

SecondTutor

Informe final del proyecto de grado

Integrantes:

Rafael Álvarez
Matías Calimares
Federico Dosil

Tutores:

Eduardo Fernández
Martín Furno

Resumen Ejecutivo

Los ambientes virtuales han tomado mayor relevancia durante los últimos años, aunque sus orígenes y usos han estado ligados más fuertemente al entretenimiento y entrenamiento basado en simulaciones que a la educación. Por su parte, las tecnologías de *e-learning*, o sea las relacionadas con las metodologías de educación presencial, semipresencial y a distancia a través de medios electrónicos, han evolucionando constantemente, especialmente en la última década. La realización de experiencias de integración de ambas tecnologías, con la finalidad de mejorar aspectos didácticos y pedagógicos, se presenta como una idea promisoría.

El propósito de este proyecto es contribuir a dicha integración, mediante el estudio del estado del arte de las herramientas de apoyo a la educación a distancia, y con la construcción de un sistema funcional que permita la interacción de un ambiente tridimensional con una herramienta de gestión de contenidos educativos.

Al comienzo del proyecto se realizó un estudio de las principales plataformas de mundos virtuales, más específicamente aquellas que puedan proveer un entorno tridimensional a través de Internet. De este estudio surgió el uso de la plataforma Second Life. A continuación se investigaron las herramientas de gestión de contenido educativo, poniendo énfasis en aquellas de código abierto, extensibles y que permitan la integración con otros sistemas a través de Internet. De esta investigación surgió ATutor, una aplicación Web de código abierto para la gestión de contenidos de aprendizaje.

Para la integración de estas dos herramientas, se diseñaron e implementaron dos módulos. Uno de ellos es un módulo de ATutor, construido en *PHP* y *MySQL*, y el otro es un conjunto de scripts alojados en el servidor de Second Life y codificados en LSL (Linden Scripting Language), el lenguaje provisto por la plataforma. La comunicación entre ellos se realiza a través del protocolo HTTP desde el módulo en Second Life, hacia el módulo instalado en el servidor de ATutor.

La solución desarrollada permite crear escenas interactivas en las que usuarios, con el rol de alumnos, participan con el propósito de adquirir conocimientos o de ser evaluados. También existen dentro del sistema usuarios con el rol de tutores, que definen las escenas creando guiones por medio de la interfaz Web y asignándolos a robots. Estos robots son objetos en el mundo virtual de Second Life, e interpretan las escenas ejecutando cada uno de los eventos que componen el guión que les fue asignado.

De esta manera, se logró crear una aplicación que integra las herramientas tradicionales de apoyo a la educación a distancia y los ambientes virtuales, logrando así los objetivos del proyecto.

Contenido

1. Introducción.....	1
1.1. Presentación del Problema.....	1
1.1.1. Problema.....	1
1.1.2. Situación actual.....	2
1.1.3. Solución deseada.....	2
1.2. Objetivos.....	3
1.3. Resultados esperados.....	3
1.4. Conclusiones.....	3
1.5. Integrantes.....	4
1.6. Organización de la documentación	4
1.7. Organización de este documento.....	4
2. Estado del arte.....	6
2.1. Estudio de alternativas de LCMS.....	6
2.2. Estudio de alternativas de plataformas de realidad virtual.....	6
2.3. Aplicaciones de realidad virtual para la educación.....	7
2.4. Conclusiones.....	7
3. Análisis de la solución.....	8
3.1. Estudio de opciones de implementación	8
3.2. Solución Planteada.....	8
3.3. Descripción.....	8
3.4. Conclusión.....	10
4. SecondTutor.....	12
4.1. Arquitectura.....	12
4.2. Base de Datos.....	13
4.3. Robots en Second Life.....	14
4.4. Módulo ATutor.....	16
4.5. Comunicación ATutor - Second Life.....	17
4.6. Pruebas.....	17
4.7. Ejemplo práctico.....	18
4.8. Precondiciones.....	18
4.9. Objetivo.....	18
4.10. Pasos a seguir.....	19
4.11. Tomar la evaluación.....	21
4.12. Conclusiones.....	23
5. Cronogramas y organización.....	25
5.1. Organización del grupo de trabajo.....	25
5.2. Cronograma.....	25
5.2.1. Diagrama de Gantt.....	25
5.2.2. Estimaciones de tiempo.....	26
6. Resultados y conclusiones.....	27
6.1. Objetivos logrados.....	27
6.2. Limitaciones.....	28
6.3. Trabajo futuro y posibles mejoras.....	28
7. Glosario.....	29
8. Referencias	31

1. Introducción

En este capítulo se plantea el problema y se motiva el trabajo realizado, se describe la situación actual en el área relacionada y luego se enumeran los que fueron los objetivos a alcanzar. Por último se mencionan brevemente las conclusiones logradas.

1.1. Presentación del Problema

1.1.1. Problema

El uso de la tecnología con fines educativos ha evolucionado de la mano con el desarrollo de Internet, debido en gran parte a las posibilidades de democratización de la información. Esto ha facilitado el acceso a información relevante a un grupo cada vez más heterogéneo de individuos. Como conclusión, desde el punto de vista tecnológico, consistió en el desarrollo de herramientas que faciliten la gestión del material educativo, más específicamente sistemas de gestión de contenidos utilizados para la enseñanza o LCMS por su sigla en inglés¹.

Una vez lograda la gestión de la información que se desea brindar a los usuarios, se avanzó en la creación de contenidos multimedia (imágenes, audio, video, animaciones, entre otros). Cumplida esta etapa, se avanza hacia la participación de varios usuarios en sincronía (a través de canales de mensajería instantánea, videoconferencias).

Estas herramientas evolucionaron, permitiendo asistir al dictado de cursos semipresenciales (*b-learning*²) así como la creación y dictado de cursos no presenciales o a distancia (*e-learning*³). Incluso en los últimos años se habla del término *e-learning 2.0*, inspirado en la Web 2.0, donde se utilizan herramientas sociales, que facilitan la interacción y comunicación, como blogs, *wikis*, o mundos virtuales.

Una característica fundamental de la educación a distancia es la separación física ente el profesorado y el alumnado. Los sistemas *e-learning* tienen la particularidad de posicionar al alumno como el centro de una formación independiente y flexible, y es él mismo, el responsable de su propio aprendizaje.

En la actualidad existe una gran cantidad de sistemas gestores de contenido educativo que se componen básicamente de los contenidos y evaluaciones en base a estos. Dicho contenido puede aparecer en forma de textos, imágenes, videos u otro contenido multimedia, y las evaluaciones están compuestas mayoritariamente por cuestionarios, normalmente de múltiple opción, para lograr una mayor automatización de la corrección de las mismas. Estos métodos, por su naturaleza, presentan ciertas características de interacción con el usuario que generan, por un lado, una serie de problemas visibles, como ser la dificultad o incapacidad de situar a la persona en la realidad que se quiere mostrar o evaluar, y por otro un conjunto de problemas intangibles, como la medida de la atención que la persona le presta al curso cuando el ambiente que la rodea, por tratarse de educación electrónica, no siempre es el más propicio para la enseñanza.

Este último punto cobra fuerza cuando escapa al ámbito académico y se tiene en cuenta la educación en los ámbitos laborales, en los que mantener el interés de la persona utilizando la herramienta puede llegar a ser de gran

1 LCMS, del inglés *Learning Content Management Systems*.

2 b-learning, del inglés *blending learning*. Es un proceso docente semipresencial, un curso dictado así incluye clases presenciales y actividades de e-learning.

3 e-learning, del inglés *electronic learning*.

importancia para una organización que quiere educar a su personal en las tareas que va a realizar.

En esta clase de situaciones el principal interesado en que la persona aprenda no es ella misma, como ocurre en el ámbito académico, sino la organización en donde trabaja. De ahí que tomar como un hecho la motivación del estudiante por ser éste el principal beneficiado del estudio, no es aplicable en todos los casos y se deben tener en cuenta variables comúnmente dejadas de lado, como ser la motivación y la situación emocional de la persona.

Otro punto a tener en cuenta es que los usuarios destinan un tiempo mínimo al aprendizaje (estudio y posterior evaluación) y es necesario maximizar la utilidad de este tiempo.

1.1.2.Situación actual

Actualmente, las herramientas más usadas de apoyo a la enseñanza utilizan como método para solventar el problema de representar correctamente una idea o una situación, la descripción por escrito de la misma. Por ejemplo, es común para evaluar cuál sería la reacción de una persona en una determinada situación, que se utilicen fórmulas del estilo: "Si usted se encontrara en tal situación, y ocurre tal evento, ¿qué haría?". Esta forma de ubicar a una persona en una situación particular posee una clara desventaja si se la compara con la recreación de la situación que se quiere describir en un ambiente tridimensional.

Con respecto a hacer más atractiva la experiencia de los usuarios, ya sea realizando un curso o bien siendo evaluados, los métodos utilizados se basan principalmente en la inclusión de imágenes acompañando los textos, así como también videos u otros contenidos multimedia, donde la mayoría de ellos tiene la característica de poseer un orden preestablecido para la exposición de los contenidos y la forma de ver los mismos.

Surge así la idea de utilizar ambientes virtuales como complemento a la educación virtual, más específicamente como herramientas de comunicación y de apoyo a las evaluaciones.

Y en lo que se refiere a este tipo de aplicaciones, de interacción entre herramientas de apoyo a la educación y mundos virtuales, existe en la actualidad una herramienta llamada Sloodle [Sloodle, 2008]. Sloodle combina el sistema de ayuda a la educación Moodle [Moodle, 2008] y el ambiente tridimensional Second Life [Second Life, 2008]. El objetivo del mismo es unir las funciones de un sistema de enseñanza basado en Web (LMS⁴) con la riqueza de interacción de un entorno virtual multiusuario en tres dimensiones (MUVE⁵). Esta herramienta no resuelve los problemas planteados en este proyecto, sino que utiliza el ambiente virtual de Second Life para crear una representación tridimensional de un conjunto de objetos de Moodle, desde noticias hasta foros, encuestas, *chats* y glosarios.

En cuanto a la integración entre Second Life y ATutor [ATutor, 2008] no se encontró a la fecha de comienzo de este proyecto ningún desarrollo activo, partiendo por lo tanto desde cero, atravesando todas las etapas de un proyecto de estas características.

