

*Centro de Informatización Universitaria  
Universidad de las Ciencias Informáticas  
Facultad 1*



*Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas*

*Título: Sistema para la ejercitación de contenidos a través  
de juegos didácticos en entornos tridimensionales.*

*Autores: Alisnay Maturell Elías*

*Guillermo Laborí Oris*

*Tutores: MSc. Dunia Suárez Ferreiro*

*Ing. Liuba Torres Almira*

*La Habana, Junio 2013*

*“En la vida hay algo peor que el fracaso: el no haber intentado nada.”*

*Franklin D. Roosevelt*



## *DECLARACIÓN DE AUTORÍA*

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año.

---

Alisnay Maturell Elías

---

Guillermo Laborí Oris

---

MSc. Dunia Suárez Ferreiro

---

Ing. Liuba Torres Almira

## *Dedicatoria*

*A mi madre por permitirme ser lo que soy hoy.*

*A mi hermana por permitir que sea su ejemplo.*

*A mi abuela por ayudarme a vencer mis miedos.*

*Alisnay*

*A mi madre, por su entrega.*

*Guillermo*

## *Agradecimientos*

*Alisnay:*

*Una inmensa colina de adoquines supera en cantidad nuestra corta travesía por la vida, aun así, debo comenzar. No llegaré al final, pero dejaré plasmada mi huella, al igual que la han dejado aquellas personas que a pesar de su inmensa colina han sido capaces de apoyarme. Personas a las que hoy, les estoy inmensamente agradecida.*

*A mi madre y a mi abuela por su total incondicionalidad y apoyo en todas las decisiones más importantes de mi vida.*

*A mi hermana por ser el motor impulsor durante estos 5 años de carrera.*

*A mi tío por ser un ejemplo de superación a seguir.*

*A mi padrastro por todo su apoyo.*

*A mi padre por sus buenos consejos.*

*A mis amigos Melvis, Royli, Rosalina, Saeret, Ariel, Geydi, Marielis y Dimaris, por haber compartido todo este tiempo y poder contar unos con los otros en los buenos y malos momentos.*

*A mi novio Yaciel, mi churry por ser mi paño de lágrimas en los momentos más tristes de mi vida.*

*A mis tías y primas por su preocupación y por alentarme durante este tiempo.*

*A mis tutoras Liuba y Dunia por toda su ayuda durante el desarrollo de este trabajo.*

*A la profe Dayris por ayudarnos en todo lo que estuvo a su alcance y alentarnos a continuar.*

## *Agradecimientos*

*A mi compañero de tesis, porque a pesar de nuestros malos entendidos hemos trabajado en equipo y logrado salir adelante con este trabajo.*

*A todos los profesores con los que tuve el placer de compartir en estos 5 años y que han logrado que yo pueda llegar a ser profesional.*

*A todas aquellas personas que de una forma u otra me han ayudado a la confección de este trabajo y han logrado que mi estancia en la universidad haya sido de una manera placentera.*

# *Agradecimientos*

## *Guillermo*

*Hay quienes adornan con su presencia los caminos sinuosos de nuestras vidas. Son pocos, pero generan tanta fuerza como el batallón más valiente de guerreros. Inspiran, animan, reviven. A esas personas que me han acompañado en los momentos felices y también en los difíciles quisiera agradecer de manera sencilla:*

- ❖ A mi madre, por la vida, la entrega, la fuerza, y la grandeza de espíritu. Por ser la inspiración de mis actos y ser lo más grande en mi vida.*
- ❖ A mi papá, por la confianza brindada, los consejos y sus desvelos en mi formación.*
- ❖ A mis hermanos.*
- ❖ A mi familia en general por su apoyo incondicional.*
- ❖ A mi novia, por su entrega, confianza, amor y soportar mis pesadeces.*
- ❖ A mis tutores por guiarme en la confección de este trabajo.*
- ❖ A todas las personas que de una forma u otra me ayudaron y apoyaron, en la confección de este trabajo.*

## Resumen

En la Universidad de las Ciencias Informáticas (UCI) la utilización de juegos serios no es generalizada, aunque los mismos pueden beneficiar el proceso enseñanza aprendizaje. Como parte de un proyecto de innovación educativa se desarrolla una aplicación con objetivos didácticos nombrada **SMProg** (**S**oftware-**M**otivación-**P**rogramación) que en su versión web permite la ejercitación y evaluación de la asignatura Técnicas de Programación de Computadoras. Este sistema facilita la creación de entornos tridimensionales a utilizar en juegos serios, lo que garantiza a los usuarios involucrados una mayor comprensión de la lógica del juego, así como una mayor satisfacción por parte de los mismos.

**SMProg** incluye el Sistema de Ejercitación de Contenidos (**SISTEC**), el cual se integra de forma sencilla con los restantes módulos que lo conforman. **SISTEC** le permite al usuario jugar y actualizar sus evaluaciones para que puedan ser consultadas por los profesores y posibilita la selección del tema y la escena en la que se requiera ejercitar, garantizando la plena navegación de los usuarios por el entorno.

En la presente investigación se exponen las características de **SISTEC**, así como los principales artefactos generados durante su desarrollo, utilizando tecnologías libres.

**Palabras claves:** aplicaciones web, entornos tridimensionales, juegos serios, sistemas de ejercitación de contenidos.



## Tabla de contenido

Introducción .....	1
Estructura de los capítulos .....	4
Capítulo I: Juegos en entornos tridimensionales aplicados a la Educación. ....	6
Introducción.....	6
1.1    Juegos serios.....	6
1.2    Mundos virtuales.....	7
1.2.1    Mundos virtuales en la Educación Superior .....	8
1.3    Sistemas de ejercitación de contenidos .....	10
1.4    Juegos serios en la Educación Superior .....	10
1.5    Tecnologías .....	13
1.5.1    Aplicaciones Web .....	13
1.5.2    Servidor Web .....	13
1.5.3    Lenguaje de programación .....	13
1.5.4    Lenguaje de marcas y estilos.....	14
1.5.5    Lenguaje Modelado .....	15
1.5.6    Herramienta CASE .....	15
1.5.7    Metodología de desarrollo.....	16
1.5.8    Entorno de desarrollo integral (IDE).....	17
1.5.9    Sistema Gestor de Base de Datos (SGBD).....	18
Capítulo 2: Propuesta e Implementación de la herramienta “Sistema para la Ejercitación de Contenidos” (SISTEC). ....	19
Introducción.....	19
2.1    Solución.....	19
2.2    Requerimientos del Sistema .....	20
2.2.1    Técnicas de Captura de Requisitos .....	20
2.2.2    Requisitos Funcionales.....	21
2.2.3    Requisitos No Funcionales (RNF).....	23
2.3    Modelado del sistema .....	27
2.3.1    Actores del sistema.....	27
2.3.2    Diagrama de Caso de Uso del Sistema .....	27
2.3.3    Descripción de caso de usos del sistema .....	28

# *Índice de Contenidos*

2.4	Diagrama de clases .....	30
2.4.1	Descripción de Clases .....	33
2.5	Diagramas de Interacción .....	34
2.6	Arquitectura de la solución .....	37
2.6.1	Patrones de Diseño .....	38
2.7	Implementación .....	39
2.7.1	Estándares de codificación .....	39
2.7.2	Implementación del Visualizador 3D .....	41
2.7.3	Trabajo con el XML para el almacenamiento de los escenarios.....	43
2.7.4	Descripción de un flujo de evento del sistema (Jugar).....	44
	Conclusiones Parciales .....	46
Capítulo 3: Pruebas a la herramienta “Sistema de Ejercitación de Contenidos” (SISTEC).....		48
	Introducción.....	48
3.1	Diagrama de componentes .....	48
3.2	Diagrama de Despliegue.....	49
3.3	Pruebas al sistema .....	49
3.3.1	Descripción de Casos de Prueba.....	50
3.3.2	Resultado de las pruebas .....	53
	Conclusiones Parciales .....	55
	Conclusiones Generales .....	56
	Recomendaciones .....	57
	Bibliografías Referenciadas .....	58
	Bibliografía Consultada .....	63
	Glosario de Términos.....	58
	Anexo #1 Especificación de Requisitos.....	65
	Anexo#2 Clases entidades del sistema.....	69
	Anexo #3 Estructura del XML.....	70

## *Índice de tablas*

Tabla 1	Tabla de Requisitos Funcionales del sistema.....	21
Tabla 2	Especificación del requisito Seleccionar asignatura .....	24
Tabla 3	Especificación de requisito Seleccionar tema.....	25
Tabla 4	Especificación de requisito Seleccionar escena .....	25
Tabla 5	Especificación de requisitos Avanzar por la escena .....	26
Tabla 6	Descripción de Actores del sistema.....	27
Tabla 7	Descripción del CUS Cumplir reglas del juego .....	28
Tabla 8	Descripción del CUS Jugar .....	29
Tabla 9	Descripción de la clase Controladora: Controladora_Jugar .....	33
Tabla 10	Descripción de la clase Entidad: Asignatura.....	33
Tabla 11	Descripción de la clase Entidad: Tema.....	33
Tabla 12	Descripción del CP Jugar .....	50
Tabla 13	Descripción CP Cumplir reglas del juego .....	51

## *Índice de Imágenes*

Figura 1 Integración al Proyecto de Innovación Pedagógica.....	20
Figura 2 Diagrama de Caso de Uso (CUS) .....	27
Figura 3 Diagrama de Clases del diseño (Vista-Controlador).....	31
Figura 4 Diagrama de Clases del diseño (Modelo).....	32
Figura 5 Diagrama de secuencia Cumplir reglas del juego .....	35
Figura 6 Diagrama de secuencia Jugar.....	36
Figura 7 Vista Lógica de la arquitectura del sistema .....	38
Figura 8 Estándar de codificación Pascal.....	40
Figura 9 Estándar de codificación LowerCamelCase .....	40
Figura 10 Parámetros de la cámara .....	42
Figura 11 Ejemplo de un archivo XML .....	43
Figura 12 Interfaz principal del sistema .....	44
Figura 13 Interfaz seleccionar Asignatura y tema.....	45
Figura 14 Interfaz Cargar escena.....	45
Figura 15 Interfaz Seleccionar directorio .....	46
Figura 16 Diagrama de componentes de SISTEC.....	48
Figura 17 Diagrama de despliegue de SISTEC.....	49
Figura 18 Resultado de No Conformidades durante las iteraciones .....	55

## Introducción

La utilización de juegos educativos se ha estimulado considerablemente durante los últimos años. En la actualidad se continúan realizando debates e investigaciones identificando juegos orientados hacia un objetivo y otros que son de carácter más social, naciendo así el término “*juegos serios*” el cual fue acuñado para describir juegos que tienen un objetivo educativo y no de ocio. Este tipo de juego posibilita al jugador desarrollar habilidades y conocimientos.

Entre los juegos que ofrecen un potencial interesante en los objetivos pedagógicos están los juegos multijugador masivo; los cuales permiten que se reúnan varios usuarios para el desarrollo de actividades en las que en ocasiones es necesaria la colaboración. En estos juegos se deja en segundo plano la diversión y se combinan los beneficios del juego con la seriedad de los conocimientos.

Los juegos serios son utilizados para incentivar la motivación de los estudiantes por las diferentes materias, teniendo en cuenta que combinan un entorno cuidadosamente trabajado en las diferentes enseñanzas. (1)

El uso de las tecnologías para la creación de estos juegos ha permitido que la apariencia de los mismos sea más real, garantizando establecer una mejor comunicación entre el jugador y el entorno. De esta forma, fueron considerados los entornos tridimensionales o 3D<sup>1</sup>, como una alternativa para simular un entorno artificial o virtual inspirado en la realidad. Los juegos creados en este tipo de entorno marcaron una manera más amena de interactuar con un ordenador. De esta manera, se comenzaron a crear entornos Multi-Usuarios<sup>2</sup>, dentro de los cuales los usuarios pueden introducirse, explorar el mundo virtual y así interactuar con otras personas. (2)

Tal como afirmara Samuel Martínez:

*“Los mundos virtuales ofrecen nuevas formas de comunicación, de reunión y socialización entre las personas.”*

Estos mundos virtuales se consideran espacios propicios para el desarrollo del trabajo en

---

<sup>1</sup> En computación las tres dimensiones son el largo, el ancho y la profundidad de una imagen. Técnicamente hablando el único mundo en 3D es el real, la computadora sólo simula gráficos en 3D. Los objetos en tercera dimensión se basan en formas gráficas (X, Y, Z).

<sup>2</sup> Se le llama multiusuario a la característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente.

equipo y el intercambio de ideas, la discusión y el respeto de las opiniones de los demás. Entre sus características principales se pueden destacar la interactividad y la persistencia y teniendo en cuenta estos aspectos fueron considerados como estrategias serias fundamentales en la modalidad educativa.

Las tecnologías se han convertido en la mejor alternativa para continuar desarrollando la calidad de la educación. Desde su surgimiento ejercen una influencia cada vez mayor en el Proceso-Enseñanza-Aprendizaje (PEA), garantizando que se beneficien todos los niveles de enseñanza, no solo en lo que respecta a la mejora del aprendizaje por parte de los estudiantes, sino que también desempeñan un papel creciente en la formación inicial y permanente del profesorado. (3)

De manera generalizada se utilizan herramientas y *software* educativo en apoyo al PEA, en particular los juegos didácticos y multimedias educativas. A pesar de la importancia que se le concede a este tipo de juegos, el aumento de la edad y el nivel de escolaridad han propiciado la disminución de la utilización de los mismos.

En la Universidad de las Ciencias Informáticas (UCI) la asignatura de Programación es en esencia compleja por el contenido que abarca, siendo históricamente causa de resultados docentes insatisfactorios, por lo que provoca un número importante de bajas del Centro. La asignatura exige al estudiante un alto razonamiento lógico para la resolución del diseño y la implementación de algoritmos, así como el dominio de las Matemáticas para la resolución de los mismos. (4)

En este centro existe una aplicación llamada **SMProg (Software-Motivación-Programación)** la cual permite la ejercitación de contenidos de manera didáctica. Esta aplicación actualmente cuenta con una serie de limitantes debido a las tecnologías con las cuales fue diseñada, estas limitaciones son:

1. Es una aplicación de escritorio, por lo que solo se puede acceder a ella en lugares donde haya sido desplegada.
2. No permite el acceso de varios usuarios de forma simultánea en la herramienta.
3. Solo permite la ejercitación de contenidos en juegos de estilo tablero.

Teniendo en cuenta lo anteriormente descrito se identifica el siguiente **problema de investigación:**

¿Cómo facilitar la ejercitación y evaluación de contenidos en la disciplina Técnicas de Programación de Computadoras en la UCI?

El **objeto de estudio** de la investigación lo constituye el desarrollo de entornos tridimensionales para la ejercitación de contenidos a través de juegos didácticos y el **campo de acción** se enmarca en los entornos tridimensionales para juegos didácticos en aplicaciones web.

Se traza como **objetivo general** de la investigación desarrollar un sistema que permita la ejercitación de contenidos a través de juegos didácticos tridimensionales en la disciplina Técnicas de Programación de Computadoras de la UCI.

Para el desarrollo de la investigación se proponen los siguientes **objetivos específicos**:

1. Construir el marco teórico de la investigación relacionado con el desarrollo de sistemas que incluyan juegos didácticos tridimensionales para la ejercitación de contenidos.
2. Diseñar una arquitectura que facilite el uso de juegos didácticos tridimensionales para la ejercitación de contenidos.
3. Verificar el funcionamiento del sistema desarrollado.

Para dar respuesta a los objetivos es necesario realizar las siguientes **tareas de investigación**:

1. Caracterización del estado del arte acerca de los sistemas que incluyan juegos didácticos tridimensionales para la ejercitación de contenidos.
2. Identificación de los recursos informáticos a desarrollar para la implementación del sistema.
3. Definición de la arquitectura del sistema.
4. Desarrollo del sistema para la ejercitación de contenidos a través de juegos didácticos tridimensionales.
5. Realización de las pruebas correspondientes al *software*.

Para el desarrollo de la investigación se utilizaron los siguientes **Métodos Científicos**:

### **Métodos Teóricos**

- *Histórico-Lógico*: con el objetivo de realizar un estudio acerca de los entornos

tridimensionales, así como su utilización para la implementación, diseño, creación y uso de juegos didácticos, haciendo énfasis en la importancia y las ventajas enfocadas al proceso de enseñanza aprendizaje.

- *Analítico-Sintético*: para analizar en cada una de sus partes los diferentes contenidos que están relacionados con entornos tridimensionales, juegos didácticos y juegos serios, para definir la relación que existe entre cada uno y permitir una fácil comprensión de sus conceptos.

## **Métodos Empíricos**

- *Encuesta*: para la realización de encuentros con profesores de Programación de la Facultad 1.

## **Justificación de la Investigación**

1. Realizar un sistema que permita la ejercitación de contenidos a través de juegos didácticos.
2. Posibilitar el acceso simultáneo de los usuarios al sistema.
3. Brindar entornos tridimensionales para mayor satisfacción del usuario.
4. Garantizar a los usuarios la comprensión de la lógica del juego de manera visual.

## **Estructura de los capítulos**

### **Capítulo 1: Juegos en entornos tridimensionales aplicados a la Educación.**

Se tratan conceptos y definiciones fundamentales para la comprensión de la investigación, además de realizar un estudio de homólogos en sistemas que utilicen la ejercitación de contenidos en entornos tridimensionales. Se analizan las herramientas y tecnologías de la Web más utilizadas en la actualidad con sus características fundamentales, así como lenguajes y metodologías escogidos para la realización de la investigación.

### **Capítulo 2: Propuesta e Implementación de la herramienta “Sistema para la Ejercitación de Contenidos” (SISTEC).**

Caracterización del sistema y descripción de los procesos para la solución, teniendo en cuenta el modelado y representación del negocio. Se realiza una propuesta de la aplicación a desarrollar y se comienza con la implementación de la misma, dejando plasmada una solución



que facilitará la ejercitación de contenidos garantizando de manera visual la lógica del juego.

