



Universidad de las Ciencias Informáticas

Facultad 2

Sistema para la localización espacial de los recursos que gestiona GRHS UCI

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autores:

Raidel Ross Rodríguez

Michael Brug Santrayll

Tutor:

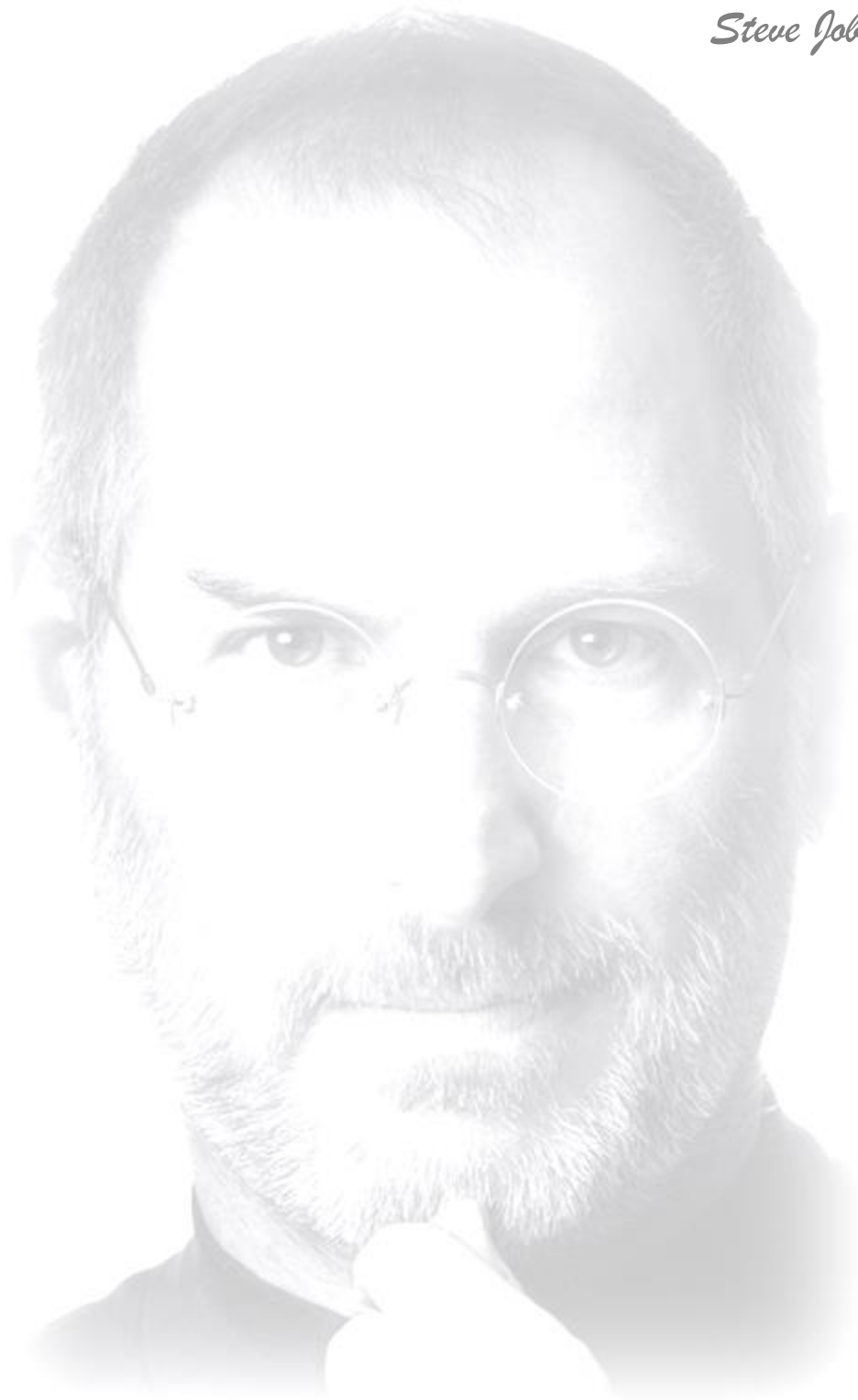
Ing. Arianna Pérez Carmenates

La Habana, marzo 2016

“Año 58 de la Revolución”

La innovación es lo que distingue a un líder de los demás.

Steve Jobs



DECLARACIÓN DE AUTORÍA

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

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

Raidel Ross Rodríguez

Firma Autor

Michael Brug Santrayll

Firma Autor

Ing. Arianna Pérez Carmenates

Firma Tutor

Agradecimientos

Primeramente, quiero agradecer a mamá por siempre estar ahí cuando la necesito, por ser madre y padre en todo momento, por la crianza y educación que me dio y por su apoyo incondicional.

A mis tías (Chavela, Margarita, Virgen Juanita, Ena y Migdalia) por brindarme tanto cariño, apoyo incondicional, consejos y regaños cuando eran necesarios.

A mis primos los mayores, por servirme de guía y a los menores por permitirme ser un ejemplo para ellos.

A toda mi familia en general, por obligarme a esforzarme para convertirme en un profesional.

A mi compañero de tesis por compartir la travesía, antes de realizar la tesis para mí era un desconocido, ahora forma parte de mi círculo cerrado de amistades.

A mis tutores por todo el esfuerzo durante la realización de este trabajo, por las correcciones de última hora, las intensas y largas llamadas por teléfono aclarando dudas, por los consejos, los alones de oreja, gracias por todo.

A mis amigos de la infancia que estuvieron ahí en los momentos difíciles, por hacerme saber que puedo contar con ellos, a los de la UCI, gracias a todo el piquete de 1er año, que hoy solo quedamos 6 de 13 que empezamos (el cheo, el butty y los pollos que me debe, Yoan, cutu y Hector) muchos han quedado en el camino a los cuales también les agradezco. Al piquete del voly, a todos los viejos, los nuevos, a Leodan y a pedro por las aclaraciones de dudas por el jabber, también ahí entra Maren, Luz María, Voltus y Estela que en varias ocasiones me brindaron su apoyo y aportaron su granito de arena.

A mis compañeros de aula, que empezamos 27 en el grupo 2105 y hoy solo quedamos los 5 (Victor, Brayan, Osmar y Asley), también a los que se incorporaron después, de los que me llevo grandes recuerdos.

A mis grandes amigas Lia y Eyidey, a JG, a Dairan, a la minion y a Oliva.

A todos los profesores que me han impartido materias a lo largo de este proceso, los que me soportaron mis locuras en clases.

En fin, a todos los que de una forma u otra han contribuido a cumplir este sueño.

Michael Brug Santrayll

Agradecimientos

A quien me dio la vida, por todo ese esfuerzo realizado para regalarme cada pedacito de su tiempo, de su cariño, de su respeto y de su confianza, elevando el significado de la palabra “Madre” a su nivel más alto. Ella es la evidencia de todo el amor que puede sentir una madre hacia sus hijos.

*A mi padre, a quien respeto y quiero mucho a pesar de haber crecido lejos de él.
A mi hermano más cercano Yuriel por sus recomendaciones, por guiarme tanto en los estudios como en mis decisiones.*

A mi abuelita Fortuna, por aguantar mis malcriadeces desde pequeño, aunque ahora me toque a mí aguantar las suyas.

A mi novia casi esposa Liyanis Peláez Baños junto a la cual he demostrado que la “felicidad” existe y casi estamos por demostrar que el “amor” también.

*A los amigos que han sido como hermanos Ayeban, Laimber, Alfredito, Roniel,
David.*

A mis parientes más cercanos aquí en la Habana Mary y Ernesto por su apoyo y su cariño.

A mi suegra, quien ha confiado en mí y me ha brindado su amistad su cariño y su mejor comida.

A todos aquellos profesores que influyeron positivamente en mi formación, en especial a mis tutores de tesis que confiaron en nosotros desde el principio.

A mi colega Maikel, por compartir esta inolvidable aventura.

Raidel Ross Rodríguez

Dedicatoria

Le dedico este trabajo de diploma a mi madre querida, por ser todo para mí, por la cual me acuesto y me levanto cada día pensando cómo hacerla feliz,

A mi tía Chavela por ser otra madre para mí

A mi papá, al que no conocí, pero me dejó enseñanzas que me convirtieron en lo que soy hoy.

A mis padrinos que hicieron función de padres.

En fin, se lo dedico a todas las personas que siempre confiaron en mí.

Michael Brug Santrayll

Le dedico este trabajo de diploma a mi madre, por haber sido mi ídolo y ejemplo a seguir.

Raidel Ross Rodríguez

Resumen

El sistema Gestor de Recursos de *Hardware* y *Software* (GRHS), es una herramienta desarrollada para inventariar y controlar medios tecnológicos conectados a la red. Este sistema carece de funcionalidades asociadas a la localización espacial de los recursos. Para resolver esta situación se desarrolló un sistema que utiliza la visualización de gráficos 3D por computadora que permite la detección de violaciones ocurridas mediante la localización de los recursos en el espacio 3D, además de la identificación de regiones donde hay mayor ocurrencia de incidencias. Posee la capacidad de combinar la información almacenada en el sistema GRHS con la ubicación espacial de las estructuras a las que pertenece dicha información, de esta manera se logró un producto final en apoyo a la toma de decisiones que facilita la búsqueda óptica de agentes con incidencias. Para desarrollar este sistema se utilizó XP como metodología de *software*, como plataforma de desarrollo se utilizó el motor de videojuegos Unity3D, el entorno de desarrollo integrado MonoDevelop y C Sharp (C#) como lenguaje de programación, para el modelado UML Visual Paradigm, como gestor de base de datos PostgreSQL, como editor de imágenes GIMP y Blender como herramienta de modelado 3D.

Palabras clave: Visualización de gráficos 3D por computadora, Gestor de Recursos de *Hardware* y *Software*, incidencias.

Summary

Resources Manager Hardware and Software (GRHS) system is a tool for inventorying and control networked technological means. This system lacks features associated with the spatial location of resources. To resolve this situation it developed a system that uses the display of 3D computer graphics that allows detection of violations that have occurred by locating resources in 3D space, in addition to identifying regions where there is greater occurrence of incidents. It has the ability to combine information stored in the GRHS system with the spatial location of structures to which such information pertains, In this way a final product was obtained in support of decision-making that facilitates optical search of agents incidents. XP was used as software methodology to develop this system, as a development platform Unity3D game engine was used, the integrated development environment MonoDevelop and C Sharp (C #) programming language, UML for modeling Visual Paradigm, as manager PostgreSQL database, as picture editor GIMP and Blender as 3D modeling tool.

Keywords: *Visualization of 3D computer graphics, Resources Manager Hardware and Software, incidences.*

Índice

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	4
1.1 Técnicas de modelado 3D.....	4
1.2 Visualización de gráficos 3D por computadora.....	6
1.3 Sistemas con visualización de gráfico por computadora.....	6
1.4 Metodología de desarrollo de <i>software</i>	7
1.5 Tecnologías de desarrollo y herramientas.....	8
1.5.1 Unity 3D.....	8
1.5.2 Visual Paradigm.....	8
1.5.3 Ambiente de desarrollo integrado.....	9
1.5.4 Lenguaje de programación.....	9
1.5.5 Sistema gestor de base de datos PostgreSQL.....	9
1.5.6 Herramienta de modelado 3D.....	9
1.5.7 Editor de imágenes.....	10
1.6 Conclusiones parciales del capítulo.....	10
Capítulo 2: Exploración y planificación.....	11
2.1 Descripción del sistema a desarrollar.....	11
2.2 Fase de exploración.....	11
2.2.1 Características funcionales del sistema.....	12
2.2.2 Características no funcionales del sistema.....	12
2.2.3 Especificación de las historias de usuario.....	13
2.3 Fase de planificación.....	14
2.3.1 Estimación del esfuerzo de cada historia de usuario.....	14
2.3.2 Plan de iteraciones.....	15
2.3.3 Plan de entregas.....	15
2.4 Conclusiones parciales del capítulo.....	16
Capítulo 3: Diseño, implementación y prueba.....	17
3.1 Fase de diseño.....	17
3.1.1 Patrón de arquitectura de <i>software</i>	17
3.1.2 Modelo físico de la base de datos.....	18
3.1.3 Tarjetas Clase-Responsabilidad-Colaborador.....	18
3.1.4 Patrones de diseño.....	19
3.2 Fase de implementación.....	19
3.2.1 Tareas de ingeniería.....	20
3.3 Fase de prueba.....	20
3.3.1 Pruebas de aceptación.....	20
3.4 Conclusiones parciales del capítulo.....	23
Conclusiones generales.....	24
Referencias.....	26
Bibliografía.....	28
Anexo I.....	32
Anexo II.....	37

Anexo III.....	42
Anexo IV	43
Glosario de Términos	51

Índice de Figuras

Figura 1 Propuesta del sistema.....	11
Figura 2 Funcionamiento del patrón MVC en el sistema.	17
Figura 3 Modelo físico de la base de datos.....	18
Figura 4 Ejemplo del patrón creador	19
Figura 5 Resultados de las pruebas de aceptación	22
Figura 6 Resultado de las pruebas unitarias.....	23

Introducción

El desarrollo de la tecnología ha reducido las barreras para realizar negocios, incrementar ingresos, mejorar procesos e implementar nuevos sistemas y herramientas dentro de las entidades (1), sin embargo, para lograr la integridad y el correcto funcionamiento de las empresas es necesaria la protección de todos sus medios tecnológicos. Con el fin de alcanzar este objetivo las entidades realizan periódicamente un inventario de sus recursos para verificar la existencia de materiales, equipos, muebles e inmuebles, a efecto de comprobar el grado de eficacia en los sistemas de control administrativo y el manejo de los materiales (2).

En la actualidad la labor de realizar el inventario se ha convertido en una tarea compleja, debido al aumento de la cantidad de medios informáticos que se utilizan en las redes de las empresas para el intercambio de recursos y la comunicación. Para agilizar esta actividad existen los inventarios de red, que consisten en obtener automáticamente la información relevante de los activos de una red de computadoras, donde se realizan controles periódicos al *hardware* y al *software*.