1.1.3.Solución deseada

La solución buscada pretende mejorar la experiencia de las personas que utilizan sistemas electrónicos de educación y el de permitir representar con mayor eficacia distintas situaciones de la vida cotidiana. Para esto se procura utilizar

4 LMS, del inglés Learning Management System.

5 MUVE, del inglés Multi User Virtual Environment

conceptos de los ambientes virtuales, permitiendo al usuario utilizar un avatar⁶ para interactuar con el entorno que lo rodea y con otros personajes en el mundo virtual. De esta forma se desea lograr una interacción más realista del usuario con escenas o situaciones que luego van a ser habituales a la hora de aplicar los conocimientos obtenidos. También se quiere lograr una forma más práctica de evaluar estos conocimientos sin tener que exponer al usuario inexperto a la situación real.

Otra característica de la solución es que debería tener la capacidad de formar parte de una herramienta de gestión de contenidos educativos, para aprovechar de esta forma las funcionalidades de manejo de usuarios, cursos y contenidos, entre otros.

1.2. Objetivos

Por tanto los objetivos planteados para el proyecto son los siguientes:

1. Realizar un estudio del estado del arte de sistemas gestores de contenido educativo y evaluar, entre otros factores, la facilidad de interacción con una aplicación externa, más específicamente un MUVE.
2. Realizar un estudio del estado del arte de los ambientes virtuales tridimensionales teniendo en cuenta la posibilidad de permitir la interacción con un LCMS.
3. A partir de las aplicaciones seleccionadas, evaluar las distintas posibilidades de implementación de un sistema que sirva de apoyo a las actividades de enseñanza, haciendo hincapié en el uso de ambientes virtuales para resolver los problemas identificados en la sección 1.1.1 de este documento.
4. Diseñar, implementar y probar el sistema resultado de la evaluación planteada en el punto anterior.

Es importante aclarar que no es un objetivo del proyecto implementar un LCMS o un ambiente tridimensional, sino utilizar herramientas de este tipo ya existentes e integrarlas de forma de aprovechar cualidades de ambas.

1.3. Resultados esperados

Se pretende contar con una versión funcional del sistema, con todas las funcionalidades de un LCMS y que permita utilizar la interfaz tridimensional para mejorar la experiencia educativa.

Otros resultados deseados son:

- Estado del arte de las herramientas de mundos tridimensionales que permitan la creación de ambientes y comunidades virtuales de usuarios, en particular Second Life.
- Estudio de herramientas LCMS, en particular ATutor por ser un LCMS completo y de código abierto.
- Una arquitectura que integre a las herramientas seleccionadas.
- Implementación del módulo de comunicación entre las herramientas seleccionadas.
- Construcción de un sistema totalmente funcional.
- Ejemplos prácticos de utilización del sistema.

1.4. Conclusiones

Las conclusiones para los distintos objetivos planteados son:

- Se estudiaron diversos sistemas gestores de contenido educativo y se escogió el LCMS ATutor, entre otras razones, por tratarse de una

⁶ Ver glosario.

herramienta de código abierto con la capacidad para añadirle módulos con nuevas funcionalidades.

- Se estudió un gran número de plataformas de ambientes virtuales tridimensionales disponibles, de las que se eligió Second Life por proveer al usuario la posibilidad de crear objetos y asignarles comportamiento, por ser gratuito y utilizar una herramienta cliente de código abierto, y por poseer la posibilidad de comunicarse con programas externos a través del protocolo HTTP.
- Se diseñó una arquitectura distribuida, compuesta por un módulo en Second Life y otro en ATutor. Este último se compone de tres partes: una para administrar escenas a ejecutar en Second Life y herramientas para evaluar el resultado de la ejecución de ellas por los usuarios; otra que brinda una interfaz de comunicación con Second Life, y finalmente una para guardar los datos en memoria persistente.
- Se desarrolló una base de datos compatible con la de ATutor, para guardar los datos de la aplicación así como historiales de ejecuciones.
- Se realizaron pruebas sobre el módulo implementado de forma sistemática e iterativa, alcanzando un producto final completamente funcional.
- Se lograron cumplir los objetivos del proyecto de grado.

1.5. Integrantes

Este proyecto fue realizado por Rafael Álvarez, Matías Calimares y Federico Dosil, estudiantes de la carrera de Ingeniero en Computación, carrera dictada en la Facultad de Ingeniería de la Universidad de la República Oriental del Uruguay.

Los tutores del proyecto fueron los profesores Eduardo Fernández y Martín Furno, docentes del mismo centro de estudios.

1.6. Organización de la documentación

A continuación se enumeran los distintos documentos que acompañan y explican el producto SecondTutor y su desarrollo.

Documentación general:

Se trata del presente documento. En el mismo se describe el proyecto de manera completa, con foco en la investigación, objetivos, solución, metodología de trabajo y conclusiones generales. El objetivo de este documento es presentar al lector la solución propuesta.

Documentación funcional del sistema:

En dicho documento se explica en detalle el funcionamiento del sistema desarrollado. Está orientado a desarrolladores que deseen conocer el funcionamiento, o incluso introducir modificaciones al producto. También se anexa información sobre la estructura de archivos, descripciones de los archivos relevantes para la comprensión de la solución, descripción de la base de datos y de los scripts que se ejecutan en los objetos dentro de Second Life.

Manual de Usuario:

Se explica cómo instalar la herramienta en ATutor y en Second Life. También se describe la forma de crear y tomar evaluaciones así como ver los resultados de estas. Introduce a los lectores la herramienta desarrollada, define los términos que puedan resultar nuevos para el lector y presenta un ejemplo práctico donde se detalla paso a paso cómo crear una evaluación.

1.7. Organización de este documento

Este documento se encuentra organizado en capítulos, cada uno presenta un

aspecto funcional del desarrollo del proyecto.

Capítulo 1, Introducción: en la introducción se define el problema en términos generales, así como los objetivos deseados a alcanzar con el proyecto y un resumen de las conclusiones alcanzadas.

Capítulo 2, Estado del Arte: en este capítulo se detalla el estado del arte para las diferentes herramientas a utilizar a lo largo del proyecto. Se culmina con una discusión sobre la selección de herramientas utilizadas para el resto del proyecto.

Capítulo 3, Arquitectura preliminar: en este capítulo se plantea la arquitectura genérica, e independiente de las herramientas a utilizar. También se presentan las distintas posibilidades estudiadas a la hora de implementar una solución.

Capítulo 4, SecondTutor: se describe la solución desarrollada, se detalla la arquitectura del sistema, el esquema de base de datos así como varias decisiones de diseño.

Capítulo 5, Cronogramas y organización: en este capítulo se presenta la organización del grupo, así como las estimaciones previas y el cronograma de actividades.

Capítulo 6, Resultados y conclusiones: en este capítulo se discuten las conclusiones generales del proyecto, limitaciones y posibles mejoras.

Capítulo 7, Glosario: se presenta el glosario de la terminología utilizada a lo largo del documento.

Capítulo 8, Referencias: guía y puntos de referencia tomados para la ejecución del proyecto.

2. Estado del arte

Durante la fase de investigación, se estudiaron distintos sistemas de gestión de contenidos educativos, así como las distintas opciones disponibles de plataformas de mundos virtuales. A continuación se expone el método empleado para comparar las herramientas, las herramientas evaluadas y el resultado obtenido en cada caso.

2.1. Estudio de alternativas de LCMS

El interés principal del estudio de las alternativas disponibles de LCMS, es encontrar una herramienta con las características de ser gratuita, de código abierto y que posea las funcionalidades de gestión de estudiantes y cursos propias de un LCMS. Durante este estudio se encontró que no existen muchos LCMS que sean simultáneamente gratuitos y de código abierto, solamente ATutor. Sin embargo sí existen varios sistemas de apoyo a la educación (LMS⁷), entre las cuales se encuentran Moodle, Ilias [Ilias, 2008] y Olat [Olat, 2008], que fueron los que se vieron con mayor detalle.

El estudio de las alternativas de LCMS tuvo entonces una fase inicial correspondiente a conocer las diferencias entre LCMS, LMS y CMS⁸. Este análisis y sus resultados se encuentran documentados en el anexo I de la documentación del proyecto.

Como resultado del estudio anterior, se resolvió utilizar un LCMS por ser más completo, y proveer de esta forma más flexibilidad a la hora de interactuar con el ambiente tridimensional elegido. Dentro de este tipo de herramientas, ATutor poseía todas las funcionalidades buscadas, además de incluir la capacidad de agregar nuevas por medio de paquetes instalables en forma de módulos⁹.

2.2. Estudio de alternativas de plataformas de realidad virtual

Se realizó un estudio de las plataformas de mundos virtuales tridimensionales más populares, más específicamente aquellas que puedan proveer un entorno tridimensional a través de Internet, tanto inmersivas como no inmersivas¹⁰. La popularidad fue considerada de gran importancia, pues existía la posibilidad que la empresa responsable dejara de desarrollar o incluso descontinuara el producto si no contaba con el apoyo de un público masivo. Entonces, para disminuir el riesgo de perder una parte esencial del proyecto en pleno desarrollo del mismo, el estudio de este tipo de herramientas se limitó a aquellas en pleno auge y a aquellas cuyo servidor estuviera disponible de forma gratuita.

Las herramientas fueron estudiadas con el objetivo de integrarse con los sistemas de apoyo a la educación como los descritos anteriormente, por tanto la prioridad de los puntos a tener en cuenta al momento de juzgar estas aplicaciones no es la misma que en una investigación comparativa con otros fines, como ser jugabilidad, rendimiento, compatibilidad o estudio de mercadeo, entre otros.

7 LMS, del inglés Learning Management System, sistema de apoyo a la educación.

8 CMS, del inglés Content Management System, sistema gestor de contenidos.

9 Más información en: <http://help.atutor.ca/developer/modules.html>

10 Los métodos inmersivos de realidad virtual con frecuencia se ligan a un ambiente tridimensional creado por ordenador, el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano. La realidad virtual no inmersiva también utiliza el computador pero permite interactuar sin la necesidad de dispositivos adicionales.