### **Capítulo 3: Pruebas a la herramienta “Sistema de Ejercitación de Contenidos” (SISTEC).**

Se realizan las pruebas al sistema garantizando un correcto funcionamiento del mismo y permitiendo la ejercitación de contenidos. Además de ofrecer las diferentes estrategias para su realización y los resultados arrojados luego de la aplicación de las mismas.

## Capítulo I: Juegos en entornos tridimensionales aplicados a la Educación.

### Introducción

En el presente capítulo se muestran los conceptos y definiciones más importantes a tratar en la investigación. Se destaca la importancia de los mundos virtuales en la educación, así como se analizan diferentes sistemas utilizados para la ejercitación de contenidos en entornos tridimensionales, aplicados a la Educación Superior. Además se definen las tecnologías a utilizar para el desarrollo de aplicaciones web que permiten crear entornos multiusuarios y juegos didácticos.

### 1.1 Juegos serios

Los juegos serios son juegos diseñados para un propósito, más que para la diversión. Un juego serio puede ser una simulación con la apariencia de un juego, pero está relacionado con acontecimientos o procesos que nada tienen que ver con los mismos. Estos son creados para modelar cualquier situación y pueden ser aplicados a la educación, la salud, en industrias y actualmente son muy utilizados para la capacitación ya sea de profesionales como de principiantes en cualquier rama profesional. (5)

Los juegos serios están dirigidos a una gran variedad de público, desde estudiantes de educación primaria y secundaria hasta profesionales y consumidores. Los mismos pueden ser de cualquier tipo o género, usar cualquier tecnología y estar desarrollados para diversas plataformas. Pueden ser considerados como un tipo de entretenimiento educativo, aunque gran parte de la comunidad los ve como una distracción o como simple ocio, pero todo está en las características del juego y el objetivo con el que fue diseñado. (1)

Los juegos serios en la actualidad son muy utilizados como apoyo al proceso de aprendizaje, contribuyendo en cualquier enseñanza y permitiendo a los estudiantes incrementar su agilidad mental. Los principios fundamentales que caracterizan los juegos son: de participación activa, de dinamismo, de entretenimiento, de interpretación de roles, de obtención de resultados concretos y de competencia. Teniendo en cuenta su utilización en la educación, históricamente se han obtenido resultados en los que se destacan:

- Elevar el estudio individual de los estudiantes.
- Mejorar los resultados docentes.

- Lograr una interacción cooperativa y competitiva entre un colectivo de estudiantes.
- Motivar al estudiantado.
- Encontrar nuevas formas de evaluación. (6)

Teniendo en cuenta sus beneficios en la enseñanza, actualmente se crean alternativas para desarrollar juegos en los que el estudiante sea capaz de ejercitar conocimientos en espacios en los que pueda compartir con otros usuarios, espacios nutridos en conocimientos y en entornos que les permitan relacionar el juego con la realidad. Por esa razón resulta de gran ayuda la integración de los mundos virtuales en la educación para la ejercitación de contenidos.

## 1.2 Mundos virtuales

Los mundos virtuales consisten en entornos tridimensionales en los que el hombre interactúa con la máquina en un entorno artificial semejante a la vida real. Estos se parecen a la realidad en términos de condición y en ocasiones también reciben el nombre de metaversos<sup>3</sup>, destacándose con mayor éxito los espacios multiusuarios. Estos mundos virtuales ofrecen la oportunidad de intercambiar de un modo que conlleva a una sensación de presencia y que otros medios audiovisuales no ofrecen. Los mismos pueden ser muy grandes en cuanto a la cantidad de usuarios que los utilizan y van aumentando su popularidad ya que emplean muchos de los elementos de la Web como forma de divulgación. (7)

Recién comenzado el siglo XXI los mundos virtuales se hacen conocidos en todo el ámbito mundial, la razón de su éxito fue la aplicación de la realidad virtual en comunidades, juegos y *software* más simples. Las realidades virtuales presentan una construcción geométrica 3D la cual permite que los objetos parezcan reales. Esta engloba cuatro aspectos fundamentales:

- **Capacidad Sintética:** Hace referencia a la capacidad que tiene el computador de representar imágenes lo más parecidas a la realidad. Se consideran imágenes sintéticas aquellas que son generadas en tiempo real de acuerdo con la posición que ocupe un usuario dentro del mundo virtual.
- **Interactividad:** Capacidad que tiene el computador de que el usuario interactúe con el sistema, en caso de que este no pueda tener el control del mismo, no sería necesaria la sintetización de imágenes. Existen dos tipos de interacción, la dinámica y la de navegación, en la primera el usuario puede desplazarse, activar mecanismos, tomar

---

<sup>3</sup> Entorno tridimensional o virtual en el cual las personas pueden relacionarse entre sí.

objetos o cualquier acción que desee y en la segunda no es más que un simple espectador que solo puede cambiar su punto de vista.

- **Tridimensionalidad:** Es una característica fundamental para que un entorno sea considerado virtual, en la cual las proyecciones de un objeto se pueden visualizar en tres dimensiones alto, ancho y profundidad y sus movimientos laterales de arriba hacia abajo son completados cuando se les proporciona la última. Esta característica es generada por el efecto conocido por perspectiva lineal, haciendo que los objetos varíen de tamaño en función de la distancia del observador.
- **Ilusión de realidad:** Esta característica es fundamental para que lo que se conoce como realidad virtual, cumpla con la condición de que se parezca a lo real. Que el usuario pueda considerar que el mundo parezca real depende tanto de factores físicos como psicológicos. Los factores físicos tienen relación con el mundo virtual (proyecciones visuales, sonoras, táctiles) y los factores psicológicos con la interacción y la influencia que se le prometa al usuario en el mundo. (8)

La integración de la realidad virtual para la creación de estos entornos, ha permitido que con gran explotación de las tecnologías se continúen desarrollando espacios que permitan a la mayoría de los usuarios en la red, la plena comunicación y la interrelación entre ellos. Por esa razón en la Web se encuentran cada vez más personas interesadas en el acceso a estos sistemas.

Los mundos virtuales cuentan con características principales que permiten una buena interacción de los usuarios en el mismo, y hace de cada uno un espacio más ameno a la hora de estos relacionarse. Estos son espacios compartidos en los que se permiten a muchos usuarios a la vez; la inmediatez resulta una de sus características más importantes ya que la relación usuario-mundo tiene lugar en tiempo real. Permiten a los usuarios modificar, desarrollar, construir, o enviar contenido personalizado mediante la interactividad existente en el espacio, además la persistencia del mundo es independiente de si son los usuarios individuales los que están conectados y mediante la socialización permite la formación de comunidades. (9)

## 1.2.1 Mundos virtuales en la Educación Superior

La educación no ha quedado exenta de estos adelantos y ha tomado los mundos virtuales como alternativa para el desarrollo y ejercitación de conocimientos en todas sus esferas,

existiendo actualmente sistemas que permiten relacionar los mundos virtuales con la enseñanza.

Los mundos virtuales pueden ser utilizados para crear espacios de aprendizaje muy eficaces. Tanto así que en la actualidad son muy generalizados y pueden aplicarse a cualquier disciplina. Se pueden crear versiones para cualquier asignatura o ámbito de estudio, además de que los objetos pueden ser tan realistas y detallados como se desee. (1)

El uso de nuevas tecnologías en los procesos de formación perfecciona la calidad del aprendizaje, permitiendo que los mundos virtuales se conviertan en un punto de interés por parte de los docentes e instituciones de educación.

Estos mundos generan nuevas formas de relación entre el estudiante y el docente, de manera que el docente deja de ser la única fuente de conocimiento para convertirse en un experto colaborando con el desarrollo de la actividad de aprendizaje. Esto va directamente relacionado con modelos antiguos en los que los estudiantes permanecían sentados presenciando una clase, ahora los mundos virtuales facilitan la adaptación de los modelos de formación con la aplicación de las tecnologías. Cuando un estudiante se sumerge en un entorno 3D, este está codificando cognitivamente<sup>4</sup> el sonido, las señales y las relaciones dentro del entorno como su conducta participativa; el estudiante está emocionalmente implicado y se comporta como en realidad lo haría. Cuando esto sucede permite que este codifique de manera más eficaz el aprendizaje para el futuro, a la vez que se le suministran las claves necesarias para aplicar las experiencias obtenidas en un ambiente 3D. Es lo que se siente como "learning by doing" (aprende haciendo). (8)

En la enseñanza el uso de las tecnologías ha creado más interés en los estudiantes por las prácticas del conocimiento y aprendizaje de las materias. Estos entornos permiten mayor interacción entre estudiantes y profesores con una intensa colaboración entre los mismos, favoreciendo la aparición de grupos de trabajo y de discusión, además de que se incorporan simuladores como una nueva herramienta de aprendizaje.

La adquisición y desarrollo de nuevas competencias por parte de los estudiantes a través de su participación en laboratorios virtuales de investigación, se hace cada vez mayor debido a la

---

<sup>4</sup> *Lo cognoscitivo es aquello que está relacionado al conocimiento. Éste a su vez es el cúmulo de información que se dispone gracias a un proceso de aprendizaje o experiencias.*

motivación que tienen los mismos por la plataforma. Estos espacios cuentan con la facilidad de disponer de fuentes de retroalimentación de la información. (7)

### **1.3 Sistemas de ejercitación de contenidos**

Los sistemas de ejercitación de contenidos son programas que contribuyen a reforzar la aplicación y retroalimentación de contenidos en los procesos instructivos. (10)

Para la ejercitación en estos sistemas, los estudiantes deben tener un previo conocimiento de conceptos y definiciones asociados con la materia de la que estarán ejercitando. Un buen sistema para la ejercitación de contenidos es aquel que cuenta con una buena cantidad de ejercicios, variedad en los formatos en los que se presenta y la retroinformación que oriente de forma indirecta a los alumnos.

Estos sistemas son de gran beneficio en escuelas y centros en los que se requiera de un estudio sistemático de una materia o contenido en específico. Sus ventajas se basan en que el sistema tenga suficientes ejercicios y bien detallados, de forma que oriente a los estudiantes y pueda lograr que llegue a una buena comprensión de la problemática. Si el estudiante no tiene buenos resultados en un aspecto determinado se podrían proponer más ejercicios en el área en que muestra mayor dificultad, brindando una mayor información para que pueda llegar a la solución. Estos programas ofrecen la posibilidad de reprocesar respuestas, dando diferentes niveles de ayuda como son pistas y criterios y cuando ya no sea posible se ofrece una solución guiada, pero nunca una respuesta directa. (11)

### **1.4 Juegos serios en la Educación Superior**

Son muchos los mundos virtuales que permiten que sus usuarios ejerciten contenidos sobre materias relacionadas con la enseñanza en la que se desempeñan. A continuación se presentan una serie de juegos de ejercitación de contenidos que están diseñados en entornos tridimensionales. Todos requieren de autenticación y/o creación de un avatar o perfil de usuario para poder interactuar en el espacio.

#### **Juegos en el mundo**

##### **✓ Dental Implant Training Solution (Solución de Entrenamiento de Implante Dental)**

Desarrollado para El Colegio Médico de Georgia, fue creado para enseñar a estudiantes y profesionales de la odontología, las mejores prácticas para evaluar al paciente siguiendo el

protocolo de diagnóstico. El juego permite practicar procedimientos de implantes dentales de forma virtual. (12)

✓ **INNOVU “El juego del emprendedor”**

Permite desarrollar los conocimientos de todo aquel que lo utilice y orientarlos a la toma de decisiones. Para ello cuenta con un simulador virtual en el cual se pueden crear productos innovadores y lanzarlos al mercado. Este simulador permite que no solo se diviertan creando y explorando mercados sino que también aprendan a cómo desarrollar negocios. (13)

✓ **Innov8 2.0**

Desarrollado para la asignatura “Simulación de Mercados Electrónicos” a los estudiantes de la facultad de Ciencias de la Administración, con el fin de permitirles ganar experiencia en el análisis, toma de decisiones en tiempo real y el trabajo en equipo para alcanzar objetivos en el negocio. Innov8 reproduce el escenario laboral en el cual se desempeñarán las empresas del futuro. (14)

✓ **Virtual Clinical Learning Lab for Health Care Training (Laboratorio Clínica Virtual para Aprendizaje y Capacitación de la Salud)**

Está destinado a estudiantes y profesionales de la salud. Consiste en un espacio simulado para la cirugía, y permite practicar las habilidades clínicas en casos de lesiones graves sufridas por accidentes y ataques terroristas. Este juego fue desarrollado para una universidad de Texas. (15)

✓ **Environmental Detectives (Detectives Medioambientales)**

Diseñado para estudiantes del área de cuidado ambiental. Orientado para que aprendan las habilidades básicas de la investigación científica, cómo los productos químicos son trasladados a través del medioambiente y cómo contrarrestar sus efectos. (16)

✓ **Second Life (Segunda Vida)**

Second Life, también conocido como SL es un programa que simula un mundo virtual en tercera dimensión, en el cual se le permite a cualquier persona por medio de un avatar (personaje virtual) crear su propia comunidad en la red. Es un entorno tridimensional en línea, simula la vida real; el acceso y la adquisición de la cuenta de usuario es gratuita aunque para tener casas, edificios, muebles y tierras, el usuario debe depositar un dinero real que luego se

convertirá en virtual para interactuar en el sistema. Second Life al ser un entorno en 3D permite crear ambientes de aprendizaje más amplios que las clases presenciales brindando la posibilidad de desarrollar simulaciones de hechos reales y clases laboratorios. (17)

Muchos de estos juegos no especifican las tecnologías con las cuales fueron desarrollados, siendo una gran desventaja ya que no se puede determinar si son herramientas libres o privadas. No dan a conocer qué mecanismos utilizan para la realización de la simulación u objetos en 3D además de que solo se pudo tener acceso a muy pocos de ellos teniendo en cuenta que no se encuentran disponibles en la red. Estos juegos a pesar de ser diseñados para la educación superior no pueden ser aplicados a la especialidad de la informática debido a que fueron desarrollados para materias específicas y totalmente diferentes con las que se imparten en la universidad.

A diferencia de los juegos mencionados anteriormente se pudieron identificar otros desarrollados en la propia universidad similar a lo que podría ser la propuesta de solución.

### ✓ **Juego de Dominó para jugadores Virtuales**

Juego desarrollado en la facultad 5 de la Universidad de las Ciencias Informáticas orientado en un juego de dominó en el cual pueden interactuar hasta cuatro jugadores. El juego fue realizado como una aplicación de escritorio, desarrollada en Visual Studio .net y con el lenguaje de programación C#. (18)

### ✓ **Software-Motivación-Programación (SMProg)**

Es una herramienta educativa que permite la ejercitación de contenidos de manera didáctica, en la cual el profesor tiene la posibilidad de diseñar los tableros de juegos en los desea que sus estudiantes ejerciten, así como poner a disposición una serie de temas o sistema de preguntas para favorecer la ejercitación. Es una aplicación de escritorio, realizada con el lenguaje de programación Java y desarrollada en NetBeans. (4)

Esta última aplicación a pesar de ser creada para la ejercitación de contenidos de la asignatura de Programación, no permite el acceso simultáneo de estudiantes a la aplicación, lo que imposibilita la ejercitación de las diferentes asignaturas desde cualquier parte de la universidad.



## 1.5 Tecnologías

### 1.5.1 Aplicaciones Web

Las aplicaciones web son soluciones informáticas que permiten interactuar con la información y a las cuales se puede acceder a través de una conexión a internet, desde cualquier lugar, sin necesidad de instalarlas previamente en una computadora. Para su utilización solo se necesita contar con un navegador dígase Mozilla, Internet Explorer, entre otros. Las aplicaciones web trabajan de manera dinámica y permiten interactuar con la información, consultar bases de datos y realizar transacciones. (19)

Para un mejor aprovechamiento, utilización del sistema y que se encuentre disponible en cualquier momento en toda la red universitaria, se decide desarrollar una aplicación web en la cual se permita la ejercitación de contenidos de manera didáctica en la asignatura de Programación. Teniendo en cuenta que toda la administración del sistema se realizará desde el lado del servidor y así la *PC* cliente solo debe contar con un navegador para acceder a la misma.

### 1.5.2 Servidor Web

#### ***Apache Tomcat (versión 7.0.27.0)***

Es un servidor web de código abierto para diversas plataformas (Linux, Windows, Unix, entre otras). Este está diseñado para que pueda funcionar en una amplia variedad de entornos y plataformas. Permite a los desarrolladores crear y comprobar fácilmente aplicaciones conectadas a internet.

Apache tiene gran relación con el servidor Tomcat ya que este es el servidor web más utilizado a la hora de trabajar con Java en entornos web; Tomcat es una implementación completamente funcional de los estándares de JSP y Servlets. También puede especificarse como el manejador de las peticiones de JSP y servlets recibidas por servidores web como el servidor Apache HTTP. Además de que Tomcat está integrado en la implementación de referencia Java 2 Enterprise Edition (J2EE). (20)

### 1.5.3 Lenguaje de programación

Un lenguaje de programación es un idioma diseñado para llevar a cabo procesos a través de una computadora. Permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis que se ponen a disposición del desarrollador para poder comunicarse con la máquina. (21)

## **Java**

Java es uno de los lenguajes utilizados para el desarrollo de aplicaciones web ya que está diseñado para soportar sistemas que serán ejecutados en los más variados entornos de red y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Entre las principales razones por las cuales se escoge como lenguaje de programación a Java es por la portabilidad de su plataforma y la propiedad para poder introducirse con otros programas. Este lenguaje de programación es muy fácil de incrustar con otros programas en forma de *applets*, donde un *applet* es un componente de Java que se ejecuta en el contexto de otros programas. Ejemplo de esto es la incrustación con navegadores web, sitios web los cuales trabajan con *applet* para representar algún contenido dentro del mismo. (22)

Para la implementación del sistema se utilizará como lenguaje de programación Java teniendo en cuenta que es un lenguaje orientado a objetos además de ser libre a partir de la liberación por la Sun MicroSystem bajo la licencia GPL. Es portable y resulta convincente para los propósitos de este proyecto teniendo en cuenta su utilización para el desarrollo de *applets*.

