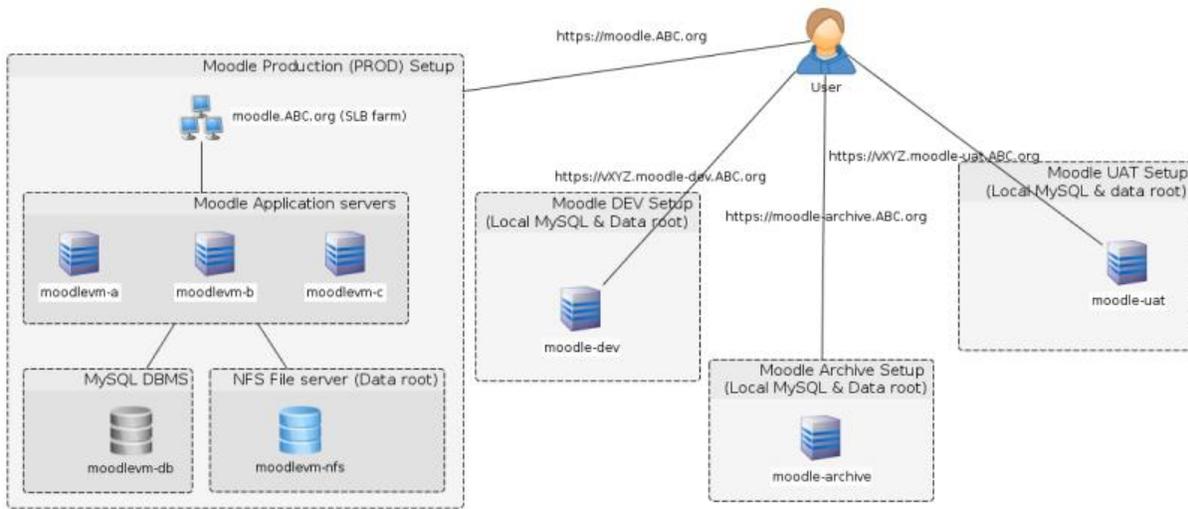


Figura 12. Diagrama de ejecución de Moodle.

Es necesario conocer también la arquitectura del servidor OpenSim para conocer mejor el funcionamiento del mismo.

3.2.2 OpenSim

OpenSimulator posee dos modos de ejecución comunes, standalone (solitario en una PC) y grid (rejilla). Se ejecutan múltiples servicios, servicio de autenticación, servicio de grid, servicio de cuentas de usuario, servicio del inventario, servicio de los objetos virtuales y otros servicios menos importantes. Luego de ejecutarse estos servicios se crean las regiones.

Si se está ejecutando en modo standalone todas las regiones se ejecutan en la misma PC que ejecuta los servicios. En la figura 13 se muestra un ejemplo.

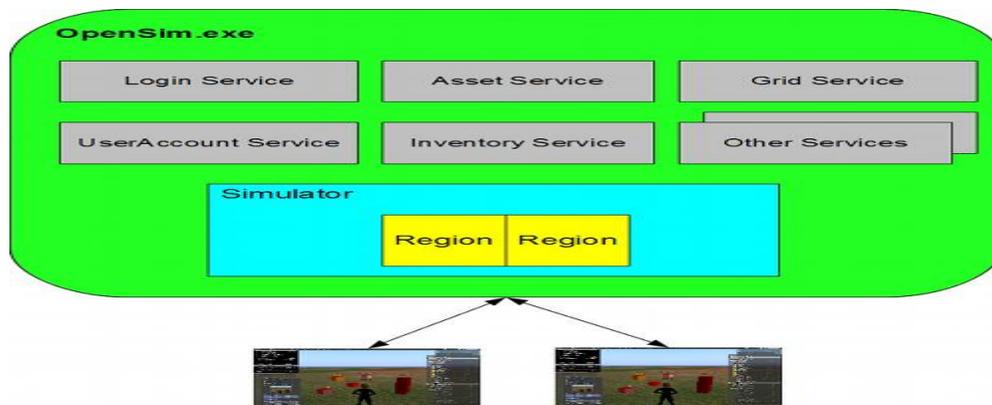


Figura 13. Modo Standalone de OpenSim.

Si se está ejecutando en modo *grid*, figura 14, la PC que ejecuta el Robust.exe es la que se encarga de ejecutar los servicios y el resto de las computadoras que se conecten en modo grid proveen la creación y conexión de las regiones.

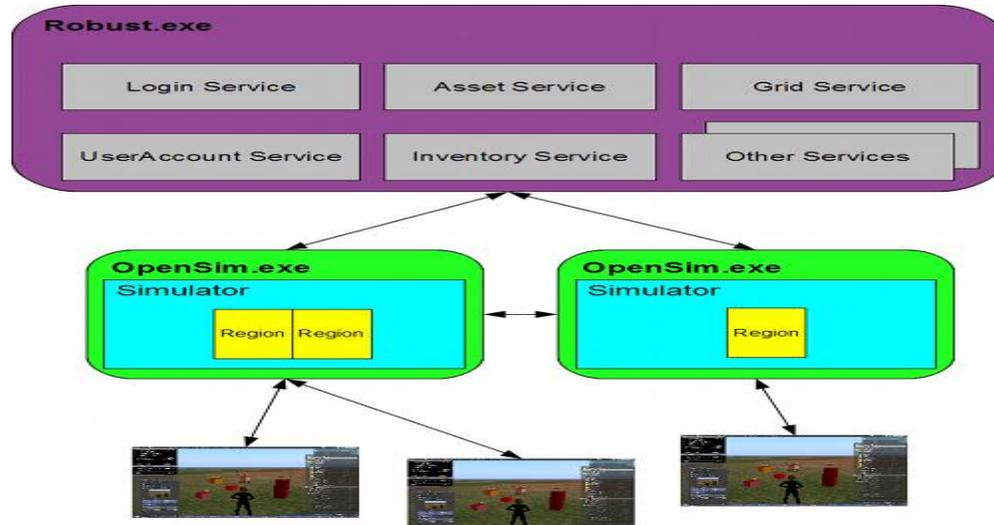


Figura 14. Modo Grid de OpenSim.

En las regiones existen objetos compuestos por *Prims* y, en versiones actuales, tienen soporte para *mesh* (malla), una tecnología de 3D que facilita la construcción y el trabajo de objetos. Estos *prims* pueden tener asociados *scripts* que se desarrollan en LSL y que poseen múltiples capacidades de comunicación en el mundo virtual y soporte para comunicación externa usando la interfaz HTTP.

3.3 Implementación

Como buena práctica de desarrollo de *software*, XP propone una arquitectura lo más flexible posible para comenzar el desarrollo, evitando grandes cambios para próximas iteraciones provocados por las modificaciones que el cliente por lo general propone.

3.3.1 Primera Iteración

En la primera iteración se pretende desarrollar las primeras cuatro HU, con el objetivo de obtener una primera versión con la que se pueda interactuar parcialmente entre el mundo virtual y la plataforma Moodle.

Se definen las siguientes tareas para esta iteración:

Historia de usuario N° 1 “Administrar el servidor y los objetos de Sloodle.”

1. Tarea N° 1: Crear el avatar para la administración del servidor.
2. Tarea N° 2: Crear y administrar los objetos de Sloodle.

Historia de usuario N° 2 “Crear laboratorio virtual.”

3. Tarea N° 3: Buscar modelos 3D para usar en el laboratorio.
4. Tarea N° 4: Crear laboratorio virtual e integrarlo al mundo virtual existente.

Historia de usuario N°3 “Conexión de OpenSim con Moodle y acceso de los usuarios al Mundo Virtual.”

5. Tarea N° 5: Crear los objetos de interconexión entre OpenSim y Moodle.
6. Tarea N° 6: Permitir la autenticación de los usuarios.

Historia de usuario N°4 “Vincular el avatar del usuario y enrolar los estudiantes.”

7. Tarea N° 7: Vincular el avatar de OpenSim con la cuenta de Moodle.
8. Tarea N° 8: Enrolar a los estudiantes en los cursos de Moodle desde OpenSim.

Tabla 7. Tarea 1: Crear el avatar para la administración del servidor.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Crear el avatar para la administración del servidor.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 2 de febrero de 2015	Fecha fin: 3 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: En la consola del servidor se utiliza el comando <i>create user</i> para crear avatares.	

Tabla 8. Tarea 2: Crear y administrar los objetos de Sloodle.

Tarea	
Número de tarea: 2	Número de HU: 1

Nombre de la tarea: Crear y administrar los objetos de Sloodle.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 4 de febrero de 2015	Fecha fin: 6 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El objeto <i>set de sloodle</i> es el encargado de crear y administrar las instancias de los objetos de Sloodle.	

La descripción de las tareas siguientes se encuentra en el Anexo 2.

3.3.2 Segunda Iteración

En la segunda iteración se pretende desarrollar las cuatro HU siguientes, con el objetivo de obtener una segunda versión que logre una mejor interacción del mundo virtual con la plataforma Moodle.

Se definen las siguientes tareas para esta iteración:

Historia de usuario N° 5 “Vincular los recursos chat y glosario.”

9. Tarea N° 9: Vincular la sala de chat.

10. Tarea N° 10: Acceder al glosario de Moodle desde el Entorno Virtual.

Historia de usuario N° 6 “Realizar preguntas de selección y permitir autenticación rápida.”

11. Tarea N° 11: Vincular las preguntas de selección de los cursos.

12. Tarea N° 12: Permitir autenticación rápida de los usuarios luego de haberse registrado.

Historia de usuario N° 7 “Realizar exámenes.”

13. Tarea N° 13: Realizar exámenes con preguntas de selección múltiples.

14. Tarea N° 14: Realizar exámenes de verdadero y falso.

15. Tarea N° 15: Realizar exámenes de repuesta rápida.

16. Tarea N° 16: Realizar exámenes del numérico.

Historia de usuario N° 8 “Permitir obtención de objetos por bonificación y el seguimiento a los usuarios.”

17. Tarea N° 17: Permitir adquirir objetos mediante la obtención de bonificaciones por respuestas.
18. Tarea N° 18: Realizar tareas de seguimientos del Usuario.

La descripción de cada una de estas tareas se encuentra en el Anexo 3.

3.3.3 Tercera Iteración

En la segunda iteración se pretende desarrollar las últimas tres HU, con el objetivo de obtener una tercera versión que logre la interacción deseada del mundo virtual con la plataforma Moodle.

Se definen las siguientes tareas para esta iteración:

Historia de usuario N° 9 “Visualizar los contenidos de Moodle desde el entorno virtual.”

19. Tarea N° 19: Visualizar contenido de audio y video.
20. Tarea N° 20: Visualizar contenido web e imágenes.

Historia de usuario N° 10 “Internacionalizar los componentes de Sloodle.”