Algunas de las características que se tuvieron en cuenta son el tipo de licencia, el estado actual y requerimientos de *hardware*. Se ponderó positivamente la facilidad de comunicación entre los usuarios. Este tipo de aplicaciones se puede ver como la evolución de los sistemas antiguos de mensajería, como la tecnología IRC¹¹. Algunas de las herramientas estudiadas son Second Life, Active Worlds, There, Interreality y Moove Online.

Para más detalles de esta investigación está disponible el anexo II de este proyecto, donde la conclusión del mismo se resume en la elección de Second Life. Una plataforma que consta básicamente de dos partes, una aplicación cliente que corre en el computador del usuario y una parte que se ejecuta en los servidores provistos por la misma empresa. Second Life es un mundo virtual tridimensional en el que cada usuario está representado por un avatar, con el que puede recorrer el vasto mundo, que en su gran mayoría es creado por los propios usuarios.

Otras de las ventajas de esta popular herramienta, en lo que al proyecto respecta, son que el código fuente de la aplicación cliente fue liberado bajo licencia GNU GPL¹², y la facilidad que brinda para crear objetos dentro del mundo virtual y asignarles comportamiento por medio de un lenguaje de *scripting* propio. Esto es, un lenguaje de programación, interpretado, basado en estados y eventos, y permite modificar los estados de los objetos así como comunicar a unos con otros. Se explica esto con más detalle en la sección 4.3 de este documento.

2.3. Aplicaciones de realidad virtual para la educación

Se realizó un estudio con el fin de analizar distintas propuestas que utilicen la realidad virtual (no inmersiva) como apoyo a actividades de enseñanza. Como punto de partida de este estudio se tomó el trabajo realizado por Livingstone y Kemp [Livingstone, 2006] donde se concluye que la integración de los ambientes virtuales y los gestores de contenido debe evitar la simple traducción de una interfaz tridimensional para los sistemas LCMS ya existentes, y como contraparte se deben estudiar nuevas formas de interacción.

Esta investigación se centró en aquellas aplicaciones que utilizan entornos virtuales tridimensionales. Como conclusión general se puede decir que no existen actualmente muchas alternativas a la hora de elegir una aplicación que integre realidad virtual a la educación a distancia. Además, dentro de estas alternativas, la gran mayoría exigen el pago de contratos antes de mostrar resultados o experiencias de usuarios asiduos.

2.4. Conclusiones

En esta etapa se estudiaron y luego se seleccionaron herramientas de gestión de contenido educativo y plataformas de ambientes virtuales tridimensionales. Se plantearon cuidadosamente las características a tener en cuenta al evaluar cada una de ellas, teniendo siempre en cuenta el objetivo global, la integración entre las herramientas seleccionadas. Finalmente se seleccionó el sistema de gestión de contenidos educativos ATutor, y el entorno virtual tridimensional Second Life.

11 IRC, Internet Relay Chat.

12 GPL, General Public License (<http://www.gnu.org/copyleft/gpl.html>).

3. Análisis de la solución

En este capítulo se presenta la solución encontrada de forma genérica, en función de sus componentes y explicando los principales puntos que componen su arquitectura y cómo se relacionan entre sí.

Se comienza mostrando los puntos principales del estudio que se realizó para escoger la aplicación a implementar, seguidamente se presenta el estudio del estado del arte y elección de las herramientas de LCMS y mundos virtuales, y finalmente se presenta una descripción en alto nivel de la aplicación, poniendo énfasis en los principales componentes que integran su arquitectura.

3.1. Estudio de opciones de implementación

Dentro del marco del proyecto, tras finalizar el estudio del estado del arte de las tecnologías a utilizar, se realizó un estudio sobre las distintas posibilidades de implementación.

Se analizó inicialmente la posibilidad de realizar un sistema que oficie de traductor de objetos de ATutor a Second Life, el cual buscaría la forma de representar los elementos brindados por ATutor de forma tridimensional para intentar mejorar así la interfaz de ATutor y hacerla más intuitiva. Posteriormente, cambiando el punto de vista, se analizó otra opción, donde se buscaba aprovechar en mayor medida las posibilidades brindadas por Second Life, se pensó entonces en un sistema que utilizara datos en ATutor y los mostrara de manera gráfica, permitiendo además interactuar con los mismos en el ambiente de Second Life.

A continuación se puso énfasis sobre las herramientas brindadas por ATutor, y se pensó en utilizar Second Life para realizar los cuestionarios de opción múltiple de ATutor de manera tridimensional. Finalmente se llegó a una opción de implementación más general, que usa en buena forma las posibilidades de ambas herramientas. Esta opción tiene como base la utilización del sistema de gestión de alumnos, profesores y cursos de ATutor para brindar, en la forma de evaluaciones, la posibilidad de crear escenas en las que los usuarios puedan participar. Estas escenas, que se ejecutan en el mundo virtual de Second Life, no necesariamente tienen que ser usadas para la evaluación de los conocimientos, también pueden utilizarse como parte de la enseñanza.

Este estudio, junto con una descripción en detalle de las etapas abordadas, se expone en el anexo III de la documentación.

3.2. Solución Planteada

En este punto se plantea la arquitectura inicial de SecondTutor que surge del estudio anterior, ya teniendo determinadas las herramientas a utilizar tras el estudio del estado del arte presentado en el capítulo 2.

3.3. Descripción

La aplicación a implementar se compone de un componente en ATutor y otro en Second Life. El componente en ATutor se encargaría de proveer a los instructores de una interfaz para generar y administrar escenas y el componente en Second Life es el que se encargaría de interpretar las escenas en las que participan los alumnos.

Las Escenas

Como punto central de la aplicación se ubican las escenas, es en su ejecución donde aparecen las posibilidades que brinda el uso de las tres

dimensiones provistas por el mundo virtual de Second Life, así como la facilidad de uso y capacidad de acoplamiento que brinda el LCMS ATutor. Las escenas son ejecutadas por objetos que ofician de "actores virtuales", estos actores tienen scripts de ejecución que permiten programar un comportamiento a cada uno de forma individual, así como la posibilidad de sincronizar la ejecución de varios actores en momentos dados por la trama de la escena para mantener la lógica de la misma.

Los "actores virtuales" serán entonces creados con las posibilidades de edición y modelado en tres dimensiones que provee la plataforma Second Life, para luego ser utilizados en las escenas de manera no exclusiva, es decir, un actor podrá realizar varios papeles para distintas escenas.

Componente ATutor

Como se mencionó anteriormente, el componente en ATutor provee la interfaz para realizar la creación y administración de las escenas. También en este componente se debía crear una base de datos capaz de guardar la información sobre las escenas. Y un sub-componente encargado de la comunicación con la plataforma de Second Life.

Para realizar estas tareas manteniendo las funcionalidades que brinda el LCMS, se aprovecharía la posibilidad que brinda ATutor de aceptar la inclusión de módulos independientes al sistema. Se crearía entonces un módulo que englobe los tres sub-componentes de SecondTutor mencionados: interfaz, base de datos y comunicación.

Sobre la interfaz de comunicación con los usuarios, era importante que fuera lo más similar posible a la de ATutor, para así facilitar su uso y extender sus funcionalidades sin dejarlas de lado.

La base de datos debía ser implementada como una extensión de la base de datos de ATutor tratando de no duplicar datos de, por ejemplo, cursos o alumnos. También era importante crear la misma de forma que sea fácilmente extensible en trabajos futuros, externos a este proyecto.

Sobre la interfaz de comunicación con Second Life, la misma debía ser visible desde uno o varios objetos en el mundo virtual y poseer la capacidad de transmitir la información sobre las escenas a los "actores virtuales", así como información adicional para la gestión de las escenas en el mundo virtual, inicio, final o alumnos habilitados.

Componente Second Life

El componente en Second Life debía poseer la información sobre el grupo de objetos que ejecutarían las escenas. En particular la información correspondiente a las características físicas de los actores y un método para referenciar a los mismos, por ejemplo su código o nombre.

Además de las características físicas, los actores virtuales debían ser provistos de una forma de interpretar las escenas guardadas en ATutor, pudiendo realizar un conjunto básico de acciones, cada una con sus parámetros (por ejemplo que para un movimiento se pudiera aceptar dirección y distancia del mismo, o coordenadas finales), y luego que se proveyera la posibilidad de combinar estas acciones para formar los guiones de cada actor de la escena.

Para finalizar el componente en Second Life, debía poseer la capacidad de comunicarse con el componente en ATutor, siendo capaz de recibir los datos de las escenas a ejecutar, y de gestión de usuarios, escenas, etcétera. También este módulo debía ser capaz de enviar datos sobre las ejecuciones de las escenas para

su posterior análisis y evaluación.

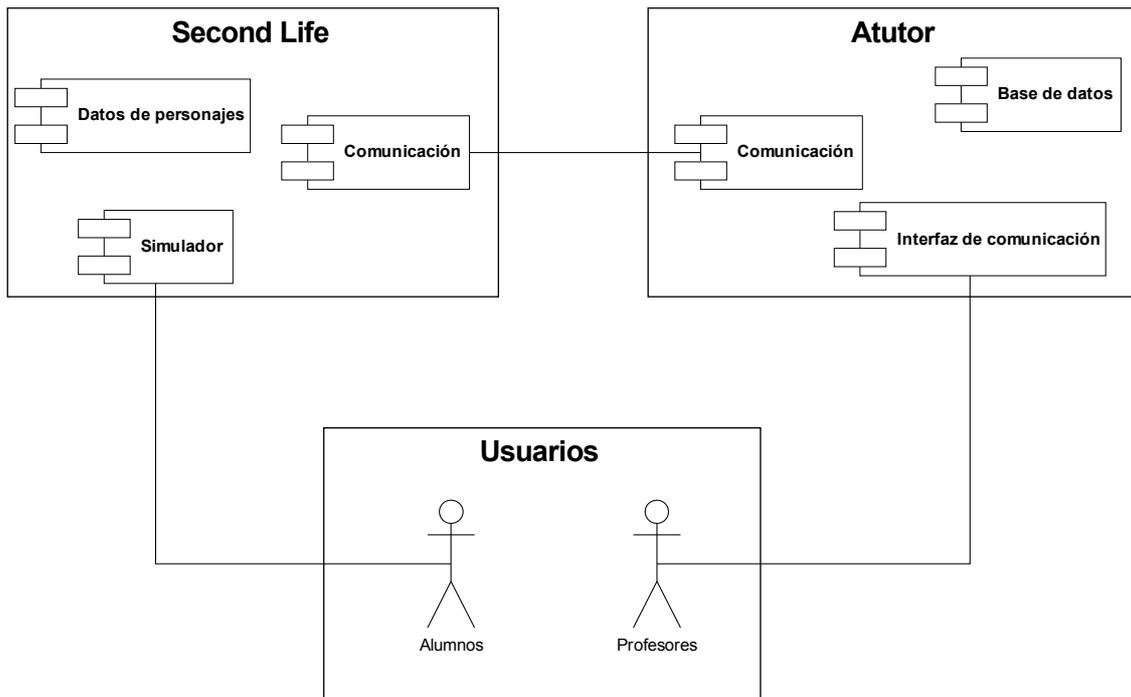


Figura 1: Diagrama de componentes.