### **1.5.4 Lenguaje de marcas y estilos**

#### ***XML (versión 1.0)***

EXtensible Markup Language (Lenguaje de Marcado Extensible) es un lenguaje de marcas que cumple determinadas reglas y que define etiquetas que estructuran e identifican las partes de un documento para un dominio específico. Esta estructura brinda gran facilidad en el momento de la interpretación, como es el caso de obtener datos lo mismo de una base de datos, o como en el sistema a desarrollar; elementos de una arquitectura que será traducido a objetos en 3D con OpenGL. Es un estándar abierto independiente de la plataforma y tiene un amplio soporte extendido a un sin número de herramientas y desarrolladores. (23)

Durante el desarrollo del sistema los XML se utilizarán para guardar los escenarios en tres dimensiones que se deseen representar en la aplicación. Al ser cargado el XML permitirá a los usuarios ver la escena en la cual se ejercitarán los contenidos, y esta se encontrará dentro de la estructura del mismo (Ver [Anexo #3](#)).

## **Hiper Text Markup Language<sup>5</sup> (HTML) (versión 4)**

El lenguaje de marcado de hipertextos es con el que se definen las páginas web. Tratándose básicamente de un conjunto de etiquetas que se utilizan para definir el texto y otros elementos que conforman la página. (24)

El sistema contará con páginas HTML en las cuales se le mostrará la información al usuario acerca del sistema, así como la ventana OpenGL donde se encontrará la escena por la que podrá interactuar y ejercitar los contenidos.

## **Cascading Style Sheets<sup>6</sup> (CSS) (versión 2.0)**

Es un lenguaje de hojas de estilos en cascada creado para mejorar la interfaz gráfica de las aplicaciones web. La utilización de CSS hace posible separar la presentación del contenido. Para la creación de esta hoja de estilo solo se necesita tener un código HTML, para poder diseñar un estilo agradable para el cliente. (25)

La hoja de estilo aplicada a las páginas HTML permitirá mejorar las interfaces de usuarios con las que contará el sistema.

### **1.5.5 Lenguaje Modelado**

El lenguaje de modelado es importante para el diseño y posterior construcción del producto de *software*. Es un conjunto estandarizado de notaciones que incluye símbolos y las distintas formas de organizarlos y disponerlos lógicamente.

## **UML (versión 2.0)**

El Lenguaje de Modelado Unificado (UML) es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Este también es considerado como un lenguaje estándar para escribir planos de *software*. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de *software*. (26)

### **1.5.6 Herramienta CASE**

Las herramientas CASE (Computer Aided Software Engineering), sus siglas traducidas al español significan Ingeniería de Software Asistida por Computadora, estas incluyen métodos, técnicas, utilidades y documentación orientadas a la automatización del ciclo de vida del *software*. Son fundamentales en la disminución del tiempo de desarrollo del *software* y en el

---

<sup>5</sup> Lenguaje de Marcado de Hipertextos

<sup>6</sup> Hojas de Estilo Cascada

aumento de la productividad, dado que permiten enfatizar en el análisis y el diseño para minimizar el esfuerzo de codificación y prueba.

### ***Visual Paradigm (versión 8.0)***

Visual Paradigm proporciona un conjunto de ayudas para el desarrollo de sistemas informáticos, desde la planificación, el análisis y diseño orientado a objetos, construcción pruebas y despliegue. Facilita el ciclo de vida completo del desarrollo de *software* y es de licencia libre y comercial, además de que posee versiones multiplataforma. El uso de Visual Paradigm como herramienta de modelado favorecerá a la creación de artefactos indispensables para la conformación del expediente de proyecto del sistema; ayudando además al desarrollo de la aplicación, permitiendo aumentar la productividad en el desarrollo de *software*. (27)

Se utiliza UML como lenguaje de modelado y Visual Paradigm como herramienta case, teniendo en cuenta que una de las restricciones del Proyecto de Innovación Pedagógica (PIP) es que se trabaje con tecnologías libres y basadas en estándares abiertos.

### **1.5.7 Metodología de desarrollo**

Una metodología de desarrollo de *software* es una filosofía o marco de trabajo para estructurar, planificar y controlar el proceso de desarrollo de *software*. Esta especifica un conjunto de pasos o procedimientos que definen quién debe hacer qué, cuándo y cómo debe hacerlo. Su misión u objetivo principal es guiar a los desarrolladores de *software* en la obtención de un producto de calidad que cumpla con los requerimientos establecidos por el cliente, ajustándose a los recursos apropiados y a un costo razonable. (33)

### ***RUP (Proceso Unificado de Rational)***

Este Proceso Unificado es un marco de desarrollo de *software* que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema. Constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Entre sus características se pueden destacar la forma disciplinada de asignar tareas y responsabilidades, realiza un desarrollo iterativo de los procesos, la administración de requisitos y pretende implementar las mejores prácticas de Ingeniería de Software. (28)

De acuerdo con las características del PIP, el enfoque que se aplica para el proceso de desarrollo de *software* es el tradicional. Entre las metodologías de desarrollo tradicionales más utilizadas se encuentra RUP, la cual presenta un modelo de desarrollo ventajoso para las características del PIP ya que es evolutivo incremental. Para la creación del sistema se realiza una adaptación de esta metodología, de la cual solo se tomarán los artefactos más importantes adaptados al proyecto en creación, como es el caso de: diagramas de caso de uso, componentes, secuencia, despliegue y diagramas de clases del diseño.

## **1.5.8 Entorno de desarrollo integral (IDE)**

Un entorno de desarrollo integrado o IDE (Integrated Development Environment), por sus siglas en inglés, es un programa que incluye un conjunto de herramientas mediante las cuales los programadores pueden codificar (34). Entre las herramientas que componen un IDE se encuentran un editor de texto, un compilador, un intérprete, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y opcionalmente un sistema de control de versiones.

### ***NetBeans (versión 7.2)***

Es un entorno de desarrollo muy completo, profesional y agradable. Contiene muchas funcionalidades para distintos tipos de aplicaciones y para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. El mismo es capaz de soportar el desarrollo de todos los tipos de aplicaciones Java como por ejemplo web. Este cuenta con un conjunto de herramientas las cuales no son necesarias instalar previamente.

Para el desarrollo se utiliza NetBeans atendiendo a que es un producto libre y gratuito sin restricciones de uso, está escrito en Java y es una herramienta que permite a sus programadores escribir, compilar, depurar y ejecutar programas.

### **Librería OpenGL**

OpenGL es una biblioteca gráfica en 3D multiplataforma, ya que puede ser utilizada en diversos sistemas operativos como Unix, GNU/Linux, Windows e incluso en móviles. Es una de las librerías más utilizadas, integrada con muchos lenguajes de programación y goza de gran popularidad entre los desarrolladores de juegos. Además de la extraordinaria calidad con que visualiza los objetos, ha sido un pilar importante en el desarrollo de aplicaciones que simulen la

realidad, lo que hace que los juegos en tres dimensiones sean los preferidos por todos en la comunidad actual. (29)

El uso de la biblioteca OpenGL para el desarrollo del sistema es primordial ya que permitirán que se puedan crear objetos en tres dimensiones, esta al poder ser incluidas en muchos lenguajes de programación da la posibilidad de trabajarlas con Java.

## **1.5.9 Sistema Gestor de Base de Datos (SGBD)**

Un SGBD permite procesar, describir, administrar y recuperar los datos almacenados en un banco de datos. Estos sistemas brindan un conjunto de programas, procedimientos y lenguajes para consultar los datos y realizar acciones sobre ellos, garantizando además la seguridad de los mismos.

### ***PostgreSQL (versión 9.1.2)***

Es un sistema de gestión de bases de datos con su código fuente disponible libremente. Utiliza el modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema, lo que implica que un fallo en un proceso no afectará que el sistema continúe funcionando. Cuenta con una arquitectura confiable es multiplataforma y de fácil conectividad con el IDE de desarrollo. (30)

Se decide utilizar este gestor debido a que la base de datos Sistema de Preguntas está desarrollada en el mismo. Para la realización de consultas se utiliza Java Data Base Connectivity (JDBC), API que permite la conexión y la ejecución de operaciones sobre la base de datos.

## **Conclusiones Parciales**

- ✓ La aplicación de juegos serios en entornos tridimensionales en la enseñanza superior ha logrado incentivar y motivar a los estudiantes a realizar un estudio sistemático.
- ✓ Se descarta la utilización de sistemas analizados por no ajustarse a las características de la solución planteada.
- ✓ El uso de herramientas libres facilita la integración con módulos existentes y futuros en el Proyecto de Innovación Pedagógica “Herramienta educativa sobre Software libre para la asignatura de Programación II y IV”.

### Capítulo 2: Propuesta e Implementación de la herramienta “Sistema para la Ejercitación de Contenidos” (SISTEC).

#### Introducción

Como parte del Proyecto de Innovación Pedagógica (PIP) “**Herramienta educativa sobre Software libre para la asignatura de Programación II y IV**” se diseña e implementa una aplicación para la ejercitación de contenidos atendiendo a sus nuevas características. En el capítulo se presenta la propuesta de solución y la arquitectura por la cual se regirá la construcción de la aplicación. Se presenta además toda la etapa de análisis, diseño e implementación de la misma, así como los requisitos funcionales y no funcionales con los cuales debe cumplir el sistema.

#### 2.1 Solución

Se propone realizar una herramienta desarrollada como aplicación web que permita la ejercitación de contenidos de manera didáctica en la asignatura Técnicas de Programación de Computadoras en la Universidad de las Ciencias Informáticas (UCI). Esta contará con la integración de dos módulos principales para su correcto funcionamiento, los cuales son:

- ✓ **Herramienta de autor para el diseño de sistemas de preguntas para juegos didácticos (GesCont)**

El módulo contará con un sistema de preguntas diseñadas por el profesor atendiendo las asignaturas y los temas (para cada una de ellas) en la cual desee que se realice la sistematización de contenidos.

- ✓ **Herramienta de autor para el diseño de entornos tridimensionales.**

El módulo permitirá el diseño de escenarios tridimensionales y en este el profesor podrá crear las escenas en las cuales los estudiantes ejercitarán. La herramienta generará un archivo con la estructura de los escenarios creados para ser utilizados posteriormente.



Figura 1 Integración al Proyecto de Innovación Pedagógica

Para la ejercitación de contenidos el estudiante tendrá la posibilidad de seleccionar el tema y el escenario en el que desee ejercitar. El sistema obtendrá las escenas del módulo *Herramienta de autor para el diseño de entornos tridimensionales* las cuales se encontrarán en una dirección física del servidor de aplicaciones y en esta se podrá realizar el desplazamiento permitiéndole al estudiante interactuar con los objetos en la misma. Esta escena contendrá objetos de tipo pregunta; que identificarán cuando realizar una llamada a la base de datos del módulo *GesCont*, mostrando de manera ascendente según el nivel de complejidad, las preguntas asociadas al tema seleccionado. Luego de concluir la ejercitación el sistema actualizará la evaluación obtenida por el estudiante en el tema que haya sistematizado.

## 2.2 Requerimientos del Sistema

Los requerimientos del sistema (o del *software*) están determinados por las necesidades del usuario, lo que él describe como lo que debe cumplir el sistema que solicita, las características que lo deben identificar y sus funcionalidades. A continuación se identifican las técnicas de captura de requisitos y requisitos funcionales (RF) y no funcionales (RNF) con los que debe cumplir el sistema.

### 2.2.1 Técnicas de Captura de Requisitos

- ✓ **Tormenta de Ideas:** Se realizaron reuniones con un grupo de personas involucradas en



el tema, donde se generaron diferentes vistas del problema y sus posibles soluciones. Estas reuniones contaron con la participación de desarrolladores y analistas de sistemas con los cuales se realizará una integración futura, definiendo las funcionalidades de cada uno.

- ✓ **Sistemas Anteriores: SMProg (Software-Motivación-Programación)** es un sistema creado y utilizado en la universidad para la ejercitación de contenidos; atendiendo a este se realiza un análisis, identificando los requerimientos con los que cuenta el sistema, agregando así novedosas funcionalidades a la nueva aplicación.

Luego de aplicar las diferentes técnicas de captura de requisitos para el sistema de ejercitación de contenidos se obtuvieron los siguientes resultados.

### 2.2.2 Requisitos Funcionales

Los requisitos funcionales de la aplicación fueron identificados a partir de las funcionalidades con las que se requiere que cuente el sistema, partiendo de las necesidades de realizar una buena ejercitación de contenidos en la asignatura de Programación de Computadoras y a su vez para otras asignaturas enmarcadas en la carrera. A continuación se muestran los requisitos funcionales, los cuales fueron organizados a partir de Casos de Uso (CU) para una mayor comprensión.

El CU\_Jugar es el encargado de realizar la mayor cantidad de funcionalidades del sistema; para poder jugar es indispensable cumplir con el CU\_Cumplir\_reglas\_de\_juego donde se garantiza el desplazamiento y funcionalidades agregadas al mismo.

Tabla 1 Tabla de Requisitos Funcionales del sistema

No_RF	Nombre	Descripción
CUS_1	Jugar	Permite al estudiante realizar la ejercitación de contenidos en la aplicación a través de un juego.
RF_1	Seleccionar Asignatura	Permite seleccionar la asignatura disponible para la ejercitación de contenidos.
RF_2	Seleccionar Tema	Permite seleccionar el tema que el usuario desee.
RF_3	Seleccionar Escena	Se selecciona el escenario 3D en el que se desee

		ejercitar.
CUS_2	Cumplir reglas de juego	Permitirá verificar que se cumplan las reglas del juego; en caso de caminar por la escena, responder preguntas, mostrar la pregunta adecuada, en caso de ser didáctica o no.
RF_4	Avanzar por la escena	Permite al usuario desplazarse por la escena seleccionada.
RF_5	Mostrar preguntas	Debe mostrar las preguntas del tema seleccionado teniendo en cuenta si hay un obstáculo de tipo pregunta.
RF_6	Evaluar preguntas	El sistema permitirá evaluar la respuesta del estudiante, comparando su respuesta con la proporcionada por el profesor.
RF_7	Responder pregunta	El usuario debe responder la pregunta que muestra el sistema.
RF_8	Avanza por la escena si responde correctamente.	Solo si responde correctamente la pregunta presentada, el usuario continúa avanzando por la escena.
RF_9	Indicar Vía de estudio.	Muestra la vía de estudio para continuar la preparación en el tema.
RF_10	Actualizar evaluación de estudiante.	Actualiza la evaluación del estudiante luego de haber realizado la ejercitación de contenidos.
RF_11	Cargar escena.	El sistema carga la escena 3D seleccionada por el usuario.

### 2.2.3 Requisitos No Funcionales (RNF)

Los RNF especifican las propiedades con las que debe cumplir el sistema; como restricciones del entorno o de la implementación, rendimiento, dependencia de la plataforma, fiabilidad. Tiene que ver con las características del sistema que incluye también interfaces de usuario.

#### **Apariencia o interfaz externa**

El sistema debe contar con una interfaz sencilla que permita al usuario interactuar de forma cómoda y que le facilite y agilice el trabajo. La interfaz será de fácil comprensión y mostrará una ayuda para el usuario.

#### **Portabilidad**

La aplicación podrá ser ejecutada en cualquier sistema operativo, por su característica de multiplataforma, permitiendo una fácil migración haciendo uso de estándares y tecnologías de código abierto.

#### **Seguridad**

Los usuarios que accedan a la aplicación deberán estar autenticados y pertenecer al dominio [uci.cu](http://uci.cu). Una vez autenticados tendrán acceso a las funcionalidades que ofrece el sistema.

#### **Disponibilidad**

El sistema se desplegará en la universidad, encontrándose disponible las 24 horas de los 7 días de la semana. Garantizará un acceso de forma rápida y fácil para los estudiantes.

#### **Software**

*PC Cliente:* Para poder utilizar la aplicación debe disponer:

- Navegador Web estándar con interpretación de *Java Script* y *CCS*.

*PC Servidor:* Se debe disponer para la instalación de la aplicación:

- Servidor Web Apache Tomcat 7.0.27.0 o superior.
- PostgreSQL 9.1.2 o superior.

#### **Hardware**

*PC Servidor:* Se debe disponer para poner a disposición la aplicación:

- Microprocesador de 2.4 GHz o superior.
- Memoria RAM de 1GB o superior.
- Capacidad de Disco duro de 80 GB o superior.

- Tarjeta de red de 100 Mbps o superior.

PC Cliente: Solo debe contar con un navegador web para tener acceso a la aplicación.

### Especificación de requisitos

La especificación de requisitos de *software* permite describir el funcionamiento completo que realiza en el sistema una determinada funcionalidad. A continuación se muestra la especificación de requisitos de algunas de las funcionalidades con las que cuenta la aplicación.

Para ver descripción de las diferentes funcionalidades del sistema ir a [Anexo 1](#).

Tabla 2 Especificación del requisito Seleccionar asignatura

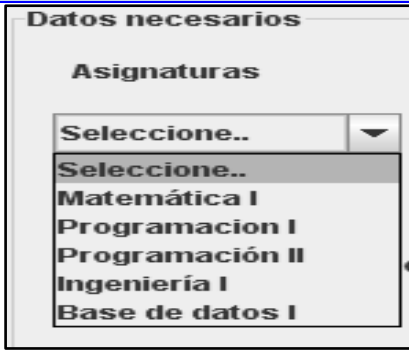
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_1]	Seleccionar Asignatura	El estudiante debe seleccionar el nombre de la asignatura de la cual desea ejercitar los contenidos. Para la selección debe desplegar el componente con la lista de asignaturas con las que cuenta la base de datos "Sistema de Preguntas".	Media	Alta
<b>Prototipo</b>				
 <p>Prototipo Seleccionar Asignatura.</p>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
<b>Observaciones</b>	1. El sistema siempre contará con asignaturas disponibles en su base de datos.			