En Cuba está establecido que toda entidad debe mantener actualizado el inventario de las tecnologías informáticas que posee (3), de ahí que este sea un proceso de vital importancia para el cumplimiento de las normas de seguridad. El sistema GRHS, es una herramienta desarrollada por el Centro de Telemática (TLM) perteneciente a la Universidad de las Ciencias Informáticas (UCI), que responde al interés de inventariar, para controlar el gran número de medios tecnológicos conectados a la red de dicha universidad. Hoy en día este sistema goza de gran aceptación entre directivos y especialistas de seguridad informática, gracias a la abundante información que brinda sobre los medios computacionales, lo que lo convierte en una potente herramienta para el apoyo a la toma de decisiones.

A pesar de las múltiples ventajas ofrecidas por el sistema GRHS este presenta carencias asociadas a la visualización y localización de los recursos, tales como locales, agentes (computadoras) y las posibles incidencias asociadas a estos. Lo cual limita la supervisión y revisión de los medios de forma intuitiva, esto trae como consecuencia que la información referente a pérdida de dispositivos, desconexión o cambio de piezas no autorizadas ocasionalmente no sean percibidos por los responsables de un área determinada o no le sea dado el nivel de importancia que requiere.

Gracias a los progresos tecnológicos, han surgido diversos programas informáticos que permiten mejorar la relación hombre – máquina, de esta forma se perfecciona la interactividad del usuario con los contenidos que se pretenden dar a conocer, una variante que ofrece facilidades visuales para este tipo de situaciones es a través de la utilización de los gráficos en tercera dimensión (3D) por computadora.

El gráfico 3D por computadora es una potente herramienta que ayuda en el proceso cognitivo de las personas, la cual provee una gran variedad de riqueza visual, que no se consiguen en representaciones estáticas o en el papel. Hoy en día existen sistemas que utilizan este tipo de

tecnología para auxiliar el trabajo de diferentes áreas como la planificación, simulación de procesos, diseño, geolocalización y otros.

Poder representar espacialmente objetos permite enriquecer y fortalecer la apreciación de la realidad, de manera que la información del mundo real se asocie con la del mundo digital. La utilización de esta forma de representación visual es muy reconocida por las personas que lo utilizan, debido a que proveen mejor comprensión y aprendizaje de las situaciones modeladas, estimula la capacidad analítica y provoca una mejora en la rapidez con la que se concibe y se analiza la información.

Actualmente el sistema GRHS no cuenta con funcionalidades que brinden la posibilidad de localizar y visualizar espacialmente los recursos que gestiona de manera que se pueda lograr un mayor control de los medios. Esto impide la capacidad de combinar la información almacenada con la ubicación espacial del dispositivo físico que la representa y que se puedan definir e identificar regiones geográficas que poseen incidencias.

La problemática previamente descrita origina la necesidad de dar respuesta al siguiente **problema a resolver**: ¿Cómo localizar espacialmente los recursos que gestiona el sistema GRHS UCI?

Teniendo en cuenta el problema antes mencionado se define como **objeto de estudio**: las técnicas de modelado y visualización de gráfico por computadoras; centrándose en el **campo de acción** la localización espacial de los recursos que gestiona el sistema GRHS UCI.

Se propone como **objetivo general**: Desarrollar un sistema informático que permita localizar espacialmente los recursos que gestiona GRHS UCI.

Para el cumplimiento de los objetivos específicos se plantean las siguientes **tareas de la investigación**:

1. Análisis de las aplicaciones que utilicen modelado y visualización de gráfico 3D por computadoras y establecer semejanzas con la presente investigación.
2. Análisis de la información que muestra el sistema GRHS para seleccionar cuáles se mostrarán en el sistema a desarrollar.
3. Análisis de herramientas, tecnologías y metodologías necesarias para la implementación del sistema que permita localizar espacialmente los recursos de *hardware* y *software* de la UCI.
4. Análisis de los tipos de prueba que propone la metodología de desarrollo de *software* seleccionada para la posterior validación del sistema.

Se propone como **idea a defender**: El desarrollo de un sistema que utilice gráfico por computadoras, permitirá localizar espacialmente los recursos de *hardware* y *software* que gestiona el sistema GRHS UCI.

Durante el desarrollo de la investigación se utilizaron los métodos científicos que a continuación se detallan:

Métodos teóricos:

Modelación: Se utiliza para representar las estructuras y entidades, las cuales se visualizarán en el entorno 3D.

Analítico-sintético: Permite realizar el estudio teórico de la investigación y la extracción de los elementos más importantes acerca del funcionamiento y proceso de desarrollo de sistemas que utilizan la visualización de gráficos 3D por computadora.

Métodos empíricos:

Observación: Se utiliza para observar cómo se visualiza actualmente la información de los recursos que gestiona GRHS.

El presente trabajo de diploma está estructurado en un total de 3 capítulos:

Capítulo 1. Fundamentación Teórica: se realiza una descripción de elementos de la visualización de gráficos 3D por computadora, además se fundamenta la selección de la metodología, tecnologías y herramientas para el desarrollo del *software*.

Capítulo 2. Exploración y planificación: se presenta la propuesta de solución para el sistema y se realizan las fases de exploración y planificación de la metodología de desarrollo de *software*, así como los artefactos generados por cada una de estas fases.

Capítulo 3. Diseño, implementación y pruebas: se desarrollan las fases de diseño, implementación y pruebas de la metodología de desarrollo de *software* y se generan los artefactos correspondientes, se evidencian los resultados de las pruebas de aceptación para las historias de usuario.

Capítulo 1: Fundamentación teórica.

En el desarrollo de este capítulo se abordan los principales conceptos asociados al dominio del problema y se realiza un análisis de algunos sistemas que utilizan la visualización de gráficos 3D por computadora. Se definen las técnicas de modelado para la creación del entorno 3D. Además, se fundamenta la selección de la metodología de desarrollo de *software*, las tecnologías y las herramientas a utilizar.

1.1 Técnicas de modelado 3D

El modelado 3D es la representación de un objeto 3D que utiliza una colección de puntos en el espacio dentro de un espacio tridimensional, conectados por varias entidades geométricas tales como triángulos, líneas, superficies curvas y otras. Hoy en día, los modelos 3D son usados en una amplia variedad de campos (4):

- ✓ La industria médica usa modelos detallados de órganos; esto puede ser creado con múltiples partes de imágenes con dos dimensiones (2D).
- ✓ La industria del cine los usa como personajes y objetos para la animación.
- ✓ La industria de los videojuegos los utiliza como recurso para videojuegos.
- ✓ El sector científico los utiliza como modelos altamente detallados de componentes químicos.
- ✓ La industria de la arquitectura los utiliza para demostrar las propuestas de edificios y panoramas.
- ✓ La comunidad ingeniera lo utiliza para diseños de nuevos artefactos, vehículos, estructuras, entre otros usos.
- ✓ En décadas recientes la comunidad de las ciencias de la tierra ha empezado a construir modelos geológicos 3D como una práctica estándar. Los modelos 3D también pueden ser la base para los elementos físicos que son construidos con impresoras 3D.

Las técnicas de modelado 3D son variadas, algunas caen en desuso al poco tiempo de nacer debido a que solo sirven para fines muy concretos mientras que otras, son muy utilizadas debido a la solvencia que poseen (5). Las técnicas de modelado 3D no son algo estático, sino que están en constante evolución. Estas facilitan el modelado al convertirlo en un proceso organizado y preciso. Es posible establecer dos tipos de modelado dependiendo de si el modelado se centra en definir únicamente las características del contorno del objeto, estos son (5):

- ✓ **Modelado sólido:** también conocido como de Geometría Sólida Constructiva (*CSG Constructed Solid Geometry*). Los modelos sólidos definen el volumen del objeto que representan, en muchos casos indican incluso el centro de masa, la densidad del material interno, etc. Se utilizan en fabricación asistida por computadoras y en aplicaciones médicas e industriales.
- ✓ **Modelado de contorno:** también conocido como de Representación de Contorno (*B-Rep – Boundary Representation*). Los modelos de contorno únicamente representan la superficie límite del objeto. Son más fáciles de definir y modificar. Además, lo interesante

para la representación del objeto es su apariencia exterior (en los casos donde interesa el interior simplemente se aproxima, como en el caso de la dispersión del subsuelo). Prácticamente todos los paquetes de diseño y animación empleados en síntesis de imagen y en aplicaciones interactivas emplean este tipo de modelos.

A continuación, se ofrece una breve descripción de algunas de las técnicas de modelado consultadas en la bibliografía (4):

- ✓ **Modelado vértice por vértice** (*Verts to verts*): consiste en crear vértices y luego acomodarlos en las zonas que se crean correctas.
- ✓ **Modelado por rotoscopia**: es una técnica muy usada para modelar objetos bien detallados. Consiste en tener dibujos o referencias de frente, perfil y si es posible de arriba de un mismo elemento para crear caras y acomodar según lo que se vea en la imagen.
- ✓ **Modelado de caja** (Box modeling): es considerada la técnica reina. Se fundamenta a partir de una figura prediseñada sencilla (llamada primitiva) como puede ser un plano o un cubo, de la que se dispone de forma inmediata en el *software*; después se añade geometría en forma de vértices y caras que hacen que el volumen gane forma y detalles.
- ✓ **Modelado escultórico** (sculptmodeling): ha ganado mucho protagonismo en los últimos tiempos. También requiere comenzar con una figura primitiva, pero en este caso la geometría se añade para simular una presión, estiramiento, aplastamiento en la malla 3D, prácticamente como si se trabajara con arcilla. Esto hace que se generen nuevas caras a nivel interno y se modifique el volumen y la forma. Tiene la particularidad de que el resultado final es poco útil para determinados fines (animación, por ejemplo) y eso obliga a rehacer el objeto con modelado de caja con un proceso denominado retopología (*retopology*).
- ✓ **Superficies y curvas NURB** (*NURB surfaces y NURB curves*): es un tipo de modelado que cada vez se ve menos. Su verdadero potencial está en el control de determinados contornos del objeto a diseñar, tiene mucho potencial para diseño de coches, aviones y otros, siempre que lo que se busque sea un modelado exacto. Sin embargo, se debe tener en cuenta que la precisión matemática no es la finalidad deseada, ya que no se trata de un *software* CAD (diseño asistido por ordenador por su significado en español) aunque presenta ciertas ayudas en este sentido. Al final es muy habitual convertir el resultado en una malla donde ya no se distingue si fue creada con superficies y curvas NURB o por modelado de caja.
- ✓ **Otras técnicas como partículas y fluidos**: incluye prácticamente todo lo que se pueda imaginar respecto a cuerpos blandos, emisión de partículas, simulaciones de viento, humo entre otros. Generalmente estas técnicas se relacionan con la animación, pero también se pueden usar para crear imágenes fijas.

Se determina utilizar las técnicas de modelado de caja y de rotoscopia, la primera se ajusta al caso que ocupa la presente investigación donde se representarán edificios, docentes y otras áreas similares gestionadas por GRHS y la segunda para los mapas de las estructuras, para de esta manera obtener un mejor detalle y exactitud en el modelo. A pesar de que el resto de las técnicas

estudiadas presentan gran utilidad en varios campos del diseño 3D, no se adecuan a las necesidades del presente trabajo.

1.2 Visualización de gráficos 3D por computadora

En las últimas décadas se ha podido constatar un resurgimiento de la investigación centrada en la visualización, debido principalmente a dos razones (6). La primera de ellas tiene que ver con la presentación de conceptos, formas, relaciones y propiedades a través de nuevos elementos y entornos de aprendizaje propios del mundo altamente tecnológico actual. Estos avances tecnológicos se convierten en potentes herramientas matemáticas y científicas al inferir dinamismo a muchas entidades que antes eran presentadas por medio de tablas, fórmulas y símbolos. La segunda de las razones antes citadas está relacionada directamente con cambios en la concepción de la propia naturaleza de la matemática, según los cuales la matemática es entendida como una búsqueda de patrones y la visualización será una herramienta fundamental para reconocer esos patrones (7).

La principal característica atribuida a la **visualización** es que “ofrece un método de ver lo invisible”, tanto en la visualización como nombre (el producto, la imagen visual) o bien como verbo (el proceso, la actividad). Cuando se habla de “ver lo invisible”, en su sentido más profundo y figurativo, se refiere a percibir un mundo abstracto (8). Consiste en visualizar algo que solo sería posible en la imaginación humana.

Por su parte el término **gráficos 3D por computadora** se refiere a trabajos de arte gráfico que son creados con ayuda de *software* y programas especiales en general, o a las técnicas y tecnologías relacionadas con los gráficos de dimensiones altamente concentradas. (9) Los sistemas que implementan gráficos 3D por computadora son una potente herramienta en el proceso cognitivo de las personas, provee una gran variedad de riqueza visual que no se consiguen en representaciones estáticas en el papel. Esta nueva forma de apreciación de la realidad combina elementos reales y virtuales, aporta nuevos puntos de vista, aumenta el aprendizaje de la geometría espacial y facilita la toma de decisiones (10).

1.3 Sistemas con visualización de gráfico por computadora

En la actualidad existen innumerables sistemas que utilizan el gráfico por computadora debido a las ventajas que brinda esta disciplina o rama de la informática en la representación de objetos irreales, reales y aquellos que solo se pueden visualizar con técnicas (ópticas o electrónicas). Estos sistemas están principalmente orientados a la educación, la salud, el deporte, la cultura y el desarrollo de videojuegos, siendo este último actualmente una de las ramas que más ha explotado la visualización de gráficos por computadora en 3D (11).

En la bibliografía consultada se encontraron algunos sistemas que utilizan la visualización de gráficos 3D por computadora, los cuales se describen a continuación:

- ✓ **VisibleBody 3D:** esta es una aplicación del ámbito educativo, para ayudar a los estudiantes de medicina, profesionales sanitarios y el público en general en la visualización de las complejidades del cuerpo humano. (11)
- ✓ **El sistema RAIOM:** es un *software* del tipo marco de trabajo a nivel prototipo para el reconocimiento, detección, identificación y suministro de información de objetos tridimensionales y la visualización mediante el uso de realidad aumentada de información contextual provistas por sensores. (8)
- ✓ **SecondLife (en español Segunda vida):** es una plataforma social lanzada en junio de 2003, desarrollada por *Linden Lab* para la creación de un ambiente virtual en 3D donde usuarios de manera gratuita pueden acceder al mismo e interactuar con él y con los demás usuarios mediante un avatar. Permite además la creación de objetos e intercambiar diversidad de productos virtuales a través de un mercado abierto. (12)
- ✓ **El museo de la informática:** es un sistema de la Escuela Superior de Informática de la Universidad de Castilla-La Mancha que permite visualizar los diferentes equipos que han quedado en el paso de la historia de la informática. Los visitantes pueden recorrer la historia de la informática no sólo a través de los equipos 3D expuestos, sino accediendo a información de diversos tipos como documentos, fotografías, videos y noticias de prensa (13).