3.4. Conclusión

Desde el punto de vista práctico, SecondTutor es una forma de introducir fácilmente la interacción de un mundo virtual en la educación a distancia. Profesores de cualquier área, sin necesidad de conocimientos ni experiencia previa en herramientas de realidad virtual tienen la posibilidad de programar guiones a robots encargados de instruir o evaluar conocimientos dentro del mundo virtual. Estos guiones son programados dentro de un módulo de ATutor, a través de una interfaz totalmente análoga a la del LCMS.

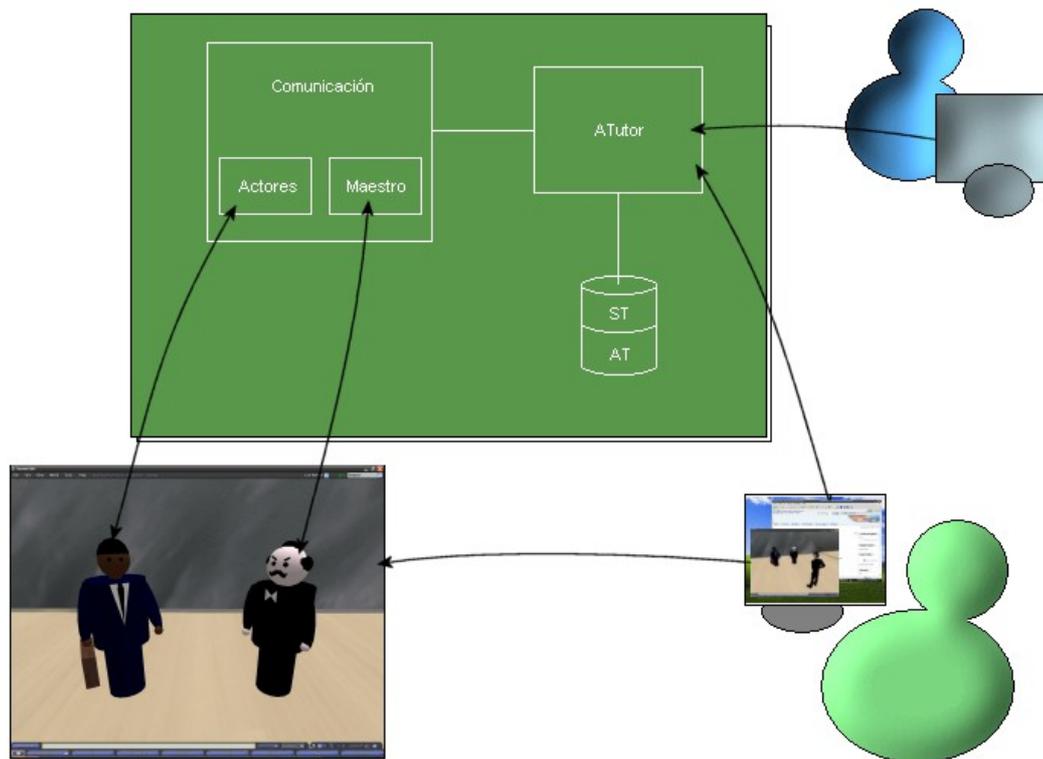


Figura 2: Distribución de los principales módulos de SecondTutor y su interacción con distintos actores.

Además, los alumnos se ven beneficiados al poder utilizar herramientas de realidad virtual como complemento de la educación a distancia, a raíz de todas las ventajas de esta clase de aplicaciones, como la interacción con objetos u otros alumnos que es sólo comparable con la del mundo real. Y todo esto sin dejar de utilizar los métodos tradicionales que ya acostumbraba, pues esta tecnología puede ser introducida de forma gradual por parte de los docentes y sólo en las situaciones en que crean conveniente.

Desde el punto de vista técnico, es una aplicación Web con una arquitectura modularizada, construida a partir de distintas tecnologías y lenguajes de programación.

4. SecondTutor

En el presente capítulo se describe la solución propuesta y desarrollada. Se exponen la arquitectura, la base de datos y otros puntos del diseño e implementación del sistema construido, detallando cada uno de los módulos que componen la solución alcanzada. En la sección Arquitectura se describe la solución en términos de componentes y comunicación entre ellos. La sección Base de Datos describe el esquema de almacenamiento de los datos del sistema. Luego se detallan las dos clases de robots creados y se describen en profundidad el módulo ATutor, el módulo Second Life y la comunicación entre ellos.

Para obtener un mayor nivel de detalle consultar el documento anexo "Documentación funcional".

4.1.Arquitectura

Si bien la arquitectura consta de varios módulos, es pertinente presentar en este documento los dos principales, el módulo ATutor y el módulo Second Life. En el siguiente diagrama se presentan estos dos componentes así como la interacción entre ellos y con los usuarios del sistema.

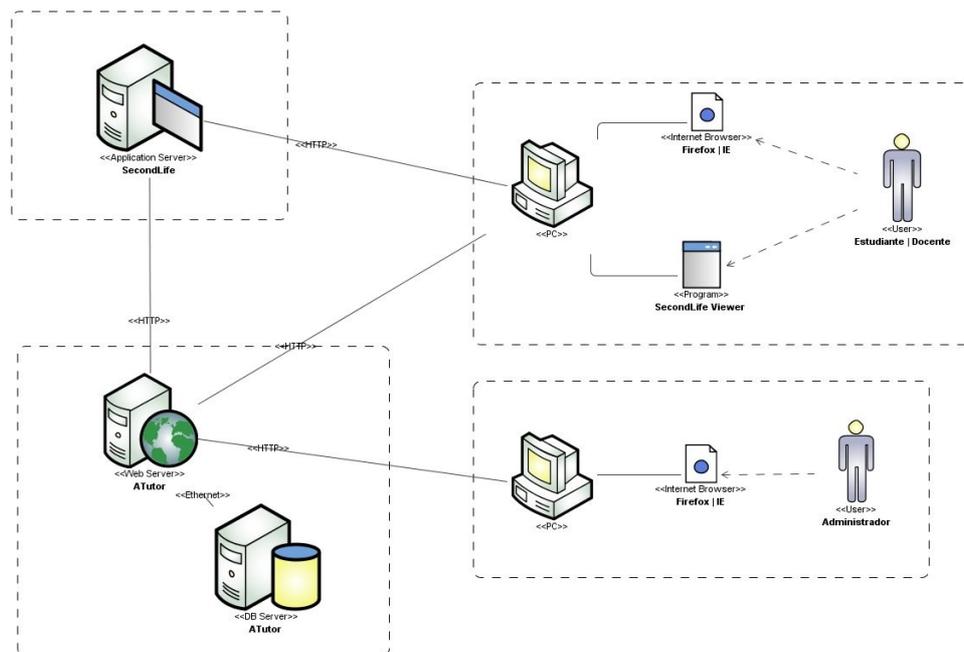


Figura 3: Arquitectura del sistema.

SecondTutor se encuentra definido en una arquitectura distribuida. El despliegue del sistema es a través de varios servidores; servidor de base de datos, servidor Web y servidor Second Life. El servidor Web y el de base de datos a menudo son el mismo, y es por lo anterior que nos referimos a ellos como servidor ATutor. También se diferencian dos clientes, el cliente Web para usuario final donde se definen dos roles (estudiante y docente) y la aplicación cliente de Second Life. Es así que cada rol tiene su interfaz de usuario independiente. Por más información sobre el despliegue de la aplicación consultar el documento anexo "Documentación Funcional".

Es un requisito para el usuario final tener un navegador Web compatible con HTTP/1.1 y el cliente de la aplicación Second Life previamente instalados.

ATutor presenta una arquitectura flexible, permitiendo añadir contenido de terceros al sistema en forma de módulos, esta funcionalidad fue explotada por el

grupo de trabajo en la creación del módulo ATutor.

El módulo Second Life es un conjunto de archivos codificados en LSL¹³, un lenguaje de programación utilizado para controlar el comportamiento de objetos en Second Life, y objetos con representación tridimensional, a los cuales llamamos robots.

La comunicación entre ambos módulos es vía HTTP, la cual se explica con más detalle en la sección 4.5.

4.2.Base de Datos

La base de datos utilizada por SecondTutor es quizás el elemento más importante, pues es el nexo principal con ATutor, la adaptabilidad y flexibilidad fueron los pilares del diseño de esta base de datos sin dejar de lado la inclusión de componentes de seguridad sobre los datos. El núcleo de la base de datos de SecondTutor es la propia base de ATutor, donde se almacena la información de los estudiantes, cursos, evaluaciones y otros elementos propios de esta herramienta. Esta se extendió de forma que soporte el almacenamiento de los datos propios de SecondTutor, más específicamente se agregaron tablas para los eventos, estudiantes, robots y evaluaciones, así como para la relación entre estos.

La gran flexibilidad del sistema radica sobre la tabla *eventos*. En la misma se definen los diferentes tipos de eventos que los robots pueden realizar (comportamiento de los robots), así como la información necesaria para ejecutarlos (parámetros de los eventos). Ejemplo de los diferentes eventos son movimiento, reproducir sonido, realizar pregunta, esperar un evento, sincronizar con otros robots, entre otros. Por una guía detallada sobre la definición y uso de los eventos realizables consultar el documento anexo "Manual de Usuario".