21. Tarea N° 21: Internacionalizar los componentes dentro del mundo virtual.
22. Tarea N° 22: Internacionalizar los componentes dentro de Moodle.

La descripción de cada una de estas tareas se encuentra en el Anexo 4.

Luego de haber analizado y descrito cada una de las tareas a realizar durante el desarrollo de la tesis se hace necesario ejemplificar los patrones de diseño utilizados.

3.4 Patrones de diseño

Un patrón de diseño se define como “... un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos”. Los patrones de diseño comunican los estilos y soluciones consideradas como “buenas prácticas”, que los expertos en el diseño orientado a objetos utilizan para la creación de sistemas (Craig Larman, 2003).

Se clasifican en patrones GRASP y GoF. Los patrones GRASP describen principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Craig, 2004). Mientras que los patrones GoF se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (E

Gamma, 1994).

Durante el desarrollo de la solución propuesta se identificaron y se hizo uso de diversos patrones para mejorar la calidad y facilitar el mantenimiento posterior de la misma.

3.4.1 Patrones GRASP

En el Anexo 5 se muestran todos los patrones y sus descripciones.

Creador: El sistema cuenta con un objeto central encargado de administrar y crear las instancias de los objetos de Sloodle en OpenSim, este objeto es denominado *set de sloodle* y contiene toda la información de las instancias construidas o que fueron ocultadas, permitiendo modificar libremente los objetos mientras él almacena la información modificada. Es capaz de realizar varias operaciones sobre los objetos y es el encargado final de la creación y destrucción de los objetos, por lo que es imposible eliminar completamente una instancia de un objeto sin usar el *set de sloodle*.

Controlador: Este patrón fue usado durante la construcción de todo el sistema y, exceptuando un elemento de Sloodle que funciona completamente en el mundo virtual, fue utilizado para separar la lógica del negocio de la interfaz con la que interactúa el usuario. Todos los objetos de Sloodle están asociados a una clase *linker.php* correspondiente encargada de ejecutar el código de Moodle necesario y de transferir la información necesaria. En el objeto *silla de exámenes* se evidencia el uso de este patrón al analizar su *linker.php* correspondiente donde se verifica que este archivo es el encargado de recibir las respuestas propuestas del usuario en el mundo virtual, se las entrega a Moodle, el cual las procesa y califica y luego envía esta calificación y cualquier mensaje de retroalimentación al usuario del mundo virtual.

Indirección: Existen algunos objetos que tienen un script de apoyo con el objetivo de reducir el acoplamiento y aumentar la cohesión, mejorando la comprensión del funcionamiento del objeto separando sus responsabilidades. El ejemplo donde más se evidencia este patrón es analizando el objeto *silla de exámenes*, el cual, además del script principal responsable de controlar el flujo de ejecución y de configurar el objeto, tiene 3 *scripts* adicionales para reducir el acoplamiento y acomodar las responsabilidades:

`sloodle_quiz_ui`, `sloodle_quiz_question_handler`, `sloodle_quiz_chair_poseball`

Variaciones Protegidas: Para disminuir los impactos de cambios y modificaciones en el

desarrollo o mantenimiento de la aplicación se trata de mantener la información que se envía entre los objetos constante, modificando la forma en la que se maneja esta información solamente. Debido a esto se cuenta con una forma universal de manejo de errores, de traslación, y de configuración, usando la sintaxis estándar *do:command* para estas operaciones.

3.4.2 Patrones GoF

En el Anexo 6 se muestran los patrones y sus categorías.

A pesar de no poseer una organización orientada a objetos y que gran parte de la funcionalidad de integración se encuentra en *scripts* se hizo uso de algunos de los patrones GOF: Observer, State, (module).

Observer: Este patrón se implementó para garantizar una integración completa entre los elementos de OpenSim y de Moodle. Es necesario garantizar que exista una comunicación simultánea entre la interfaz de Moodle y la interfaz de OpenSim. El ejemplo más destacado del uso de este patrón es para la vinculación de la sala de chat con el chat de OpenSim. Es necesario que la sala de chat de Moodle muestre el chat de las personas de OpenSim y viceversa, esta es una situación idónea para la implementación de este patrón.

```

373 touch_start( integer total_number)
374 {
375     // Activating this requires access permission
376     if (sloodle_check_access_ctrl(llDetectedKey(0)) == FALSE) {
377         sloodle_translation_request(SLOODLE_TRANSLATE_SAY, [0],
378             "nopermission:ctrl", [llDetectedName(0)], NULL_KEY, "");
379         return;
380     }
381     llListenRemove(listenctrl);
382     listenctrl = llListen(SLOODLE_CHANNEL_AVATAR_DIALOG, "",
383         llDetectedKey(0), "");
384     sloodle_translation_request(SLOODLE_TRANSLATE_DIALOG,
385         [SLOODLE_CHANNEL_AVATAR_DIALOG, "0", "1"], "webintercom:ctrlmenu",
386         ["0", "1"], llDetectedKey(0), "webintercom");
387     llSetTimerEvent(10.0);
388 }

```

Figura 15. Fragmento de código ejemplificando el patrón Observer.

Este segmento de código, figura 15, se encarga de registrar la comunicación de los usuarios y procesarla, para luego mostrarla en la sala de chat de Moodle.

```

564
565     timer()
566     {
567         // Get updated chat from Moodle
568         if (httpchat == NULL_KEY) {
569             string body = "sloodlecontrollerid=" +
570                 (string)sloodlecontrollerid;
571             body += "%sloodlepwd=" + sloodlepwd;
572             body += "%sloodlemoduleid=" + (string)sloodlemoduleid;
573
574             httpchat = llHttpRequest(sloodleserverroot +
575                 SLOODLE_CHAT_LINKER, [HTTP_METHOD, "POST", HTTP_MIMETYPE,
576                 "application/x-www-form-urlencoded"], body);
577
578         }
579
580         // Purge any expired command dialogs
581         sloodle_purge_cmd_dialog();
582     }
583
584     http_response(key id, integer status, list meta, string body)

```

Figura 16. Fragmento de código ejemplificando el patrón Observer.

Este segmento, figura 16, es el encargado de recibir la información de la sala de chat cada un intervalo de tiempo, el evento definido al final del segmento, *http_response*, es el encargado de procesar la información recibida por HTTP de Moodle conteniendo el estado actual de la sala de chat.

```

40  $sloodle = new SloodleSession();
41  $sloodle->authenticate_request();
42  $sloodle->load_module('chat', true);
43  // Attempt to validate the user
44  // (this will auto-register/enrol users where necessary and allowed)
45  // If server access level is public, then validation is not essential... otherwise, it is
46  $sloodleserveraccesslevel = $sloodle->request->get_server_access_level(false);
47  if ($sloodleserveraccesslevel == 0) $sloodle->validate_user(false);
48  else $sloodle->validate_user(true);
49
50
51  $sloodle->validate_requirements();
52
53
54  // Has an incoming message been provided?
55  $message = sloodle_clean_for_db($sloodle->request->optional_param('message', null));
56  if ($message != null) {
57      // Add it to the chatroom - if it fails add a negative side effect code to our response.
58      // The positive side effect will be added by the function if successful.
59      if (!$sloodle->module->add_message($message)) {
60          $sloodle->response->add_side_effect(-10101);
61          add_to_log($sloodle->course->get_course_id(), 'sloodle', 'add message', '', 'Failed to add chat message to
62      } else {
63          add_to_log($sloodle->course->get_course_id(), 'sloodle', 'add message', '', 'Added chat message to chatroom
64      }
65  }

```

Figura 17. Fragmento de código ejemplificando el patrón Observer.

Y finalmente es en este segmento de código, figura 17, donde se accede a la sala de chat y se actualiza la información con la generada en el mundo virtual. Un vistazo rápido a las dos últimas líneas de código revela que el objeto que hace de observer en este caso es el log de Moodle

State: Este patrón se utiliza para dar un control más fino a los *scripts* de los objetos de OpenSim. Debido a que el lenguaje LSL no puede usar una POO ya que es orientado a *scripts* y rutinas se utiliza este patrón para definir diferentes formas de interacción y de ejecución según el estado del objeto.

```

261     state ready
262     {
263         state_entry()
264         {
265             // Clear existing data
266             clearUserQuizData();
267             //tell other scripts we are in the ready
                state
268             llMessageLinked(LINK_SET,
                SLOODLE_CHANNEL_QUIZ_STATE_ENTRY_READY, "", NULL_KEY);
269             // This is now handled by a separate poseball
270             // llSitTarget(<0,0,.5>, ZERO_ROTATION);
271         }
272
273         // Wait for the script that handles the sitting to tell us that
                somebody has sat on us.
274         // Normally a sit will immediately produce a link message
275         // But variations on the script may do things differently,
276         // eg. the awards script doesn't want to start the quiz until
                it's got a Game ID
277         link_message(integer sender_num, integer num, string str, key id)
278         {
279             if (num == SLOODLE_CHANNEL_QUIZ_START_FOR_AVATAR) {
280
281                 sitter = id;
282                 // Make sure the given avatar is allowed to use this
                object
283                 if (!sloodle_check_access_use(sitter)) {
284                     sloodle_translation_request(SLOODLE_TRANSLATE_SAY,
                [0], "nopermission:use", [llKey2Name(sitter)],
                NULL_KEY, "");

```

Figura 18. Fragmento de código ejemplificando el patrón State.

```

307 // Fetching the general quiz data from the server
308 state load_quiz_for_user
309 {
310     state_entry()
311     {
312         llMessageLinked(LINK_SET,
313             SLOODLE_CHANNEL_QUIZ_STATE_ENTRY_LOAD_QUIZ_FOR_USER, "",
314             sitter);
315         sloodle_translation_request(SLOODLE_TRANSLATE_SAY, [0],
316             "fetchingquiz", [], NULL_KEY, "quiz");
317         // Request the quiz data from Moodle
318         sloodlehttpvars = "sloodlecontrollerid=" +
319             (string)sloodlecontrollerid;
320         sloodlehttpvars += "%sloodlepwd=" + sloodlepwd;
321         sloodlehttpvars += "%sloodlemoduleid=" +
322             (string)sloodlemoduleid;
323         sloodlehttpvars += "%sloodleuuid=" + (string)sitter;
324         sloodlehttpvars += "%sloodleserveraccesslevel=" +
325             (string)sloodleserveraccesslevel;
326         sloodlehttpvars += "%sloodleavname=" +
327             llEscapeURL(llKey2Name(sitter));
328         httpquizquery = llHTTPRequest(sloodleserverroot +
329             sloodle_quiz_url, [HTTP_METHOD, "POST", HTTP_MIMETYPE,
330             "application/x-www-form-urlencoded"], sloodlehttpvars);
331         llSetTimerEvent(0.0);
332         llSetTimerEvent((float)request_timeout);
333     }
334
335     state_exit()
336     {
337         llSetTimerEvent(0.0);
338     }
339 }

```

Figura 19. Fragmento de código ejemplificando el patrón State.

```

331 timer()
332 {
333     sloodle_translation_request(SLOODLE_TRANSLATE_SAY, [0],
334         "httptimeout", [], NULL_KEY, "");
335     state ready;
336 }
337 link_message(integer sender_num, integer num, string str, key
338 id) {
339     if (num == SLOODLE_CHANNEL_QUIZ_STOP_FOR_AVATAR){
340         state ready;
341     }
342 }

```

Figura 20. Fragmento de código ejemplificando el patrón State.