Luego de analizar las herramientas mencionadas, se determina que, aunque ninguna de ellas responde a la problemática en cuestión, se tendrán en cuenta algunas de sus características y principios de funcionamiento para ser aplicadas en el sistema a implementar. La navegación por el entorno 3D se realizará de forma similar a la utilizada por el sistema Secondlife y la forma de modelación de los dispositivos informáticos del museo de la informática. VisibleBody 3D y RAIOM fueron descartados debido a que presentan desarrollo privativo y fueron creadas con fines totalmente diferentes a los requeridos para la herramienta a desarrollar. Secondlife no se centra en la localización espacial de recursos y el museo de la informática no cuenta con un ambiente de navegación 3D.

1.4 Metodología de desarrollo de *software*

La metodología *Extreme Programming* (XP) consiste en una programación rápida o extrema centrada en la simplicidad de las soluciones implementadas y versatilidad para enfrentar los cambios, se enfoca en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*, promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo (14).

Se selecciona esta metodología puesto que independientemente de ser una de las más exitosas en la actualidad, posee documentación abundante y el equipo cuenta con experiencia en su utilización, es especialmente adecuada para proyectos pequeños con requisitos imprecisos y muy cambiantes, en el desarrollo del presente sistema los requisitos tienden a cambiar frecuentemente conforme

avanza el trabajo, el equipo es pequeño formado por una pareja de programadores donde la comunicación con el cliente es satisfactoria. Esta metodología es eficiente, de bajo riesgo, flexible y reduce el tiempo de implementación, en el presente trabajo se cuenta con aproximadamente 4 meses para el desarrollo, por lo que es necesario enfocarse en la programación para obtener un producto funcional en un breve intervalo de tiempo.

1.5 Tecnologías de desarrollo y herramientas

Las tecnologías en el desarrollo de *software* son un conjunto de herramientas que facilitan la realización del análisis, diseño, desarrollo e implementación de *software*. Para implementar el presente sistema se ha seleccionado la plataforma para el desarrollo de videojuegos Unity 3D, C Sharp (C#) como lenguaje de programación, el entorno integrado de desarrollo (IDE *Integrate Development Environment* por sus siglas en inglés) MonoDevelop, como herramienta de modelado Blender y dentro de las herramientas de diseño se utiliza el programa de manipulación de imagen (*Image Manipulation Program* por sus siglas en inglés) GIMP.

1.5.1 Unity 3D

Unity 3D es una plataforma para el desarrollo de videojuegos 3D y 2D, se utiliza en varios de los proyectos del centro VERTEX perteneciente a la facultad 5 de la UCI por ser uno de los mejores motores de videojuegos en la actualidad. Dentro de las ventajas que presenta se encuentran, tener una versión libre de desarrollo lo cual posibilita la creación de aplicaciones sin la necesidad de pagar licencia, permite exportar para múltiples plataformas entre las que se destacan *Windows*, *Mac*, *IOS*, *Android* y la web, además dispone de una interfaz de desarrollo muy bien definida e intuitiva y un editor exclusivo para 3D y 2D con el cual se puede probar sin necesidad de compilar, lo que brinda una gran ventaja debido a que permite ver cómo se comporta el entorno, además el equipo de desarrollo posee experiencia y conocimientos en la creación de aplicaciones con dicho motor de videojuegos, debido a estas características se determina utilizar Unity 3D en su versión 5.0. para desarrollar el presente sistema.

1.5.2 Visual Paradigm

Dentro de la familia de las herramientas de ingeniería de *software* asistida por computadora (CASE *Computer Aided Software Engineering* por sus siglas en inglés), Visual Paradigm es una aplicación profesional que soporta el ciclo de vida completo del desarrollo de *software*, presenta licencia gratuita y comercial, es fácil de instalar, actualizar y es compatible entre sus distintas versiones (15). Se utiliza en este trabajo de diploma la herramienta en su versión 8.0, para modelar la base de datos (BD) y transformar los diagramas Entidad-Relación a tablas de BD, esto posibilita la creación y diseño del modelo de datos, así como el *script* de la BD utilizada en el desarrollo del sistema de visualización 3D.

1.5.3 Ambiente de desarrollo integrado

Un ambiente de desarrollo integrado IDE, es una aplicación que ofrece un conjunto de herramientas que facilitan el desarrollo de aplicaciones informáticas de distintos tipos, en uno o varios lenguajes. MonoDevelop es un entorno de desarrollo libre y gratuito creado por desarrolladores provenientes del Proyecto Mono, quienes basándose en el popular IDE SharpDevelop para *Windows* lo adaptaron al entorno Linux mediante el uso de las bibliotecas Gtk.6

Se utiliza en este trabajo de diploma el IDE MonoDevelop en su versión 4.2 ya que es uno de los IDEs de desarrollo que soporta Unity 3D como editor de texto, diseñado primordialmente para C# y otros lenguajes .NET, cuenta con soporte completo para GNU/Linux, *Windows* y *Mac* (16).

1.5.4 Lenguaje de programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos *hardware* y *software* existentes (17).

Para la implementación de la nueva solución se empleó C# como lenguaje de programación, por ser el que utiliza por defecto el IDE MonoDevelop. C# es un lenguaje orientado a objetos¹ desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

1.5.5 Sistema gestor de base de datos PostgreSQL

Un sistema gestor de BD es un programa de ordenador que facilita una serie de herramientas para manejar bases de datos y obtener resultados (información) de ellas. Además de almacenar la información se le pueden hacer preguntas sobre su contenido, obtener listados impresos, generar pequeños programas de mantenimiento, o ser utilizado como servidor de datos para programas más complejos realizados en cualquier lenguaje de programación (18).

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution* por sus siglas en inglés) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado, utiliza un modelo cliente/servidor y multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará su funcionamiento (19).

1.5.6 Herramienta de modelado 3D

Se determina la utilización de la herramienta Blender en su versión 2.72, este es un programa informático multiplataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos, es libre y de código abierto. El motor de videojuegos Unity 3D presenta gran vínculo con la herramienta debido a que permite la importación de modelados,

¹OO: Orientado a objetos.

animaciones entre otras funcionalidades. Actualmente es compatible con todas las versiones de *Windows*, *Mac*, *GNU/Linux*, *Solaris*, *FreeBSD* e *IRIX* (20).

1.5.7 Editor de imágenes

Se decide utilizar *GIMP* en su versión 2.6.8, este es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Forma parte del proyecto *GNU* y está disponible bajo la Licencia pública general de *GNU* y *GNU Lesser General Public License*. Es el programa de manipulación de gráficos disponible en más sistemas operativos (*Unix*, *GNU/Linux*, *FreeBSD*, *Solaris*, *Microsoft Windows* y *Mac OS X*, entre otros) (21).

1.6 Conclusiones parciales del capítulo

En este capítulo se realiza un análisis de algunos sistemas que emplean la visualización de gráficos 3D por computadora, de esta manera se determinan las herramientas y técnicas a utilizar en el sistema a implementar. Durante este proceso de análisis no se hallaron evidencias de sistemas de este tipo que mostraran información de los medios tecnológicos de una determinada entidad, por tanto, es necesaria la realización de un sistema con estas características para los recursos que gestiona el sistema *GRHS*. Se determina el empleo de la metodología de desarrollo de *software XP* a partir de las características del sistema a desarrollar y su conjunto de buenas prácticas. Se define el ambiente de desarrollo que permite establecer las herramientas y tecnologías para implementar la solución.

Capítulo 2: Exploración y planificación

En este capítulo se describe la propuesta de solución del sistema a desarrollar, se realizan las fases de exploración, planificación y diseño pertenecientes a la metodología XP y se generan los artefactos correspondientes a cada una de ellas. Se describen las características funcionales y no funcionales del sistema, se detallan las historias de usuario (HU), el plan de iteraciones en el que se mencionan las HU asociadas a cada iteración y el plan de entrega para definir las fechas de terminación de las versiones funcionales que serán entregadas al cliente.

2.1 Descripción del sistema a desarrollar

El sistema debe permitir a partir de la utilización de la visualización de gráficos 3D por computadora, la ubicación espacial de los recursos que gestiona el sistema GRHS. Los agentes y los locales de cada entidad estarán representados por estructuras físicas modeladas en el entorno 3D, esto permitirá la identificación de su localización espacial. También se mostrará la información referente a los locales, los agentes y sus incidencias.

El sistema utiliza la información almacenada por GRHS para asociarla a las estructuras físicas del entorno 3D, lo que permite que al seleccionar dichas estructuras estas muestren la información que les corresponde. Cuenta con una BD donde se almacenan sus configuraciones. El cliente interactúa con el sistema, el sistema realiza peticiones al servidor, este devuelve la información resultante de las peticiones, el sistema procesa la información enviada y la entrega al cliente. El acceso al sistema se realiza mediante un navegador con soporte para WebGL.

Webgl es una especificación estándar que está siendo desarrollada actualmente para mostrar gráficos 3D en navegadores web, permite mostrar gráficos en 3D acelerados por *hardware* (GPU) en páginas web, sin la necesidad de plug-ins (22). Esto posibilita correr el actual sistema compilado en Unity en un navegador sin necesidad de instalar ningún plug-in.



Figura 1 Propuesta del sistema.

2.2 Fase de exploración

En esta fase se precisan las historias de usuario y se determina su prioridad en correspondencia con la importancia que posee para el desarrollo del sistema. Al mismo tiempo el equipo de desarrollo

se familiariza con las herramientas, la tecnología y las prácticas a ser utilizadas. Posteriormente se fundamentan los elementos y artefactos generados en esta fase.

2.2.1 Características funcionales del sistema

En la metodología XP los requerimientos funcionales del sistema son modelados como HU, describen lo que se espera de cada funcionalidad y a su vez son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

Las HU generadas son las siguientes:

- **HU 1.** Configurar datos.
- **HU 2.** Gestionar entidades lógicas.
- **HU 3.** Mostrar docente.
- **HU 4.** Gestionar agentes de un local.
- **HU 5.** Gestionar los locales.
- **HU 6.** Administrar ordenamiento.
- **HU 7.** Navegar por el entorno 3D.
- **HU 8.** Mostrar la información de un local.
- **HU 9.** Mostrar la información de un agente.

2.2.2 Características no funcionales del sistema

Las características no funcionales son aquellos requisitos que se desean del sistema, señalan una restricción de este y son fundamentales en el éxito del producto, son propiedades o cualidades que el producto debe tener y normalmente se vinculan a requerimientos funcionales. Para el presente sistema se han determinado los siguientes requerimientos no funcionales:

Apariencia o interfaz externa: la aplicación debe estar diseñada con una interfaz de fácil uso, de forma tal que el usuario navegue sin dificultad alguna.

Usabilidad: se necesita una preparación previa, aunque no extensa para operar con el sistema. Se requiere un nivel medio de conocimientos de computación, aunque el manejo de la aplicación es sencillo y permite la fácil comprensión por el usuario.

Servidor

Hardware:

- ✓ 2GB de RAM o superior, microprocesador Intel Corei3 o superior.

Software:

- ✓ Servidor Apache 2.2.31 o superior.
- ✓ 150 GB de capacidad de disco duro.

Cliente

Hardware:

- ✓ 1GB de RAM o superior, microprocesador Intel Pentium 4 a 2.5 GHz.

- ✓ En buen estado los periféricos teclado y mouse, este último con clic central.
- ✓ Para Firefox sobre el sistema operativo *Windows* se requieren las siguientes versiones mínimas de tarjeta gráfica: NVIDIA ≥ 257.21 , ATI de AMD ≥ 10.6 y versiones de Intel de septiembre de 2010 o más reciente.

Software:

- ✓ Navegador web con soporte para JavaScript, OpenGL 2.0 u OpenGL ES 2.0, se recomienda el navegador Mozilla Firefox 43 o superior con la característica WebGL activada en la configuración about:config.
- ✓ Para Firefox sobre el sistema operativo *Windows* requiere *Windows* XP o superior.

2.2.3 Especificación de las historias de usuario

Las especificaciones de las HU se realizan con el objetivo de determinar cuáles son las de mayor importancia para el sistema a fin de poder realizar una correcta planificación de su implementación, cada HU es clasificada según su prioridad para el negocio en:

Alta: se le otorga a las HU que resultan fundamentales en el desarrollo del sistema. Estas son: Configurar datos, Gestionar entidades lógicas y Mostrar docente.

Media: se le otorga a las HU que resultan necesarias, pero no imprescindibles para el funcionamiento del sistema. Estas son: Gestionar agentes de un local, Gestionar los locales y Administrar ordenamiento.

Baja: se le otorga a las HU que son de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo. Estas son: Navegar por el entorno 3D, Mostrar la información de un local y Mostrar la información de un agente.

De igual modo se tiene en cuenta la dificultad y posible existencia de errores durante la implementación de cada HU, para ello se clasifica a cada una según el riesgo de su desarrollo:

Alta: cuando en la implementación de las HU se considera la posible existencia de errores que lleven a la inoperatividad del sistema. Las HU que se encuentran en esta clasificación son: Mostrar docente y Administrar ordenamiento.

Media: cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión. En el sistema poseen esta clasificación las siguientes HU: Configurar datos, Gestionar entidades lógicas, Gestionar agentes de un local y Mostrar la información de un agente.

Baja: cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto. Ellas son: Gestionar los locales, Navegar por el entorno 3D y Mostrar la información de un local.

Las HU son representadas mediante tablas con la estructura de la tabla 2.1, donde los puntos estimados representan días hábiles:

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Configurar datos.


Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 1
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 9	
Descripción: El administrador puede realizar asociaciones de entidades lógicas del sistema GRHS con los locales físicos del entorno 3D, definir el tipo de ordenamiento aplicado a un local determinado, establecer un intervalo de fechas para los datos que desea visualizar y guardar los cambios mencionados anteriormente.	
Observaciones: Debe existir conexión con el servidor.	
Prototipo de interfaz:	
	

Tabla 2.1 HU 1. Configurar datos.

Para consultar el resto de las HU ver [Anexos I.](#)

2.3 Fase de planificación

Durante esta fase se realiza una estimación del esfuerzo que conlleva la implementación de cada HU, se crea el plan de iteraciones y se acuerda la posible fecha de entrega del producto.

2.3.1 Estimación del esfuerzo de cada historia de usuario

Para medir el desempeño del proyecto se utiliza la estimación por puntos. Un punto es considerado como una semana de trabajo (5 días), donde los miembros del equipo de desarrollo trabajan sin interrupciones. En los casos donde se requiera solo un número de días inferior a una semana de trabajo se utilizará la fracción $x/5$ donde x representa el número de días necesarios y el 5 el número de días hábiles de trabajo en una semana. Los puntos fueron asignados teniendo en cuenta la complejidad de las HU y la experiencia de los programadores.

Historia de Usuario	Puntos de estimación
HU 1. Configurar datos.	9

HU 2. Gestionar entidades lógicas.	3
HU 3. Mostrar docente.	8
HU 4. Gestionar agentes de un local.	8
HU 5. Gestionar los locales.	5
HU 6. Administrar ordenamiento.	7
HU 7. Navegar por el entorno 3D.	5
HU 8. Mostrar la información de un local.	5
HU 9. Mostrar la información de un agente.	10

Tabla 2.2 Estimación del esfuerzo por cada HU.

2.3.2 Plan de iteraciones

Una vez identificadas las HU y estimado el esfuerzo dedicado a la realización de cada una, se efectúa la planificación de la etapa de implementación. En esta las HU serán desarrolladas y probadas en un ciclo iterativo, representadas en tareas de ingeniería.

En la tabla 2.3 se muestra el plan de iteraciones, donde se representa la duración de cada una de las tres iteraciones y las HU asociadas a cada una de estas.

Iteración	HU a implementar	Duración de la iteración
Iteración 1	HU 1. Configurar datos.	4 semanas
	HU 2. Gestionar entidades lógicas.	
	HU 3. Mostrar docente.	
Iteración 2	HU 4. Gestionar agentes de un local.	4 semanas
	HU 5. Gestionar los locales.	
	HU 6. Administrar ordenamiento.	
Iteración 3	HU 7. Navegar por el entorno 3D.	4 semanas
	HU 8. Mostrar la información de un local.	
	HU 9. Mostrar la información de un agente.	

Tabla 2.3 Plan de Iteraciones.

2.3.3 Plan de entregas

En el plan de entregas se establecen aquellas HU que serán creadas para cada versión del producto, así como las fechas en las que serán entregadas al cliente.

La tabla 2.4 muestra la planificación de las entregas establecidas para el desarrollo del sistema.

Fecha	Historia de Usuario	Versión
	HU 1. Configurar datos.	v0.1

Finalizada la iteración 1 20/03/2016	HU 2. Gestionar entidades lógicas.	
	HU 3. Mostrar docente.	
Finalizada la iteración 2 17/04/2016	HU 4. Gestionar agentes de un local.	v0.2
	HU 5. Gestionar los locales.	
	HU 6. Administrar ordenamiento.	
Finalizada la iteración 3 15/05/2016	HU 7. Navegar por el entorno 3D.	v1.0
	HU 8. Mostrar la información de un local.	
	HU 9. Mostrar la información de un agente.	

Tabla 2.4 Plan de Entregas.

2.4 Conclusiones parciales del capítulo

En este capítulo se han expuesto las principales características de la aplicación, en función de realizar un sistema capaz de cumplir con las expectativas del cliente. Se elabora el plan de entregas donde se indican las HU creadas para cada versión del producto y las fechas en las que se publican estas versiones. Se expone la planificación que se planteó el equipo de desarrollo, compuesta por iteraciones donde de forma incremental se implementará el sistema propuesto de acuerdo con la prioridad asignada a cada HU, de esta manera se proporciona una mejor planificación y optimización del tiempo necesario para desarrollar el sistema.

Capítulo 3: Diseño, implementación y prueba

En el presente capítulo se generan los artefactos correspondientes a las fases de diseño, implementación y prueba. En la etapa de diseño se describe el patrón de arquitectura a utilizar, se detalla el diseño de la BD y se crean las tarjetas CRC (Clase Responsabilidad Colaboración) las cuales reflejan cuales son las clases que tienen las mayores responsabilidades y sus relaciones con otras clases, también se mencionan y explican los patrones de diseño utilizados. En la etapa de implementación se exponen las tareas de ingeniería generadas por cada historia de usuario, con el propósito de facilitar su comprensión. En la etapa de prueba se comprueba el correcto funcionamiento del *software* y se realizan las pruebas pertinentes para garantizar la calidad del sistema y se exponen los resultados obtenidos.

3.1 Fase de diseño

En esta fase se realiza el diseño del sistema. Con las facilidades que brinda la metodología XP la estrategia de diseño es simple, con el menor número de clases y métodos posibles, lo que propicia su fácil implementación con menos tiempo y esfuerzo.

3.1.1 Patrón de arquitectura de *software*

En la actualidad se han perfeccionado los algoritmos de implementación a tal punto, que los códigos ya no son un problema para el desarrollo de *software*. El reto principal es el diseño de los sistemas, por ello los patrones arquitectónicos expresan la estructura fundamental de estos. Para el sistema a desarrollar el patrón de arquitectura de *software* es Modelo Vista Controlador (MVC) que se estructura en tres capas: Modelo, Vista y Controlador. En el **modelo** se encuentran las clases con funciones para consultar y transformar la información de la BD, la cual se transfiere hacia el **controlador** donde se aplica toda la lógica de negocio, se procesa y se envían los datos a la **vista**. En el sistema la **vista** está compuesta por objetos de tipo *gameobject* que se encargan de mostrar la información al usuario. En el **controlador** se encuentran aquellas clases que procesan las peticiones de los usuarios, realizan las operaciones con las clases del modelo y conforman la vista que será mostrada. En el **modelo** se encuentra la clase *Conexion* responsable de enviar y recibir información del servidor, procesarla y entregarla al controlador.

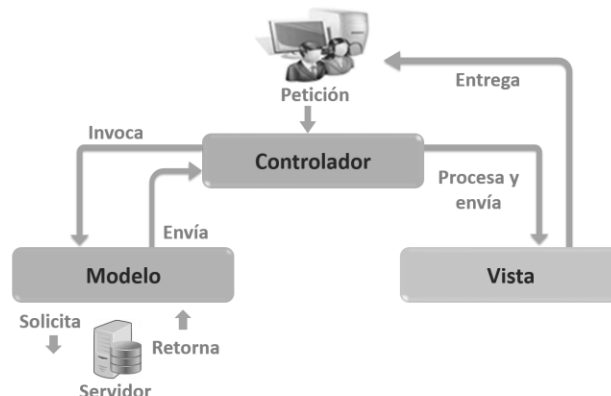


Figura 2 Funcionamiento del patrón MVC en el sistema.

3.1.2 Modelo físico de la base de datos

Para la implementación del sistema es necesario el uso de una nueva BD (Figura 3), utilizada para almacenar datos de configuración como son las relaciones entre los locales físicos del entorno 3D y las entidades lógicas del sistema GRHS, la información de las regiones de cada local y su configuración de ordenamiento para la asignación de posiciones a los agentes dentro del mismo.

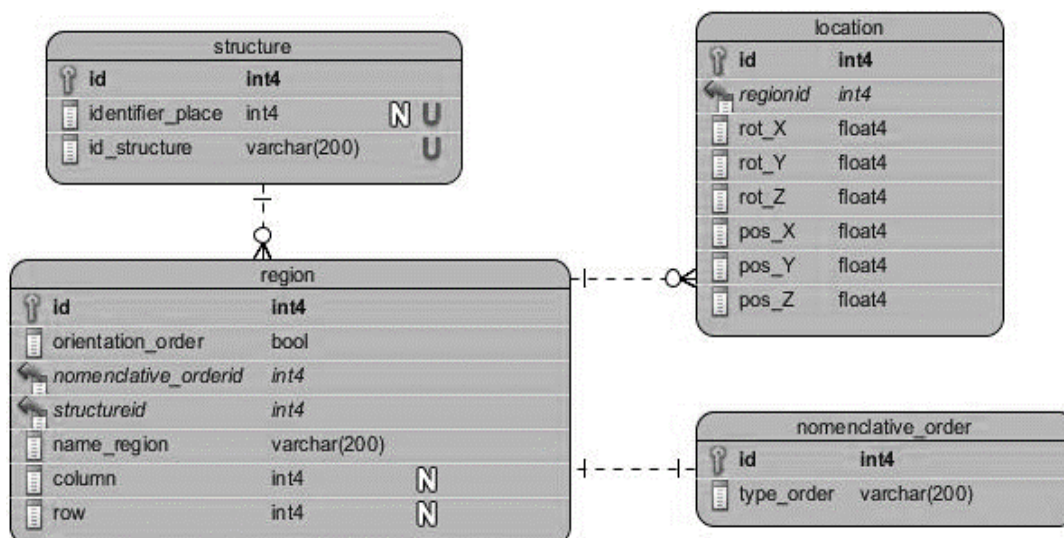


Figura 3 Modelo físico de la base de datos.

La tabla “*struture*” almacena los datos de un local en el entorno 3D, esta se relaciona de uno a mucho con la tabla “*region*”, la cual es utilizada para almacenar los datos de las regiones sobre las cuales se ubican los agentes de un local. La tabla “*location*” almacena los datos necesarios para ubicar un agente sobre una región y la tabla “*nomenclative_order*” se utiliza para estandarizar el nombre de los tipos de ordenamiento.

3.1.3 Tarjetas Clase-Responsabilidad-Colaborador

Una tarjeta CRC es en realidad una colección de tarjetas índices estándar que representan clases. Estas se dividen en tres secciones, generalmente el nombre de la clase se coloca en el borde superior en forma de título, en la parte derecha los colaboradores (clases que se implican en cada funcionalidad) y las responsabilidades (funcionalidades) de la clase en el extremo izquierdo (23).

Clase: Local	
Responsabilidad: Representa los locales físicos en el entorno 3D encargados de controlar los agentes en su interior.	Colaborador: Place Docente

Tabla 3.1 Tarjeta CRC 1. Local.

Para consultar el resto de las tarjetas CRC ver [Anexo III](#).

3.1.4 Patrones de diseño

Los patrones de diseño se han convertido en una técnica importante para el reuso del conocimiento de *software*. Cada patrón provee información sobre su diseño para describir las clases, métodos y relaciones que resuelven un problema de diseño en particular, estos han sido agrupados y organizados en catálogos cada uno para dar diferentes clasificaciones y descripciones (24).

Los patrones para asignar responsabilidades GRASP (*General Responsibility Assignment Software Patterns*, por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos en una aplicación. Se considera que más que patrones, son una serie de "buenas prácticas" recomendadas en el diseño de *software*. (25)

A continuación, se describen los patrones GRASP presentes en el sistema:

Creador: Debería ser el responsable de la creación de una nueva instancia de alguna clase, o sea la clase B es un creador de los objetos de la clase A. Este patrón se evidencia en las clases Parce_Agente y Parce_Places que son las encargadas de crear instancias de las clases Agente y Place respectivamente.

```
13     places = new Place[plac.Length / 3];
14     int ii = 0;
15     for (int i = 0; i < plac.Length; i = i + 3)
16     {
17         if (!plac[i].Equals(""))
18         {
19             places[ii++] = new Place(plac[i], plac[i + 1], plac[i + 2]);
20         }
21     }
```

Figura 4 Ejemplo del patrón creador

Alta cohesión: Mantiene la complejidad manejable, o sea asigna una responsabilidad de manera que la cohesión permanezca alta. La clase "Conexion" presenta la funcionalidad de cargarLocales y colabora con la clase Parce_Place (para comprobar que la información enviada desde el servidor es correcta y devolver los objetos de tipo Place) para obtener la información correctamente.

Controlador: Es el responsable de gestionar un evento de entrada al sistema, por tanto, asigna la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. Algunas de las clases que implementan el patrón Controlador son: Panel_AdministradorLocal, Panel_Agentes, Panel_Entidades, Local, Docente.

3.2 Fase de implementación

Para la implementación de las HU se chequea a cada una en conjunto con el plan de iteraciones, de esta forma inicia la creación de las tareas de ingeniería. Estas son para el uso estricto de los programadores y pueden ser escritas en lenguaje técnico. Teniendo en cuenta las iteraciones determinadas para la implementación del sistema se expone en la tabla 3.2 la tarea de ingeniería 1 correspondiente a cada HU 1.

3.2.1 Tareas de ingeniería

Las tareas de ingeniería son escritas por el equipo de desarrollo a partir de las historias de usuario elaboradas por el cliente. Las tareas describen a las historias de usuario, detallándolas a profundidad para facilitar su implementación, se estima un tiempo más cercano de la realidad para realizarlas. A continuación, se muestra la tarea de ingeniería 1 correspondiente a la HU 1, las restantes son expuestas en el [Anexo II](#).

Tarea de ingeniería.	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Crear la base de datos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se diseña y crea la BD. Para el diseño se seleccionan los datos que se van a almacenar, como son los referentes a las estructuras, las regiones y las localizaciones. Seguidamente se crea el diagrama entidad relación con la herramienta Visual Paradigm, después se exporta el código sql y se importa en el Pgadmin.	

Tabla 3.2 Tarea de ingeniería 1. Crear la base de datos

3.3 Fase de prueba

El proceso de pruebas es un elemento fundamental de la metodología XP, este reduce el número de errores no detectados, así como el tiempo entre su introducción en el sistema y su detección. Dicha metodología divide las pruebas en varios grupos entre los que se encuentran:

Pruebas de Aceptación (o pruebas funcionales): Son las pruebas destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

Pruebas Unitarias: Son las pruebas implementadas por los desarrolladores, encargadas de verificar el código.

