



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

**Título: Desarrollo de una aplicación web para facilitar la
interacción entre los usuarios de la herramienta Centros
Expositivos Virtuales (CEV) y el grupo de desarrollo de Paseos
Virtuales.**

Autor: Manuel Orlando García Perdomo.

Tutor: MSc. Lidiexy Alonso Hernández.

Cotutor: Ing. Minardo Gollún González López.

La Habana

2011

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al tutor Minardo Gollún González López y al Centro de Informática Industrial (CEDIN), de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Manuel Orlando García Perdomo.

Firma del Autor

MSc. Lidiexy Alonso Hernández.

Firma del Tutor

Ing. Minardo Gollún González López.

Firma del Tutor

Nombre y Apellidos: Minardo Gollún González López

Edad: 28 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Ciencias Informáticas

Categoría Docente: Profesor Instructor

E-mail: mgonzalezl@uci.cu

Graduado de la UCI, con cinco años de experiencia en el tema de Realidad virtual y jefe del proyecto Paseos Virtuales en el Centro de Desarrollo de Informática Industrial.

Nombre y Apellidos: MsC. Lidiexy Alonso Hernández

Edad: 31 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Lic. Ciencias de la Computación

Categoría Docente: Profesor Instructor

E-mail: lidiexy@uci.cu

Graduado en la Universidad Central de las Villas en el año 2003, Máster en Ciencias desde el 2009 y Profesor Instructor de la UCI, con 8 años de experiencia en el desarrollo de temas asociados con la realidad virtual.

Agradecimientos:

A Enier y Anacelia, por su apoyo incondicional e imprimirme fuerzas en los momentos más difíciles.

A mi tutor y cotutor por la ayuda brindada durante el desarrollo de la tesis.

A todos mis compañeros: Andy, Evert, Reinier, Fumero, Nidelso, Rufino, Tony, Jose, Frank, Ramiro y otros tantos, por estos magníficos 5 años de carrera.

A todos aquellos que contribuyeron, de una forma u otra, a mi formación como ingeniero y como persona.

Dedicatoria:

A mis padres, Silvia, Andrés y Manuel Antonio, por siempre estar a mi lado.

A mis abuelos, siempre los llevo presente donde quieran que estén.

A toda mi familia.

Resumen:

El desarrollo de un módulo de edición de escenas para una herramienta de creación de Centros Expositivos Virtuales supone un avance para el proyecto Paseos Virtuales del Centro de Informática Industrial, debido a la optimización que le brinda al proceso de crear un paseo virtual. Sin embargo, si se desea ampliar el alcance de esta herramienta con relación a la entrega de un producto final de calidad a un usuario determinado y realizar dicha acción de forma automatizada, se necesita de una aplicación donde los clientes puedan interactuar de cierta manera con cada elemento de sus Centros Expositivos Virtuales de forma interactiva junto al equipo de desarrollo.

A través de la realización de esta investigación, se desarrolló exitosamente una aplicación web a la cual los usuarios pueden acceder mediante el uso de un navegador Web. Dicha aplicación permite a clientes, de manera sencilla e intuitiva recrear la apariencia que desean de sus escenarios virtuales e interactuar con los elementos que lo van a conformar. Por otra parte, permitirá mediante un sistema de votación y comentario expresar criterios sobre la conformación de los modelos en tres dimensiones que formarán parte del paseo virtual final. De esta forma se producirá un flujo de comunicación entre cliente y equipo de desarrollo, que permite una mayor participación de ambas partes en la realización de un producto. Simplemente se vería al cliente como un integrante más del equipo.

Palabras clave: Aplicaciones enriquecidas de Internet, editor de escenas, paseos virtuales.

Índice:

Introducción	12
Capítulo 1.Fundamentación Teórica.	16
1.1 Tendencias y tecnologías para el desarrollo web.	16
1.2 Tecnologías de visualización 3D en la Web.....	30
1.3 Sistema de gestión de contenidos (CMS)	32
1.4 Herramientas de desarrollo.....	34
1.5 Consideraciones del capítulo.	34
Capítulo 2.Características del sistema. Análisis y Diseño.	35
2.1 Modelo del dominio.....	35
2.2 Soluciones Técnicas.....	35
2.3 Prototipo de interfaz.....	36
2.4 Especificación de los requisitos de software.	43
2.5 Definición de los Casos de Uso.	48
2.6 Diagrama de Casos de Uso.	48
2.7 Descripción textual de los Casos de Uso.....	49
2.8 Patrón "Modelo-Vista-Controlador".	56
2.9 Diseño.	56
2.10 Diagramas de clases del diseño.....	57
Diagrama de clases del diseño (Autenticar Usuario) ver Anexo 3.	57
2.10.1 Diagrama de clases del diseño (Gestionar Proyecto).	57
2.10.2 Diagrama de clases del diseño (Administrar Modelo 3D).....	58
2.10.3 Diagrama de clases del diseño (Administrar Comentario sobre modelo 3D).....	58
2.10.4 Diagrama de clases del diseño (Crear Escena).	59
2.10.5 Diagrama de clases del diseño (Visualizar modelo 3D).	59
2.11 Diagramas de secuencia.....	60