A continuación, figura 21, se muestra el Diagrama de Despliegue para un mejor entendimiento del funcionamiento del sistema.

3.5 Diagrama de despliegue

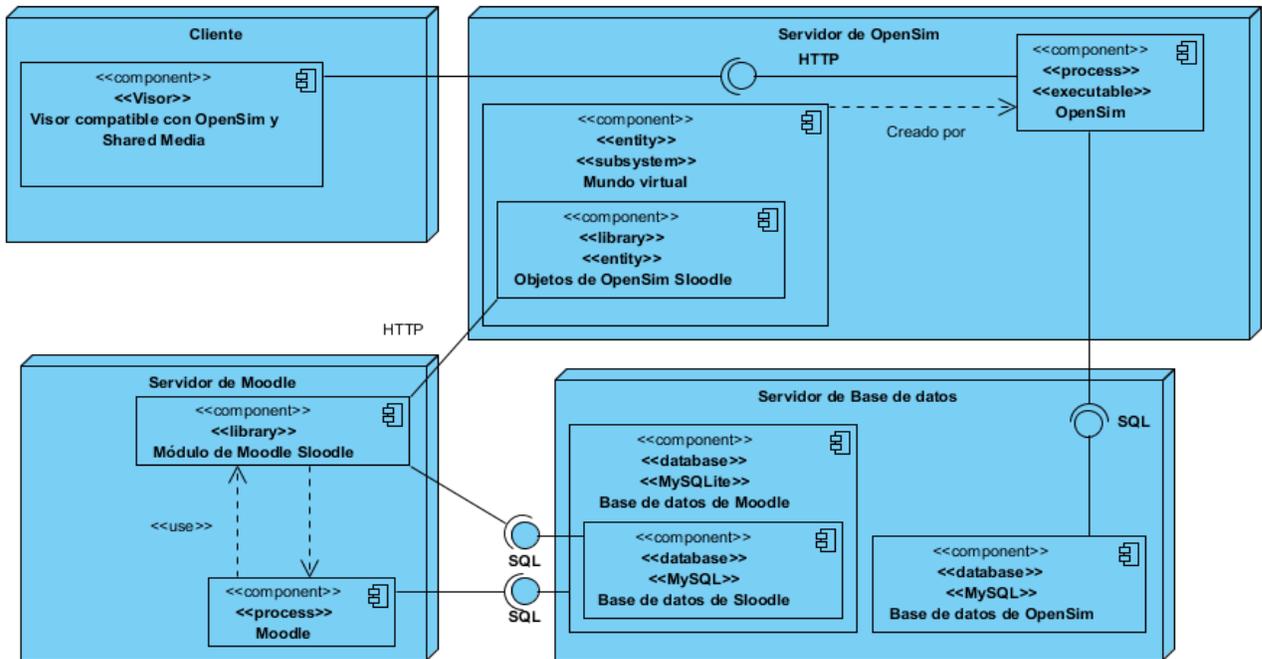


Figura 21. Diagrama de despliegue.

El Diagrama de Despliegue muestra cómo funciona el sistema mostrando como los componentes y procesos son configurados para la ejecución en los nodos de procesamiento.



Figura 22. Nodo Cliente.

El cliente necesita para conectarse a la aplicación de un Visor compatible con la interfaz propuesta por OpenSim y para poder utilizar todas las funcionalidades de la aplicación es necesario que tenga soporte para Shared Media, una tecnología de vanguardia que implementa OpenSim y que es usada por el mundo virtual en el Presenter y en otros objetos

para mostrar información y mejorar la comunicación. El cliente debe ser capaz de conectarse por HTTP a el servidor OpenSim, la configuración por defecto establece que el puerto de conexión inicial es el puerto 9000, algunos visores tienen soporte para configuraciones avanzadas, permitiendo el uso de proxies y similares.

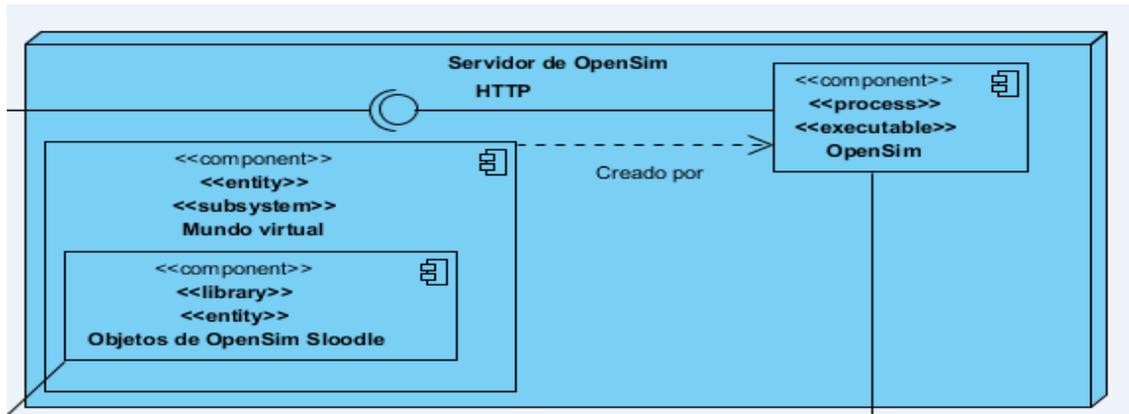


Figura 23. Nodo Servidor de OpenSim.

En el servidor de OpenSim debe ejecutarse el servidor con el ejecutable OpenSim.exe, el cual crea un proceso que genera el Mundo Virtual y permite administrarlo. OpenSim tiene soporte para SQLite, MySQL, MSSQL y PGSQL, en esta aplicación se usó una base de datos MySQL. Dentro de este mundo virtual se deben crear los Objetos de OpenSim que provee Sloodle los cuales se conectarán con el módulo de Moodle Sloodle y permitirán una interacción constante entre los mismos.

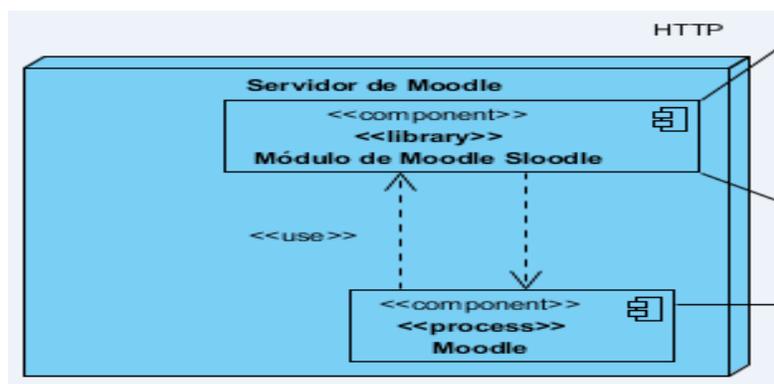


Figura 24. Nodo Servidor de Moodle.

En el servidor de Moodle se debe tener acceso administrativo a Moodle para poder instalar el módulo de Sloodle, el cual proveerá diversos servicios a los objetos de OpenSim

mediante conexiones HTTP, tal y como si se estuviese trabajando desde un navegador.

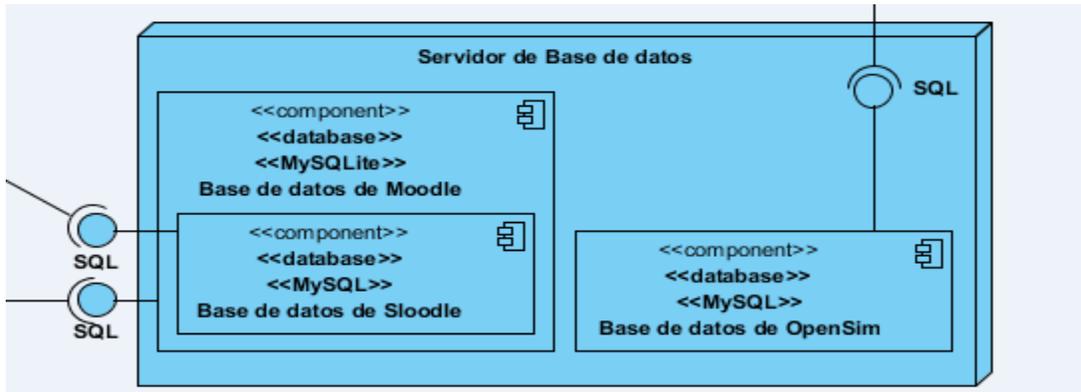


Figura 25. Nodo Servidor de Bases de datos.

Finalmente se debe contar con un servidor de Base de datos que administre las bases de datos de OpenSim, Sloodle y Moodle. Todas estas bases de datos usan la interfaz SQL para recibir peticiones.

Luego de haber profundizado en el funcionamiento del sistema es necesario pasar a la etapa siguiente, las pruebas de *software* para poder verificar que producto cumpla con los requerimientos del cliente.

3.6 Pruebas

Las pruebas del *software* son un elemento crítico para la garantía de calidad del *software* y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, 2002). Tienen como objetivo principal el descubrir errores en el *software* realizado, el éxito de dichas pruebas está en encontrar alguna deficiencia al producto que se esté desarrollando, estas deben planificarse mucho antes de que comiencen a realizarse y deben comenzar desde lo pequeño y progresan hasta lo más grande.

Para garantizar la calidad del software se realizaron pruebas al mismo pero todas tuvieron que realizarse a la interfaz de forma manual. Al utilizar un lenguaje orientado a eventos y debido a que el código PHP encargado de la integración no funciona con objetos y clases, realizar pruebas automatizadas sería demasiado costoso puesto que cada una entorpecería el código de la lógica de negocio trayendo consecuencias negativas posteriormente (Software Testing Fundamentals, 2011). Por todo lo dicho anteriormente se realizaron pruebas de aceptación al *software*.