3.3.1 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada historia de usuario, estas tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en XP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención. (26)

Estas pruebas deben su importancia, en especial, a que miden el nivel de satisfacción del cliente con cada iteración concluida, además de que marcan el final de esta y el comienzo de la próxima. A continuación se explican los elementos que componen un prototipo de caso de prueba, estos se describirán en tablas que contendrán los siguientes campos:

- ✓ **Clases válidas:** se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas válidas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ✓ **Clases inválidas:** se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las posibles entradas inválidas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado y cómo responde el sistema.
- ✓ **Resultado esperado:** se hará una breve descripción del resultado que se espera ya sea para entradas válidas o entradas inválidas.
- ✓ **Resultado de la prueba:** se hará una breve descripción del resultado que se obtiene.
- ✓ **Observaciones:** algún señalamiento o advertencia que sea necesario hacerle a la sección que está en prueba.

La evaluación de la prueba realizada se hará según el resultado de estas, los cuales se describen a continuación:

- ✓ **Satisfactorio:** cuando el resultado de la prueba es exactamente el esperado por el usuario.
- ✓ **No Satisfactorio:** cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, esto trae como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la HU.

En la **Tabla 3.3**, se muestra un ejemplo de la aplicación de estas pruebas para la HU1.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario selecciona guardar los cambios después de haber realizado alguno en la configuración del sistema		El sistema muestra un mensaje de confirmación "¿Quiere guardar los cambios realizados?"	Satisfactorio	Debe haber realizado algún cambio en la configuración
	El usuario selecciona guardar los cambios sin haber realizado alguno	El sistema muestra un mensaje de confirmación "¡No se han realizado cambios!"	Satisfactorio	No debe haber realizado cambios

	en la configuración del sistema			
El usuario selecciona guardar los cambios después de haber realizado alguno en la configuración del sistema y en la ventana con el texto de confirmación selecciona "Aceptar"		El sistema guarda los cambios realizados	Satisfactorio	
El usuario selecciona guardar los cambios después de haber realizado alguno en la configuración del sistema y en la ventana con el texto de confirmación selecciona "Cancelar"		El sistema no guarda los cambios	Satisfactorio	

Tabla 3.3 Prueba de aceptación HU1_P1

Para consultar el resto de las pruebas de aceptación ver [Anexo IV](#).

Las pruebas de aceptación se realizaron a las HU 1,2 y 3 en una primera iteración, las que arrojaron 3 no conformidades las cuales fueron corregidas en su totalidad, en la segunda iteración se aplicaron a las HU 4,5 y 6, en estas se encontraron 10 no conformidades que fueron resueltas, en la tercera y última iteración se aplicaron 7 pruebas correspondientes a las HU 7, 8 y 9, en este caso no se encontraron no conformidades. Estas pruebas han permitido validar la aplicación de manera que la calidad del *software* sea la requerida por el cliente, libre de errores que podrían causar insatisfacciones. En la **figura 5** se muestran los resultados de estas pruebas en sus tres iteraciones correspondientes.

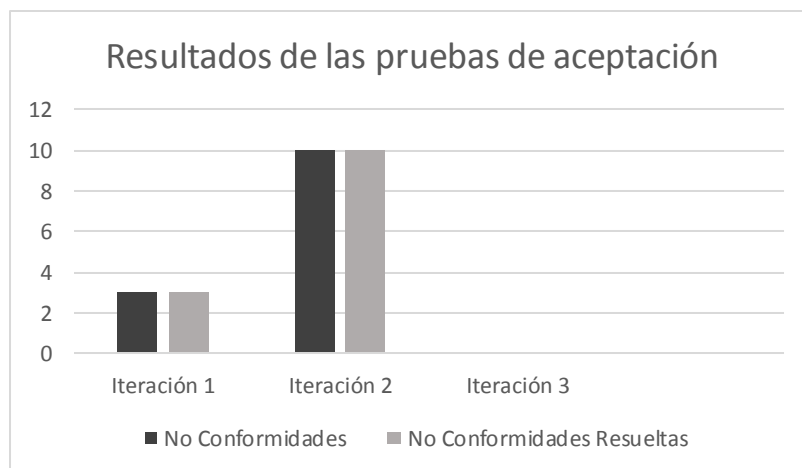


Figura 5 Resultados de las pruebas de aceptación

3.3.2 Pruebas unitarias

Se utiliza la herramienta UnityTestTools para validar las funcionalidades del sistema dentro de las que se encuentran: entrar con la cámara en un local seleccionado, activar o desactivar el interior de un local al entrar o salir con la cámara, desactivar todos los interiores de los locales al inicio de la aplicación, el movimiento hacia el interior del local hasta el punto de observación de este, se valida su rotación y posición.

A continuación, en la Figura 6 se muestra en el panel izquierdo inferior evidencias del resultado de las pruebas realizadas, en el panel izquierdo superior se encuentran las vistas pertenecientes a cada funcionalidad validada, el panel derecho superior representa el entorno 3D y el panel derecho inferior arroja información adicional de cada una de las pruebas realizadas.

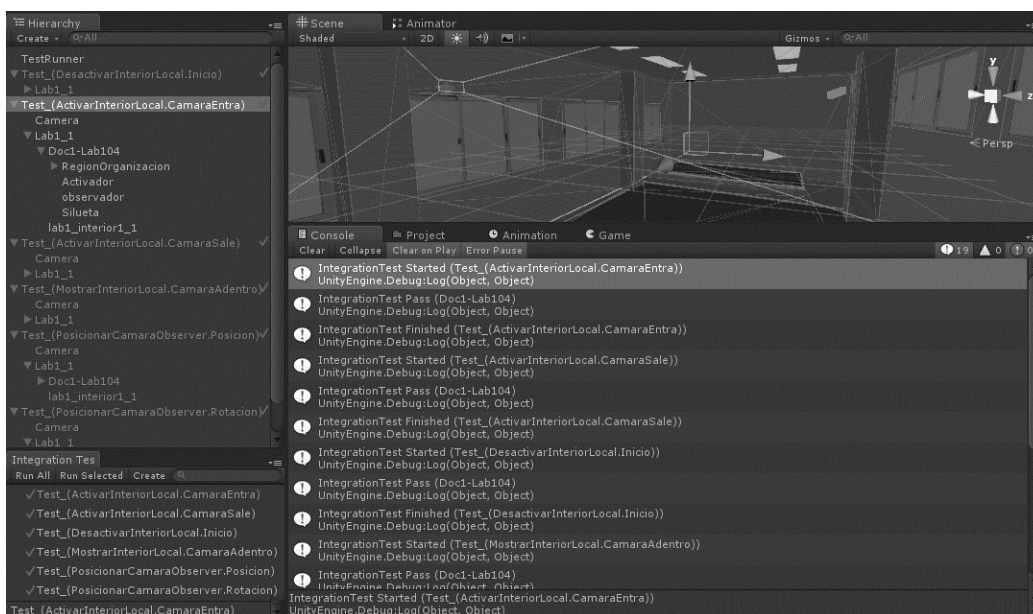


Figura 6 Resultado de las pruebas unitarias.

3.4 Conclusiones parciales del capítulo

Con el desarrollo del capítulo se elige la arquitectura de *software* MVC por sus características determinantes en la implementación del sistema, así como el uso de patrones de diseño que contribuyen a la ejecución de buenas prácticas en el desarrollo de este. Se implementaron las tareas de ingeniería en correspondencia con los requerimientos funcionales descritos, lo que permite obtener una primera versión funcional del producto. Una vez realizadas las pruebas se muestra una gráfica con los resultados obtenidos en las tres iteraciones de las pruebas de aceptación, así como la evidencia ilustrada de las pruebas al código mediante la utilización de la herramienta UnityTestTools, lo que garantiza la calidad del producto desarrollado.

Conclusiones generales

En el desarrollo de la investigación realizada se analizaron sistemas que utilizan la visualización de gráficos 3D por computadora, lo que brindó un mejor entendimiento de esta disciplina o rama de la informática, no obstante, estos sistemas fueron creados con fines totalmente diferentes a los requeridos para la herramienta a desarrollar, sin embargo, se tomaron aportes para incluir en la solución propuesta. Para desarrollar el sistema se seleccionó un conjunto de herramientas que propiciaron el buen desempeño del equipo en el proceso del *software*, así como el empleo correcto de la metodología XP brindó un conjunto de procedimientos que guiaron el desarrollo hacia un producto con las características deseadas por el cliente. La realización de las pruebas permitió validar el correcto funcionamiento del sistema, lográndose un producto final para el apoyo a la toma de decisiones que facilita la búsqueda óptica de agentes con incidencias. Este permitirá que usuarios autorizados sin noción de la estructura física de su entidad puedan utilizar el sistema y lograr ubicarse rápidamente en su objetivo, lo que favorece la detección de violaciones ocurridas, la identificación de regiones de mayor concurrencia de incidencias y el apoyo a la planificación del sistema de seguridad. Se podrá contar con un sistema que permite localizar espacialmente los recursos que gestiona el sistema GRHS para un mayor control visual de estos. Podrá combinar la información almacenada con la ubicación espacial de las estructuras a las que pertenece dicha información, lo que favorece la capacidad de análisis ante situaciones extraordinarias que se presenten debido al mal uso de las tecnologías.

Recomendaciones

Se propone continuar con el desarrollo de la aplicación a partir de la incorporación de funcionalidades tales como:

- ✓ La construcción de estructuras virtuales dentro del sistema.
- ✓ La representación de las incidencias en gráficas dentro del entorno 3D.
- ✓ La notificación de zonas de altas concentraciones de incidencias.

Referencias

1. **EUROINNOVA FORMACIÓN S.L.** El Tiempo. [En línea] [Citado el: 4 de noviembre de 2015.] <http://www.eltiempo.com/archivo/documento/MAM-1532000>.
2. **Fernández , A.** *Diccionario de Términos Económicos Contables y Agroeconómicos.* s.l. : Universidad de Oriente, 2005.
3. **RESOLUCION No. 204 / 96, capítulo IV, artículo 19.** *RESOLUCION No. 204 / 96, capítulo IV, artículo 19.*
4. **EducaLab.** [En línea] http://descargas.pntic.mec.es/mentor/visitas/DemoModeladoBlender/111_tipos_de_modelado.html.
5. **ESI.** Escuela superior de informática. [En línea] <http://www.esi.uclm.es/www/cglez/fundamentos3D/02.02.Tecnicas.html>.
6. **Phillips, L M, Norris, S P y Macnab, J S.** *Visualization in mathematics, reading and science education.* 2010.
7. **Fernández Blanco, Teresa .** *La investigación en visualización y razonamiento espacial. Pasado, presente y futuro.* 2013.
8. **Arcavi, A.** The role of visual representations in the learning of mathematics. 2003, págs. 215-241.
9. **Arquigrafico.** [En línea] <http://www.arkigrafico.com/los-graficos-por-computadora-en-3d/>.
10. **Chirstou.** *Developing student spatial ability with 3-dimensional applications.* 2008.
11. **Hugo Ramón, y otros.** *El uso de los Entornos Virtuales 3D como una herramienta innovadora en propuestas educativas mediadas con tecnología.* Buenos Aires : s.n., 2013.
12. **Calonge Arias, Luisa Fernanda.** *USO DE LA REALIDAD AUMENTADA (RA) CON IMÁGENES EN 2D Y 3D COMO APOYO AL PROCESO DE ENSEÑANZA – PRENDIZAJE DEL SISTEMA ÓSEO HUMANO CON ESTUDIANTES DE GRADO QUINTO DEL COLEGIO SURORIENTAL DE BOSTON.* BOSTON : s.n., 2014.
13. **Museo 3D informática.** [En línea] museo.inf.uva.es/museo3d/.
14. **Canós, José H, Letelier , Patricio y Penad, M^a Carmen.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n., 2005.
15. **Visual Paradigm.** [En línea] [Citado el: 26 de 2 de 2016.] https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html.
16. **NOVEL Enterprise.** *MonoDevelop IDE.* 2005.
17. **definición.org.** [En línea] <http://definicion.org/lenguaje-de-programacion>.
18. **Elmasri, R y Navatthe, S.** *Fundamentos de Sistemas de Base de Datos.* 2007.
19. **PostgreSQL.** [En línea] [Citado el: 23 de 2 de 2016.] http://www.postgresql.org/es/sobre_postgresql.
20. **Blander Foundation.** *Herramienta Blender.* 2006.
21. **Kimball.** *Herramientas de diseño.* 2002.

22. unity3dtutorial. [En línea] 30 de 11 de 2015.
<https://unity3dtutorial.wordpress.com/2015/11/30/publicar-un-videojuego-de-unity-en-webgl/>.
23. Pressman, Roger S. *Software Engineering, a practitioner's approach*. . 2011. Vol. 7. ISBN 9780071267823.
24. Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. *ARQUITECTURAS DE SOFTWARE*. 2004.
25. RODRIGUEZ, J. *PATRONES DE DISEÑO Y FRAMEWORKS*. 2014.
26. Beck, K. *Extreme Programming Explained. Embrace Change*. s.l. : Pearson Education, 1999.