La base de datos se puede definir en término de regiones, donde cada una está compuesta por un conjunto de tablas estrechamente relacionadas entre sí. A continuación se detalla el conjunto de regiones:

Región	Tablas	Funcionalidad
Cursos	at_courses	Vincula las evaluaciones existentes en el sistema con un curso existente en el sistema ATutor.
Estudiantes	st_estudiantes, st_estudiantes_corridas	Vincula al estudiante existente en ATutor con su representante virtual (avatar) en Second Life.
Evaluaciones	st_corridas, st_evaluaciones, st_robots_evaluaciones	Define evaluaciones de tipo SecondTutor, indica los robots participantes para una evaluación definida, registra las diferentes instancias en las que son tomadas las evaluaciones.
Históricos	st_historicos, st_historico_respuestas	Registra el histórico de un usuario en una evaluación determinada. Entre otros, registra las respuestas brindadas por un estudiante y texto ingresado por el usuario

13 LSL, Linden Scripting Language (http://wiki.secondlife.com/wiki/LSL_Portal).

		en Second Life
Robots	st_robots, st_robots_evaluaciones	Registra información de los robots y de participación de estos en las evaluaciones.
Eventos	st_eventos, st_eventos_aleatorios, st_eventos_movimientos, st_eventos_mensajes, st_eventos_estudiantes, st_eventos_detectar, st_eventos_timeout, st_eventos_caminos, st_caminos, st_preguntas, st_respuestas, st_respuestas_texto, st_respuestas_timeout, st_funciones, st_parametros_funciones, st_eventos_die, st_eventos_sincronizar, st_sincronizaciones	Define las posibles acciones que puede tomar un robot a lo largo de la ejecución de una evaluación.

Los eventos se comportan según el siguiente esquema:

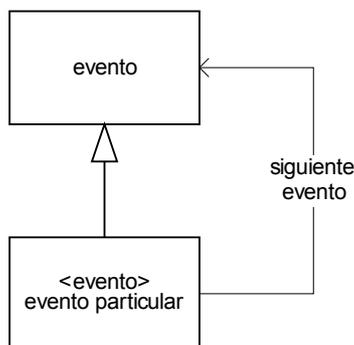


Figura 4: Eventos en SecondTutor.

Para definir un nuevo tipo de eventos, a nivel de base de datos, se debe crear una nueva tabla, que a su vez debe contener una referencia a la tabla *st_eventos*. Por más información consultar el documento Documentación Funcional.

4.3.Robots en Second Life

En el ambiente de Second Life es posible representar objetos del mundo real. Por ejemplo, un teléfono en el mundo virtual estaría representado por medio de un objeto con forma de teléfono con el que el usuario puede interactuar. A estos objetos se le puede agregar comportamiento por medio de *scripts* que se adjuntan al mismo. Se puede entonces, por ejemplo, crear un objeto con forma de humano que reaccione cuando haya un usuario cerca e interactúe con él de varias maneras e incluso participar de la conversación por medio de mensajería instantánea o de sonidos.

Esta posibilidad fue usada para crear objetos, llamados robots, con un conjunto de acciones posibles preestablecidas, llamados eventos, accionados a partir de guiones creados y almacenados en el componente ATutor del sistema.

Se cuenta con un grupo de robots a los cuales se les asignan eventos a realizar, llamados *robots actores*. Además se tiene otro tipo de robot, el cual realiza la función de gestionar las escenas creando o eliminando robots actores según sea necesario, este robot será referido como *robot maestro*.

A continuación se explican las características y el comportamiento de los dos

tipos de robots existentes en SecondTutor, creados en este proyecto de grado.

El robot maestro

Este robot realiza las funciones de gestión de las evaluaciones y es único en el sistema. Se le llama robot por ser un objeto con un *script* adjunto, sin embargo no realiza ninguna acción visible para el usuario y puede aparecer en la escena como un cubo, una planta, una alfombra, o cualquier otro objeto que se pueda crear con las herramientas de modelado que brinda Second Life.

Posee dos funciones principales, controlar la presencia de estudiantes y monitorear a los robots actores. Estas funciones las realiza en dos pasos. En el primer paso recibe la lista de estudiantes y la de robots actores activos desde el servidor de ATutor. Luego inspecciona el área de la escena para identificar estudiantes conectados y actores activos, con el fin de analizar el estado actual de la escena y tomar acciones correctivas en caso de necesidad. Esto lo hace de manera continua cada cinco segundos aproximadamente, no es posible controlar exactamente este tiempo por la forma en que se realiza en Second Life la comunicación con sistemas externos (esto se explica con más detalle en el punto de comunicación ATutor – Second Life).

Sobre los estudiantes: el maestro recibe la lista de estudiantes que tengan una sesión activa actualmente en ATutor y hayan iniciado una evaluación. Con esta lista inspecciona la escena para identificar los estudiantes presentes en la misma. Si algún estudiante no está presente, envía esta información al módulo ATutor para dar de baja al estudiante de la evaluación actual (se guarda la hora de finalización de la actividad del estudiante en la escena, y se finaliza la escena en caso de ser el último estudiante). Si hay un estudiante y la evaluación no ha sido iniciada, es tarea del robot maestro crear los robots actores correspondientes a la escena que se va a ejecutar.

Sobre la lista de actores activos: el robot maestro, una vez que recibe la información de que hay por lo menos un estudiante conectado y que lo haya detectado en Second Life, inicia el proceso de pedir la lista de actores activos. Esta lista indica qué actores deben ser creados en el mundo virtual, para esto el robot maestro tiene en su inventario¹⁴ todos los robots actores existentes en el sistema y los crea y elimina según sea el caso.

Cuando inicia la evaluación crea todos los robots y cuando detecta que ya no hay usuarios conectados o se terminó la evaluación desde ATutor, elimina todos los robots de la escena. Si un robot actor falta por algún motivo el robot maestro lo crea nuevamente tantas veces como sea necesario. Este robot actor "faltante" seguirá con la escena donde finalizó anteriormente sin necesidad de volver a iniciarla.

Los robots actores

Esos robots son los que realizan las escenas en Second Life ejecutando los guiones creados en el módulo ATutor. Componen la parte visible del sistema en el mundo virtual.

Los robots actores son identificados dentro del sistema por su nombre. Dentro de Second Life los actores poseen además características físicas correspondientes al objeto utilizado para darle forma al robot, que pueden ser representaciones más o menos fieles del objeto en la realidad dependiendo de la habilidad del diseñador del mismo. Además de la forma física, es posible agregarles sonidos a los robots actores, que pueden ser luego ejecutados como parte del guión de una escena.

¹⁴ Ver glosario.

La creación de los robots actores se compone de tres pasos. Como primer paso un objeto es creado en Second Life con un nombre que lo identifica, se le agrega a su inventario el script LSL de robots actores de SecondTutor. A continuación son agregados al inventario de objetos del robot maestro y finalmente son agregados en el módulo ATutor para poder ser utilizados en las evaluaciones. Dependiendo del propósito del objeto, es posible agregar sonidos al mismo.

A modo de ejemplo, para crear un robot actor que oficie de teléfono, se crea un objeto con dicha forma, se lo nombra "Teléfono", luego se le agrega un sonido acorde, se lo agrega al inventario del robot maestro y se lo da de alta en el módulo SecondTutor de ATutor.

Una vez en el sistema, los robots actores pueden estar activos o no. Un robot actor activo es un robot actor que está participando en la ejecución de una escena y aún no terminó los pasos que conforman su guión en la misma. Como contraparte un robot actor inactivo es un robot actor que no está ejecutando un guión y por lo tanto no es visible en Second Life, sino que se encuentra contenido en el inventario del robot maestro.

Es importante aclarar que un robot actor posee sólo un nombre y una apariencia física, pero es capaz de representar para cada escena, un guión diferente o incluso ninguno (en el caso de que en una escena el robot actor no se utilice). Esta posibilidad de reutilizar los actores, es lo que le da al sistema la capacidad de crear infinitas escenas para diferentes propósitos con un conjunto reducido de robots.

4.4. Módulo ATutor

El módulo ATutor es un conjunto de páginas Web dentro de ATutor, y realiza la gestión de la información relevante a las evaluaciones que se llevan a cabo en Second Life.

La creación de evaluaciones para un curso, la asignación de fecha de la evaluación, así como la asignación de estudiantes, es realizado por medio del sistema ATutor. Con esto se pretende reducir el impacto sobre el aprendizaje de la herramienta para los usuarios asiduos a la plataforma de *e-learning*.

Dentro de una evaluación, como se explicó en puntos anteriores, conviven varios robots. Cada uno contiene un conjunto de eventos a ejecutar. Dichos eventos deben ser programados dentro de este módulo. Se provee una interfaz intuitiva y flexible para la creación de evaluaciones así como asignación de guiones para cada robot participante de una evaluación determinada.

La funcionalidad para crear robots también es responsabilidad del presente módulo. Finalmente, cada usuario es capaz de visualizar un conjunto de resultados sobre una evaluación SecondTutor.

Desde un punto de vista técnico, es un módulo ATutor, codificado en su totalidad en PHP¹⁵ y MySQL¹⁶. También se utilizó JavaScript¹⁷ para enriquecer la interfaz de algunas páginas, específicamente se utilizó Mootools [Mootools, 2008], un potente *framework* JavaScript que permite crear con facilidad código totalmente funcional en los navegadores de Internet más conocidos. También se utilizó la

15 PHP, PHP Hypertext Pre-processor, lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas.

16 MySQL, sistema de gestión de base de datos relacional, multihilo, multiusuario y multiplataforma.

17 JavaScript, lenguaje de programación interpretado, utilizado principalmente en páginas Web, con una sintaxis semejante a la de Java o C.

técnica conocida como Ajax¹⁸ con el mismo propósito, el de aumentar la interactividad, velocidad y usabilidad.

En resumen, el módulo ATutor proporciona una serie de herramientas para interactuar con el sistema, ya sea desde el punto de vista de un docente (creación de evaluaciones, creación de robots, asignación de guiones a robots en evaluaciones, obtener estadísticas sobre las evaluaciones ya tomadas) así como del punto de vista de un estudiante (consultar resultados y configurar parámetros de Second Life).

4.5. Comunicación ATutor - Second Life

La comunicación entre ATutor y Second Life se realiza de forma unidireccional, mediante llamadas HTTP desde los robots en Second Life, invocando funcionalidades provistas por archivos PHP en el servidor de ATutor.