3.6.1 Pruebas de aceptación

Las pruebas de aceptación son realizadas una vez terminado e integrado el producto que se esté llevando a cabo, pero lo que la diferencia de las demás pruebas que se le realiza a un *software* es que estas están concebidas para que sea el cliente o usuario final quien detecte los posibles errores que pueda presentar el producto. Estas pruebas están enfocadas principalmente a probar las historias de usuario, es decir a demostrar que dichas historias de usuarios no cumplen con lo acordado con el cliente.

Destacar que se toma como resultado satisfactorio en la evaluación de la prueba, los casos donde el resultado esperado y el resultado de la evaluación sean el mismo.

A continuación, se muestra una de las pruebas de aceptación realizadas en la **primera iteración de la solución propuesta**, las demás se encuentran en el anexo 7. Se realizaron siete casos de prueba de aceptación, de las cuales dos arrojaron resultados no satisfactorios.

Tabla 9. CP Creación del avatar de administración del servidor de OpenSim.

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Creación del avatar de administración del servidor de OpenSim.	
Descripción: Prueba para verificar la creación del avatar de administración.	
Condiciones de Ejecución: El usuario debe haber ejecutado el servidor OpenSim previamente.	
Entrada/ Pasos de ejecución: Al ejecutarse el servidor se le muestra al usuario los parámetros de creación del avatar.	
Resultado Esperado: Se crea en el mundo virtual el avatar de administración.	
Evaluación de la Prueba: Prueba satisfactoria.	

En la **segunda iteración de la solución propuesta** se realizaron todas las pruebas de aceptación de la primera iteración para identificar si algún cambio realizado afectase las

funcionalidades anteriores, todas arrojaron resultados satisfactorios. Después de realizados estos casos de prueba se realizaron los once casos de pruebas correspondientes con la segunda versión del sistema, de estas dos arrojaron resultados no satisfactorios. Anexo 8.

En la **tercera iteración de la solución propuesta** se realizaron todas las pruebas de aceptación de las iteraciones anteriores para identificar si algún cambio realizado afectase las funcionalidades anteriores, todas arrojaron resultados satisfactorios. Después de realizados estos casos de prueba se realizaron los dos casos de pruebas correspondientes con la segunda versión del sistema arrojando resultados satisfactorios. Anexo 9. En la figura 26 se muestran los resultados obtenidos en forma de gráfica en las tres iteraciones.

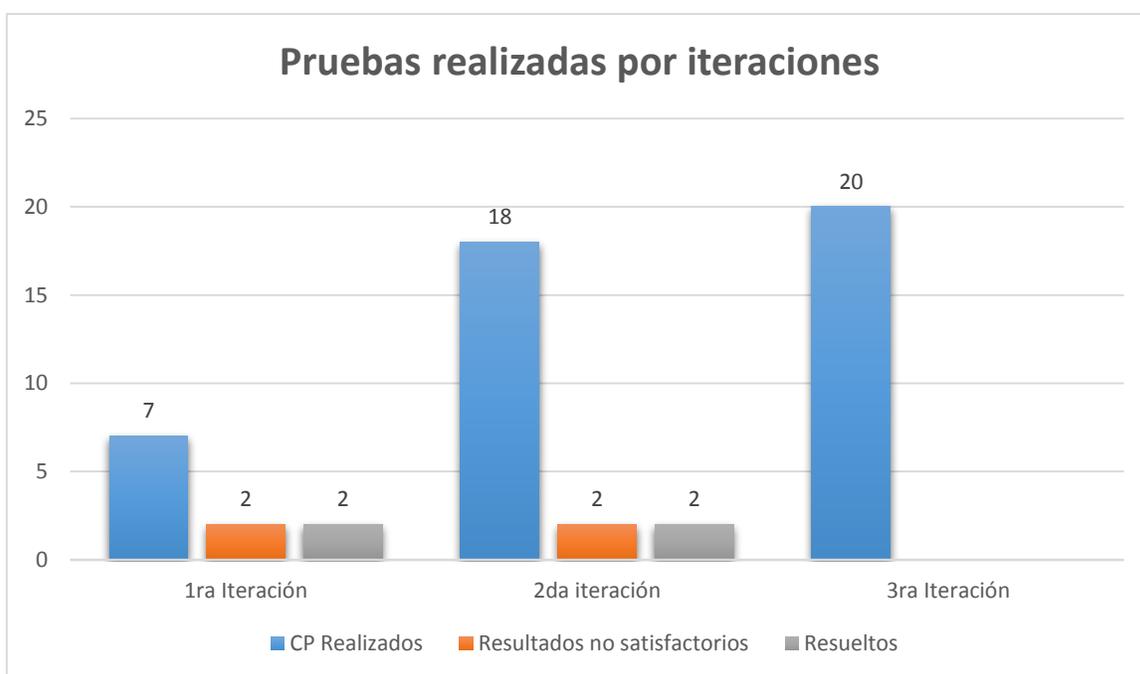


Figura 26. Resultados de las pruebas de aceptación.

Luego de haber realizado las pruebas, todas satisfactorias después de la tercera iteración, se obtuvo una carta de aceptación expresando la satisfacción del cliente con el producto desarrollado.

3.7 Conclusiones parciales

Realizadas las iteraciones, se reconoció la importancia que tuvo la organización su desarrollo por tareas, evitando duplicar esfuerzos por parte del equipo de desarrollado. A medida que se implementaban las tareas de las iteraciones, se lograba un avance y orden

Capítulo 3

en el proceso de desarrollo de la solución; el tiempo que se le asignó a cada una de estas iteraciones fue justo el que se necesitó para su desarrollo. Se realizaron pruebas de caja negra o aceptación, lo que permitió que el cliente obtuviera un mejor resultado en el producto final de la aplicación. Todas las pruebas que se realizaron tuvieron éxito.

Conclusiones generales

Con la realización del presente trabajo se logró sentar las bases para la integración de un mundo virtual con la plataforma Moodle utilizando la biblioteca Sloodle sin soporte desde el año 2012. Luego de analizado el proceso de desarrollo de *software* se llegaron a las siguientes conclusiones:

- ✓ Se realizó el análisis del estado del arte para adoptar desde el punto de vista teórico las definiciones de: mundos virtuales, LMS, Moodle, interacción e integración.
- ✓ Utilizando el ambiente de desarrollo y realizando las correcciones pertinentes en la biblioteca Sloodle se logró la integración a partir de las necesidades detectadas en la planificación de la investigación.
- ✓ Al término del proceso de pruebas realizado a la solución propuesta, mediante el empleo de pruebas de aceptación, se demostró la correcta implementación de sus funcionalidades, dando cumplimiento a las necesidades y expectativas del cliente.

Recomendaciones

Para dar continuación con el desarrollo de la aplicación se recomienda:

- ✓ Debe ser posible editar el nombre de las tareas asociadas al objeto seguidor y añadir soporte a los demás tipos de preguntas de los exámenes.
- ✓ Incluir poder visualizar y manipular paquetes scorm desde el mundo virtual.

Glosario de Términos

Avatares: Personas virtuales que representa a un individuo real en el metaverso.

Ciberespacio: Ámbito artificial creado por medios informáticos.

OpenSim: Servidor 3D de código abierto que permite crear ambientes virtuales.

SCORM: Conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados.

Mundo Virtual: Un ambiente simulado en una computadora en el cual los usuarios son capaces de sumergirse y dentro de los cuales son capaces de, mediante sus avatares (representaciones computarizadas de ellos o de personas alternativas), experimentar, manipular, interactuar con y/o crear objetos virtuales y lugares que están gráficamente mostrados en tres dimensiones. Los objetos y lugares dentro de un mundo virtual pueden ser modelados acorde a aquellos en el mundo real o pueden estar basados en una fantasía. La mayoría de las aplicaciones de los mundos virtuales permiten múltiples usuarios e incluyen facultades que permiten a los usuarios comunicarse e interactuar unos con los otros dentro del ambiente virtual. Posee sincronía y persistencia.

- ✓ Sincronía: los elementos del mundo virtual deben existir simultáneamente en el mismo espacio tiempo para todos los usuarios.
- ✓ Persistencia: el mundo virtual debe existir y continuar funcionando de manera síncrona y constante incluso si todos los usuarios se retiran del mismo.

Licencia BSD: Berkeley *Software* Distribution, licencia de *software* libre que nos permite ver el código y modificarlo pero también cerrar el sistema o la aplicación.

Interacción: Acción que se ejerce recíprocamente entre dos o más objetos, agentes, fuerzas, funciones, etc.

Integrar: Hacer que alguien o algo pase a formar parte de un todo.

IDEs: Entorno de desarrollo integrado es una aplicación de *software*, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de *software*.

HTTPS: En el protocolo HTTP las URLs comienzan con “http://” y utilizan por defecto el puerto 80, Las URLs de HTTPS comienzan con “https://” y utilizan el puerto 443.

OSSL: Extensión de LSL, el lenguaje de scripting en OpenSimulator y en Second Life.

MOAP: Medios en un prim se implementó en OpenSimulator desde 0.7.1. Esta facilidad le permite visualizar y navegar las páginas web de la superficie / textura de un prim.

Bibliografía

2013. ¿Qué son las historias de usuario?: 7 preguntas y respuestas - La Oficina de Proyectos de Informática. [En línea] 24 de 4 de 2013. [Citado el: 11 de 12 de 2013.] <http://www.pmoinformatica.com/2013/04/que-son-las-historias-de-usuario-7.html>.

AIDA. 2011. Guía de iniciación Linden Scripting Language. [En línea] 2011. [Citado el: 2 de febrero de 2015.] http://aida.ii.uam.es/virtuam/wp-content/uploads/Descargas/Tutoriales/guia_sobre_linden_scripting_language.pdf.

Álvarez, Solís y Díaz, Figueroa. 2007. Metodologías Tradicionales vs. Metodologías Ágiles. [En línea] 2007. http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.1515.

Álvarez, Solís; Díaz, Figueroa. 2007. Metodologías Tradicionales vs. Metodologías Ágiles. [En línea] 2007. http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.1515.

Apache. 2013. Apache. [En línea] 2013. [Citado el: 15 de enero de 2015.] http://httpd.apache.org/docs/2.0/new_features_2_0.html.

CALIDAD. 2008. *D. D. Diagnostico 2008 - Resultado de la revisión UCI - Herramientas de modelado, Universidad de las Ciencias Informáticas.* s.l. : 3.5.4, 2008.

Carlos, J. 2010. *Intercambios virtuales en busca del conocimiento.* 2010.

Carlos, J. 2010. Intercambios virtuales en busca del conocimiento. [En línea] 2010.