Bibliografía

1. **EUROINNOVA FORMACIÓN S.L.** El Tiempo. [En línea] [Citado el: 4 de noviembre de 2015.] <http://www.eltiempo.com/archivo/documento/MAM-1532000>.
2. **Fernández , A.** *Diccionario de Términos Económicos Contables y Agroeconómicos.* s.l. : Universidad de Oriente, 2005.
3. **RESOLUCION No. 204 / 96, capitulo IV, articulo 19.** *RESOLUCION No. 204 / 96, capitulo IV, articulo 19.*
4. **EducaLab.** [En línea] http://descargas.pntic.mec.es/mentor/visitas/DemoModeladoBlender/111_tipos_de_modelado.html.
5. **ESI.** Escuela superior de informática. [En línea] <http://www.esi.uclm.es/www/cglez/fundamentos3D/02.02.Tecnicas.html>.
6. **Phillips, L M, Norris, S P y Macnab, J S.** *Visualization in mathematics, reading and science education.* 2010.
7. **Fernández Blanco, Teresa .** *La investigación en visualización y razonamiento espacial. Pasado, presente y futuro.* 2013.
8. **Arcavi, A.** The role of visual representations in the learning of mathematics. 2003, págs. 215-241.
9. **Arquigrafico.** [En línea] <http://www.arkigrafico.com/los-graficos-por-computadora-en-3d/>.
10. **Chirstou.** *Developing student spatial ability with 3-dimensional applications.* 2008.
11. **Hugo Ramón, y otros.** *El uso de los Entornos Virtuales 3D como una herramienta innovadora en propuestas educativas mediadas con tecnología.* Buenos Aires : s.n., 2013.
12. **Calonge Arias, Luisa Fernanda.** *USO DE LA REALIDAD AUMENTADA (RA) CON IMÁGENES EN 2D Y 3D COMO APOYO AL PROCESO DE ENSEÑANZA – PRENDIZAJE DEL SISTEMA ÓSEO HUMANO CON ESTUDIANTES DE GRADO QUINTO DEL COLEGIO SURORIENTAL DE BOSTON.* BOSTON : s.n., 2014.
13. **Museo 3D informática.** [En línea] museo.inf.uva.es/museo3d/.
14. **Canós, José H, Letelier , Patricio y Penad, M^a Carmen.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n., 2005.
15. **Visual Paradigm.** [En línea] [Citado el: 26 de 2 de 2016.] https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html.
16. **NOVEL Enterprise.** *MonoDevelop IDE.* 2005.
17. **definición.org.** [En línea] <http://definicion.org/lenguaje-de-programacion>.
18. **Elmasri, R y Navatthe, S.** *Fundamentos de Sistemas de Base de Datos.* 2007.
19. **PostgreSQL.** [En línea] [Citado el: 23 de 2 de 2016.] http://www.postgresql.org/es/sobre_postgresql.
20. **Blander Foundation.** *Herramienta Blender.* 2006.
21. **Kimball.** *Herramientas de diseño.* 2002.

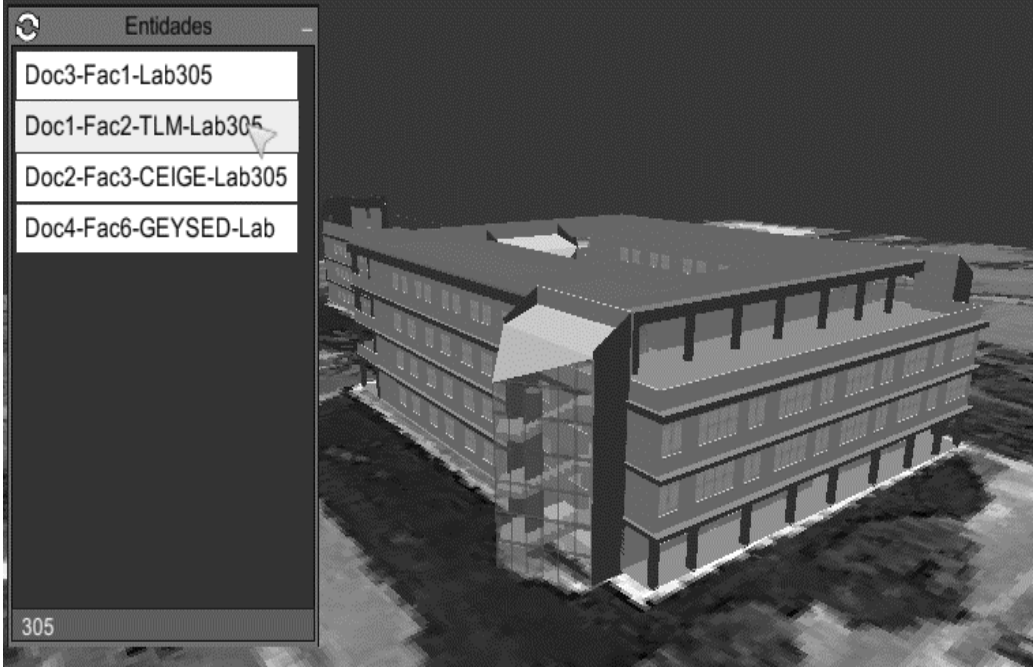
22. unity3dtutorial. [En línea] 30 de 11 de 2015.
<https://unity3dtutorial.wordpress.com/2015/11/30/publicar-un-videojuego-de-unity-en-webgl/>.
23. Pressman, Roger S. *Software Engineering, a practitioner's approach*. . 2011. Vol. 7. ISBN 9780071267823.
24. Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. *ARQUITECTURAS DE SOFTWARE*. 2004.
25. RODRIGUEZ, J. *PATRONES DE DISEÑO Y FRAMEWORKS*. 2014.
26. Beck, K. *Extreme Programming Explained. Embrace Change*. s.l. : Pearson Education, 1999.
27. González y Flores. 1997.
28. Gutiérrez, A. *Tendencias actuales de investigación en geometría y visualización*. 1998. Text of an invited conference in "Encuentro de Investigación en Educación Matemática" TIEM-98. Centre de Recerca Matemática.
29. Hershkowitz, R, Parzysz, B y Dormolen, J Van. *Space and shape*. London : International Handbook of Mathematics Education., 1996, págs. 161-204.
30. Battista, M T. *The development of geometric and spatial thinking*. s.l. : International Handbook of Mathematics Education., 2007, págs. 843-908.
31. Presmeg, N C. *An overarching theory for research in visualization in mathematics education*. [En línea] 16 de julio de 2008. <http://tsg.icme11.org/document/get/97>).
32. Azuma. 1997.
33. Ordoñez, Yoanni, y otros. *GRHS: Gestor de Recursos de Hardware y Software*. La Habana : s.n., 2014.
34. Hernández Pérez, Julio Antonio, Ordoñez Leyva, Yoanni y Aviles Vazquez, Ernesto. *GESTIÓN DE INCIDENCIAS EN INVENTARIOS DE RED*. 2014.
35. ISFTIC España. *CAPACIDAD DE INMERSIÓN DE LOS ENTORNOS VIRTUALES TRIDIMENSIONALES*. 2015.
36. Pressman, Roger S. *INGENIERÍA DE SOFTWARE. Un enfoque práctico.(quinta edición)*.Madrid. 5. : s.n., . . [ed.] Concepción Fernández . 2002. ISBN 0-079677-0.
37. Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico.(sexta edición)*. México : Editorial McGrawHill, 2006. Vol. 6.
38. Boeras Velázquez, Mairelys. *Aplicando el método de Boehm y Turner*. . [En línea] 15 de junio de 2012. <http://publicaciones.uci.cu> | <http://seriecientifica@uci.cu>. ISSN | RNPS.
39. Flores, Ervin. *Ingeniería de Software*. [En línea] 3 de abril de 2015.
http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.
40. Beck, K. *Extreme Programming Explained. Embrace Change*. . s.l. : Pearson Education, 1999.
41. Rodriguez, J. *PATRONES DE DISEÑO Y FRAMEWORKS*. ingenieriasw2. [En línea] 1 de abril de 2014. <http://ingenieriasw2.blogspot.com..>
42. Informatizate. [En línea] [Citado el: 20 de 11 de 2015.]
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html..

43. Gredos. [En línea] 2 de 2002. [Citado el: 8 de 2 de 2016.] <http://gredos.usal.es/jspui/handle/10366/21733>.
44. Mecanismos de extensión de PostgreSQL. [En línea] [Citado el: 23 de 2 de 2016.] <http://postgresql.uci.cu/wp-content/uploads/2014/11/Extendiendo-PostgreSQL.pdf>.
45. Sistema para la actualización de aplicaciones médicas. [En línea] [Citado el: 12 de 11 de 2015.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592012000200011.
46. Facorro, Juan Martín Muñoz. *Clojure IDE*. Buenos Aires : s.n., 2012.
47. Reyna, Rafael. Scribd. *Librería del mundo digital*. [En línea] Attribution Non-commercial, 23 de 5 de 2013. [Citado el: 6 de 4 de 2015.] <https://es.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
48. López, Denys López. *Desarrollo de una herramienta para el diseño gráfico de políticas de seguridad*. La Habana : UCI, 2012.
49. Viruslist. Viruslist.com. *Todo sobre seguridad en internet*. [En línea] 5 de 12 de 2013. [Citado el: 1 de 4 de 2015.] <http://www.viruslist.com/sp/analysis?pubid=207271238>.
50. Yeja, Ing. Adrian Hernández. *Estrategia para la detección de vulnerabilidades en aplicaciones web durante la fase de implementación del ciclo de desarrollo de software*. La habana : UCI, 2014.
51. Mecias, Lilian Teresa Castro. *Guía de gestión del riesgo tecnológico para el tratamiento de la seguridad durante el proceso de desarrollo de software*. La Habana : UCI, 2014.
52. Marta Castellaro, Susana Romaniz, Juan Carlos Ramos, Carlos Feck, Ivana Gaspoz. *Aplicar el Modelo de Amenazas para incluir la Seguridad*. Santa Fe : Universidad Tecnológica Nacional, 2008.
53. McGraw, Gary. *Software Security: Building Security In*. Boston : Addison Wesley Professional, 2006.
54. Ivar Jacobson, Grady booch, James Rumbaugh. *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, S.A, 2000.
55. Gómez, Miguel Ángel Amutio. *Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información*. Madrid : Ministerio de Hacienda y Administraciones Públicas, 2012.
56. José A. Mañas, Carlos Belso. *Gestión Dinámica de Riesgos: Seguridad de la Red de Servicios*. Madrid : Universidad Politécnica de Madrid, 2013.
57. Security, Department of Homeland. Build Security In. *Setting a higher standard for software assurance*. [En línea] Homeland Security, 18 de 9 de 2008. [Citado el: 20 de 4 de 2015.] <https://buildsecurityin.us-cert.gov/bsi/>.
58. Mellado, Daniel. *Un Proceso de Ingeniería de Requisitos de Seguridad en la Práctica*. Madrid : IEEE Latin America, 2007.
59. Paradigm, Vsual. Manual Online. [En línea] Visual Paradim, 23 de 12 de 2014. [Citado el: 8 de 4 de 2015.] <http://www.visual-paradigm.com/features/>.
60. Firesmith, Donald. *Security Use Cases*. U.S.A : Journal of Object Technology, 2003.
61. Microsoft. Aplicación de Microsoft amenaza Modeling Blog. *Análisis y modelado (TAM) v3.0*. [En línea] Microsoft Corporación, 20 de 7 de 2009. [Citado el: 18 de 5 de 2015.] <http://blogs.msdn.com/b/threatmodeling/archive/2009/07/20/threat-analysis-and-modeling-tam-v3-0-learn-about-the-new-features.aspx>.
62. P.F, Daniel. *Análisis y Modelado de Amenazas*. Asturias : -, 2006.

63. Laebel, Gabi. Practical Threat Analysis for Information Security Experts. *PTA Technologies*. [En línea] Eldan Software Systems Ltd., - de - de 2013. [Citado el: 15 de 5 de 2015.] <http://www.ptatechnologies.com/default.htm>.
64. NetBeans. Gembeta Dev. [En línea] 9 de 1 de 2014. [Citado el: 15 de 4 de 2015.] <http://www.genbetadev.com/herramientas/netbeans-1>.
65. Cuba. Ecured. *Enciclopedia Cubana*. [En línea] 14 de 2 de 2010. [Citado el: 10 de 4 de 2015.] http://www.ecured.cu/index.php/Lenguaje_de_programaci%C3%B3n_Java.
66. González, Héctor Suárez. *Manual Hibernate*. - : Documentación de javaHispano, 2013.
67. Andalucía, Junta de. Marco de Desarrollo. *Hibernate*. [En línea] ©Junta de Andalucía, 1 de 3 de 2013. [Citado el: 19 de 5 de 2015.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97>.
68. Álvarez, Cecilio. Genbeta:dev. *Desarrollo y Software*. [En línea] -, 12 de 8 de 2014. [Citado el: 12 de 4 de 2015.] <http://www.genbetadev.com/frameworks/que-es-spring-framework>.
69. Leibys Oria Pardo, José Miguel Noa Cobas. *Herramienta para el Análisis y Modelado de Amenazas en sistemas informáticos*. La Habana : UCI, 2012.
70. Gestión de proyectos y desarrollo de software. *Desarrollo de software. Tarjetas CRC*. [En línea] Jummp, 10 de 1 de 2012. [Citado el: 12 de 5 de 2015.] <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
71. Microsoft. Developer Network. *Revisiones de código y estándares de codificación*. [En línea] Microsoft, - de - de 2003. [Citado el: 2 de 5 de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
72. Cukerman, Diego. *Testing en eXtreme Programming*. Uruguay : Universidad de la República - Uruguay, 2006.
73. Junit. [En línea] 4 de 12 de 2014. [Citado el: 25 de 5 de 2015.] <http://junit.org/>.
74. Lorena Sanchez . Que es Microsoft Visio y sus Características. [En línea] 13 de 1 de 2013. [Citado el: 2 de 6 de 2015.] <http://lorehidal.blogspot.com/>.
75. Zepeda, Vianca Vega. *Ingeniería de Requerimientos para Productos Seguros*. Antofagasta : Unversidad Católica del Norte, 2006.
76. IDEF0 Integrated Definition Methods. *IDEF0*. [En línea] KBSI Community, 2010. [Citado el: 1 de 6 de 2015.] <http://www.idef.com/IDEF0.htm>.
77. Martinez, Rafael. PostgreSQL-es. *Sobre PostgreSQL*. [En línea] Creative Commons, 2 de 10 de 2010. [Citado el: 20 de 5 de 2015.] http://www.postgresql.org.es/sobre_postgresql.
78. CubaDebate. [En línea] 17 de septiembre de 2015. <http://mesaredonda.cubadebate.cu/temas/ciencia-y-tecnologia-temas/2015/09/17/avances-en-la-informatizacion-de-la-sociedad-cubana/>.
79. Mena Rodríguez, Adrian E, y otros. Informaticahabana. [En línea] <http://www.informaticahabana.cu/sites/default/files/ponencias/SEG14.pdf>.
80. Portal de la Universidad de las Ciencias Informáticas. [En línea] www.uci.cu.
81. Canós, José H, Letelier, Patricio y Penad, M^a Carmen. *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n.
82. Tinoco Gómez, Oscar , Rosales López, Pedro Pablo y Salas Bacalla, Julio . *Criterios de selección de metodologías de desarrollo de software*. 2010. ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico) .