Estos archivos PHP proveen una interfaz de comunicación que es utilizada por las dos clases de robots en el sistema y por lo tanto está dividida en dos partes, una encargada de la gestión de las escenas en Second Life, que será utilizada por el robot maestro, llamada interfaz del maestro. Y otra utilizada para proveer a los actores los guiones a realizar, llamada interfaz de actor.

La interfaz del maestro es utilizada para recibir en Second Life la lista de estudiantes activos, la lista de robots actores activos, dar de alta robots actores en el sistema, y dar de alta coordenadas en la escena con un nombre.

La interfaz de actor, por otra parte, provee a los robots actores la funcionalidad de obtener su siguiente evento, repetir el evento actual del robot, enviar la respuesta de una pregunta, y guardar la conversación de mensajería instantánea en la escena (estas dos últimas responden a requerimientos particulares de dos tipos de evento, el evento pregunta y el evento registrar diálogos). Debido a que HTTP es un protocolo sin estado, todas las funciones expuestas por esta interfaz requieren que se les proporcione el identificador del robot que las invocan.

El método utilizado para realizar esta comunicación, es mediante la función *IIHTTPRequest* provista por LSL. Esta función hace una petición, mediante el protocolo HTTP, a una URL¹⁹ dada, y luego de realizado esto acciona el evento *http_response*, el cual posee el código necesario para decodificar el mensaje obtenido del servidor ATutor. De aquí que la comunicación entre Second Life y ATutor se realiza de forma asíncrona y no de forma sincrónica, como es habitual en la comunicación por HTTP.

4.6. Pruebas

Las pruebas realizadas al sistema fueron varias e incluyeron pruebas unitarias, pruebas de integración y pruebas funcionales. En la siguiente figura se muestra la implementación de las pruebas sobre el sistema.

¹⁸ Ajax, Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas.

¹⁹ URL, Uniform Resource Locator, es decir, localizador uniforme de recurso.

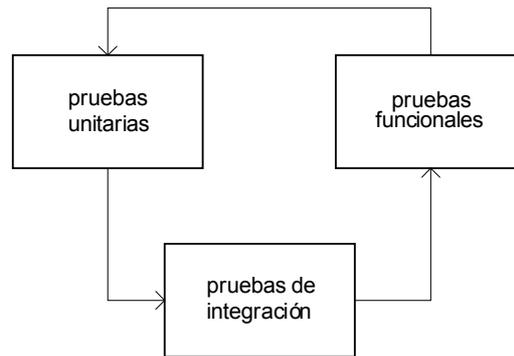


Figura 5: Esquema de las pruebas realizadas.

Pruebas unitarias:

Respecto a las pruebas unitarias, cabe destacar que al culminar la etapa de implementación cada funcionalidad pasaba por un conjunto de pruebas preliminares por el propio desarrollador antes de liberarla a la comunidad. Una vez versionado el código de la funcionalidad, era responsabilidad del resto del equipo de trabajo realizar las pruebas cruzadas de las funcionalidades, evitando así la inclusión subjetiva de casos de prueba.

Una vez detectados errores en los módulos en las pruebas unitarias, los mismos eran introducidos en la planilla de reporte de errores.

Pruebas de integración:

Finalizada la etapa de pruebas unitarias se proseguía con las pruebas de integración, esto se realizaba al momento de integrar las funcionalidades al sistema ya existente. Para ello, el equipo completo definía una serie de pruebas a efectuar y se ejecutaban de manera independiente. Nuevamente, los errores encontrados en esta etapa eran reportados en la planilla de errores.

Pruebas de función (funcionales):

En esta fase se probaban casos de uso completos. La metodología de trabajo para esta etapa fue dividir los casos de uso posibles sobre el sistema en función de los desarrolladores de manera tal que quien implementaba la mayoría de las funcionalidades necesarias para ese caso no participara en la prueba. Una vez reportados los errores comunes en la planilla de errores, se realizaba una comprobación sobre los errores encontrados, se priorizaban y se resolvían para volver sobre las pruebas unitarias.

4.7. Ejemplo práctico

Se presenta a continuación un ejemplo práctico, donde se pueden apreciar las principales características de SecondTutor. Básicamente se creará una escena en donde se le presenta a un estudiante un teléfono con el cual deberá interactuar.

4.8. Precondiciones

En este ejemplo se supone que se cuenta con un robot maestro y un robot actor, este último llamado *Teléfono* al cual se le añadió un archivo de sonido llamado *ring* en su inventario.

4.9. Objetivo

La idea principal de este ejemplo es mostrar cómo crear una evaluación Second Tutor, donde intervenga un robot actor que tenga el siguiente comportamiento:

- Que repiquetee como un teléfono.

- Que registre la manera en que el estudiante lo contesta.
- Una vez el estudiante conteste, que se le pregunte hasta qué hora está abierta la oficina, con un par de opciones.
- Que se despida.

4.10.Pasos a seguir

A continuación se presentan una posible secuencia de pasos a seguir para conseguir este comportamiento y un conjunto de imágenes tomadas a partir de la realización de esta escena.

1. Crear un nuevo curso, utilizando la opción que provee ATutor por defecto.

[My Courses](#) | [Browse Courses](#) | [Create Course](#)

Create Course

[Create Course](#)

★ Title
Curso ejemplo

Primary Language
English

Description
Curso utilizado para la creación del ejemplo de interacción.

Category
[Uncategorized]

Export Content
 Not available on any of the pages
 Available only for top level pages
 Available on every page

Syndicate Announcements
 Enable Announcements syndication via RSS
 Disable Announcements syndication

Access
 Public: Available to all users with or without an ATutor System Account. Login not required. Enrollment is optional.
 Protected: Available only to users with an ATutor System Account. Login required. Enrollment is optional.

Figura 6: Nuevo curso.

2. En la sección Evaluaciones hacer clic sobre *Crear Evaluación* y completar los datos para la nueva evaluación.

Crear Evaluación

★ Título
Evaluación simple Telefono

Fecha comienzo
19 July 2007 at 8:00 24hr

Fecha fin
31 July 2008 at 23:55 24hr

Descripción
Evaluación de ejemplo de simple interacción entre los usuarios y el robot Telefono

Robots asignados
 Araña
 Computadora
 Escritorium
 Pibe
 Cliente 1
 Telefono
 Viejero
 Vieja

Guardar Cancelar

Figura 7: Nueva evaluación.

3. Ahora, en la sección *Evaluaciones* se listará la nueva evaluación, a la cual se le puede asignar un script haciendo clic sobre el botón *Asignar Script*.
4. Crear un evento de tipo mensaje. Ingresar el Id del evento en cuestión, por ejemplo 1. Elegir en *Tipo del mensaje* la opción *Sonido* y en *Texto del mensaje* el nombre del sonido, en este caso ingresar *ring*. Elegir un Id del evento siguiente, por ejemplo 2. Establecerlo como inicial, haciendo clic en *asignar inicial* e ingresar en *Id del evento inicial* el Id del evento creado, 1. En la Figura 8 se puede observar el estado luego de crear el primer evento.

Asignar Script

Eventos del robot: Telefono

Eventos asignados:

Id	Evento	Siguiente		
cambiar evento inicial				
1	Mensaje "ring"	2		
2		*		

▼ Eventos sin asignar:

Id	Evento	Siguiente

[Go to Top](#)

Nuevo Evento

Tipo del nuevo evento

- Aleatorio
- Detección
- Espera
- Final
- Link

Figura 8: Sección de Asignar Script luego de crear el primer evento.

5. Crear un evento de tipo detección, esperando que el alumno conteste, asignándole el Id 2. Elegir en *Tipo de detección* la opción *Detectar interacción* y en *Tiempo de detección* el intervalo entre los tonos del teléfono, por ejemplo 5 segundos. En *Siguiente por timeout* elegir el Id del primer evento, 1, para que el teléfono vuelva a sonar. Y en *Siguiente por interacción* el evento detección a crear a continuación, en este caso 3.
6. Crear un nuevo evento de tipo detección, pero en este caso de tipo de detección *Registrar diálogos*. Asignarle 3 como Id, y en tiempo de detección el tiempo máximo que el estudiante puede demorar en responder, por ejemplo 5 segundos. En *Siguiente por timeout* ingresar el Id del evento a ejecutarse si el estudiante no dice nada en los 5 segundos siguientes a contestar el teléfono. En este caso ingresar el Id de la pregunta a crear luego, más exactamente el 4.
7. Luego crear el evento para el caso en que el estudiante demore demasiado en contestar, en este caso puede ser una despedida que demuestre la disconformidad del posible cliente. Para ello se puede crear un evento de tipo mensaje, asignarle como Id el número 5, y el tipo *Público* para que pueda ser leído en la sección del chat de Second Life e ingresar como texto: "Gracias por nada.". Luego elegir como Id del siguiente evento, un evento final que aún no está creado, por ejemplo el 6.
8. Crear el evento final mencionado. Elegir en *Tipo del nuevo evento* la opción *Final* y en la sección *Editar evento* ingresar el Id 6.
9. Crear después el evento para el caso en que el estudiante responda con alguna de las opciones. En este caso se puede elegir un mensaje, crear uno nuevo, con Id 7 y texto "Muchas gracias, muy amable.". Y como Id del evento siguiente utilizar el Id del evento final recién creado, el 6.
10. Crear la pregunta, ingresar como Id el número 4 y en *Texto de la pregunta* "Buenas tardes, ¿podría decirme hasta qué hora atienden al público hoy?".

Marcar la pregunta como *Pregunta con timeout* e ingresar en *Siguiente* el Id del mensaje de disconformidad, el número 5. En *Tiempo* ingresar el tiempo máximo que esperamos por una respuesta, por ejemplo 5 segundos, y en *Puntaje* el puntaje asignado por esta pregunta en caso que se demore este tiempo, por ejemplo -10. En la sección *Respuestas* crear las posibles respuestas, en este ejemplo sólo se necesitan dos. Hacer clic en *nueva respuesta* elegir en *Tipo múltiple-opción* y en texto una de las opciones, por ejemplo "A las 19:00", en puntaje 10 (la opción correcta), y en siguiente el Id del mensaje de agradecimiento, en este ejemplo el 7. De la misma manera crear una nueva respuesta con el texto "A las 18:30" y puntaje -5, y con el mismo Id de evento siguiente.