Tabla 3 Especificación de requisito Seleccionar tema

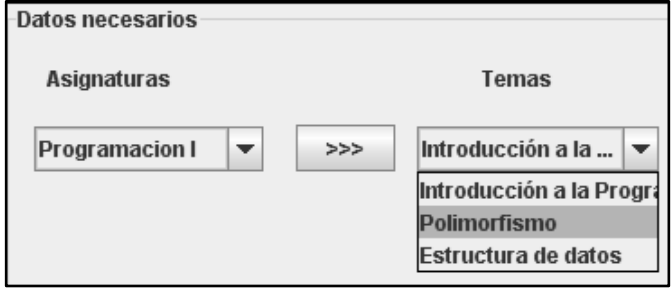
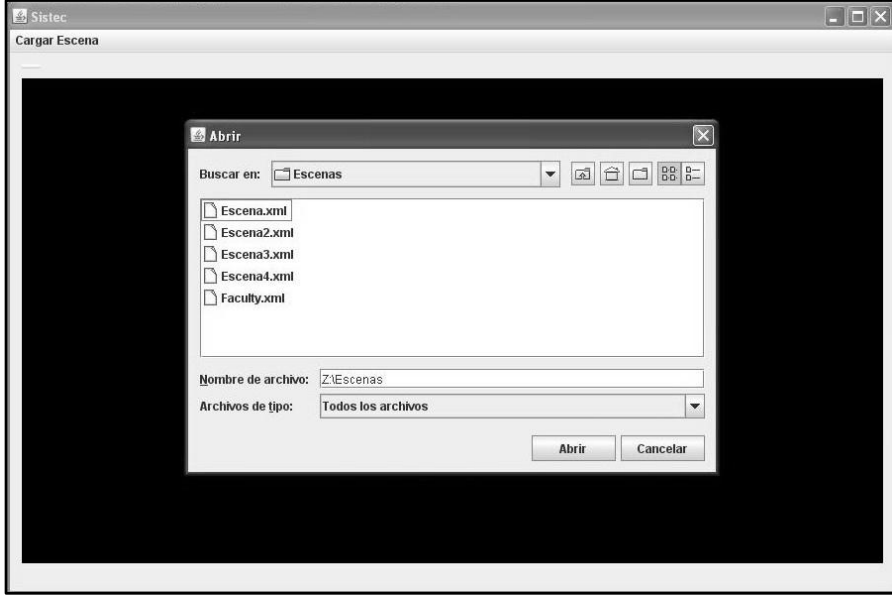
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_2]	Seleccionar Tema	Se muestra un listado de temas correspondientes a la asignatura que fue seleccionada. El estudiante debe seleccionar el tema que quiere ejercitar y automáticamente se carga el sistema de preguntas con las que cuenta el mismo.	Media	Alta
<b>Prototipo</b>				
 <p><b>Prototipo Seleccionar Tema.</b></p>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
<b>Observaciones</b>		<ol style="list-style-type: none"> <li>1. El sistema siempre contará con asignaturas disponibles en su base de datos.</li> <li>2. Cada asignatura contará con un grupo de preguntas por cada tema.</li> </ol>		

Tabla 4 Especificación de requisito Seleccionar escena

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_3]	Seleccionar Escena	Se muestra una ventana con la opción Cargar escena. Al presionar la opción el sistema permite la selección de un escenario a través de un directorio, en una dirección física de la PC.	Media	Alta
<b>Prototipo</b>				

 <p style="text-align: center;"><b>Prototipo Seleccionar Escena.</b></p>		
<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>
<b>Observaciones</b>	<ol style="list-style-type: none"> <li>El sistema siempre contará con escenas disponibles para la ejercitación.</li> </ol>	

**Tabla 5 Especificación de requisitos Avanzar por la escena**

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_4]	Avanzar por la escena	Durante la ejercitación el estudiante tendrá la posibilidad de desplazarse por la escena seleccionada, interactuando con cada uno de los objetos que se encuentran en la misma. Para su desplazamiento debe presionar las teclas a la derecha (flechas arriba, abajo, derecha, izquierda o PageDown y PageUp).	Media	Alta
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>Para realizar una correcta ejercitación el estudiante debe interactuar con la escena y avanzar por ella.</li> </ol>		

## 2.3 Modelado del sistema

### 2.3.1 Actores del sistema

A continuación se realiza una breve descripción del actor que interviene en el sistema; haciendo referencia a las acciones que puede realizar en el mismo.

Tabla 6 Descripción de Actores del sistema

Actor	Descripción
Estudiante	Puede realizar la ejercitación de contenidos seleccionando un tema y una escena en la aplicación; permitiéndosele la navegación por la misma e interactuar con los objetos que la conforman.

### 2.3.2 Diagrama de Caso de Uso del Sistema

El diagrama de Caso de Uso del Sistema (CUS) que se representa a continuación permite determinar quiénes son los actores que intervienen en la aplicación así como las acciones que pueden realizar en la misma.

**SISTEC** permitirá el acceso de usuarios de tipo estudiante y en él se podrán realizar la ejercitación de contenidos a través de juegos didácticos. La acción principal de los usuarios en el sistema será jugar y para ello deberán cumplir una serie de reglas orientadas al funcionamiento del mismo y la correcta ejercitación.

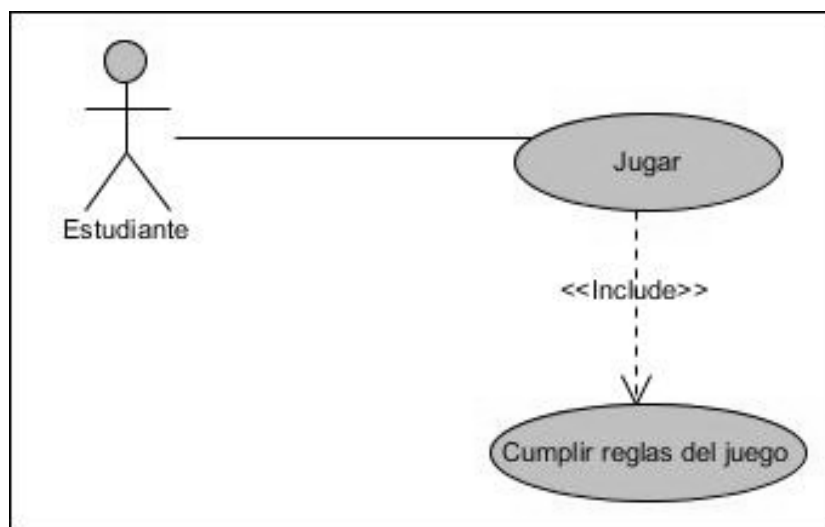


Figura 2 Diagrama de Caso de Uso (CUS)

### 2.3.3 Descripción de caso de usos del sistema

#### 1. CUS Cumplir reglas del juego

Tabla 7 Descripción del CUS Cumplir reglas del juego

<b>Caso de Uso:</b>	Cumplir reglas del juego	
<b>Actores:</b>	Estudiantes	
<b>Resumen:</b>	El caso de uso inicia luego que el estudiante comienza a jugar. Camina por la escena respondiendo las preguntas correspondientes. Si la pregunta es respondida correctamente el estudiante avanza y continúa jugando.	
<b>Precondiciones:</b>	El usuario debe estar autenticado en el sistema.	
<b>Referencias</b>	RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11	
<b>Prioridad</b>	Crítica	
<b>Flujo Normal de Eventos</b>		
<b>Sección “Cumplir reglas de juego”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Avanzar por la escena.	2. Muestra una pregunta didáctica.	
3. Responde la pregunta.	4. Evalúa Pregunta.	
5. Avanza por la escena.	6. Termina el juego.	
	7. Actualizar evaluaciones generales.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>	
	2.1 Si el objeto en la escena es de tipo pregunta.	
3.1 Si la pregunta es correcta.	4.1 Actualiza la evaluación de la pregunta.	
3.2 Si la pregunta es incorrecta.	4.2 Muestra otra pregunta, hasta que sea respondida correctamente.	
5.1 Avanza por la escena respondiendo las preguntas.	6.1 El sistema continúa realizando las operaciones <b>2.1</b> , <b>4.1</b> , <b>4.2</b> hasta que sean respondidas todas las preguntas del tema.	
	7.1 Si concluye con calificación satisfactoria. Muestra el mensaje “Has obtenido resultados satisfactorios. Felicidades has ganado!!!!”.	
	7.2 Si concluye con calificaciones no muy buenas Muestra una vía de estudio en el siguiente mensaje “Debes estudiar un poco más, visita el entorno de aprendizaje <a href="http://eva.uci.cu">eva.uci.cu</a> ”.	
<b>Pos condiciones</b>	Al cumplir las Reglas correspondientes con el juego, se estará realizando una correcta	



	ejercitación de contenidos
--	----------------------------

## 2. Jugar

Tabla 8 Descripción del CUS Jugar

<b>Caso de Uso:</b>	Jugar
<b>Actores:</b>	Estudiantes
<b>Resumen:</b>	El caso de uso inicia cuando el estudiante accede a las opciones que le permiten Jugar. Selecciona el tema de la asignatura que desea ejercitar así como la escena con la que quiere interactuar. El sistema ejecuta las acciones correspondientes y carga el juego según la configuración especificada y permite la interacción del usuario en el mismo.
<b>Precondiciones:</b>	RF1, RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11
<b>Referencias</b>	
<b>Prioridad</b>	Crítica
<b>Flujo Normal de Eventos</b>	
<b>Sección “Jugar”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. Selecciona la opción “Jugar”	2. Muestra la interfaz de usuario con la lista de Asignaturas y Escenas que puede seleccionar para jugar.
3. Selecciona la “Asignatura”	4. Busca la asignatura seleccionada y muestra listado de temas de esa asignatura.
5. Selecciona “Tema”	6. Carga un sistema de preguntas del tema seleccionado.
7. Selecciona la “Escena”	8. Carga la “Escena” seleccionada.
9. Pulsa el botón “Aceptar”	10. Carga la configuración especificada, con el tema y la escena seleccionada por el usuario. 11. Muestra la escena escogida integrada con un sistema de preguntas correspondientes con el tema seleccionado.
12. Avanza por la escena.	13. Verifica objeto de la escena. 14. Mostrar Pregunta.
15. Responder Pregunta	16. Evalúa Pregunta
17. Ejercitar	18. Terminar juego
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
3.1 Si no selecciona una “Asignatura”	4.1 Muestra el mensaje “Debe seleccionar una Asignatura.”

5.1 Si no selecciona un "Tema"	6.1 Muestra el mensaje "Debe seleccionar un Tema"
7.1 Si no selecciona "Escena"	8.1 Muestra el Mensaje "Debe seleccionar una escena."
	13.1 Si el objeto es didáctico, muestra pregunta. 13.2 Si no es didáctico (Volver al paso 12)
15.1 Si la pregunta es correcta.	16.1 Actualiza evaluación de la pregunta.(Volver al paso 12)
15.2 Si la pregunta es incorrecta.	16.2 (Volver al paso 14, mostrando otra pregunta)
17.1 Continúa ejercitando (pasos del 10-14)	18.1 Si concluye con calificación satisfactoria. Muestra el mensaje "Has obtenido resultados satisfactorios. Felicidades has ganado!!!!!!".
	18.2 Si concluye con calificaciones no muy buenas Muestra una vía de estudio en el siguiente mensaje "Debes estudiar un poco más, visita el entorno de aprendizaje <a href="http://eva.uci.cu">eva.uci.cu</a> ".

## 2.4 Diagrama de clases

El diagrama de clases de diseño es un diagrama que describe gráficamente las especificaciones de las clases existentes en un producto de *software* así como las interfaces involucradas.

En el diagrama de clase siguiente se identifican las clases con las que cuenta el sistema y como se relacionan. Para una mejor visión del diagrama de clases de la herramienta **SISTEC**, este se divide en tres paquetes, el paquete vista, controlador y el modelo.

El paquete Vista cuenta con todas las vistas que tiene el sistema así como las relaciones con la clase controladora. El paquete controlador contiene las clases controladoras como es el caso de las Server Page, encargadas del manejo de las vistas y las clase Controladora del sistema: Jugar.

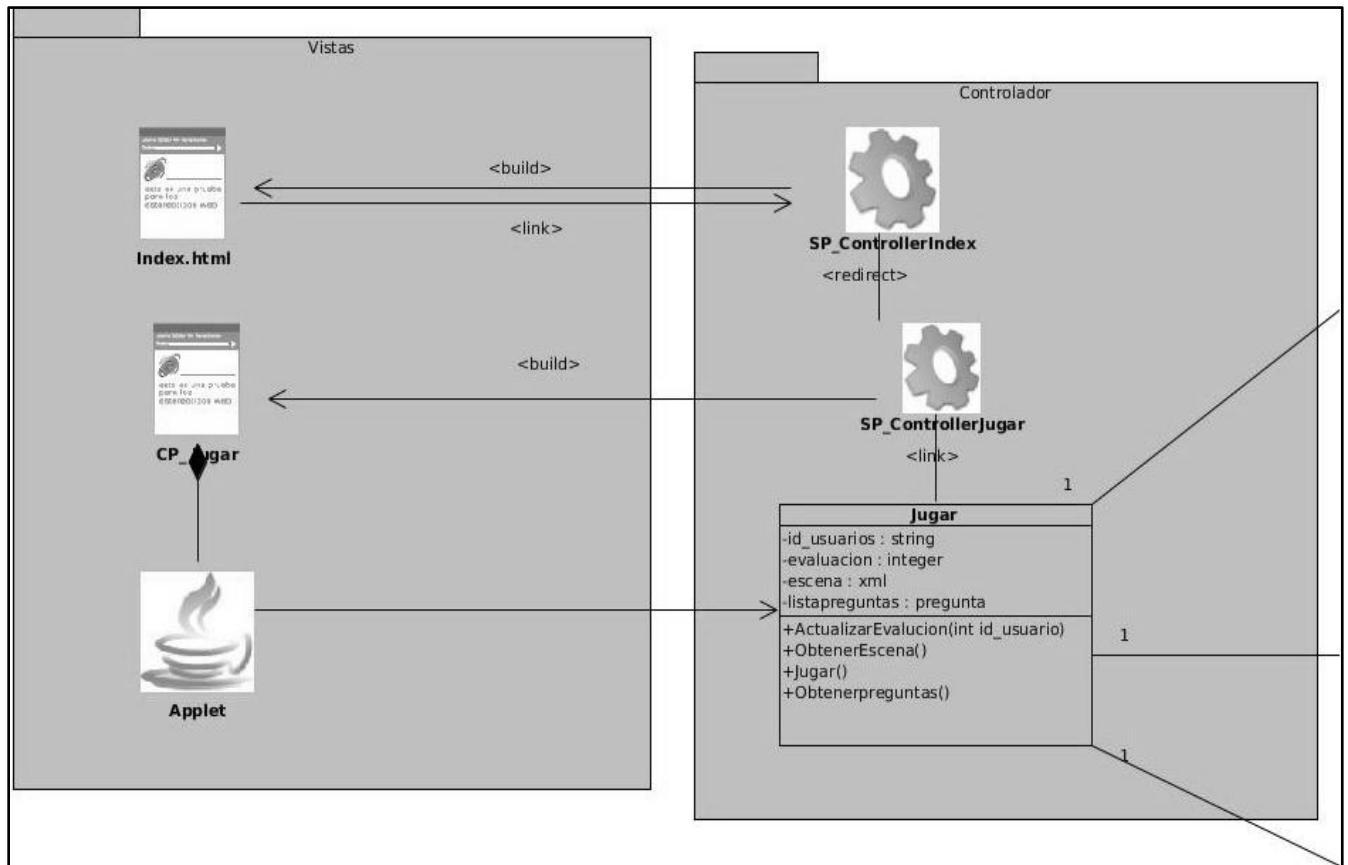


Figura 3 Diagrama de Clases del diseño (Vista-Controlador)

La figura siguiente muestra el modelo de datos del sistema, representado en el diagrama de clases. Las clases de este diagrama tienen relaciones con la clase Jugar la cual se encarga de toda las operaciones del sistema y obtiene los datos necesarios del modelo para su funcionamiento.