Company Headquarters. 2012. Visual Paradigm. [En línea] 2012. [Citado el: 25 de enero de 2015.] <http://www.visual-paradigm.com>.

Craig Larman. 2003. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* s.l. : Pearson Educación, 2003. ISBN 9788420534381.

Craig, Larman. 2004. *UML y patrones.* s.l. : Prentice Hall., 2004.

Cruz, Albano. 2010. Comunidades virtuales. Metaversos y Second Life. [En línea] abril de 2010. [Citado el: 16 de enero de 2015.] http://www.pacap.net/es/publicaciones/pdf/comunidad/12/recursos_3.pdf.

Diana Guerra, Jhonatan Villacís, Ramiro Delgado, Diego Marcillo. 2010.

Bibliografía

Implementación de Mundos Virtuales como apoyo a la formación por competencias de la carrera de Ingeniería de Sistemas e Informática. 2010.

E Gamma, E. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software.* s.l. : Addison-Wesley Professional. 416, 1994.

e-ABC. 2011. Definición de e-Learning. [En línea] 2011. [Citado el: 15 de enero de 2015.] <http://www.e-abclearning.com/definicion-e-learning>.

EdITLib. 2012. Managing 3D Multi User Learning Environments - A Case Study on Training Disaster Management. [En línea] 2012. <http://www.editlib.org/p/45290/>.

Enrique González. 2015. ¿Es PHP un buen lenguaje de programación? ¿Cuáles son las versiones de PHP? Historia. [En línea] 2015. [Citado el: 11 de febrero de 2015.] <http://www.aprenderaprogramar.com>.

Escartín, Emilio R. La realidad virtual, una tecnología educativa a nuestro alcance. [En línea] [Citado el: 15 de enero de 2015.] <http://www.sav.us.es/pixelbit/pixelbit/articulos/n15/n15art/art151.htm>.

González Seco, Jose Antonio. 2011. El Lenguaje de Programación C#. [En línea] 2011. [Citado el: 30 de enero de 2015.] <http://es.slideshare.net/catalan21/lenguaje-de-programacion-c-10408424>.

Hermes Romero. 2012. [En línea] 2012. <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>.

Hernández Sampieri, Roberto. 2006. *Metodología de la investigación.* Mexico, DF : s.n., 2006.

Hidalgo, Diana Guerra. Repositorio Digital ESPE. Implementación de mundos virtuales como apoyo a la formación por competencias de la carrera de Ingeniería de Sistemas e Informatic. [En línea] [Citado el: 3 de febrero de 2015.] <http://repositorio.espe.edu.ec/bitstream/21000/5233/1/AC-SISTEMAS-ESPE-033179.pdf>.

Imprudence Viewer. 2013. [En línea] Informer Technologies, 2013. [Citado el: 4 de febrero de 2015.] <http://imprudence-viewer.software.informer.com/>.

Ingeniería de Software. Programación Extrema XP. [En línea] [Citado el: 20 de enero de 2015.] http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.

Integrating Virtual Worlds and Virtual Learning Environments through Sloodle: from theory to practice in a case of study for teaching of algorithms. **Felipe Becker Nunes, Gleizer**

Bibliografía

Bierhalz Voss, Fabrício Herpich, Érico Marcelo Hoff do Amaral, Roseclea Duarte Medina, Liane M. R. Tarouco. 2013. Brasil : s.n., 2013.

ISEA. 2008. INTERNET 3D. *Análisis prospectivo de las potenciales aplicaciones asociadas a los Mundos Virtuales.* [En línea] octubre de 2008. [Citado el: 15 de enero de 2015.] http://www.iseamcc.net/eISEA/Vigilancia_tecnologica/informe_3.pdf.

JACOBSON, I. 2000. *El Proceso Unificado del Desarrollo de Software.* Malaga : s.n., 2000.

La Oficina de Proyectos de Informática. 2013. ¿Qué son las historias de usuario?: 7 preguntas y respuestas. [En línea] 24 de abril de 2013. [Citado el: 9 de marzo de 2015.] <http://www.pmoinformatica.com/2013/04/que-son-las-historias-de-usuario-7.html>.

Lic. Ervin Flores. 2015. AUP Ingeniería de Software. [En línea] 2015. [Citado el: 11 de febrero de 2015.] http://ingenieriadesoftware.mex.tl/63758_AUP.html.

Linden_Lab. 2015. Second Life. [En línea] 2015. <http://www.secondlife.com>.

Lopez Guzmán, Clara. 2005. *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. Tesis doctoral.* España. Salamanca : s.n., 2005.

Luzcando, Diego Riofrío. 2012. Archivo Digital UPM. Diseño e Implementación de un Laboratorio Virtual de Biotecnología. [En línea] 2012. [Citado el: 25 de enero de 2015.] http://oa.upm.es/13700/2/TESIS_MASTER_DIEGO_RIOFRIO_LUZCANO.pdf.

Magisterio Virtual. Del aprendizaje colaborativo al aprendizaje por inmersión. [En línea] [Citado el: 25 de febrero de 2015.] <http://www.magisterio.com.co>.

Mark J. W. Lee, Barney Dalgarno, Lauren Carlson. 2010. Australasian Journal of Educational Technology. [En línea] 2010. <http://ascilite.org.au/ajet/ajet27/dalgarno.html>.

Mayor, Alicia Cañellas. 2012. Centro de comunicación y pedagogía. [En línea] diciembre de 2012. [Citado el: 15 de abril de 2015.] <http://www.centrocp.com/lms-y-lcms-funcionalidades-y-beneficios/>.

Moodle. 2015. Moodle.net. [En línea] 2015. <https://moodle.net/stats/>.

—. **2014.** Moodle.org. [En línea] 2014. https://docs.moodle.org/all/es/Acerca_de_Moodle.

—. **2006.** Moodledocs. [En línea] 2006. https://docs.moodle.org/all/es/Arquitectura_de_Moodle.

OpenSim. 2012. OpenSimulator. [En línea] 2012. [Citado el: 4 de febrero de 2015.] <http://opensimulator.org>.

Bibliografía

- . **2012**. OpenSimulator. [En línea] 2012. <http://opensimulator.org/wiki/Arquitectura>.
- Pressman, Roger S. 2002**. *Ingeniería del Software. Un enfoque práctico*. Madrid : Concepción Fernández, 2002. 0-07-709677-0.
- Real Academia Española. 2012**. Real Academia Española. [En línea] 2012. <http://lema.rae.es/drae/?d=drae&val=avatar&x=0&y=0#>.
- . **2012**. Real Academia Española. [En línea] 2012. <http://lema.rae.es/drae/?d=drae&val=interacci%F3n&x=41&y=11>.
- . **2012**. Real Academia Española. [En línea] 2012. <http://lema.rae.es/drae/?d=drae&val=integrar&x=0&y=0#>.
- Rodríguez, Dra. Maria Morales. 2011**. Aplicaciones de los Mundos Virtuales. [En línea] 27 de septiembre de 2011. http://www.academia.edu/1709123/Aplicaciones_de_los_Mundos_Virtuales.
- Rubén González Crespo, Sergio Rios Aguilar, Roberto Ferro Escobar, Nicolás Torres. 2012**. *Dynamic, ecological, accessible and 3D Virtual Worlds-based Libraries using OpenSim and Sloodle along with mobile location and NFC for Checking in*. 2012.
- SCRUM. 2014**. Qué es SCRUM, proyectos Ágiles. [En línea] 2014. [Citado el: 25 de enero de 2015.] <http://www.proyectosagiles.org/que-es-scrum>.
- Sloodle. 2012**. Simulation Linked Object-Oriented Dynamic Learning Environmet. [En línea] 2012. [Citado el: 3 de febrero de 2015.] <http://www.sloodle.org>.
- Software Testing Fundamentals. 2011**. [En línea] Enero de 2011. <http://softwaretestingfundamentals.com/unit-testing/>.
- SOMMERVILLE. 2005**. *I. Ingeniería de Software.7ma*. Madrid : s.n., 2005.
- Stephenson, Neal. 1992**. *Snow Crash*. 1992.
- SXP, Metodología ágil para el Desarrollo de Software*. **G. Peñalver, A. Meneses, S. García. 2010**. Ciudad de La Habana, Cuba : s.n., 2010.
- Virtual Worlds Research: Past, Present & Future*. **Bell, Mark W. 2008**. 2008.

Anexos

Anexo 1: Historias de usuarios.

Tabla 10. HU 1: Administrar el servidor y los objetos de Sloodle.

Historia de Usuario	
No.: 1	Nombre: Administrar el servidor y los objetos de Sloodle.
Usuario: Administrador	
Prioridad en el Negocio: alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 1
Descripción: Debe ser posible acceder a la configuración del servidor y ser creados los objetos de Sloodle.	
Información adicional (Observaciones): Da cumplimiento a las funcionalidades “Crear el avatar para la administración del servidor” y “Crear y administrar los objetos de Sloodle”.	

Tabla 11. HU 2: Crear laboratorio virtual.

Historia de Usuario	
No.: 2	Nombre: Crear laboratorio virtual.
Usuario: Administrador	
Prioridad en el Negocio: alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 1
Descripción: El entorno virtual será un laboratorio que se incluirá en un entorno virtual mucho más grande en existencia y deberá poder comunicarse con los recursos del Moodle.	
Información adicional (Observaciones): Da cumplimiento a la funcionalidad “Crear laboratorio virtual e integrarlo al mundo virtual existente”.	

Tabla 12. HU 3: Conexión de OpenSim con Moodle y acceso de los usuarios al Mundo Virtual.

Historia de Usuario	
No.: 3	Nombre: Conexión de OpenSim con Moodle y acceso de los usuarios al Mundo Virtual.
Usuario: Profesor	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 1
Descripción: El sistema debe permitir creación de objetos que permitan la conexión entre los objetos de Moodle y OpenSim. Además el sistema debe permitir que los usuarios puedan acceder satisfactoriamente.	
Información adicional (Observaciones): Los usuarios deben estar creados previamente.	

Tabla 13. HU 4: Vincular el avatar del usuario y enrolar los estudiantes.

Historia de Usuario	
No.: 4	Nombre: Vincular el avatar del usuario y enrolar los estudiantes.
Usuario: Usuarios autenticados	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Baja
Estimación: 1 semana	Iteración Asignada: 1
Descripción: La propuesta de solución debe permitir que el profesor pueda vincular el usuario en OpenSim a una cuenta en Moodle. Además debe permitir a los estudiantes incorporarse a los cursos de Moodle desde OpenSim.	
Información adicional (Observaciones): Da cumplimiento a las funcionalidades “Vincular el avatar de OpenSim con la cuenta de Moodle” y “Enrolar a los estudiantes en los cursos de Moodle desde OpenSim”.	