Course Server
Curso ejemplo
 Home Forums Glossary File Storage Manage

Asignar Script

Eventos del robot: Telefono (agregar)

Eventos asignados:

Id	Evento	Siguiente
	cambiar evento inicial	
1	Mensaje "ring"	2
2	Detectar interacción	3
3	Detectar registro de diálogos	*
4	Pregunta "Buenas tardes, ¿podría decirme hasta qué hora atienden al público hoy?"	*
7	Mensaje "Muchas gracias, muy amable"	6
6	Evento final.	*
5	Mensaje "Gracias por nada."	6

Nuevo Evento

Tipo del nuevo evento
 Aleatorio
 Detección
 Espera
 Final
 Link

Crear

Editar evento

Figura 9: Eventos del robot Telefono del ejemplo.

Se debe tener en cuenta que muchas de las decisiones tomadas pueden variar y conseguir el mismo resultado final.

4.11. Tomar la evaluación

Para tomar la evaluación, el usuario (alumno del curso en cuestión) ingresa al sistema y en la sección de evaluaciones puede observar la evaluación Second Tutor recién creada como una evaluación ATutor más. Una vez haya hecho clic sobre la evaluación, se le presentan las preguntas de la evaluación, pero en este tipo de evaluaciones hay sólo una pregunta, que es un hipervínculo al sitio virtual donde se lleva a cabo la escena de la evaluación. Un clic en este vínculo ejecuta la aplicación cliente de Second Life tomando como punto inicial el lugar preestablecido para la evaluación.



The screenshot shows the 'Course Server' interface for 'Curso ejemplo'. It includes navigation buttons for Home, Forums, Glossary, File Storage, and Manage. The main content area is titled 'Take Test' and contains a section for 'Evaluacion simple Telefono'. This section includes instructions, a question titled 'Question 1: Multiple Answer' worth 10 points, a link to 'Tomar evaluacion (ahora)', and a 'Submit' button. A 'Go to Top' link is visible at the bottom right of the page.

Figura 10: Tomar evaluación.

Una vez en Second Life es responsabilidad del estudiante situarse en su puesto, y el robot Maestro identificará que él está tomando la evaluación y creará los robots que intervienen en la evaluación, en este caso el teléfono. En la evaluación creada el teléfono reproducirá el sonido, sin embargo en la Figura 11 también se puede observar en el chat que se muestra el nombre del archivo, y si este es mnemotécnico facilita la comunicación en caso de no contar con equipamiento multimedia, siendo también útil para interactuar con personas con problemas de audición.



Figura 11: Estudiante tomando una evaluación dentro de Second Life.

En este momento el estudiante debe interactuar con los robots, utilizando los conocimientos adquiridos en el curso. En este punto debe contestar la llamada, para eso sólo debe hacer un clic sobre el teléfono. En la Figura 12 se puede ver la pregunta efectuada, que había sido programada como se explicó en el apartado 4.10. También se puede observar el mensaje de disconformidad que se había

programado para el caso en que el estudiante demoraba más de cinco segundos en responder la pregunta.



Figura 12: Estudiante tomando una evaluación dentro de Second Life.

Una vez finalizada la evaluación, el alumno puede verificar el resultado de la misma dentro del módulo de ATutor.

Course Server
Curso ejemplo
[Home](#) [Forums](#) [Glossary](#) [File Storage](#) [Manage](#)

Wednesday January 02, 2008 - 16:39

My SecondTutor

Seleccione la evaluación:

Corrida del día 02/01/2008

Hora	Robot	Pregunta	Respuesta	Puntaje	Tipo
13:29:16	Telefono	Buenas tardes, ¿podría decirme hasta qué hora atienden al público hoy?	Pregunta vencida a los 5 segundos.	-10	TO
Total:				-10	

Content Navigation

Home
 Welcome To ATutor

Related Topics

None Found.

Users Online

matias callmares
 Guests are not listed

Glossary

N/A

[Go to Top](#)

Figura 13: Resultados de la evaluación.

4.12. Conclusiones

SecondTutor es una aplicación que permite el uso de Second Life como apoyo a la educación a distancia. Su arquitectura es distribuida, y el diseño de la base de datos permite la extensión de la aplicación sin la necesidad de una reingeniería total.

Los robots, objetos con comportamiento dentro de Second Life, juegan un papel de vital importancia a la hora de la evaluación o capacitación del estudiante.

Su creación se realiza de manera simple, al igual que la asignación de los guiones que rigen su comportamiento.

El módulo ATutor a su vez se divide en dos módulos, uno de comunicación, que permite el intercambio de información con los robots, y el módulo Web, que permite la configuración por parte de los docentes así como otras funcionalidades para el resto de los usuarios. Este módulo presenta un alto acoplamiento con ATutor, pero esto también ayudó a una buena integración y a generar una interfaz depurada y usable.

La comunicación entre los dos módulos principales es sencilla y permite fácilmente agregar soporte para nuevas aplicaciones de mundos virtuales.

El producto final pasó por varias etapas de pruebas, y por la naturaleza distribuida de la arquitectura, y la forma en que fue implementado, se realizó una gran cantidad de pruebas de integración entre los módulos.

5. Cronogramas y organización

En este capítulo se presenta la organización del grupo, así como las estimaciones previas y el cronograma de actividades.

5.1. Organización del grupo de trabajo

Durante el transcurso del proyecto se realizaron distintas reuniones de coordinación entre los integrantes del equipo. La frecuencia de las mismas se vio modificada dependiendo de la etapa en la que se encontraba el desarrollo del proyecto. Durante el estudio del estado del arte y las herramientas a utilizar las reuniones se realizaron en forma semanal. Durante la fase de análisis y diseño de la solución elegida, las reuniones presenciales continuaron siendo semanales y se agregó la modalidad de reuniones virtuales a través de mensajería instantánea, video-conferencias y reuniones en mundos virtuales de forma casi diaria. Durante la fase de implementación y pruebas, las reuniones presenciales pasaron a ser quincenales, continuando con las reuniones virtuales cada tres o cuatro días.

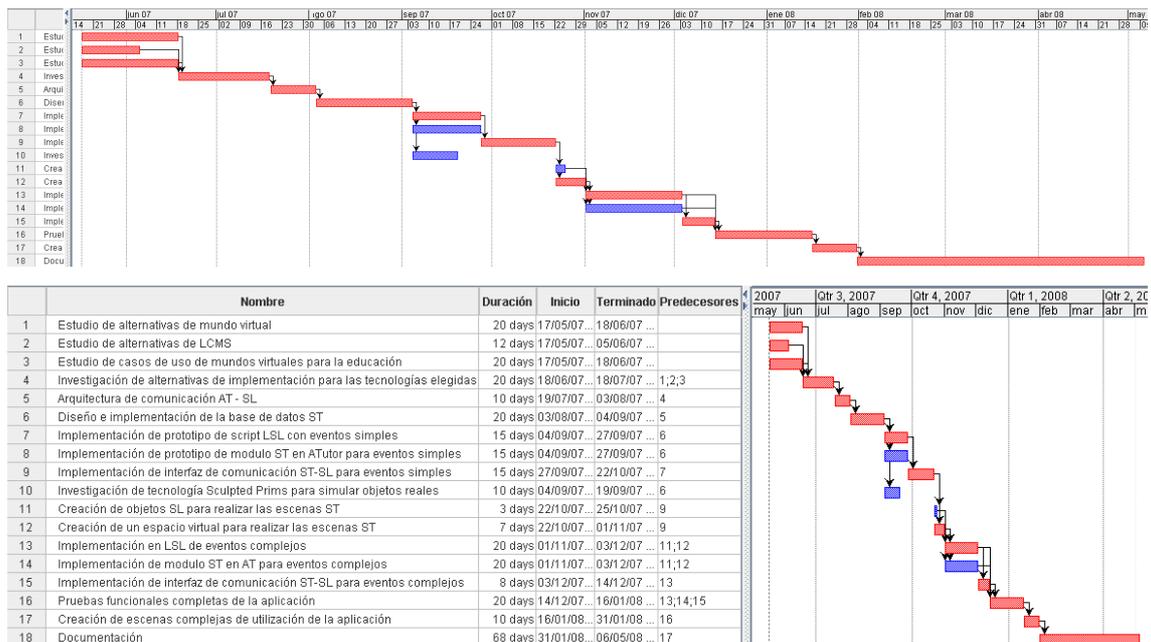
Como complemento de las reuniones mencionadas, se utilizaron documentos compartidos en Internet para registrar los errores encontrados y hacer seguimiento de los mismos.

También se utilizó un sistema de control de versiones (CVS²⁰) para mantener los cambios de todo el trabajo y poder colaborar de forma remota y cómoda [Sourceforge, 2008].

Además, a lo largo del proyecto se realizaron reuniones con los tutores, comenzando de manera semanal, pasando luego a reuniones quincenales y después cuando se presentaba la necesidad.

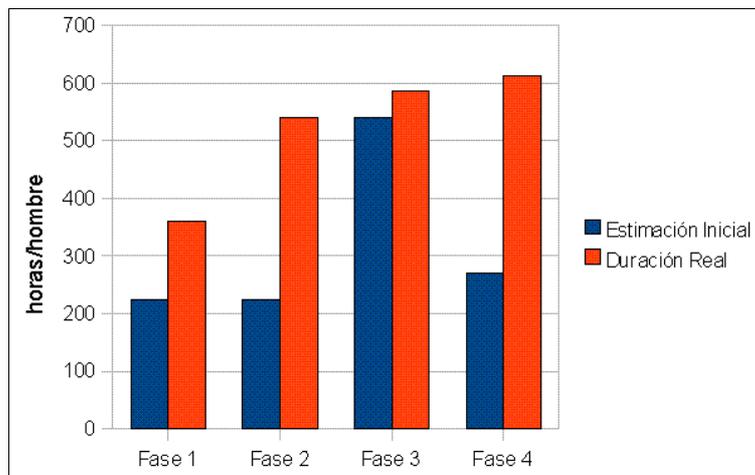
5.2. Cronograma

5.2.1. Diagrama de Gantt



²⁰ CVS, Concurrent Versions System.