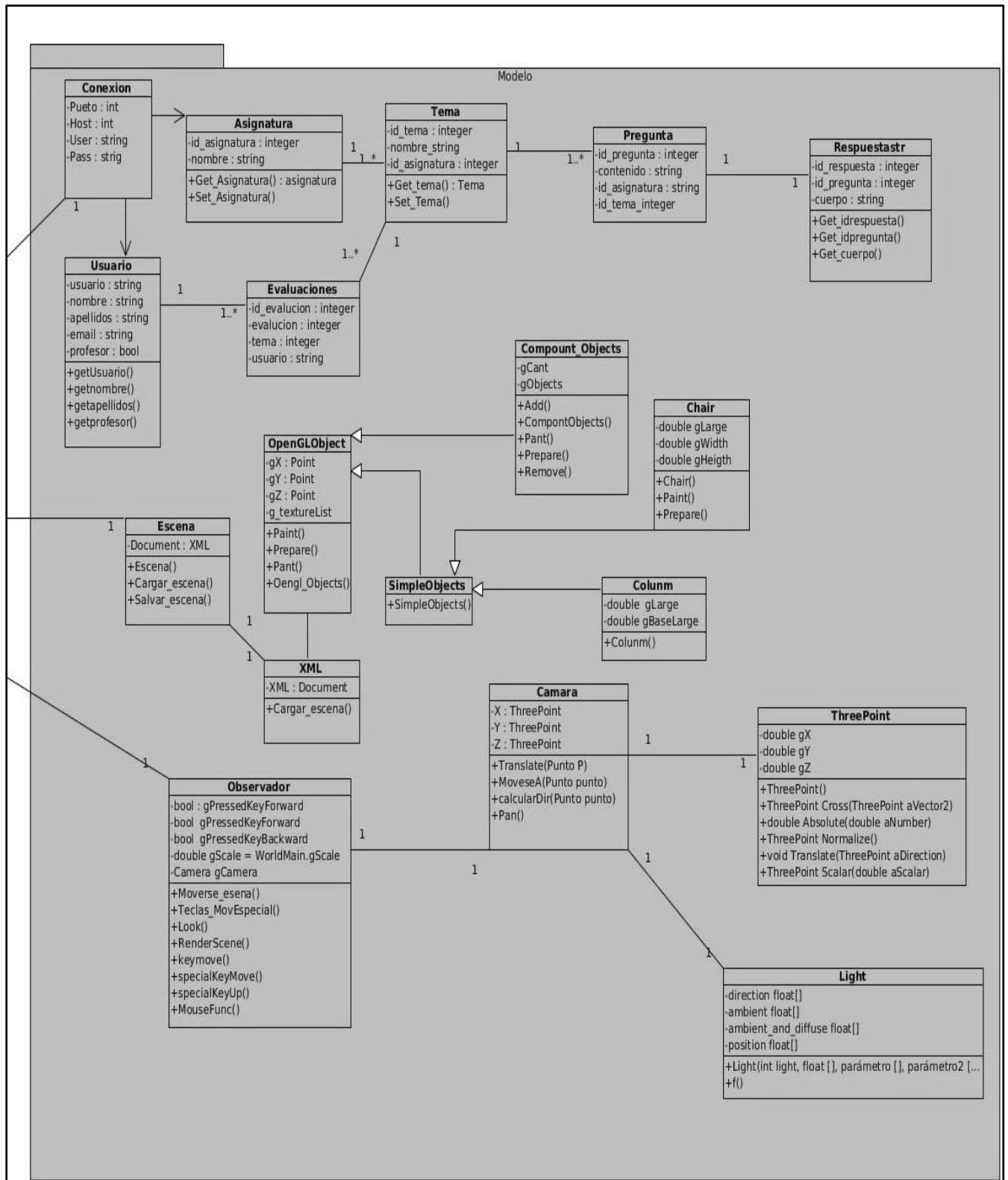


Figura 4 Diagrama de Clases del diseño (Modelo)

## 2.4.1 Descripción de Clases

### Clase Controladora

Tabla 9 Descripción de la clase Controladora: Controladora\_Jugar

<b>Nombre:</b> Controladora_Jugar	
<b>Tipo de clase:</b> Controladora	
<b>Atributos</b>	<b>Tipo</b>
Id_Usuario	String
evaluación	Integer
listapreguntas	Lista<Preguntas>
escena	Document
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
getId_Usuario	Retorna el id del usuario
get_evalucion	Retorna la evaluación obtenida
Obtener_Preguntas	Retorna el listado de preguntas
Jugar	Garantiza la ejercitación de contenidos
Obtener_escena	Retorna la escena en la que se ejercitará

Tabla 10 Descripción de la clase Entidad: Asignatura

<b>Nombre:</b> Asignatura	
<b>Tipo de clase:</b> Entidad persistente	
<b>Atributos</b>	<b>Tipo</b>
nombre	String
Id	Integer
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
get_Nombre	Retorna el nombre de la asignatura
get_ID	Retorna el identificador de la asignatura

Tabla 11 Descripción de la clase Entidad: Tema

<b>Nombre:</b> Tema	
<b>Tipo de clase:</b> Entidad persistente	
<b>Atributos</b>	<b>Tipo</b>

nombre	String
id	Integer
id_asignatura	Integer
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
get_Nombre	Retorna el nombre del tema.
get_ID	Retorna el identificador del tema.
get_IDAsignatura	Retorna el identificador de la asignatura a la que pertenece.

### 2.5 Diagramas de Interacción

Los diagramas de interacción muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esto implica modelar instancia de clases interfaces, componentes y nodos junto con los mensajes enviados entre ellos, a los que se les denomina flujo de control. Entre los diagramas de interacción que existen se encuentran los de Secuencia y los de Colaboración. (31)

Para la representación del flujo de control en el sistema se utilizan los diagramas de secuencia debido a que el número de mensajes es muy grande y se hace difícil su comprensión en un diagrama de colaboración.

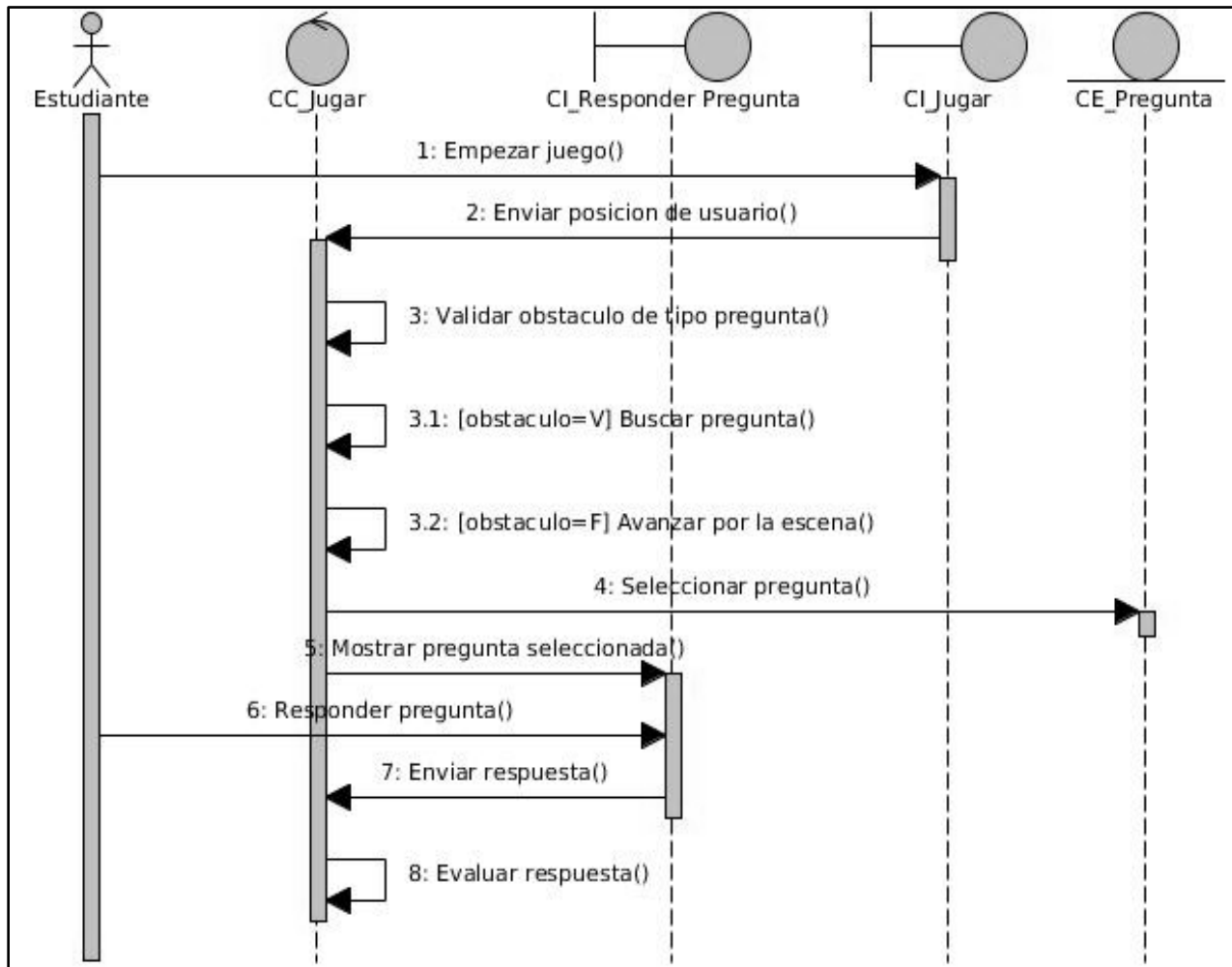


Figura 5 Diagrama de secuencia Cumplir reglas del juego

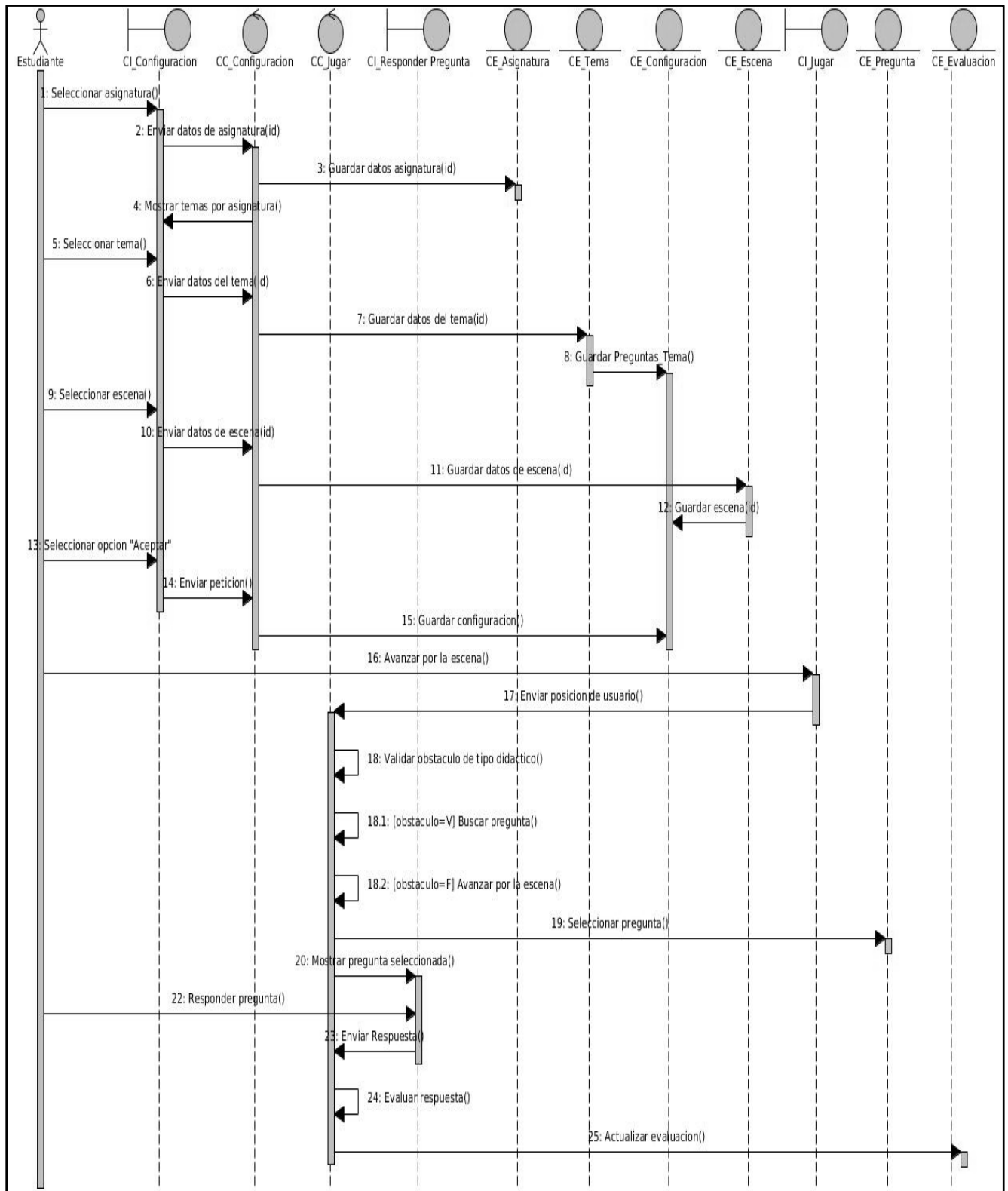


Figura 6 Diagrama de secuencia Jugar



### 2.6 Arquitectura de la solución

La arquitectura de la solución está basada en el patrón arquitectónico Modelo-Vista-Controlador (MVC), el cual se encargará de separar los datos de la aplicación, la interfaz del usuario y la lógica del control en tres componentes distintos. Este patrón es muy frecuente en la construcción de aplicaciones web y sus tres componentes se definen en:

- ✓ **Modelo:** Representa la información con la que trabaja la aplicación, la cual es obtenida de la base de datos con la que esté trabajando el sistema. En la solución propuesta se evidencia a través de las clases entidades que contendrán la información con la que trabajará el sistema, como es el caso de Escena, la cual cuenta con toda la información necesaria para la representación de los escenarios en la aplicación, además de una clase *Conexión* que permitirá el acceso a la base de datos donde se encuentra parte de la información necesaria para la ejercitación dígame asignaturas, temas y preguntas. Así como la clase *XMLTools* que garantiza la ubicación de los escenarios en una dirección física del servidor y a través de ella se realizan las llamadas a esas escenas para previa representación en el sistema.
- ✓ **Vista:** Es la responsable de recibir datos del modelo y mostrarlos al usuario, en la propuesta de solución se evidencia a través de las páginas HTML con las cuales interactuará el estudiante, y la hoja de estilo que permitirá mejorar la apariencia de las interfaces. Esta capa contiene además un *applet* de Java que mostrará la ventana OpenGL que permitirá la configuración del juego mostrando la selección de la asignatura y el tema a ejercitar, así como la escena por la cual se desplazará el jugador para ejercitar el contenido.
- ✓ **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca peticiones al modelo y probablemente a las vistas. En la solución propuesta se evidencia a través de las clases encargadas del funcionamiento del sistema, como es el caso de la clase *Controladora\_Jugar* que tiene la responsabilidad de actualizar las evaluaciones del estudiante y las funcionalidades de mostrar y evaluar las preguntas; así como la clase *GLRender* encargada del desplazamiento por toda la escena.

En la siguiente figura se muestra la vista lógica de la arquitectura de **SISTEC**, definida en tres capas: la capa Vista, capa Modelo y capa Controlador; reduciendo al máximo el acoplamiento entre ellas.

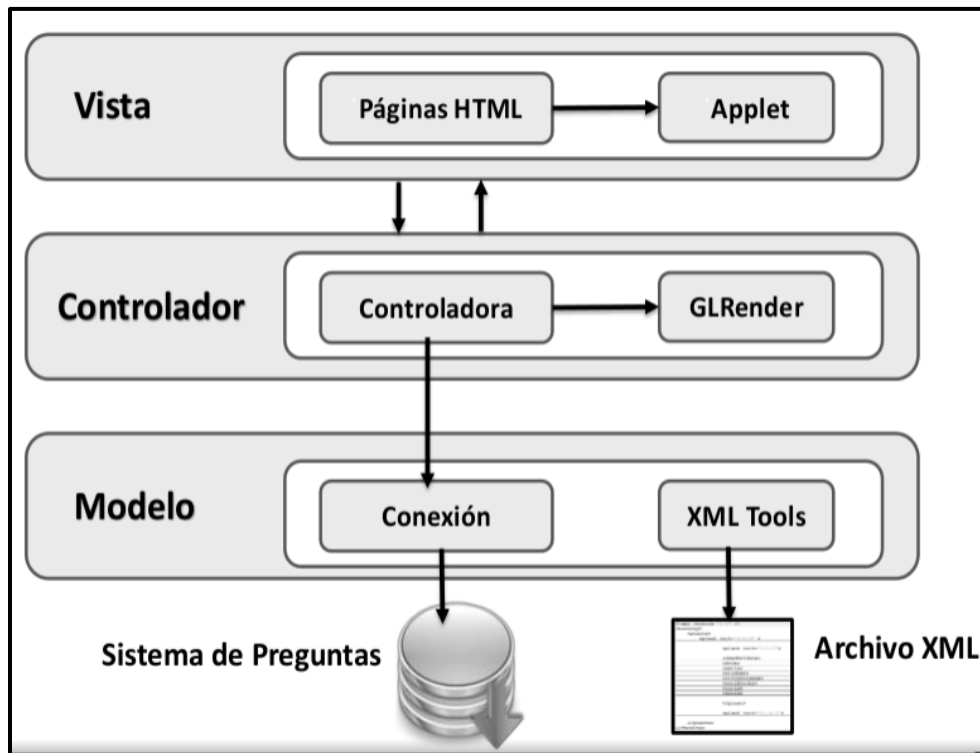


Figura 7 Vista Lógica de la arquitectura del sistema

### 2.6.1 Patrones de Diseño

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva y reusable (se puede aplicar a diferentes problemas y en distintas circunstancias).

#### Patrones GRASP

Los patrones generales de *software* para asignar responsabilidades, GRASP por sus siglas en inglés (General Responsibility Assignment Software Paterns), describen en su totalidad los principios fundamentales sobre la asignación de responsabilidades a objetos.

**Controlador:** La función de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. En este caso la clase **Controladora\_Jugar** es la encargada de realizar todas las funcionalidades del sistema, garantizando un buen funcionamiento.

**Creador:** Patrón que guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debamos conectar con el objeto producido en cualquier evento. Ejemplos del patrón creador en SISTEC:

- ✓ Jugar es la clase encargada de asumir la responsabilidad de crear instancias de XML, Observer, Canvas.

**Experto:** Asignar una responsabilidad a la clase que tiene la información necesaria para cumplirla. Ejemplo de patrón creador en SISTEC:

- ✓ *Observer* : Tiene la responsabilidad de realizar el desplazamiento por la escena
  - ✓ *Jugar*: Responsable de interpretar el XML con la información de las escenas, así como pintar dichas escenas.
  - ✓ *XML*: Encargado del trabajo con los objetos de la escena, esta clase tiene la responsabilidad de guardar los objetos en un documento XML, así como cargar dichos documentos.
  - ✓ *Conexión*: Responsable de hacer todas las peticiones a la base de datos.
- Implementación.

**Bajo Acoplamiento:** Se utilizó para la asignación correcta de responsabilidades de modo, que no exista mucha relación entre clases, de manera que un cambio en una produzca pocos o ningún cambio en aquellas clases con las que esté relacionada.

**Alta cohesión:** Basado en la asignación de responsabilidades teniendo en cuenta que la cohesión permanezca alta. Su utilización facilita la comprensión del diseño e incrementa las capacidades de realización.