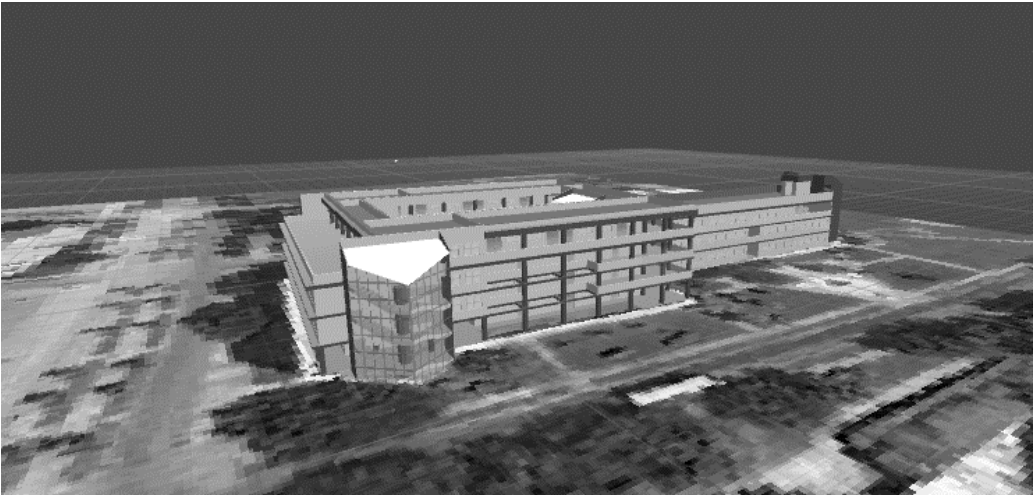
Anexo I

HU 2. Gestionar entidades lógicas.

Historia de Usuario	
Número: 2	Nombre de Historia de Usuario: Gestionar entidades lógicas.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 1
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 3	
Descripción: El administrador puede cargar, listar y filtrar las entidades lógicas existentes.	
Observaciones: Debe existir conexión con el servidor y deben existir entidades creadas.	
Prototipo de interfaz:	
 <p>The screenshot shows a Blender interface. On the left, there is a panel titled 'Entidades' with a list of four items: 'Doc3-Fac1-Lab305', 'Doc1-Fac2-TLM-Lab305', 'Doc2-Fac3-CEIGE-Lab305', and 'Doc4-Fac6-GEYSED-Lab'. A mouse cursor is pointing at the second item. Below the list is a search bar containing the number '305'. To the right of the list is a 3D model of a multi-story building with a modern architectural style, featuring a glass facade and a flat roof with several chimneys or ventilation stacks.</p>	

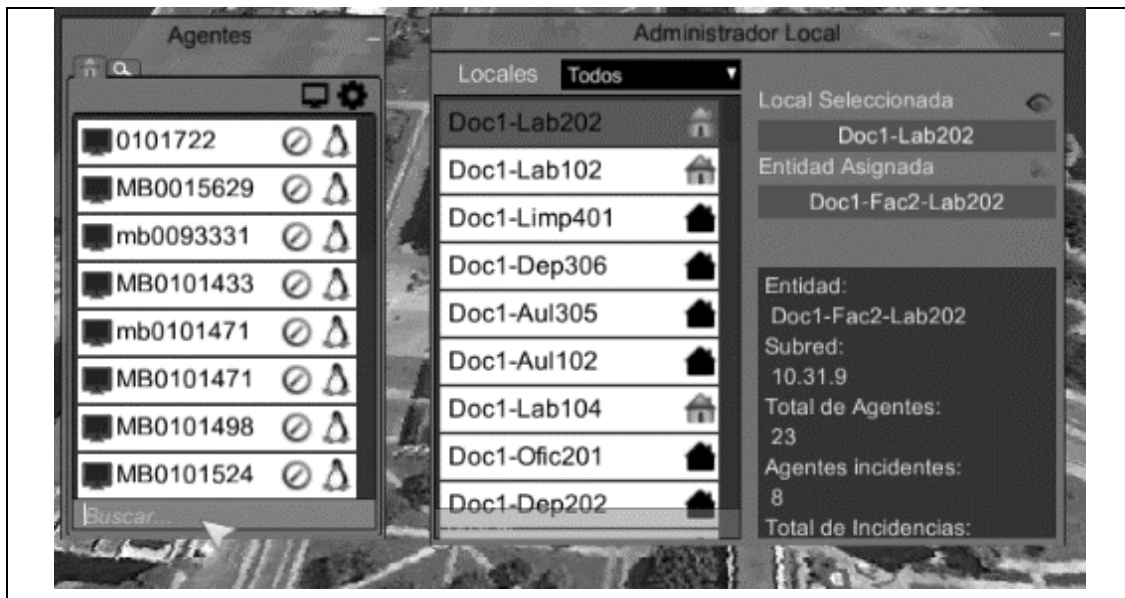
HU 3. Mostrar docente.

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Mostrar del docente 1
Modificación de Historia de Usuario Número: Ninguna	

Usuario: Administrador	Iteración Asignada: 1
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 8	
Descripción: El administrador puede visualizar el docentes 1 de la UCI en un entorno 3D donde puede navegar y visualizar la ubicación de los locales y los agentes.	
Observaciones:	
Prototipo de interfaz:	
	

HU 4. Gestionar agentes de un local.

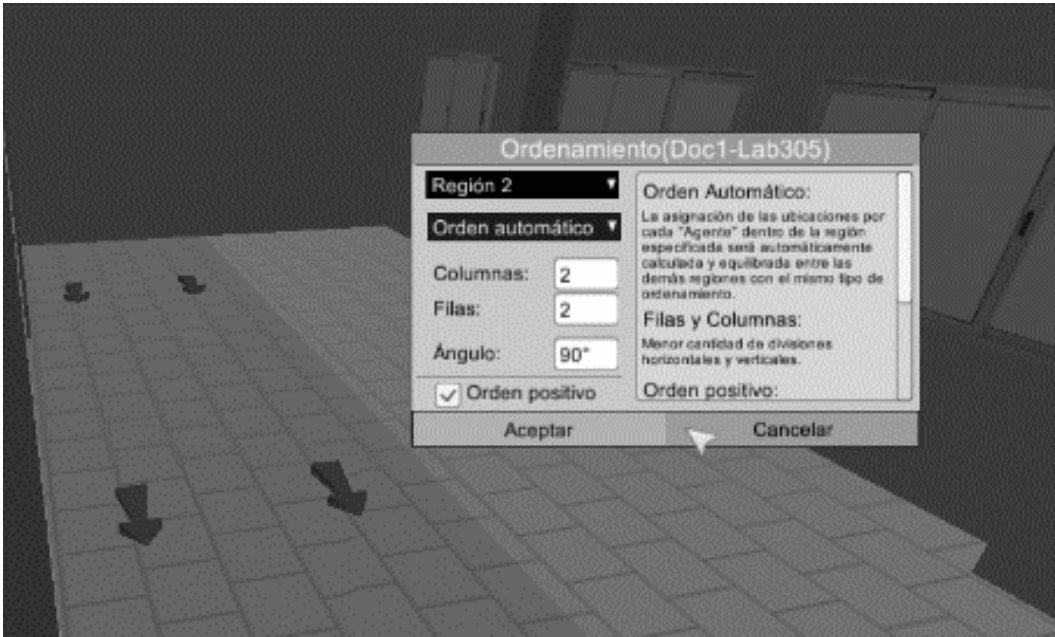
Historia de Usuario	
Número: 4	Nombre de Historia de Usuario: Gestionar agentes de un local
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 2
Prioridad en negocio: Media	Riesgo en Desarrollo: Medio
Puntos estimados: 8	
Descripción: El administrador puede cargar, listar, filtrar, ubicar en conjunto e individual los agentes de un local.	
Observaciones: Debe existir conexión con el servidor y deben existir entidades creadas.	
Prototipo de interfaz:	



HU 5. Gestionar los locales.

Historia de Usuario	
Número: 5	Nombre de Historia de Usuario: Gestionar los locales
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 2
Prioridad en negocio: Media	Riesgo en Desarrollo: Bajo
Puntos estimados: 5	
Descripción: El administrador puede cargar, listar, filtrar, los locales por dos criterios de búsqueda como son el nombre y la tenencia o no de incidencias.	
Observaciones:	
Prototipo de interfaz:	

HU 6. Administrar ordenamiento.


Historia de Usuario	
Número: 6	Nombre de Historia de Usuario: Administrar ordenamiento
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 2
Prioridad en negocio: Media	Riesgo en Desarrollo: Alto
Puntos estimados: 7	
<p>Descripción: El administrador puede seleccionar y modificar los valores del tipo de ordenamiento para cada región en cada local. Puede seleccionar uno de los siguientes ordenamientos: automático o manual. En el ordenamiento automático el sistema define las ubicaciones para cada uno de los agentes y el usuario puede establecer el ángulo en el que se verán y el orden a partir del que comenzarán a ubicarse los agentes.</p> <p>En el ordenamiento manual el usuario puede crear y eliminar las ubicaciones para los agentes, así como modificar su posición y rotación.</p>	
Observaciones:	
Prototipo de interfaz:	
	

HU 7. Navegar por el entorno 3D.

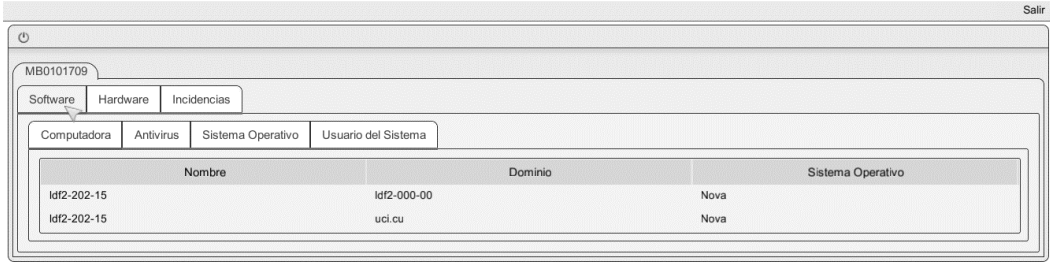
Historia de Usuario	
Número: 7	Nombre de Historia de Usuario: Navegar por el entorno 3D

Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 3
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos estimados: 5	
Descripción: El administrador puede moverse por el entorno 3D a través de los periféricos mouse y teclado.	
Observaciones:	
Prototipo de interfaz: No aplica.	

HU 8. Mostrar la información de un local.

Historia de Usuario	
Número: 8	Nombre de Historia de Usuario: Mostrar la información de un local
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 3
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos estimados: 5	
Descripción: El administrador puede visualizar información del local como son (nombre, subred, total de agentes, agentes con incidencias y total de incidencias)	
Observaciones:	
Prototipo de interfaz:	
	

HU 9. Mostrar la información de un agente.

Historia de Usuario	
Número: 9	Nombre de Historia de Usuario: Mostrar la información de un agente
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Administrador	Iteración Asignada: 3
Prioridad en negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 10	
Descripción: El administrador puede visualizar la información de los dispositivos de <i>hardware</i> , de los <i>software</i> instalados y las incidencias de un agente.	
Observaciones:	
Prototipo de interfaz:	
	

Anexo II

Tarea de ingeniería.	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Consultar la información de la BD_GRHS	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se seleccionan los datos que requieren ser cargados, de las bases de datos BD_GRHS y BD_GRHS_3D.	

Tarea de ingeniería.	
Número Tarea: 3	Número Historia de Usuario: 1
Nombre Tarea: Guardar datos de configuración	

Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: En la interfaz visual se crean las vías para guardar datos en la BD.	

Tarea de ingeniería.	
Número Tarea: 4	Número Historia de Usuario: 2
Nombre Tarea: Gestionar entidades lógicas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se crean las vías para cargar, listar, filtrar y seleccionar las entidades lógicas.	

Tarea de ingeniería.	
Número Tarea: 5	Número Historia de Usuario: 3
Nombre Tarea: Diseñar los planos de la estructura.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se obtienen los planos digitalizados de la estructura a modelar.	

Tarea de ingeniería.	
Número Tarea: 6	Número Historia de Usuario: 3
Nombre Tarea: Modelar el docente.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se modela la estructura en la herramienta Blender. Para mejor rendimiento se modela el interior y exterior de cada local como objetos independientes.	

Tarea de ingeniería.	
Número Tarea: 7	Número Historia de Usuario: 3
Nombre Tarea: Importa en Unity el modelo del docente.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	

Descripción: Se importa el modelo de una estructura y se define para cada local las regiones de activación, las regiones de resaltado y se crea el observador.

Tarea de ingeniería.	
Número Tarea: 8	Número Historia de Usuario: 4
Nombre Tarea: Modelar los agentes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se modela el agente en la herramienta Blender.	

Tarea de ingeniería.	
Número Tarea: 9	Número Historia de Usuario: 4
Nombre Tarea: Gestionar agentes de una estructura seleccionada	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se crean las vías para cargar, listar, filtrar, resaltar en conjunto e individual, seleccionar y ubicar los agentes.	

Tarea de ingeniería.	
Número Tarea: 10	Número Historia de Usuario: 5
Nombre Tarea: Delimitar las regiones en los locales	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se crean las regiones de ordenamiento donde se ubicaran los agentes.	

Tarea de ingeniería.	
Número Tarea: 11	Número Historia de Usuario: 5
Nombre Tarea: Gestionar local	

Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se crean las vías para cargar, listar, filtrar, resaltar en conjunto e individual y seleccionar los locales.	

Tarea de ingeniería.	
Número Tarea: 12	Número Historia de Usuario: 6
Nombre Tarea: Gestionar ordenamiento	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se definen los tipos de ordenamiento para cada región en cada local. En el ordenamiento automático se definen la menor cantidad de filas y columnas en las que será dividida la región para el cálculo de las ubicaciones, también se tienen en cuenta el ángulo en el que se verán los agentes y el orden que define a partir de donde comienzan a ubicarse los agentes. Este tipo de ordenamiento recalcula las posiciones en caso de que la cantidad de agentes sea mayor que la cantidad mínima de ubicaciones de una región. El ordenamiento manual permite crear y eliminar ubicaciones, el tipo de orden que define a partir de donde comienzan a ubicarse los agentes, así como modificar posición y rotación manualmente. La cantidad de ubicaciones para este ordenamiento es fija por lo que no se recalcula.	