5.2.2. Estimaciones de tiempo



	Fase 1	Fase 2	Fase 3	Fase 4
Estimación Inicial	225	225	540	270
Duración Real	360	540	585	612

Fase 1: Estudio del estado del arte y posibilidades de implementación

Fase 2: Diseño y prototipado

Fase 3: Implementación y pruebas

Fase 4: Documentación

6. Resultados y conclusiones

En este proyecto se plantea el desarrollo de un ambiente virtual tridimensional de apoyo a las actividades de la enseñanza. Se cumplió con el objetivo al desarrollar un prototipo avanzado que permite utilizar el mundo virtual de Second Life como complemento del gestor de contenidos educativos ATutor. Al producto desarrollado se lo bautizó con el nombre de SecondTutor. La elección de estas herramientas se tomó luego de un estudio a profundidad entre varias aplicaciones de estos géneros.

Una vez elegida ATutor como la herramienta de LCMS a utilizar, se intentó en todo momento que la interfaz Web de SecondTutor sea lo más similar posible a la de ATutor con el fin de facilitar su uso por parte de docentes habituados a la herramienta. De esta manera sin instalar ninguna aplicación extra, desde el navegador de Internet, se pueden programar los guiones para los robots, que van a ser posteriormente ejecutados en Second Life.

Otra característica de SecondTutor es el alto acoplamiento con las herramientas Second Life y ATutor. Esto no significa que sea una desventaja, pues permitió que se aprovecharan funcionalidades propias de cada una. Por otro lado, la arquitectura modularizada permite de manera muy simple extender SecondTutor para otras plataformas de mundos virtuales, incluso soportar varias de estas simultáneamente. Con ATutor, donde la dependencia es más fuerte, si se quisiera cambiar la herramienta LCMS utilizada, habría que rescribir casi completamente el módulo creado, donde sólo quedaría la idea, probablemente parte de la base de datos y quizás parte de la comunicación.

A continuación se presentan los objetivos logrados, las limitaciones del sistema diseñado e implementado y algunas de las posibles mejoras y las razones por las que no están disponibles en esta versión.

6.1. Objetivos logrados

Se logró cumplir con la totalidad de los objetivos planteados. Se realizó un estudio del estado del arte, se evaluaron distintas posibilidades de implementación y luego se desarrolló una aplicación de ejemplo.

Se realizó un estudio del estado del arte de las herramientas de gestión de contenidos educativos así como de las de ambientes virtuales tridimensionales y de sus posibilidades de extensión e interacción entre ellas, teniendo en cuenta los objetivos del proyecto. El resultado de este estudio determinó la elección de las dos herramientas sobre las que se desarrolló el producto final sin grandes inconvenientes, probando así que el estudio realizado cumplió con su cometido.

Se evaluaron varias opciones de implementación, donde cada una de ellas se enfocaba sobre las carencias de la anterior. Por lo tanto SecondTutor es el resultado de la evolución de varias ideas, aprovechando así las ventajas que ofrecen las herramientas de LCMS y realidad virtual, sin pretender replicar ni mejorar sus funcionalidades, sino utilizarlas para crear algo nuevo.

En cuanto al último de los objetivos planteados, se logró desarrollar un prototipo totalmente funcional. La aplicación permite a docentes crear escenas con fines educativos dentro de un entorno virtual tridimensional, en el que es posible situar a un estudiante en un contexto determinado para impartirle o evaluar sus conocimientos. Como ventaja adicional, al haber utilizado una herramienta como Second Life, en lugar de crear una nueva, se aprovechan los esfuerzos de la misma para no perder vigencia en cuanto al uso de tecnologías, y por ejemplo permitir la

comunicación mediante VoIP²¹.

6.2.Limitaciones

Por restricciones de tiempo, no se tomó en cuenta en la etapa de verificación que existen dos evaluaciones ejecutándose de manera simultánea. A pesar de que el sistema fue diseñado y construido con esto en mente, no se recomienda ni se garantiza una correcta ejecución en esta situación. Otra limitación del sistema es que un maestro sólo puede estar en una única evaluación activa.

Una funcionalidad que pudo haber sido útil es la de controlar el acceso al predio en la que se está realizando una escena a usuarios que no son estudiantes o que no pertenecen al curso en cuestión. Esto fácilmente pudo haber sido implementado en el robot maestro, o en algún otro que cumpliera con ese rol, e incluso ser configurable desde el módulo ATutor. La razón por la que no se realiza dicho control es que para implementarlo es necesario ser propietario del terreno en Second Life (ver documento anexo "Documentación Funcional").

6.3.Trabajo futuro y posibles mejoras

Una de las posibles mejoras sobre el sistema desarrollado es la creación de módulos para otras plataformas de mundos virtuales.

Otro punto pendiente sería realizar pruebas con usuarios y cursos reales para así poder corregir errores en la interfaz y utilizar sus comentarios y sugerencias para mejorar la aplicación en general.

También se podrían incluir más tipos de evento que, como se mencionó en el capítulo 4.2, puede ser realizado de manera simple.

Otra de las mejoras posibles que admite el sistema desarrollado es la de ejecutar varias escenas simultáneamente. Se podría extender para que se permitan varios maestros en distintos puntos del mundo virtual, cada uno de ellos en escenas diferentes, y todos de forma simultánea.

También se podría trabajar en la concurrencia de los eventos en un robot actor, dicho de otra forma, agregar lógica (sobre el conjunto de eventos) para que más de una acción desencadene un evento. Este cambio impactaría sobre la interfaz del módulo ATutor, para ofrecer esta posibilidad a quien programa las escenas.

El módulo en ATutor, tiene soporte multi-idioma, aunque actualmente no hayan traducciones disponibles. Estas traducciones deben ser insertadas directamente en la base de datos.

A pesar de que el módulo Web es bastante intuitivo, se podría construir una aplicación de escritorio que utilice la misma base de datos. Una de las ventajas es que con una aplicación escritorio se le puede presentar al profesor una vista en dos, o incluso tres, dimensiones de dónde se va a ejecutar la escena, lo que facilitaría aún más las tareas. De esta forma se podrían posicionar los actores involucrados en la escena y simplificaría la creación de eventos de tipo movimiento, además de poder pre-visualizar toda o parte de la escena creada.

21 VoIP del inglés Voice over Internet Protocol.

7. Glosario

- **ATutor** es un LCMS de código abierto basado en la Web y diseñado con el objetivo de lograr accesibilidad y adaptabilidad.
- **Avatar** Es una representación gráfica mediante un dibujo, representación tridimensional o una fotografía de una persona para su identificación.
- **b-learning (blended learning)** Es un proceso docente semipresencial. Un curso dictado en este formato incluirá tanto clases presenciales como actividades de e-learning.
- **CMS (Content Management System)** Sistema de gestor de contenidos, es un programa que permite crear una estructura de soporte para la creación y administración de contenidos por parte de los participantes principalmente en páginas Web.
- **e-learning (electronic learning)** Enseñanza a distancia caracterizada por una separación física entre profesorado y alumnado (sin excluir encuentros físicos puntuales), entre los que predomina una comunicación de doble vía asíncrona donde se usa preferentemente Internet como medio de comunicación y de distribución del conocimiento, de manera que el alumno es el centro de una formación independiente y flexible, al tener que gestionar su propio aprendizaje, generalmente con ayuda de tutores externos.
- **Escena** Conjunto formado por robots, sus guiones, y los alumnos (representados por su avatar) para una evaluación en curso.
- **Evento** Instrucción que puede ser ejecutada por un robot en Second Life.
- **GPL (General Public License)** Licencia pública orientada principalmente a proteger la libre distribución, modificación y uso de software.
- **Guión** Conjunto ordenado de eventos asignados a un robot.
- **Inventario** El inventario de un personaje o robot dentro de Second Life es un repositorio de elementos (sonidos, scripts u otros objetos virtuales) que se puede organizar con carpetas de forma análoga a un sistema de archivos.
- **LCMS (Learning Content Management System)** Sistema de gestor de contenidos (CMS) para la enseñanza.
- **LMS (Learning Management System)** Sistema de apoyo a la enseñanza, un programa que sirve para administrar, distribuir y controlar las actividades de formación presencial o e-learning de una organización.
- **Robot** Objeto de Second Life con un script asociado, el cual determina su comportamiento dentro del mundo virtual.
- **Robot actor** Robot encargado de realizar las escenas en Second Life, ejecutando los guiones creados en el módulo ATutor.
- **Robot maestro** Robot encargado de controlar la presencia de estudiantes y monitorear a los robots actores.
- **Script** Es un guión o conjunto de instrucciones.

- **Second Life** (abreviado como SL), cuya traducción sería "Segunda Vida", es un mundo virtual tridimensional de interacción social creado por Linden Labs y fundado por Philip Rosedale. Es un mundo que está distribuido en una amplia red de servidores y al que se puede acceder a través de Internet. Este programa proporciona a sus usuarios o "residentes" herramientas para modificar el mundo y participar en su economía virtual.

8. Referencias

ATutor, 2008. Learning Content Management System.

URL: <http://www.atutor.ca/>

Último acceso: setiembre de 2008.

Ilias, 2008. Ilias Learning Management.

URL: <http://www.ilias.de/>

Último acceso: setiembre de 2008.

Livingstone, 2006. Putting a second life "metaverse" skin on learning Management systems. (Jeremy Kemp and Daniel Livingstone).

URL: <http://www.sloodle.org/whitepaper.pdf>

Último acceso: setiembre de 2008.

LSL, 2008. LSL Portal.

URL: http://wiki.secondlife.com/wiki/LSL_Portal

Último acceso: setiembre de 2008.

Moodle, 2008. Moodle.

URL: <http://moodle.org/>

Último acceso: setiembre de 2008.

Mootools, 2008. The compact JavaScript framework.

URL: <http://mootools.net/>

Último acceso: setiembre de 2008.

Olat, 2008. Open source LMS.

URL: <http://www.olat.org/>

Último acceso: setiembre de 2008.

Second Life, 2008.

URL: <http://secondlife.com/>

Último acceso: setiembre de 2008.

Sloodle, 2008. Learning System for Virtual Environments.

URL: <http://www.sloodle.org/>

Último acceso: setiembre de 2008.

Sourceforge, 2008. Sitio del proyecto.

URL: <http://sourceforge.net/projects/secondtutor/>

Último acceso: setiembre de 2008.