## 2.7 Implementación

### 2.7.1 Estándares de codificación

Para lograr un mejor entendimiento del código en la implementación de la aplicación es necesario establecer un estándar de codificación a usar. Los estándares de codificación no son más que políticas aplicadas para permitir un mejor entendimiento en el código de un sistema. Estos estándares permitirán que el sistema sea totalmente independiente del autor y cualquier persona, ya sea desarrollador o no, pueda entender lo que se plasmó en la construcción del sistema. Con la aplicabilidad de estos estándares a cualquier aplicación, se mantiene un estilo en la programación y sirve de referencia a otros programadores para futuras integraciones o mantenimiento.

A continuación se describen los estándares utilizados para el desarrollo de la aplicación así como las normas a seguir.

**Pascal:** El primer carácter de cada palabra va con mayúscula. El estándar es utilizado en la declaración de métodos y clases, véase ejemplo en la siguiente imagen.

```
public ThreePoint SubtractionOperator(ThreePoint afirst, ThreePoint asecond)
{
    double a = afirst.getgX() - asecond.getgX();
    double b = afirst.getgY() - asecond.getgY();
    double c = afirst.getgZ() - asecond.getgZ();
    ThreePoint rest = new ThreePoint(a, b, c);
    return rest;
}
```

Figura 8 Estándar de codificación Pascal

**CamelCase (lowercamelcase):** La declaración de variables se realiza con este estándar ya que la primera letra es en minúscula seguida por la otra palabra cuyo primer carácter será en mayúscula. Véase ejemplo.

```
protected boolean glightEnabled = true;
protected boolean gPressedKeyForward = false;
protected boolean gPressedKeyBackward = false;
protected boolean glButtonDown = false;
protected boolean grButtonDown = false;
```

Figura 9 Estándar de codificación LowerCamelCase

### **Norma General**

Todo el código de este sistema se desarrollará en el lenguaje de programación Java, usando las librerías GLUEGEN-RT y JOGL, para la representación gráfica de objetos 3D.

### **Reglas a seguir:**

1. Para la declaración de los métodos en el código del sistema se realizará con la primera letra en mayúscula. Ejemplo (**Camera, Observer**).
2. Los nombres de las clases deben ser sustantivos o frase de sustantivo que no pueden usar prefijos. Ejemplo (**Light, Utils**)
3. Las clases y paquetes del sistema se nombran en inglés.
4. Las clases quedarán organizadas por los paquetes que representen su funcionalidad

en el sistema. Ejemplo (Movement [Camera, Observer])

5. Para los parámetros y campos no constantes se deben usar nombres descriptivos que deben determinar el significado de la variable y su tipo.
6. Las propiedades deben ser nombradas utilizando frases o palabras en forma sustantiva.
7. Para entrada de datos se recomienda no poner prefijos. Por ejemplo en el caso de parámetros para métodos no sería necesario declararlos como: [p\_Nombre].

### 2.7.2 Implementación del Visualizador 3D

En el paquete **Movement** se encuentran implementadas las clases encargadas de lograr los movimientos en la escena.

#### **Observador**

El observador es el mediador entre la cámara y el usuario asociados a la escena que se representa, mediante este se definen los movimientos que el usuario desea realizar y le permite desplazarse por los escenarios, basando sus movimientos mediante el teclado de la PC.

#### ***Movimientos por el teclado***

Desplazamiento hacia delante con la tecla: ↑

Desplazamiento hacia atrás con la tecla: ↓

Desplazamiento hacia la derecha con la tecla: →

Desplazamiento hacia la izquierda con la tecla: ←

Desplazamiento hacia arriba con la tecla: PageUp

Desplazamiento hacia abajo con la tecla: PageDown

Para la implementación del Observador se define una clase denominada **Observer** la cual contiene el código para el desplazamiento de las escenas representadas en el sistema. En esta clase se trabaja directamente con la **Cámara** implementada, atendiendo a sus posiciones dentro de la escena.

El observador es uno de los aspectos más importantes con los que cuenta un entorno tridimensional, ya que va a permitir a los usuarios interactuar con el sistema y poder realizar los recorridos en la escena, de manera que se aparente una forma real.

#### **Cámara**

En los entornos tridimensionales las cámaras 3D son aquellas que van a permitir mostrar los elementos en proporción a sus magnitudes, de acuerdo a como deben ser vistas por el ojo humano. La cámara contará con tres parámetros *gEyes*, *gUp* y *gDirection*, los cuales guiarán a la misma en los movimientos que debe realizar sobre la escena.

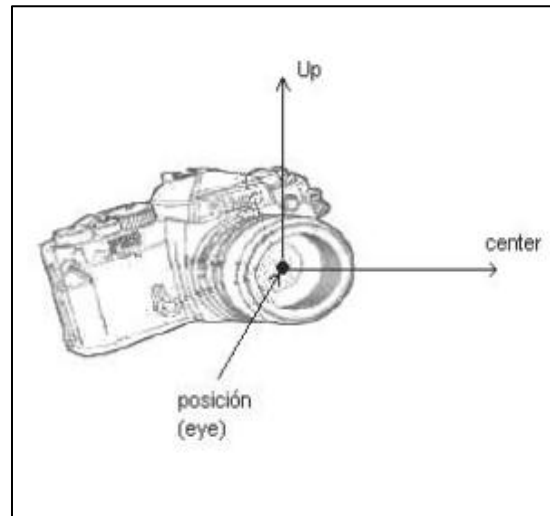


Figura 10 Parámetros de la cámara

Teniendo en cuenta estos parámetros, la cámara con el *eyes* apuntará hacia delante, con el *up* para arriba dando la dirección en altura y con el *direction* o *center* apuntará hacia el centro. Con estos puntos se puede lograr que la cámara tenga un buen funcionamiento permitiendo lograr la efectividad en los movimientos.

Los movimientos serían solo de la cámara y no de la imagen o escena que se quiere representar ya que permitiría que se pueda simular un movimiento real. Inicialmente se garantiza que la cámara se posicione (observe) en un punto sobre la escena para lo que se utilizaría la función *Look ()*, y dentro de ella la sentencia OpenGL *gl.gluLookAt* pasándole como parámetros los tres puntos de cada uno de sus parámetros iniciales, teniendo en cuenta que son de tipo *ThreePoint*. Si se desea trasladar a una posición determinada, el método *Translate (direction)*, asegura que pueda hacerlo.

Durante la implementación la clase **Cámara** se interactúa con la clase **ThreePoint**, la cual contará con tres puntos (*x*, *y*, *z*), además de poseer los métodos de Rotación, Traslación, Normalización y Norma de los vectores; permitiendo que la cámara pueda convertir sus parámetros.

### 2.7.3 Trabajo con el XML para el almacenamiento de los escenarios

La estructura de objetos con la cual será creada la escena se encontrará en un documento XML, en un formato definido por el sistema *Herramienta de autor para el diseño de escenas tridimensionales*. Estos escenarios estarán conformados por una jerarquía de objetos, creados para simular escenas reales en la aplicación. El XML quedaría diseñado con las características de cada uno de los objetos y las posiciones en las que se encontrarán en la escena. Este documento estará estructurado por etiquetas que representan los objetos que se muestran. La escena se encuentra dividida por facultades y las mismas contienen, por separado, las diferentes plantas, cada uno con sus respectivos objetos, dígame paredes, ventanas, puertas, columnas entre otros. En esta ocasión se decide visualizar cualquier escena siempre que se encuentran definidos los objetos y su estructura se encuentre en un documento con formato XML. Para observar la estructura y el esquema para el tipo de documentos XML que recibe la aplicación realizada, remitirse [Anexos #3](#).

A continuación se muestra un ejemplo del archivo XML en el cual se almacenan los escenarios. En este archivo se crea un etiqueta por cada uno de los objetos que la componen y cada uno comprende sus características dígame; propiedades, posiciones en la escena, así como el largo, alto y ancho.

```
<?xml version="1.0" ?>
<faculty>
  <plants>
    <plant init="0:0:0" >
      <quadColumns>
        <quadColumn init="-1.125:0:-0.877625" Large="2.5" BaseLarge="0.25" Width="0.35" Anchor="0.
        </quadColumns>
      <circlColumns>
        <circlColumn init="-1.125:-0.6:-8.0875" Large="10" Ratio="0.50" BaseLarge="1.25" TextureNa
        </circlColumns>
      <arcs>
        <arc init="-2.6125:2.5:-0.697375" Angle="90" Height="1.25" Width="2.6625" Thickness="0.35"
        </arcs>
      <walls>
        <wall init="-1.125:0:-1.915125" Large="0.90" Width="2.0875" Height="10" TextureName="r1" G
        </walls>
      <stairs>
        <stair init="5.5075:-0.43:-4.5075" Angle="0" StepH="0.18" StepW="0.35" Height="31.83" NumS
        </stairs>
      <blocks>
        <block init="26.875:-2.43:-12.1" Large="1.7" Height="2.34" Width="3.4" TextureName="r" Gen
        </blocks>
      <windows>
        <window init="26.025:2.75:0.5551625" Angle="90" Mark="0.1" Large="2" Width="0.1" Height="1
        </windows>
      <doors>
        <door init="26.025:0:18.9113125" Angle="90" Large="2" Mark="0.1" Width="0.1" Height="3" Te
        </doors>
      <wallBlocks>
        <wallBlock init="10.45:3.75:24.865675" Large="8.5002" Height="0.25" Width="0.93" BlockWid
        </wallBlocks>
    </plant>
  </plants>
</faculty>
```

Figura 11 Ejemplo de un archivo XML

#### 2.7.4 Descripción de un flujo de eventos del sistema (Jugar).

El estudiante accede a la interfaz principal del sistema donde encuentra toda la información acerca de la aplicación. La opción Inicio le permite conocer los objetivos y misión del sistema así como la institución donde fue desarrollada. Ver figura 11.



Figura 12 Interfaz principal del sistema

Al seleccionar la opción Jugar el sistema muestra la página que permite la configuración de la ejercitación, donde el estudiante debe seleccionar una asignatura. Para la selección de esta se le desplegará una lista con todas las asignaturas disponibles en la base de datos "Sistema de Preguntas". Luego de seleccionada, corresponde la selección del tema. Al presionar la opción (>>>) siguiente; el sistema muestra los temas correspondientes con la asignatura seleccionada. Ver figura 12.



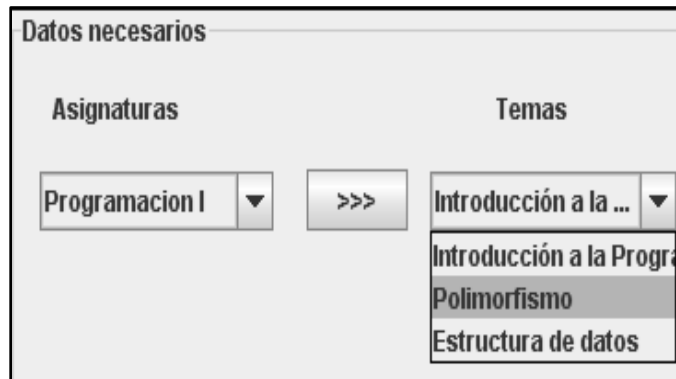


Figura 13 Interfaz seleccionar Asignatura y tema

Luego de seleccionar el tema, se selecciona la opción **Abrir** la cual mostrará un ventana en la que se permitirá la selección de la escena a ejercitar. Ver figura 13.



Figura 14 Interfaz Cargar escena

Para cargar la escena se selecciona la opción **Cargar escena**, la cual mostrará un directorio del que se puede seleccionar el archivo con el escenario en 3D para la ejercitación. Estos archivos se encuentran en una dirección física en el servidor de aplicaciones.

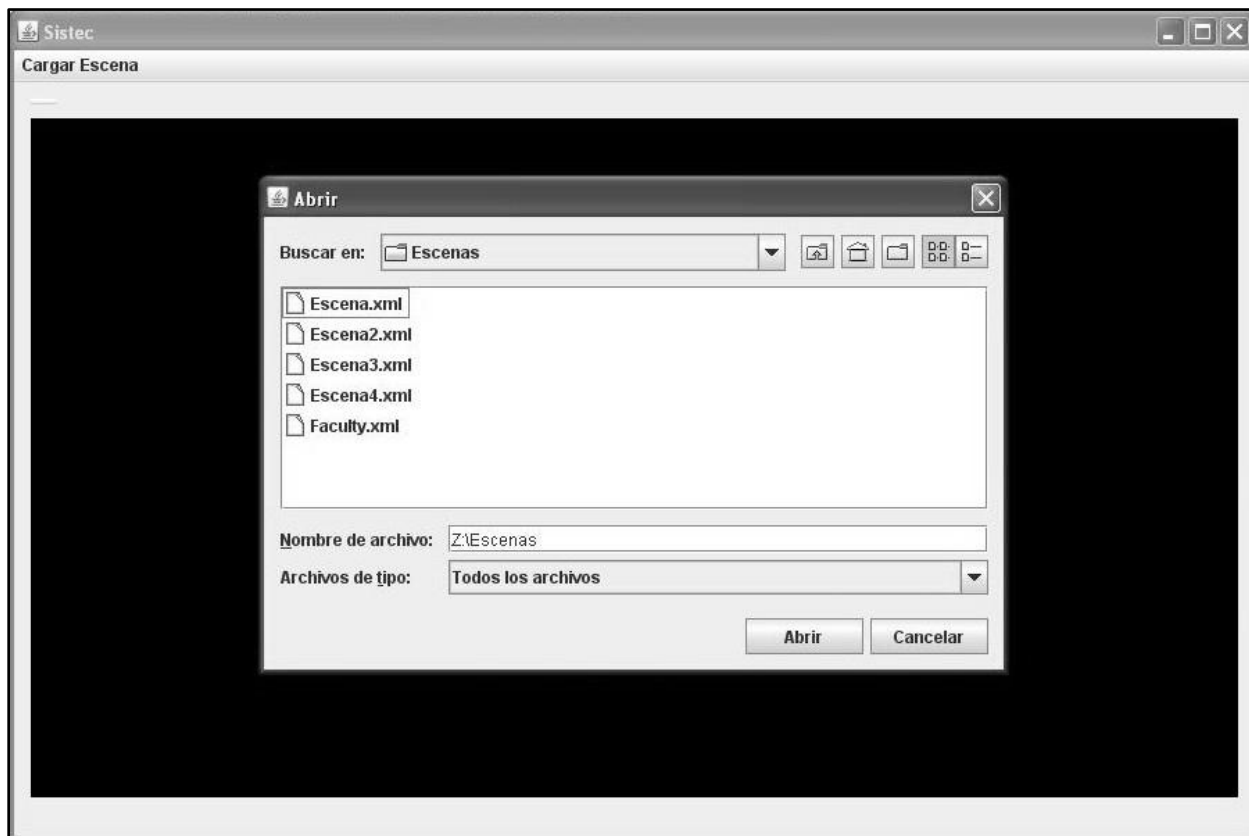


Figura 15 Interfaz Seleccionar directorio

Al concluir la configuración del juego, luego de haber seleccionado la asignatura, tema y escena en la que se desea ejercitar, el sistema guarda el listado de preguntas correspondientes al tema seleccionado. Para la ejercitación el usuario deberá desplazarse por la escena interactuando con los objetos existentes en la misma. Al encontrarse en la posición de un objeto de tipo pregunta, el sistema realiza una llamada a la base de datos de preguntas y se muestra una problemática con un nivel de complejidad bajo, garantizando una ejercitación de contenidos desde niveles bajos de complejidad. Al estudiante responder correctamente la pregunta; el sistema evalúa y actualiza en la base de datos su puntuación. Si la pregunta no es respondida correctamente el sistema muestra otra pregunta del mismo nivel de complejidad de la anterior; cuando sea respondida correctamente se permite continuar con el desplazamiento por el escenario.

### Conclusiones Parciales

- ✓ La captura de requisitos permitió una mejor comprensión de las características del sistema, garantizando el funcionamiento del mismo.

- ✓ El desarrollo de la arquitectura posibilitó obtener una mayor visión de la estructura del sistema.
- ✓ La generación de los diferentes artefactos garantizó un desarrollo más ágil del sistema.
- ✓ La aplicación del módulo SISTEC garantiza la ejercitación de contenidos en la asignatura Técnicas de Programación de Computadoras, brindan además una mayor motivación a los usuarios involucrados teniendo en cuenta que ofrecen escenarios basados en entornos tridimensionales.

## Capítulo 3: Pruebas a la herramienta “Sistema de Ejercitación de Contenidos” (SISTEC).

### Introducción

En el presente capítulo se presentan los diagramas de componentes correspondientes a cada uno de los paquetes del sistema; el diagrama de despliegue, representando los recursos físicos necesarios para el despliegue de la aplicación. Se definen las pruebas que se realizan al sistema así como los casos de pruebas pertinentes para darle validez a los requisitos funcionales y así garantizar el óptimo funcionamiento de la aplicación.

### 3.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, además de que muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Estos diagramas representan todos los tipos de elementos de *software* que entran en la fabricación de aplicaciones informáticas.

El diagrama de componentes que se muestra a continuación representa las relaciones entre las clases definidas en la herramienta para la ejercitación de contenidos a través de juegos didácticos **SISTEC**.