Tarea de ingeniería.	
Número Tarea: 13	Número Historia de Usuario: 7
Nombre Tarea: Movimiento en el entorno 3D	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se crean los mecanismos para mover la cámara mediante la utilización del teclado, también se crean funcionalidades para controlar la cámara mediante el mouse.	

Tarea de ingeniería.	
Número Tarea: 14	Número Historia de Usuario: 7
Nombre Tarea: Tooltip con información	

Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se crean los tooltip para mostrar información referente al local, agente o estructura seleccionada.	

Tarea de ingeniería.	
Número Tarea: 15	Número Historia de Usuario: 8
Nombre Tarea: Entrar en un local seleccionado	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se realiza el movimiento de la cámara hacia el local seleccionado.	

Tarea de ingeniería.	
Número Tarea: 16	Número Historia de Usuario: 8
Nombre Tarea: Mostrar la información de un local seleccionado	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: De la BD_GRHS se carga la información referente al local seleccionado y se muestra al entrar en el mismo.	

Tarea de ingeniería.	
Número Tarea: 17	Número Historia de Usuario: 9
Nombre Tarea: Entrar en un agente seleccionado	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	
Descripción: Se realiza el movimiento de la cámara hacia el agente seleccionado.	

Tarea de ingeniería.	
Número Tarea: 18	Número Historia de Usuario: 9
Nombre Tarea: Mostrar la información del agente	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Programador responsable: Michael Brug Santrayll, Raidel Ross Rodríguez	

Descripción: De la BD_GRHS se carga la información referente al agente seleccionado y se muestra.

Anexo III

Tarjeta CRC 2. Place.

Clase: Place	
Responsabilidad: Representa aquellas particiones lógicas en las que está dividida una entidad en la BD como laboratorios, oficinas o aulas.	Colaborador: Agente Local

Tarjeta CRC 3. Agente.

Clase: Agente	
Responsabilidad: Representa a los agentes lógicos dentro de la BD.	Colaborador: Place

Tarjeta CRC 4. Docente.

Clase: Docente	
Responsabilidad: Representa un docente como entidad físicas en el entorno 3D.	Colaborador: Local Conexion

Tarjeta CRC 5. Conexion

Conexion	
Responsabilidad: Permite la comunicación del sistema con el servidor.	Colaborador: Controladora_Entorno Docente

Tarjeta CRC 6. Controladora_Entorno

Controladora_Entorno	
Responsabilidad: Es la encargada de cargar la configuración inicial y la inicialización del entorno.	Colaborador: Global Conexion

Tarjeta CRC 7. Global

Global	
Responsabilidad: Posee métodos y variables que podrán ser llamados desde cualquier clase.	Colaborador: Controladora_Entorno

Anexo IV

➤ **HU 2.** Gestionar entidades lógicas.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
En la ventana entidades el usuario selecciona la opción de cargar		El sistema debe listar las entidades en el panel, si están asignadas debe mostrarlas con color de fondo oscuro de lo contrario de color claro	Satisfactorio	Debe haber al menos una entidad en la BD
En usuario escribe en el cuadro de texto cualquier combinación de letras, números o caracteres especiales		El sistema debe listar solo las entidades que contengan en el nombre la combinación de caracteres escritas por el usuario	Satisfactorio	Debe haber al menos una entidad en la BD

➤ **HU 4.** Gestionar agentes de un local.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
En la ventana agentes el usuario selecciona la opción mostrar los agentes (icono de la casita)		El sistema debe listar los agentes pertenecientes al local seleccionado en el panel. Por cada agente listado se muestra un icono de monitor de color rojo si tiene incidencias y verde en caso contrario, un icono de linux o de <i>windows</i> según el sistema operativo que posea y un	Satisfactorio	Debe haber seleccionado un local el cual a su vez debe estar asociado con una entidad y esta debe tener al menos un agentes

		icono que indica si está activo o inactivo dicho agente		
En la ventana agentes el usuario selecciona la opción buscar agente (icono de la lupa), escribe en el cuadro de texto cualquier combinación de letras, números y caracteres especiales		El sistema debe listar solo los agentes que contengan en el nombre la combinación de caracteres escritas por el usuario	Satisfactorio	Debe haber seleccionado un local el cual a su vez debe estar asociado con una entidad y esta debe tener al menos un agentes
El usuario para resaltar todos los agentes dentro de un local posiciona el mouse encima del icono del bombillo en el panel izquierdo superior		El sistema debe resaltar todos los agentes, los que tienen incidencias de color rojo y los que no tienen de color verde	Satisfactorio	La cámara debe estar dentro de un local, este debe poseer algún tipo de ordenamiento, debe estar asociado con una entidad y esta última debe tener al menos un agente
En la ventana agentes el usuario selecciona con el mouse uno de los agentes listados y la cámara está dentro del local seleccionado		El sistema debe resaltar el agente físico que representa al agentes seleccionado, si tiene incidencias lo resalta de color rojo sino de color verde	Satisfactorio	Debe haber seleccionado un local el cual a su vez debe estar asociado con una entidad, esta debe tener al menos un

				agente y el local debe tener algún tipo de ordenamiento
En la ventana agentes el usuario selecciona con el mouse uno de los agentes listados y la cámara está fuera del local seleccionado		El sistema debe resaltar el local donde está ubicado el agente seleccionado, si tiene incidencias lo resalta de color rojo sino de color verde	Satisfactorio	Debe haber seleccionado un local el cual a su vez debe estar asociado con una entidad, esta debe tener al menos un agente y el local debe tener algún tipo de ordenamiento
En la ventana agentes el usuario selecciona con doble <i>click</i> uno de los agentes listados y la cámara está fuera del local seleccionado		El sistema debe llevar la cámara hasta el local donde se está ubicado el agente seleccionado	Satisfactorio	Debe haber seleccionado un local el cual a su vez debe estar asociado con una entidad, esta debe tener al menos un agente y el local debe tener algún tipo de ordenamiento
	En la ventana agentes el usuario selecciona con doble <i>click</i> uno de los agentes listados, la cámara está dentro del local seleccionado y	El sistema muestra un mensaje de información "No hay espacio para ubicar el agente : (nombre agente)"	Satisfactorio	Debe haber seleccionado un local el cual a su vez debe estar asociado con una entidad, esta debe tener al menos un agente

	no hay espacio para ubicar el agente			
En la ventana agentes, el usuario selecciona la pestaña buscar (icono de la lupa) y escribe cualquier combinación de letras, números y caracteres especiales		El sistema debe listar solo los agentes que contengan en el nombre la combinación de caracteres escritas por el usuario. Por cada agente listado se muestra un icono de monitor, debe aparecer de color rojo si tiene incidencias y verde en caso contrario, un icono de linux o de <i>windows</i> según el sistema operativo que posea y un icono que indica si está activo o inactivo dicho agente	Satisfactorio	Debe existir al menos un agente que cumpla con el patrón de búsqueda
	En la ventana agentes, en la pestaña buscar (icono de lupa) al seleccionar con doble <i>click</i> a un agente listado y la entidad a la que pertenece no está asignada a ningún local	El sistema muestra un mensaje de un información "La Entidad del agente seleccionado no está asignada a un local"	Satisfactorio	Debe existir al menos un agente que cumpla con el patrón de búsqueda

➤ **HU 5.** Gestionar local.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
En usuario para ver el listado de locales		El sistema debe mostrar un listado de los locales, si no están	Satisfactorio	Debe existir al menos un local

despliega la ventana Administrar local		asociado con ninguna entidad se muestra un icono de una casita negra, si están asociado y tienen incidencias se muestra una casita con el techo rojo y si están asociado y no tienen incidencias se muestra una casita con el techo verde		
En la ventana Administrador local el usuario escribe en el cuadro de texto cualquier combinación de letras, números y caracteres especiales		El sistema debe listar solo los locales que contengan en el nombre la combinación de caracteres escritas por el usuario	Satisfactorio	Deben existir al menos un local que contenga en el nombre la combinación de caracteres escritas por el usuario
El usuario para resaltar todos los locales posiciona el mouse encima del icono del bombillo en el panel izquierdo superior		El sistema debe resaltar todos los locales, los que tienen incidencias con color rojo y los que no tienen con color verde	Satisfactorio	
En la ventana Administrador local el usuario selecciona con el mouse uno de los locales listados		El sistema debe resaltar la ubicación del local físico de color rojo si tienen incidencias y de color verde en caso contrario	Satisfactorio	
En la ventana Administrador local el usuario posiciona el mouse en el icono del ojo		El sistema debe resaltar el local físico de color rojo si tienen incidencias y de color verde en caso contrario	Satisfactorio	

El usuario para visualizar información referente al local, selecciona uno de los locales listados que tengan entidad asociada		El sistema en el panel de información de la ventana Administrador local debe mostrar la información del nombre, subred, total de agentes, agentes con incidencias y total de incidencias	Satisfactorio	Debe tener una entidad asignada
	El usuario para visualizar información referente al local, selecciona uno de los locales listados que no tengan entidad asociada	El sistema en el panel de información de la ventana Administrador local debe mostrar información de cómo realizar una asignación	Satisfactorio	
El usuario para asignar una entidad a un local selecciona un local en la ventana Administrar local y selecciona una entidad en la ventana Entidades y presiona el botón Asignar Entidad		El sistema debe asociar el local con la entidad seleccionada	Satisfactorio	El local no debe tener ninguna entidad asignada

➤ **HU 6.** Administrar ordenamiento.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario para establecer el tipo de ordenamiento de un local, en la ventana Agentes presiona el icono gestionar		El sistema debe mostrar la interfaz de Ordenamiento	Satisfactorio	Debe de estar dentro del local seleccionado

En la ventana ordenamiento el usuario para configurar el ordenamiento de un local, selecciona la región a configurar, el tipo de ordenamiento en automático, cantidad mínima de filas y columnas, ángulo y orden en que se van a ordenar los agentes		El sistema debe ubicar los agentes en el espacio disponible de la región, a través del cálculo automático de la cantidad de filas y columnas necesarias para la cantidad de agentes a ubicar	Satisfactorio	
En la ventana Ordenamiento el usuario para configurar el ordenamiento de un local, selecciona la región a configurar, el tipo de ordenamiento en manual y orden en que se van a ordenar los agentes		El sistema debe permitir crear, eliminar y mover posiciones, modificar el ángulo y el orden en que se ordenaran los agentes	Satisfactorio	
El usuario para crear una posición debe presionar Ctrl + click izquierdo		El sistema debe crear una nueva posición en la región seleccionada	Satisfactorio	Debe haber seleccionado una región
El usuario para eliminar una posición debe presionar la tecla Alt + click izquierdo en la posición deseada		El sistema debe eliminar la posición seleccionada	Satisfactorio	Debe haber al menos una posición
El usuario para rotar una posición debe presionar con el click izquierdo la posición, después presionar la tecla Shift y después mover el mouse horizontalmente hasta que quede en el ángulo deseado		El sistema debe permitir rotar la posición	Satisfactorio	Debe haber al menos una posición

El usuario para mover una posición debe mantener presionado el click izquierdo y después mover el mouse hasta la ubicación		El sistema debe permitir mover la posición a cualquier lugar dentro de la región seleccionada	Satisfactorio	Debe haber al menos una posición
--	--	---	---------------	----------------------------------

➤ **HU 7.** Navegar por el entorno 3D.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario para mover la cámara por el entorno debe presionar las teclas W, A, S, D		El sistema debe permitir mover la cámara hacia delante, detrás, izquierda y derecha según las teclas W, S, A, D respectivamente	Satisfactorio	
El usuario para mover el ángulo de la cámara debe presionar el click derecho y mover el mouse		El sistema debe permitir cambiar el ángulo de la cámara	Satisfactorio	
El usuario para acercar o alejar la cámara debe usar el scroll del mouse		El sistema debe permitir acercar la cámara con el scroll hacia delante y alejarla con el scroll hacia detrás	Satisfactorio	
El usuario para visualizar si un local posee incidencias debe deslizar el mouse por la estructura		El sistema debe resaltar los locales por donde pasa el mouse con incidencias de color rojo y de color verde en caso contrario	Satisfactorio	
El usuario para visualizar el nombre de un local debe deslizar el mouse por la estructura		El sistema debe mostrar el nombre del local por donde pasa el mouse	Satisfactorio	
El usuario para entrar en un local debe		El sistema debe entrar en el local seleccionado	Satisfactorio	

seleccionarlo al dar click sobre él				
-------------------------------------	--	--	--	--

- **HU 9.** Mostrar la información de un agente.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario para ver la información de <i>hardware</i> y <i>software</i> del agente debe dar click sobre alguno		El sistema debe mostrar la información de <i>hardware</i> y <i>software</i> del agente seleccionado	Satisfactorio	

Glosario de Términos

Para una mejor comprensión de las descripciones y explicaciones que se exponen a lo largo del documento es necesario conocer los principales conceptos que se abordarán:

- ✓ **Agente:** es una representación lógica de las computadoras inventariadas en el sistema GRHS.
- ✓ **Recurso:** son todos los componentes que gestiona el sistema GRHS; piezas, programas instalados o sistema operativo.
- ✓ **Componente:** pieza de computadora, programa instalado o propiedad del sistema operativo.
- ✓ **Cambio:** es la sustitución, adición o sustracción de un componente.
- ✓ **Incidencia:** es un cambio no autorizado.
- ✓ **Administrador:** persona encargada de controlar las computadoras en la entidad.
- ✓ **Cliente:** persona que utiliza las computadoras en la entidad.
- ✓ **Visualización en 3D:** este proceso consiste en todas las operaciones que conlleva el despliegue de un objeto tridimensional en un dispositivo gráfico de salida. Por lo general, el objeto es concebido como una serie de triángulos, que serán procesados por un motor gráfico 3D con soporte de *hardware* gráfico (11).