Tabla 14. HU 5: Vincular los recursos chat y glosario.

Historia de Usuario	
No.: 5	Nombre: Vincular los recursos chat y glosario.
Usuario: Estudiante	
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema debe tener vinculada la sala de chat y debe permitir acceder al glosario que posee Moodle desde el Entorno Virtual.	
Información adicional (Observaciones): Debe existir al menos un curso con un controlador de Moodle. Da cumplimiento a las funcionalidades “Vincular la sala de chat” y “Acceder al glosario de Moodle desde el Entorno Virtual”.	

Tabla 15. HU 6: Realizar preguntas de selección y permitir autenticación rápida.

Historia de Usuario	
No.: 6	Nombre: Realizar preguntas de selección y permitir autenticación rápida.
Usuario: Estudiante	
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema debe tener vinculadas las preguntas de selección de los cursos de Moodle y además hay que realizar las actividades del tipo tarea.	
<ol style="list-style-type: none"> Información adicional (Observaciones): Debe existir al menos un curso con un controlador de Moodle. Da cumplimiento a las funcionalidades “Vincular las preguntas de selección de los cursos” y “Permitir autenticación rápida de los usuarios luego de haberse registrado”. 	

Tabla 16. HU 7: Realizar exámenes.

Historia de Usuario	
No.: 7	Nombre: Realizar exámenes.
Usuario: Estudiante	
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema deberá permitir la realización de exámenes de verdadero y falso y exámenes de selección múltiples.	
Información adicional (Observaciones): Debe existir al menos un curso con un controlador de Sloodle.	

Tabla 17. HU 8: Permitir obtención de objetos por bonificación y el seguimiento a los usuarios.

Historia de Usuario	
No.: 8	Nombre: Permitir obtención de objetos por bonificación y el seguimiento a los usuarios.
Usuario: Estudiantes	
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 2
Descripción: El sistema debe permitir que los usuarios puedan adquirir objetos mediante la obtención de bonificaciones por respuestas realizadas y además debe permitir poder hacer seguimiento a los usuarios.	
Información adicional (Observaciones): Da cumplimiento a las funcionalidades “Permitir adquirir objetos mediante la obtención de bonificaciones por respuestas” y “Realizar tareas de seguimientos del Usuario”.	

Tabla 18. HU 9: Visualizar los contenidos de Moodle desde el entorno virtual.

Historia de Usuario	
No.: 9	Nombre: Visualizar los contenidos de Moodle desde el mundo virtual.
Usuario: Estudiante	
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 2 semanas	Iteración Asignada: 3
Descripción: El sistema debe permitir que los contenidos de Moodle se visualicen en el Mundo Virtual.	
Información adicional (Observaciones):	

Tabla 19. HU 10: Internacionalizar los componentes de Sloodle.

Historia de Usuario	
No.: 10	Nombre: Internacionalizar los componentes de Sloodle.
Usuario: Administrador	
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimación: 1 semana	Iteración Asignada: 3
Descripción: Deberán ser traducidos los contenidos de Sloodle para su mejor comprensión.	
Información adicional (Observaciones): Se realizará la internacionalización de cada uno de los componentes de Sloodle que se incluyeron en el mundo virtual.	

Anexo 2: Descripción de las tareas de la primera iteración.

Tabla 20. Tarea 1: Crear el avatar para la administración del servidor.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Crear el avatar para la administración del servidor.	
Tipo de tarea: Desarrollo	Estimación: 2 días

Fecha de inicio: 2 de febrero de 2015	Fecha fin: 3 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: En la consola del servidor se utiliza el comando <i>create user</i> para crear avatares.	

Tabla 21. Tarea 2: Crear y administrar los objetos de Sloodle.

Tarea	
Número de tarea: 2	Número de HU: 1
Nombre de la tarea: Crear y administrar los objetos de Sloodle.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 4 de febrero de 2015	Fecha fin: 6 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El objeto set de sloodle es el encargado de crear y administrar las instancias de los objetos de Sloodle.	

Tabla 22. Tarea 3: Buscar modelos 3D para usar en el laboratorio.

Tarea	
Número de tarea: 3	Número de HU: 2
Nombre de la tarea: Buscar modelos 3D para usar en el laboratorio.	
Tipo de tarea: Preparación	Estimación: 1 día
Fecha de inicio: 9 de febrero de 2015	Fecha fin: 9 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: Se buscarán en internet objetos 3D como sillas, mesas y computadoras acordes a la temática del mundo virtual para usar en el laboratorio.	

Tabla 23. Tarea 4: Crear laboratorio virtual e integrarlo al mundo virtual existente.

Tarea	
Número de tarea: 4	Número de HU: 2
Nombre de la tarea: Crear laboratorio virtual e integrarlo al mundo virtual existente.	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 10 de febrero de 2015	Fecha fin: 13 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: Se creará un laboratorio con los modelos encontrados y los objetos de Sloodle. Este laboratorio se introducirá en el mundo virtual existente.	

Tabla 24. Tarea 5: Crear los objetos de interconexión entre OpenSim y Moodle.

Tarea	
Número de tarea: 5	Número de HU: 3
Nombre de la tarea: Crear los objetos de interconexión entre OpenSim y Moodle.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 16 de febrero de 2015	Fecha fin: 18 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El <i>set de sloodle</i> le transfiere a cada instancia creada los parámetros proporcionados a él anteriormente para la interconexión.	

Tabla 25. Tarea 6: Permitir la autenticación de los usuarios.

Tarea	
Número de tarea: 6	Número de HU: 3
Nombre de la tarea: Permitir la autenticación de los usuarios.	

Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 19 de febrero de 2015	Fecha fin: 20 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: <i>Quiosco de registro e inscripción</i> , este objeto será el encargado de autenticar los usuarios y registrarlos en el curso.	

Tabla 26. Tarea 7: Vincular el avatar de OpenSim con la cuenta de Moodle.

Tarea	
Número de tarea: 7	Número de HU: 4
Nombre de la tarea: Vincular el avatar de OpenSim con la cuenta de Moodle.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 23 de febrero de 2015	Fecha fin: 25 de febrero de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El <i>quiosco de registro e inscripción</i> será el encargado de vincular la cuenta con la primera autenticación del usuario.	

Tabla 27. Tarea 8: Enrolar a los estudiantes en los cursos de Moodle desde OpenSim.

Tarea	
Número de tarea: 8	Número de HU: 4
Nombre de la tarea: Enrolar a los estudiantes en los cursos de Moodle desde OpenSim.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 26 de febrero de 2015	Fecha fin: 27 de febrero de 2015

Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García

Descripción: El *quiosco de registro e inscripción* es el encargado de registrar al usuario en el curso correspondiente.

Anexo 3: Descripción de las tareas de la segunda iteración.

Tabla 28. Tarea 9: Vincular la sala de chat.

Tarea	
Número de tarea: 9	Número de HU: 5
Nombre de la tarea: Vincular la sala de chat.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 2 de marzo de 2015	Fecha fin: 3 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: <i>Chat virtual</i> es el objeto que vincula la sala virtual de Moodle con el chat nativo de OpenSim.	

Tabla 29. Tarea 10: Acceder al glosario de Moodle desde el Entorno Virtual.

Tarea	
Número de tarea: 10	Número de HU: 5
Nombre de la tarea: Acceder al glosario de Moodle desde el Entorno Virtual.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 4 de marzo de 2015	Fecha fin: 6 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: <i>Glosario</i> es el objeto que permite buscar en el glosario del curso definiciones y sinónimos usando el comando <code>/def</code> .	

Tabla 30. Tarea 11: Vincular las preguntas de selección de los cursos.

Tarea	
Número de tarea: 11	Número de HU: 6
Nombre de la tarea: Vincular las preguntas de selección de los cursos.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 9 de marzo de 2015	Fecha fin: 10 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El objeto <i>pregunta</i> es el encargado de mostrar las preguntas de selección de Moodle en el mundo virtual mostrando las estadísticas de las respuestas.	

Tabla 31. Tarea 12: Permitir autenticación rápida de los usuarios luego de haberse registrado.

Tarea	
Número de tarea: 12	Número de HU: 6
Nombre de la tarea: Permitir autenticación rápida de los usuarios luego de haberse registrado.	
Tipo de tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 11 de marzo de 2015	Fecha fin: 13 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: La <i>zona de autenticación</i> es el objeto que se encarga de permitir una autenticación más rápida luego de que el usuario se haya registrado.	

Tabla 32. Tarea 13: Realizar exámenes con preguntas de selección múltiples.

Tarea	
Número de tarea: 13	Número de HU: 7
Nombre de la tarea: Realizar exámenes con preguntas de selección múltiples.	

Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 16 de marzo de 2015	Fecha fin: 16 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: La <i>silla de exámenes</i> es el objeto que permite tomar exámenes del curso que contenga preguntas del tipo selección múltiples.	

Tabla 33. Tarea 14: Realizar exámenes de verdadero y falso.

Tarea	
Número de tarea: 14	Número de HU: 7
Nombre de la tarea: Realizar exámenes de verdadero y falso.	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 17 de marzo de 2015	Fecha fin: 17 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: La <i>silla de exámenes</i> es el objeto que permite tomar exámenes del curso que contenga preguntas del tipo verdadero y falso.	

Tabla 34. Tarea 15: Realizar exámenes de respuesta corta.

Tarea	
Número de tarea: 15	Número de HU: 7
Nombre de la tarea: Realizar exámenes de respuesta corta.	
Tipo de tarea: Desarrollo	Estimación: 1 día
Fecha de inicio: 18 de marzo de 2015	Fecha fin: 18 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: La <i>silla de exámenes</i> es el objeto que permite tomar exámenes del curso que contenga preguntas del tipo respuesta corta.	

Tabla 35. Tarea 16: Realizar exámenes del tipo numérico.

Tarea	
Número de tarea: 16	Número de HU: 7
Nombre de la tarea: Realizar exámenes del tipo numérico.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 19 de marzo de 2015	Fecha fin: 20 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: La <i>silla de exámenes</i> es el objeto que permite tomar exámenes del curso que contenga preguntas del tipo respuesta numérica.	

Tabla 36. Tarea 17: Permitir adquirir objetos mediante la obtención de bonificaciones por respuestas.

Tarea	
Número de tarea: 17	Número de HU: 8
Nombre de la tarea: Permitir adquirir objetos mediante la obtención de bonificaciones por respuestas.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 23 de marzo de 2015	Fecha fin: 25 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El <i>distribuidor</i> es el objeto de Sloodle que permite de acuerdo a los puntos que posea el usuario comprar objetos que estén en su inventario dentro el mundo virtual.	

Tabla 37. Tarea 18: Realizar tareas de seguimientos del Usuario.