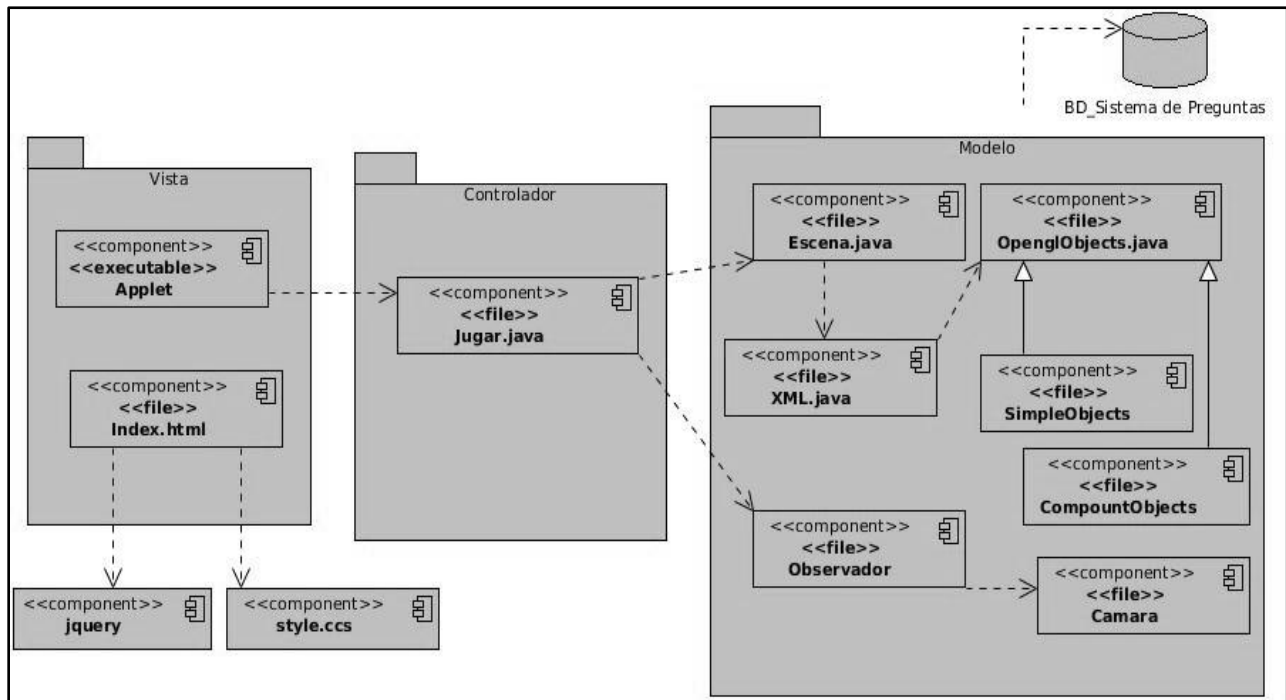


Figura 16 Diagrama de componentes de SISTEC

## 3.2 Diagrama de Despliegue

El diagrama de despliegue es el artefacto generado durante el modelado del sistema que permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. Este diagrama se representa con nodos, a través de los cuales se modela la relación entre las diferentes estaciones de trabajo con las que se debe contar para un correcto funcionamiento del sistema. Se representan las *PC* clientes, servidores de aplicaciones así como los servicios y protocolos mediante los cuales se comunicarán.

El diagrama de despliegue que se muestra a continuación representa como se encontrarán distribuidas las estaciones de trabajo de **SISTEC**, el mismo está compuesto por solo tres nodos, el servidor de aplicaciones, en el cual se encuentra un componente llamado XML, que no es más que el lugar donde se encuentran las escenas en 3D que utilizará la aplicación; la *PC* cliente y el servidor de base de datos. Atendiendo a los protocolos de comunicación utilizados en la universidad se definen TCP/IP entre los servidores y HTTPS entre el servidor de aplicaciones y la *pc* cliente.

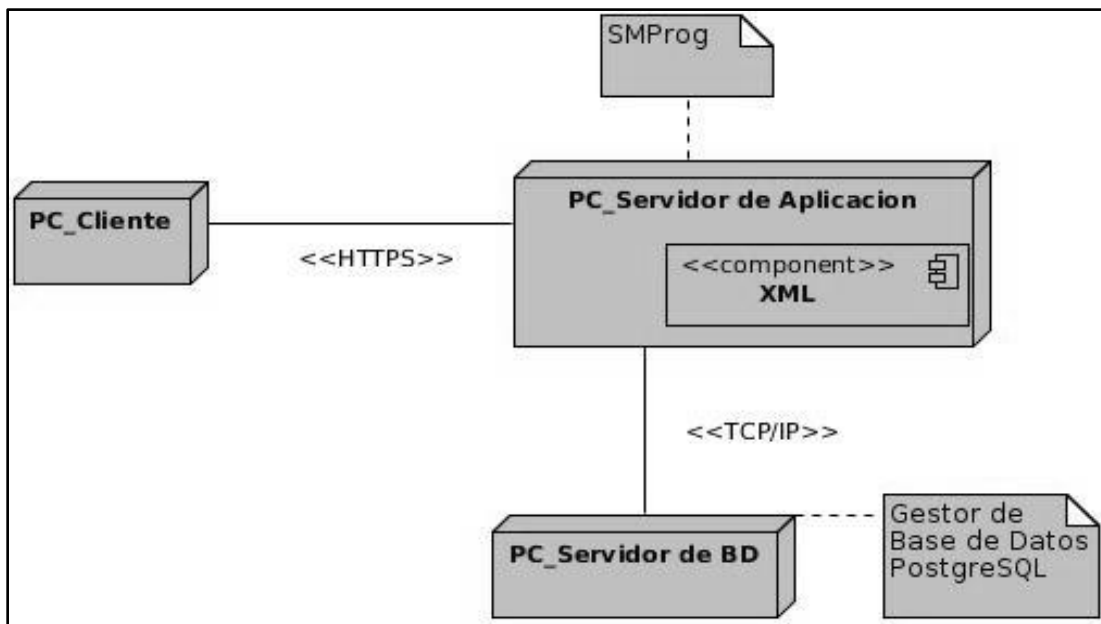


Figura 17 Diagrama de despliegue de SISTEC

## 3.3 Pruebas al sistema

Las pruebas que se realizan en un producto de *software* son necesarias para verificar el funcionamiento, la calidad y seguridad del mismo, de modo que una vez sea entregado al cliente incurra en el menor número de inconvenientes posibles. Estas tienen como objetivo que

el sistema cumpla con los requerimientos y se comporte como realmente se desea. Para su realización es necesario establecer el tipo que se va a realizar a la aplicación para ello se especifica a continuación:

1. **Pruebas de Caja Negra:** Las pruebas de caja negra son las que se realizan sobre la interfaz del programa, entendiendo por interfaz la muestra visual con la que interactúa el usuario, validando las entradas y salidas del programa. Estas pruebas no se enfocan en el diseño interno del programa sino en los requerimientos establecidos y en la funcionalidad del sistema. (32)

### 3.3.1 Descripción de Casos de Prueba

Tabla 12 Descripción del CP Jugar

<b>Caso de Prueba</b>
<b>Nombre de Caso de uso:</b> Jugar
<b>Nombre de la persona que realiza la prueba:</b> Alisnay Maturell Elías
<b>Descripción de la Prueba:</b> Funcional
<b>Condiciones de Ejecución:</b> El sistema debe estar funcionando y el usuario debe estar autenticado
<b>Entrada/ Pasos de Ejecución:</b>  <b>Para acceder al juego:</b> El usuario accede al sistema y en la vista de menú ubicada en la parte superior selecciona la opción “Jugar”.  El sistema muestra una ventana de configuración que permite la selección obligatoria de los siguientes elementos: <ul style="list-style-type: none"><li>• Asignatura (campo de selección)</li><li>• Tema (campo de selección)</li></ul> <b>Para configurar juego:</b> El usuario debe seleccionar la asignatura de la que requiere ejercitación y una vez seleccionada la asignatura, el sistema muestra la disponibilidad de temas que tiene la misma. Luego tendrá que seleccionar la opción “Abrir”, ubicada en la ventana de configuración.  El sistema guarda la configuración seleccionada y carga la ventana encargada de mostrar la

<p>escena.</p> <p><b>Para cargar la escena:</b> Una vez que la ventana carga la escena, el usuario debe seleccionar la opción “Abrir” y seleccionar la escena en la que desea ejercitar.</p> <p>El sistema permite la selección del escenario desde una dirección física, mostrando el directorio para que el usuario pueda seleccionar.</p>		
Escenarios	Resultados Esperados	Evaluación de la Prueba
SC1_Acceder al juego	El sistema muestra la ventana para la configuración del juego.	Satisfactoria
SC2_Configurar Juego	El sistema muestra las opciones de selección (asignatura, tema, escena).	Satisfactoria
SC3_Cargar escena	El sistema permite seleccionar el directorio donde se encuentra la escena, para previamente realizar la ejercitación.	Satisfactoria

### Caso de prueba Cumplir reglas de juego

A continuación se diseña el Caso de Prueba (CP) Cumplir reglas del juego, en el mismo no pudieron ser probados todos los escenarios por problemas de integración con los diferentes módulos que componen al Proyecto de Innovación Pedagógica “*Herramienta educativa sobre Software libre para la asignatura Programación II y IV*”.

Tabla 13 Descripción CP Cumplir reglas del juego

<b>Caso de Prueba</b>
<b>Nombre de Caso de uso:</b> Cumplir reglas del juego
<b>Nombre de la persona que realiza la prueba:</b> Alisnay Maturell Elías
<b>Descripción de la Prueba:</b> Funcional
<b>Condiciones de Ejecución:</b> El sistema debe estar funcionando y el usuario debe estar autenticado

### Entrada/ Pasos de Ejecución:

**Para avanzar por la escena:** El usuario debe desplazarse en la escena por las teclas ubicadas en la derecha del teclado (flechas del teclado o por *Page Down* y *Page Up*).

El sistema recorre la escena en dependencia de los movimientos (arriba, abajo, derecha, izquierda) del usuario.

**Para mostrar pregunta:** Al usuario estar en presencia de un objeto de tipo preguntas, el sistema realiza una consulta a la base de datos del “Sistema de Preguntas”, mostrando una problemática referente al tema seleccionado.

**Para responder pregunta:** El usuario selecciona la respuesta que estime, para luego ser evaluada. El sistema guarda la respuesta obtenida del usuario.

**Para evaluar pregunta:** El sistema evalúa la respuesta enviada por el jugador, comparándola con la respuesta de esa misma pregunta ubicada en la base de datos.

Si la respuesta es correcta, el sistema le permite al usuario continuar avanzando por la escena; si la respuesta es incorrecta éste muestra otra pregunta del mismo nivel de complejidad que la anterior.

**Para actualizar evaluación:** La aplicación actualiza la evaluación obtenida por el usuario durante la ejercitación del tema, en la base de datos del “Sistema de Preguntas”.

**Para indicar vía de estudio:** La aplicación promedia las evaluaciones obtenidas por el jugador y en caso de no ser satisfactorias envía el mensaje: “Debes estudiar un poco más, visita el entorno de aprendizaje [eva.uci.cu](http://eva.uci.cu)”.

De lo contrario envía: “Has obtenido resultados satisfactorios. Felicidades has ganado!!!!!!”.

Escenarios	Resultados Esperados	Evaluación de la Prueba
SC_1 Avanzar por la escena	El sistema permite el desplazamiento por el escenario seleccionado.	Satisfactoria
SC_2 Mostrar Pregunta	El sistema muestra una pregunta del tema seleccionado.	
SC_3 Responder pregunta	El sistema muestra la ventana	



	con la problemática permitiendo la selección de la respuesta.	
SC_4 Evaluar pregunta	El sistema evalúa la pregunta elegida por el usuario comparándola con la añadida por el profesor.	
SC_5 Actualizar evaluación	El sistema actualiza la evaluación del estudiante en la base de datos.	
SC_6 Indicar vía de estudio	Al concluir el juego el sistema muestra una vía de estudio en caso de resultados insatisfactorios.	

### 3.3.2 Resultado de las pruebas

Durante el desarrollo de la aplicación se fueron realizando pruebas funcionales al sistema, permitiendo identificar las No Conformidades (NC), con ayuda de los casos de prueba para cada funcionalidad. Las NC fueron clasificadas según el impacto que tenían durante el desarrollo de la aplicación.

#### Impacto de las No Conformidades:

**Alta:** Define a aquellas funcionalidades que impiden un funcionamiento total o parcial del sistema.

- ✓ En el módulo de juego la evaluación obtenida por el estudiante no era almacenada o se sobrescribía en la base de datos del Sistema de Preguntas.
- ✓ El escenario donde se ejercitará no era mostrado al usuario.
- ✓ No se permite el desplazamiento por la escena.

**Media:** Errores que impiden la conclusión de un flujo alterno.

- ✓ El sistema no muestra las preguntas por orden de complejidad impidiendo la continuidad del juego.

- ✓ Luego de responder una pregunta incorrectamente no se mostraba otra problemática con el mismo orden de complejidad.
- ✓ El sistema no contaba con las validaciones necesarias para su correcto funcionamiento.

**Baja:** No conformidades que no afectan directamente el funcionamiento del sistema.

- ✓ Errores Ortográficos en las interfaces.
- ✓ Redacción.

Al sistema se le realizaron pruebas funcionales por parte del grupo de desarrollo, las mismas se efectuaron en 3 iteraciones arrojando alrededor de 40 no conformidades.

**1ra Iteración:** Se detectaron un promedio de 20 no conformidades, las cuales se clasificaron en: Error de idioma e interfaz, errores de redacción, validaciones, opciones que no funcionan entre otros.

**2da Iteración:** Se encontraron 15 no conformidades, entre las cuales se destacan: la opción de cargar escena que no mostraba el escenario seleccionado, así como errores de código e inconformidades por parte del cliente en cuanto a las validaciones y excepciones que debe arrojar el sistema.

**3ra Iteración:** Se detectaron 5 no conformidades en cuanto a operaciones funcionales del sistema.

A continuación se muestra una imagen donde se describen las no conformidades encontradas y resueltas durante las tres iteraciones.

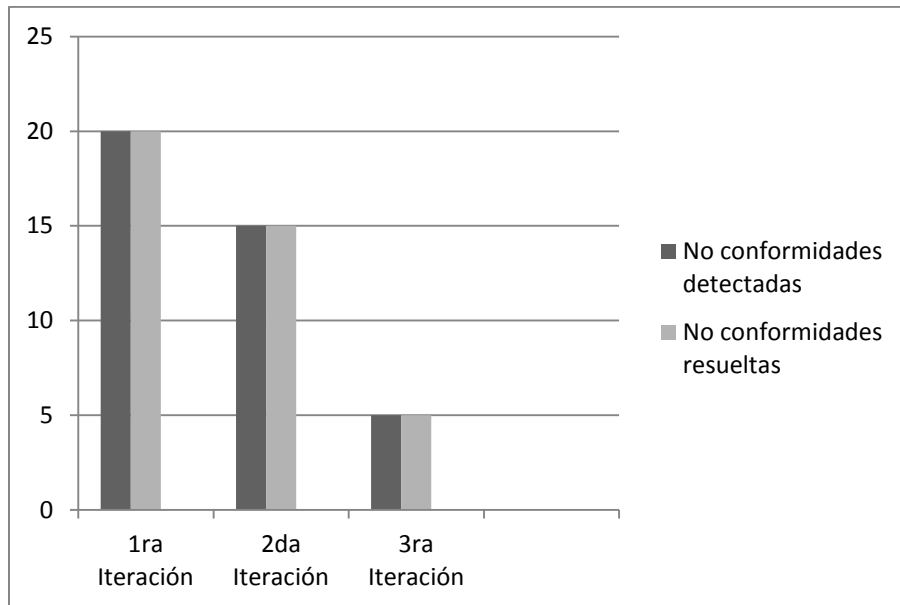


Figura 18 Resultado de No Conformidades durante las iteraciones

## Conclusiones Parciales

- ✓ Las pruebas realizadas posibilitaron el correcto funcionamiento de la aplicación, permitiendo un fácil manejo por parte de los usuarios involucrados.
- ✓ Los diagramas de despliegue y de componentes reflejan los recursos físicos y la estructura de los paquetes en la solución propuesta.

### **Conclusiones Generales**

- ✓ Los juegos serios constituyen una alternativa para el Proceso-Enseñanza-Aprendizaje (PEA), facilitando la evaluación y ejercitación de contenidos en la Educación Superior; creando una mayor motivación en los estudiantes y profesores.
- ✓ Los sistemas homólogos estudiados no están en correspondencia con la solución a implementar, debido a que son diseñados para disciplinas específicas que no son las impartidas en la universidad, por lo que se descarta su utilización y se hace necesario el desarrollo de un nuevo sistema para la ejercitación de contenidos.
- ✓ El diseño de la arquitectura permitió tener una mejor visión de la aplicación, así como una mayor organización de los componentes que la integran.
- ✓ Las pruebas realizadas posibilitaron verificar el correcto funcionamiento del sistema, permitiendo una fácil interacción de los usuarios en el mismo.
- ✓ La aplicación desarrollada permite la ejercitación y evaluación de contenidos a través de juegos didácticos tridimensionales en la disciplina Técnicas de Programación de Computadoras de la UCI.

### **Recomendaciones**

Luego de la creación del sistema para la ejercitación de contenidos se recomienda:

- ✓ Integrar el entorno del juego a la Web mediante la utilización de applet.
- ✓ Desarrollar las funcionalidades mostrar y evaluar pregunta.
- ✓ Actualizar la evaluación del estudiante luego de haber realizado la ejercitación de contenido.
- ✓ Crear niveles de ayuda para los diferentes tipos de preguntas, en los que se les muestre a los estudiantes una solución guiada en caso de no saber responder una problemática.
- ✓ Crear nuevos escenarios tridimensionales cumpliendo con el formato de archivos XML.

## Glosario de Términos

**API:** Interfaz de programación de aplicaciones o Application Programming Interface por sus siglas en inglés. Es un conjunto de funciones (métodos) y procedimientos que ofrece una biblioteca para ser utilizado por otro *software*.

**Applet:** Es un componente de *software* que se ejecuta en el contexto de otro programa como por ejemplo en un navegador web. El *applet* debe ejecutarse en un contenedor que lo proporciona un programa anfitrión, mediante un *Plug-in*, o en aplicaciones que soportan el modelo de programación por *applets*.

**Bases de datos:** Es un conjunto de datos junto con una serie de aplicaciones para su manejo, accesible simultáneamente por diferentes usuarios y programas.

**CUS:** Casos de Uso del Sistema.

**GPL:** Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente GNU GPL). Es la licencia más utilizada en el *software*.

**HTTPS:** HiperText Transfer Protocol Secure (en español Protocolo de transferencia segura de hipertextos). Es la versión segura del protocolo http. El sistema http utiliza un cifrado basado en la Secure Socket Layers (SSL) con el que se destina a páginas web en las que hay entrega de datos personales de un usuario a una web o viceversa.

**Hardware:** Son consideradas todas las partes físicas y tangibles de una computadora, es decir cables, cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

**Multiplataforma:** Se refiere a programas informáticos que pueden funcionar en diversas plataformas.

**PIP:** Proyecto de Innovación Pedagógica creado en la Universidad de las Ciencias Informáticas con el fin de ofrecer una vía de motivación a los estudiantes a través de juegos didácticos, para la sistematización de los contenidos.

**PEA:** Proceso-Enseñanza-Aprendizaje es la ciencia que estudia la educación como unos procesos consientes, organizado y dialéctico de apropiación de los contenidos.

**PC:** Personal Computer o Computadora Personal. Máquina electrónica personal rápida y exacta que es capaz de aceptar datos a través de un medio de entrada, procesarlos

automáticamente bajo el control de un programa previamente almacenado, y proporcionar la información resultante a un medio de salida.

**Plugins:** Es un programa o aplicación que añade funcionalidad al programa principal donde está hospedado. Es común también llamarlos extensiones y se consideran una aplicación que desempeñan una tarea muy específica dentro de un conjunto de todas las funcionalidades del programa principal en el cual se desempeñen.

**UCI:** Universidad de las Ciencias Informáticas.

**Software:** Se conoce como software al equipamiento o soporte lógico de una computadora digital; comprende el conjunto de componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos del sistema, llamados *hardware*.

**TCP/IP:** Protocolo de Control de Transmisión/Protocolo de Internet (Transmission Control Protocol/Internet Protocol), es un sistema de protocolos que hacen posibles los servicios de Internet, correo y otros; además de que hacen posible la transferencia de datos entre redes de ordenadores.

### Bibliografías Referenciadas

1. HORIZON. *El Informe Horizon*. California: 2007.
2. GARCÍA, L. A. V. Uso de los videojuegos como auxiliar didáctico en la educación superior. *enero-junio*, 2013, nº p. 39.
3. ADRIANA FIDEL, J. V. *Aprender aprender en 3D*. Misnisterio de Educación GCBA-Argentina. Universidad Pedagógica Experimetal Libertador (UPEL), 2009.
4. FERREIRO, D. S. *Herramienta para la creación y uso de juegos didácticos en la enseñanza de la Programación*. Departamento de Técnicas de Programación. Universidad de las Ciencias Informáticas, 2011.
5. SALTAREN, H. *Juegos Serios Colombia*: Facultad de Ciencias de la Administración, Disponible en: <http://cuse.univalle.edu.co/index.php/juegos-serios/itemlist/user/48-henrysaltaren>.
6. CASTRO, L. *Módulo para la construcción de tableros de juegos didácticos de la herramienta educativa SoMProg para el Entorno Virtual de Aprendizaje Moodle*. Universidad delas Ciencias Informáticas, 2008.
7. BLOG, D. *¿Porque usar mundos virtuales en la Educación?* Disponible en: <http://o3dsoft.com/blog/es/2011/04/por-que-usar-mundos-virtuales-en-la-educacion/>.
8. ALTAMIRANO, E. *Mundos Virtuales y educación a distancia*. 2008, nº Disponible en: <http://cerv-uaq.blogspot.com/2008/09/ventajas-y-desventajas-de-los-mundos.html>.
9. JAIRO. *Historia y Características de los mundos virtuales* Disponible en: <http://www.tecnolives.com/historia-y-caracteristicas-de-los-mundos-virtuales/>.
10. LEÓN, U. D. *Sistema de ejercitación y práctica* Disponible en: <http://ticleon.wikispaces.com/Sistemas+de+ejercitaci%C3%B3n+y+pr%C3%A1ctica>.
11. JUAN C. QUINCHE, F. L. G. *Entornos Virtuales 3D, Alternativa Pedagógica para el Fomento del Aprendizaje Colaborativo y Gestión del Conocimiento en Unminuto*. 2011, nº p. 10.
12. GAME, S. *Dental Implant Training Solution* Disponible en: <http://www.breakawayltd.com/serious-games/solutions/healthcare/>.



## Bibliografía Referenciada

---

13. ---. *INNOVU "El emprendedor"* Disponible en: <http://lochnessh.wordpress.com/2010/12/15/ibm-innov8-un-juego-de-simulacion-de-bpm-business-process-management/>.
14. IBM. *"INNOV 8"* Disponible en: <http://www-01.ibm.com/software/solutions/soa/innov8/index.html>.
15. GAME, S. *"Virtual Clinical Learning Lab for Health Care Training"* Disponible en: <http://www.breakawayltd.com/serious-games/solutions/healthcare/>.
16. ---. *"Environmental Detectives"* Disponible en: <http://www.educationarcade.org/gtt/Handheld/Intro.htm>.
17. LIFE, S. *Second Life* Disponible en: <http://secondlife.com>.
18. RUBIO, R. L. *"Entornos Virtuales" Juego de Dominó para Jugadores Virtuales*. Universidad de la Ciencias Informáticas, 2009.
19. HUMANA, E. *Ventajas y Beneficios de las aplicaciones web* Disponible en: <http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html>.
20. PROMOB. *Visión general y beneficios del servidor Tomcat* Disponible en: <http://es.prmob.net/apache-tomcat/apache-software-foundation/microsoft-personal-web-server-2505698.html>.
21. MARIN, M. D. A. *Definición de lenguajes de programación. Tipos. Ejemplos* (Catedra de Programación). Disponible en: <http://catedraprogramacion.foroactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
22. MARAÑÓN, G. Á. *Características del lenguaje Java* Disponible en: <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
23. DUNIA SUÁREZ FERREIRO. *Representación Tridimensional de Edificaciones*. Facultad de Matemática y Computación Universidad de la Habana, 2005.
24. ALVARÉZ, M. A. *¿Qué es HTML?* Disponible en: <http://www.desarrolloweb.com/articulos/que-es-html.html>.
25. BOS, B. *Cascading Style Sheets Level 2*. Editado por: Çelik, T. 2011.
26. CAMEJO, J. E. G. *¿Que es UML?* Disponible en: <http://www.docirs.cl/uml.htm>.

## Bibliografía Referenciada

---

27. MANAGER, F. D. *Paradigma visual para UML* Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%5Bcuenta\\_de\\_Plataforma\\_de\\_Java\\_14715\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/).
28. HERNÁNDEZ, Y. *El Proceso Unificado de Desarrollo de Software (RUP)* Disponible en: <http://yaqui.mx/uabc.mx/~molquin/as/RUP.htm>.
29. ORTEGA, A. L. *Introducción a OpenGL. ¿Que es OpenGL?* Disponible en: <http://www.alobbs.com/revistas/opengl1>.
30. MARTÍNEZ, R. *Sobre PostgreSQL* Disponible en: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
31. OROZCO, S. *Diagramas de Interacción* Disponible en: <http://sg.com.mx/content/view/415>.
32. ADAM, V. G. Pruebas de caja negra. 2012, nº Disponible en: <http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>.
33. Figueroa, Roberth G., Solís, Camilo J. y Cabrera, Armando A. METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. [En línea] [Citado el: 17 de Enero de 2013.] Disponible en: <http://www.google.com/cu/url?sa=t&rct=j&q=metodologias+de+desarrollo+de+software+pesadas&source=web&cd=6&ved=0CEkQFjAF&url=http%3A%2F%2Fadonisnet.files.wordpress.com%2F2008%2F06%2Farticulo-metodologia-de-sw+formato.doc&ei=bTn0ULbfJqH0QHcmIDACg&usg=AFQjCN>.
34. Ramos, Isidro Salavert y Lozano, María Dolores Pérez. Entornos de Desarrollo Integrado. Ingeniería Del Software Y Bases de Datos: Tendencias Actuales. S.I.: Univ. de Castilla. La Mancha, 2000.

## Bibliografía Consultada

1. PÉREZ, J. F. B. *¿Como elegir un servidor web?* 2010, Disponible en: <http://www.slideshare.net/programadorphp/cmo-elegir-un-servidore-web>.
2. LEÓN, U. D. *Sistema de ejercitación y práctica* Disponible en: <http://ticleon.wikispaces.com/Sistemas+de+ejercitaci%C3%B3n+y+pr%C3%A1ctica>.
3. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) por Visual Paradigm International Ltd. - reporte y descarga. [Sitio de descarga libre de software]. Última actualización: 2007/03/05 de 2013/05/07/14:55:25. Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/files/223/Paradigma\\_Visual\\_para\\_UML\\_\(Ml\)\\_14720\\_p.html](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/files/223/Paradigma_Visual_para_UML_(Ml)_14720_p.html).
4. COLOMÉ CEDEÑO, I. D. M.; ESTRADA SANTÍ, D. V., et al. Ambiente tecnológico para la creación de objetos de aprendizaje en apoyo al proceso docente de las universidades cubanas. ACIMÉD, 2012/03/21/de 2012, vol. 23, nº p. 116-129. [Consultado el: 2013/01/20/17:40:33]. Disponible en: <http://new.medigraphic.com/cgi-bin/resumenMain.cgi?IDARTICULO=35871files/23/resumenMain.html>. ISSN 1024-9435.
5. FERNÁNDEZ, E. F. y HERNÁNDEZ, A. G. Módulo de gestión de asignaturas de la herramienta educativa SMProg para el EVA - UCI. Tutor: Ferreiro., L. D. S. y Chapman, I. a. H. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Facultad 10. Universidad de las Ciencias Informáticas, 2010.
6. JIMÉNEZ, M. O. *Netbeans con PostgreSQL* Disponible en: <http://nuclearcode.blogspot.com/2011/10/conectar-netbeans-con-postgresql.html>.
7. CARLOS MARIO ZAPATA, C. P., NATALÍ AYALA. *UNC- Analista: Hacia la captura de un corpus e requisitos a partir de la aplicación del experimento Mago de Oz*. Medellín: Colombia, 2007, vol. Número 7.
8. CHAVES, M. A. *La ingeniería de requisitos y su importancia en el desarrollo de proyectos de software*. Costa Rica: Costa Rica, 2006, vol. Número 10, Disponible en: [www.intersedes.ucr.ac.cr](http://www.intersedes.ucr.ac.cr).
9. MARÍA JOSÉ ESCALONA, N. K. *Ingeniería de Requisitos en Aplicaciones para la Web- Un estudio comparativo*. Sevilla: España, 2002, 26 p.
10. MUÑIZ, M. D. *Ingeniería de Requisitos para Xerberos*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2012.
11. HORIZON. *The Horizon Report*. Stanford, California: California, 2011.
12. HORIZON. *Horizon Report Higher Education Edition*. Stanford: California, 2012.

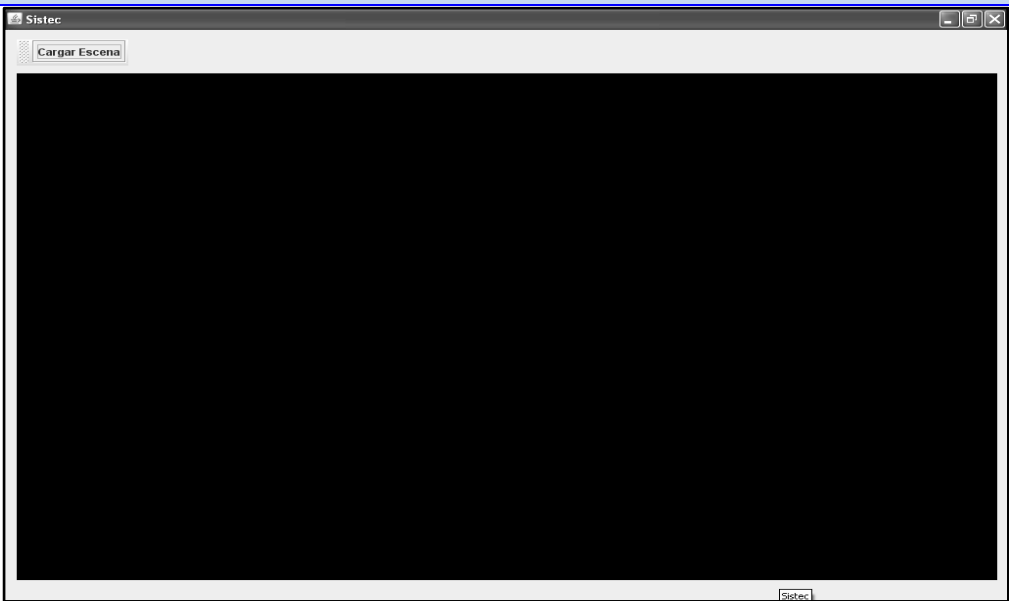
## Bibliografía Consultada

---

13. JOHNSON, L., ADAMS. *Resumen Informe Horizon 2012. Enseñanza Universitaria*. 2012.
14. JOHNSON, L., ADAMS. *Resumen Informe Horizon 2012. Enseñanza Primaria y Secundaria*. 2011.
15. EDUARDO FAILDE FERNÁNDEZ, A. G. H. *Módulo de gestión de asignaturas de la herramienta SMProg para el EVA-UCI*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2010.
16. BÁRBARA MARTA ZERQUERA, Y. M. M. *Módulo SMProg para el Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2010.
17. LUDIEL CASTRO BARRIOS, M. B. B. *Módulo para la construcción de tableros de juegos didácticos de la herramienta educativa SoftMotProg para el Entorno Virtual de Aprendizaje Moodle*. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2008.
18. MARCANO, B. *Juegos Serios y entrenamiento en la sociedad digital*. [Revista Electrónica Teoría de la Educación. Educación y Cultura en la Sociedad de la Información.]. 2008,
19. LARMAN, C. *UML y Patrones. Introducción al análisis y Diseño orientado a objetos*. México: 1999, 536 p. ISBN: 970-17-0261-1.
20. SOMMERVILLE, I. *Ingeniería de Software Séptima Edición*. ISBN: 84-7829-074-5 ed. Madrid: 2005. 712 p.

## Anexo #1 Especificación de Requisitos.

### Descripción RF\_3 Seleccionar escena

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_3]	Seleccionar Escena	Se muestra un listado de escenas tridimensionales que se encuentran disponibles en la aplicación y mediante las cuales se realiza la ejercitación.	Media	Alta
<b>Prototipo</b>				
 <p>Prototipo Seleccionar Escena.</p>				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
<b>Observaciones</b>		2. El sistema siempre contará con escenas disponibles para la ejercitación.		

### Descripción RF\_5 Mostrar preguntas

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_5]	Mostrar preguntas	Se muestra una pregunta en caso de que el objeto sea didáctico, está será en	Media	Alta

		correspondencia con el tema seleccionado.		
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	1. Para poder avanzar por la escena se debe responder correctamente la pregunta.		

#### Descripción RF\_6 Evaluar preguntas

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_6]	Evaluar pregunta	La respuesta del estudiante será comparada con la orientada por el profesor y si es correcta se le otorgarán todos los puntos correspondientes a esa pregunta.	Media	Baja
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	1. La respuesta del estudiante debe estar en correspondencia con la publicada por el profesor que creó el sistema de preguntas. 2. Luego de evaluada la pregunta el estudiante puede continuar jugando.		

#### Descripción RF\_7 Responder preguntas

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_7]	Responder pregunta	Se muestra la pregunta y el estudiante responde correctamente, para poder continuar la ejercitación. Para ello selecciona la respuesta que sea correcta.	Media	Alta
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	1. Las evaluaciones por cada respuesta serán acumuladas y resultará una evaluación final del tema.		

#### Descripción RF\_8 Avanzar por la escena si responde correctamente

Nº	Nombre	Descripción	Complejidad	Prioridad para
----	--------	-------------	-------------	----------------

				cliente
[RF_8]	Avanzar por la escena si responde correctamente	Si la respuesta del estudiante es correcta avanza por la escena, continuando con la ejercitación. De lo contrario el sistema mostrará otra pregunta hasta que sea respondida correctamente.	Media	Alta
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	1.		

#### Descripción RF\_9 Indicar vía de estudio

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_9]	Indicar vía de estudio	Se muestra una vía de estudio en el caso de que estudiante no haya alcanzado la evaluación correspondiente atendiendo a la cantidad de preguntas. Esta vía de estudio es el Entorno Virtual de Aprendizaje (EVA) para continuar la preparación de los estudiantes.	Media	Baja
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	1. La vía de estudio indicará al eva.uci.cu haciendo énfasis en la asignatura del tema seleccionado.		

#### Descripción RF\_10 Actualizar evaluación de estudiante

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_10]	Actualizar evaluación de estudiante	Luego de la ejercitación el sistema actualiza la evaluación del estudiante que estuvo ejercitando. Garantizando que luego su profesor pueda consultar la evaluación de la última ejercitación.	Media	Baja
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	1. La actualización de esta evaluación es importante, debido a que se		

		puede realizar un seguimiento al estudiante de temas en los que necesite fortalecer los conocimientos.
--	--	--



## Anexo#2 Clases entidades del sistema.

<b>Nombre:</b> Pregunta	
<b>Tipo de clase:</b> Entidad persistente	
<b>Atributos</b>	<b>Tipo</b>
cuerpo	String
complejidad	String
id	Integer
id_tema	Integer
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
get_Cuerpo	Retorna el cuerpo de la pregunta
get_ID	Retorna el identificador de la pregunta.
get_Complejidad	Retorna la complejidad de la pregunta
get_IDTema	Retorna el identificador del tema al que pertenece.

<b>Nombre:</b> Respuesta	
<b>Tipo de clase:</b> Entidad persistente	
<b>Atributos</b>	<b>Tipo</b>
cuerpo	String
id	Integer
id_pregunta	Integer
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Descripción:</b>
get_Cuerpo	Retorna el cuerpo de la respuesta
get_ID	Retorna el identificador de la respuesta.
get_IDTema	Retorna el identificador del tema al que pertenece.

Anexo #3 Estructura del XML.