Tarea	
Número de tarea: 18	Número de HU: 8

Nombre de la tarea: Realizar tareas de seguimientos del Usuario.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 26 de marzo de 2015	Fecha fin: 27 de marzo de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: Los <i>seguidores</i> son los objetos de Sloodle que permiten dar seguimiento al usuario en una región determinada o al hacer clic.	

Anexo 4: Descripción de las tareas de la tercera iteración.

Tabla 38. Tarea 19: Visualizar contenido de audio y video.

Tarea	
Número de tarea: 19	Número de HU: 9
Nombre de la tarea: Visualizar contenido de audio y video.	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha de inicio: 30 de marzo de 2015	Fecha fin: 3 de abril de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: El <i>presenter</i> es el objeto de Sloodle que permite mostrar los contenidos de audio y video en el mundo virtual.	

Tabla 39. Tarea 20: Visualizar contenido web e imágenes.

Tarea	
Número de tarea: 20	Número de HU: 9
Nombre de la tarea: Visualizar contenido web e imágenes.	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha de inicio: 6 de abril de 2015	Fecha fin: 10 de abril de 2015

Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García

Descripción: El *presenter* es el objeto de Sloodle que permite mostrar los contenidos web e imágenes en el mundo virtual.

Tabla 40. Tarea 21: Internacionalizar los componentes dentro del mundo virtual.

Tarea	
Número de tarea: 21	Número de HU: 10
Nombre de la tarea: Internacionalizar los componentes dentro del mundo virtual.	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha de inicio: 13 de abril de 2015	Fecha fin: 17 de abril de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: Se internacionalizarán todos los componentes de Sloodle que se encuentren dentro del mundo virtual.	

Tabla 41. Tarea 22: Internacionalizar los componentes dentro de Moodle.

Tarea	
Número de tarea: 22	Número de HU: 10
Nombre de la tarea: Internacionalizar los componentes dentro de Moodle.	
Tipo de tarea: Desarrollo	Estimación: 5 días
Fecha de inicio: 20 de abril de 2015	Fecha fin: 24 de abril de 2015
Programador responsable: J. Rafael Aguilera Zaldivar y Eilder Jorge García	
Descripción: Se internacionalizarán todos los componentes de Sloodle que se encuentren dentro de Moodle.	

Anexo 5: Patrones GRASP.

Patrones de Software Generales para la Asignación de Responsabilidades (GRASP)	
<i>Patrón</i>	<i>Descripción</i>
Experto en Información	<p>¿Un principio general del diseño de objetos y la asignación de responsabilidades?</p> <p>Asigne una responsabilidad al experto en información, —la clase que tiene la información necesaria para llevar a cabo la responsabilidad.</p>
Creador	<p>¿Quién crea? (Nótese que la Factoría es una solución alternativa frecuente.)</p> <p>Asigne a la clase B la responsabilidad de crear una instancia de la clase A si se cumple alguno de los puntos siguientes:</p> <ol style="list-style-type: none"> 1. B contiene a A 2. B agrega a A 3. B tiene los datos de inicialización de A 4. B registra a A 5. B utiliza estrechamente a A
Controlador	<p>¿Quién gestiona un evento del sistema?</p> <p>Asigne la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que represente una de estas opciones:</p> <ol style="list-style-type: none"> 1. Representa el sistema global, dispositivo o un subsistema (controlador de fachada). 2. Representa un escenario de caso de uso en el que tiene lugar el evento del sistema (controlador de caso de uso o sesión).
Bajo Acoplamiento (evaluativo)	<p>¿Cómo dar soporte a las bajas dependencias y al incremento de la reutilización?</p> <p>Asigne responsabilidades de manera que el acoplamiento (innecesario) se mantenga bajo.</p>
Alta Cohesión (evaluativo)	<p>¿Cómo mantener manejable la complejidad?</p> <p>Asigne responsabilidades de manera que la cohesión permanezca alta.</p>
Polimorfismo	<p>¿Quién es el responsable cuando el comportamiento varía en función del tipo?</p> <p>Cuando las alternativas o comportamientos relacionados varían según el tipo (clase), asigne la responsabilidad del comportamiento —utilizando operaciones polimórficas— a los tipos para los que varía el comportamiento.</p>
Fabricación Pura	<p>¿Quién es el responsable cuando está desesperado, y no quiere violar los principios de alta cohesión y bajo acoplamiento?</p> <p>Asigne un conjunto altamente cohesivo de responsabilidades a una clase de "comportamiento" artificial o de conveniencia que no representa un concepto del dominio del problema —algo inventado—, para dar soporte a la alta cohesión, bajo acoplamiento y la reutilización.</p>
Indirección	<p>¿Cómo asignar responsabilidades para evitar el acoplamiento directo?</p> <p>Asigne la responsabilidad a un objeto intermedio para mediar entre otros componentes o servicios, de manera que no se acoplan directamente.</p>
Variaciones Protegidas	<p>¿Cómo asignar responsabilidades a los objetos, subsistemas, y sistemas de manera que las variaciones o inestabilidad en estos elementos no influya de manera no deseable en otros elementos?</p> <p>Identifique los puntos de variaciones predecibles o inestabilidad; asigne las responsabilidades para crear una "interfaz" estable alrededor de ellos.</p>

Figura 27. Patrones GRASP.

Anexo 6: Patrones GoF.

Clasificación de los patrones GoF (gang of Four) [Gamma]

Propósito Ámbito	Creación	Estructural	Comportamiento
Clase	✓ Factory Method	✓ Adapter	Interpreter Template Method
Objeto	Abstract Factory Builder Prototype ✓ Singleton	✓ Adapter Bridge ✓ Composite Decorator ✓ Facade Flyweight Proxy	Chain of Responsibility Command ✓ Iterator Mediator Memento ✓ Observer State ✓ Strategy Visitor

Figura 28. Patrones GoF.

Además de estos patrones definidos por Gamma existen otros tres que son utilizados frecuentemente:

- ✓ Delegation.
- ✓ Interface.
- ✓ Marker Interface.

Anexo 7: Pruebas de aceptación de la primera iteración.

Tabla 42. CP Creación el set de Sloodle.

Caso de prueba de aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Creación el set de Sloodle.	
Descripción: Se creará el set de Sloodle usando el avatar de administración.	

Condiciones de Ejecución: El usuario necesita estar conectado al mundo virtual.
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> • El usuario importa el archivo .iar que contiene el set de Sloodle. • El usuario arrastra el objeto del inventario al mundo virtual.
Resultado Esperado: Se crea el set de Sloodle en el mundo virtual.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 43. CP Creación el set de Sloodle.

Caso de prueba de aceptación	
Código: HU1_P3	Historia de Usuario: 1
Nombre: Creación el set de Sloodle.	
Descripción: Se creará el set de Sloodle usando el avatar de administración.	
Condiciones de Ejecución: El usuario necesita estar conectado al mundo virtual.	
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> • El usuario importa el archivo .iar que contiene el set de Sloodle. • El usuario arrastra el objeto del inventario al mundo virtual. 	
Resultado Esperado: Se crea el set de Sloodle en el mundo virtual.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 44. CP Conexión del set de Sloodle con Moodle.

Caso de prueba de aceptación	
Código: HU3_P4	Historia de Usuario: 3
Nombre: Conexión del set de Sloodle con Moodle.	
Descripción: Se introducirá en la descripción del set de Sloodle la dirección http de	

Moodle con el objetivo de conectarse.
Condiciones de Ejecución: El usuario necesita estar conectado al mundo virtual y el set de Sloodle debe existir en el mundo virtual.
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> • El usuario hace clic derecho en el objeto y selecciona la opción editar. • El usuario introduce en el cuadro de texto “descripción” la dirección http de Moodle.
Resultado Esperado: Se muestra una url con la ventana de administración de Sloodle.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 45. CP Verificar la autenticación de los usuarios en Moodle desde el mundo virtual.

Caso de prueba de aceptación	
Código: HU3_P5	Historia de Usuario: 3
Nombre: Verificar la autenticación de los usuarios en Moodle desde el mundo virtual.	
Descripción: Se verificará que el usuario se autentique en Moodle usando el <i>quiosco de registro e inscripción</i> .	
Condiciones de Ejecución: El usuario debe estar conectado al mundo virtual y su avatar debe estar vinculado a una cuenta de Moodle.	
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> • El usuario hace clic en el objeto <i>quiosco de registro e inscripción</i>. 	
Resultado Esperado: Se muestra un mensaje mostrando que el usuario ya se encuentra autenticado y en Moodle se muestra el usuario como conectado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 46. CP Vincular el avatar de OpenSim con la cuenta de Moodle.

Caso de prueba de aceptación	
Código: HU4_P6	Historia de Usuario: 4
Nombre: Vincular el avatar de OpenSim con la cuenta de Moodle.	
Descripción: Se verificará que el avatar se vincule con la cuenta de Moodle usando el <i>quiosco de registro e inscripción</i> .	
Condiciones de Ejecución: Debe existir el <i>quiosco de registro e inscripción</i> y el usuario debe poseer una cuenta en Moodle.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario hace clic al <i>quiosco de registro e inscripción</i>. • El usuario accede a la url que te provee el <i>quiosco de registro e inscripción</i>. • El usuario se autentica con su cuenta Moodle 	
Resultado Esperado: Se crea una relación entre el avatar y la cuenta autenticada.	
Evaluación de la Prueba: Prueba no satisfactoria. Si en el navegador se encuentra el usuario autenticado en Moodle la url que provee el <i>quiosco de registro e inscripción</i> no es correcta.	

Tabla 47. CP Enrolar a los estudiantes en los cursos de Moodle desde OpenSim.

Caso de prueba de aceptación	
Código: HU4_P7	Historia de Usuario: 4
Nombre: Enrolar a los estudiantes en los cursos de Moodle desde OpenSim.	
Descripción: Se verificará que el usuario se pueda enrolar en un curso de Moodle.	
Condiciones de Ejecución: Debe existir el <i>quiosco de registro e inscripción</i> y el usuario debe estar autenticado en Moodle.	

<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario hace clic al <i>quiosco de registro e inscripción</i>. • El usuario accede a la url que te provee el <i>quiosco de registro e inscripción</i>. • El usuario hace clic en el mensaje de aceptación para enrolarse.
<p>Resultado Esperado: El usuario se enrola en el curso asociado.</p>
<p>Evaluación de la Prueba: Prueba no satisfactoria. Si en el navegador se encuentra el usuario autenticado en Moodle la url que provee el <i>quiosco de registro e inscripción</i> no es correcta.</p>

Anexo 8: Pruebas de aceptación de la segunda iteración.

Tabla 48. CP Verificar que la sala de chat Moodle se transmita al chat del mundo virtual.

Caso de prueba de aceptación	
Código: HU5_P8	Historia de Usuario: 5
Nombre: Verificar que la sala de chat Moodle se transmita al chat del mundo virtual.	
Descripción: Se verificará que lo que los usuarios escriban en la sala de chat de Moodle se muestre en el chat local del mundo virtual.	
Condiciones de Ejecución: Debe existir en el mundo virtual el objeto <i>sala de chat</i> asociado a la sala de chat de Moodle con la que se desea vincular.	
<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario escribe en la sala de chat de Moodle. 	
Resultado Esperado: Se muestra lo escrito por el usuario en el mundo virtual.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 49. CP Verificar que la sala de chat del mundo virtual se transmita al chat de Moodle.

Caso de prueba de aceptación	
Código: HU5_P9	Historia de Usuario: 5
Nombre: Verificar que la sala de chat del mundo virtual se transmita al chat de Moodle.	
Descripción: Se verificará que lo que los usuarios escriban en el chat local y se muestre en la sala de chat de Moodle.	
Condiciones de Ejecución: Debe existir en el mundo virtual el objeto <i>sala de chat</i> asociado a la sala de chat de Moodle con la que se desea vincular.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario hace clic en el objeto <i>sala de chat</i>. • El usuario acepta la pregunta ¿Desea ser grabado? • El usuario escribe en el chat local. 	
Resultado Esperado: Se muestra lo escrito por el usuario en el mundo virtual en la sala de chat de Moodle.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 50. CP Verificar que se acceda al glosario desde el mundo virtual.

Caso de prueba de aceptación	
Código: HU5_P10	Historia de Usuario: 5
Nombre: Verificar que se acceda al glosario desde el mundo virtual.	
Descripción: Se verificará que al usar el objeto <i>glosario</i> en el mundo virtual se puedan acceder a las definiciones que contenga el glosario de Moodle.	
Condiciones de Ejecución: Debe existir en el mundo virtual el objeto <i>glosario</i> con un glosario de Moodle asociado.	

<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario hace clic en el objeto <i>glosario</i> para activarlo. • El usuario escribe <i>/def</i> [término], por ejemplo: <i>/def ram</i>.
<p>Resultado Esperado: Se muestra al usuario la definición del término en el chat privado del usuario.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>

Tabla 51. CP Verificar que se muestren las preguntas de selección en el mundo virtual.

Caso de prueba de aceptación	
Código: HU6_P11	Historia de Usuario: 6
Nombre: Verificar que se muestren las preguntas de selección en el mundo virtual.	
Descripción: Se verificará el objeto <i>pregunta</i> muestre una pregunta de selección de Moodle.	
Condiciones de Ejecución: Debe existir el objeto <i>pregunta</i> con una pregunta de selección asociada de Moodle.	
<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario hace clic en la respuesta deseada. 	
Resultado Esperado: Se muestra un mensaje diciéndole al usuario que su respuesta ha sido guardada satisfactoriamente. Se guarda la respuesta en Moodle.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 52. CP Verificar la autenticación rápida de los usuarios en Moodle desde el mundo virtual.

Caso de prueba de aceptación	
Código: HU6_P12	Historia de Usuario: 6
Nombre: Verificar la autenticación rápida de los usuarios en Moodle desde el mundo virtual.	

<p>Descripción: Se verificará que el usuario se autentique rápidamente en Moodle usando la <i>zona de autenticación</i>.</p>
<p>Condiciones de Ejecución: El usuario debe estar conectado al mundo virtual y su avatar debe estar vinculado a una cuenta de Moodle.</p>
<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario entra a la <i>zona de autenticación</i>.
<p>Resultado Esperado: Se muestra un mensaje mostrando que el usuario ya se encuentra autenticado y en Moodle se muestra el usuario como conectado.</p>
<p>Evaluación de la Prueba: Prueba no satisfactoria. Si la zona es demasiado pequeña no realiza el intento de autenticar.</p>

Tabla 53. CP Verificar que se realicen los exámenes de selección múltiple.

<p>Caso de prueba de aceptación</p>	
<p>Código: HU7_P13</p>	<p>Historia de Usuario: 7</p>
<p>Nombre: Verificar que se realicen los exámenes de selección múltiple.</p>	
<p>Descripción: Se verificará que el objeto <i>silla de exámenes</i> realice exámenes con preguntas de selección múltiples.</p>	
<p>Condiciones de Ejecución: Debe existir el objeto <i>silla de exámenes</i> asociado a un examen que contenga preguntas de selección múltiples.</p>	
<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario hace clic derecho en el objeto <i>silla de exámenes</i> y selecciona la opción sentarse. • El usuario responde las preguntas del examen usando los cuadros de diálogos. 	
<p>Resultado Esperado: Se muestra un mensaje diciéndole al usuario su calificación en el examen. El examen es guardado en Moodle para poder ser revisado por el profesor.</p>	

Evaluación de la Prueba: Prueba satisfactoria.

Tabla 54. CP Verificar que se realicen los exámenes de verdadero y falso.

Caso de prueba de aceptación	
Código: HU7_P14	Historia de Usuario: 7
Nombre: Verificar que se realicen los exámenes de verdadero y falso.	
Descripción: Se verificará que el objeto <i>silla de exámenes</i> realice exámenes con preguntas de verdadero y falso.	
Condiciones de Ejecución: Debe existir el objeto <i>silla de exámenes</i> asociado a un examen que contenga preguntas de verdadero y falso.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none">• El usuario hace clic derecho en el objeto <i>silla de exámenes</i> y selecciona la opción sentarse.• El usuario responde las preguntas del examen usando los cuadros de diálogos.	
Resultado Esperado: Se muestra un mensaje diciéndole al usuario su calificación en el examen. El examen es guardado en Moodle para poder ser revisado por el profesor.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 55. CP Verificar que se realicen los exámenes de tipo numérico.

Caso de prueba de aceptación	
Código: HU7_P15	Historia de Usuario: 7
Nombre: Verificar que se realicen los exámenes de tipo numérico.	
Descripción: Se verificará que el objeto <i>silla de exámenes</i> realice exámenes con preguntas de tipo numérico.	
Condiciones de Ejecución: Debe existir el objeto <i>silla de exámenes</i> asociado a un examen que contenga preguntas de tipo numérico.	

<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario hace clic derecho en el objeto <i>silla de exámenes</i> y selecciona la opción sentarse. • El usuario responde las preguntas del examen usando los cuadros de diálogos.
<p>Resultado Esperado: Se muestra un mensaje diciéndole al usuario su calificación en el examen. El examen es guardado en Moodle para poder ser revisado por el profesor.</p>
<p>Evaluación de la Prueba: Prueba no satisfactoria. Existen dificultades con el manejo de las unidades de medidas.</p>

Tabla 56. CP Verificar que se realicen los exámenes de respuesta corta.

<p>Caso de prueba de aceptación</p>	
<p>Código: HU7_P16</p>	<p>Historia de Usuario: 7</p>
<p>Nombre: Verificar que se realicen los exámenes de respuesta corta.</p>	
<p>Descripción: Se verificará que el objeto <i>silla de exámenes</i> realice exámenes con preguntas de respuesta corta.</p>	
<p>Condiciones de Ejecución: Debe existir el objeto <i>silla de exámenes</i> asociado a un examen que contenga preguntas de respuesta corta.</p>	
<p>Entrada/ Pasos de ejecución:</p> <ul style="list-style-type: none"> • El usuario hace clic derecho en el objeto <i>silla de exámenes</i> y selecciona la opción sentarse. • El usuario responde las preguntas del examen usando los cuadros de diálogos. 	
<p>Resultado Esperado: Se muestra un mensaje diciéndole al usuario su calificación en el examen. El examen es guardado en Moodle para poder ser revisado por el profesor.</p>	
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>	

Tabla 57. CP Verificar la adquisición de objetos por bonificación.

Caso de prueba de aceptación	
Código: HU8_P17	Historia de Usuario: 8
Nombre: Verificar la adquisición de objetos por bonificación.	
Descripción: Se verificará que el objeto <i>distribuidor</i> entregue objetos si tiene las bonificaciones correspondientes.	
Condiciones de Ejecución: Debe existir el objeto <i>distribuidor</i> y que el estudiante debe tener bonificaciones.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario hace clic en el objeto <i>distribuidor</i> y selecciona el objeto deseado. 	
Resultado Esperado: Se añade al inventario del estudiante el objeto seleccionado.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 58. CP Verificar el seguimiento de los usuarios.

Caso de prueba de aceptación	
Código: HU8_P18	Historia de Usuario: 8
Nombre: Verificar el seguimiento de los usuarios.	
Descripción: Se verificará que el objeto <i>seguidor</i> registre la presencia de un usuario.	
Condiciones de Ejecución: Debe existir el objeto <i>seguidor</i> y el estudiante debe estar autenticado.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario entra en el rango del objeto <i>seguidor radial</i>. <p style="text-align: center;">O</p> <ul style="list-style-type: none"> El usuario hace clic en el objeto <i>seguidor botón</i>. 	

Resultado Esperado: Se registra la presencia del usuario en la región donde esté ubicado el objeto *seguidor* para ser revisado por un profesor desde Moodle.

Evaluación de la Prueba: Prueba satisfactoria.

Anexo 9: Pruebas de aceptación de la tercera iteración.

Tabla 59. CP Verificar la visualización del contenido de audio y video.

Caso de prueba de aceptación	
Código: HU9_P19	Historia de Usuario: 9
Nombre: Verificar la visualización del contenido de audio y video.	
Descripción: Se verificará que el objeto <i>presenter</i> muestre información multimedia.	
Condiciones de Ejecución: Debe existir el objeto <i>presenter</i> y debe tener archivos multimedia vinculados. El usuario debe poseer un visor que soporte la tecnología <i>shared media</i> .	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none">• El usuario debe ubicarse de tal forma que pueda visualizar el objeto <i>presenter</i>.	
Resultado Esperado: Se muestra en el objeto <i>presenter</i> los archivos multimedia vinculados.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 60. CP Verificar la internacionalización de Sloodle.

Caso de prueba de aceptación	
Código: HU10_P20	Historia de Usuario: 10
Nombre: Verificar la internacionalización de Sloodle.	
Descripción: Se verificará que al realizarse cambios de idioma la interfaz acepte el lenguaje español tanto como el inglés.	
Condiciones de Ejecución:	

Entrada/ Pasos de ejecución:

- El usuario debe seleccionar el idioma deseado en Moodle.

Resultado Esperado: Se muestra la interfaz en el idioma seleccionado.

Evaluación de la Prueba: Prueba satisfactoria.