Diagrama de secuencia: Autenticar Usuario ver Anexo 4.....	60
2.11.1 Diagrama de secuencia: Gestionar Proyecto (Insertar Proyecto).....	60
2.11.2 Diagrama de secuencia: Gestionar Proyecto (Cargar Proyecto).....	61
2.11.3 Diagrama de secuencia: Gestionar Proyecto (Eliminar Proyecto).....	61
2.11.4 Diagrama de secuencia: Crear Escena (Cargar Imagen).....	62
2.11.5 Diagrama de secuencia: Crear Escena (Crear Región).	62
2.11.6 Diagrama de secuencia: Crear Escena (Insertar Área).....	63
2.11.7 Diagrama de secuencia: Crear Escena (Visualizar Modelo).	63
2.11.8 Diagrama de secuencia: Administrar Modelo (Insertar Modelo).....	64
2.11.9 Diagrama de secuencia: Administrar Modelo (Eliminar Modelo).	64
2.12 Diseño de la Base de Datos.....	65
2.13 Consideraciones del capítulo.	65
Capítulo3. Implementación y Validación del Sistema.	67
3.1 Modelo de implementación.....	67
3.2 Diagrama de Componentes.....	67
3.3 Modelo de Despliegue.....	68
3.4 Modelo de Prueba.....	69
3.5 Descripción de los casos de prueba de integración.....	69
3.5.1 Diseño de caso de prueba. CU Administrar modelo 3D.....	69
3.5.2 Diseño de caso de prueba. CU Visualizar modelo 3D.....	70
3.5.3 Diseño de caso de prueba. CU Crear Escena.....	70
3.5.4 Diseño de caso de prueba. CU Gestionar Proyecto.....	72
3.5.5 Diseño de caso de prueba. CU Realizar votación sobre modelo 3D.....	73
3.6 Consideraciones del capítulo.....	73
Conclusiones.....	74
Recomendaciones.....	75
Referencias Bibliográficas.....	76

Glosario de Términos.....	79
Anexos.....	80
Anexo1. Descripciones textuales de los casos de uso.....	80
Anexo2. Diseños de prueba.....	86
Anexo 3. Diagrama de clases del diseño (Autenticar Usuario).....	91
Anexo 4. Diagrama de secuencia: Autenticar Usuario.....	91

Índice de tablas:

Tabla 1.Requisitos Funcionales.	43
Tabla 2. Actores del Sistema.	48
Tabla 3. CU Administrar Modelo.	49
Tabla 4.CU Visualizar Modelo 3D.	51
Tabla 7.CU Crear Escena.	51
Tabla 8.CU Gestionar Proyecto.	53
Tabla 9.Realización Votación sobre modelo 3D.	55
Tabla 10.Diseño de caso de prueba. CU Gestionar modelo 3D.	69
Tabla 11. Diseño de caso de prueba. CU Visualizar modelo 3D.	70
Tabla 14.Diseño de caso de prueba. CU Crear Escena.	71
Tabla 15.Diseño de caso de prueba. CU Gestionar Proyecto.	72
Tabla 16. Diseño de caso de prueba. CU Realizar votación sobre modelo 3D.	73
Tabla 17.CU Autenticar Usuario.	80
Tabla 18.CU Gestionar Usuario.	80
Tabla 19.CU Visualizar galería de imágenes.	82
Tabla 20.Gestionar imagen.	82
Tabla 6.CU Administrar comentarios sobre modelo 3D.	84
Tabla 5.CU Interactuar con modelo 3D.	85
Tabla 21.Diseño de caso de prueba. CU Autenticar Usuario.	86
Tabla 22.Diseño de caso de prueba. CU Gestionar usuario.	87
Tabla 23.Diseño de caso de prueba. CU Visualizar galería de imágenes.	88
Tabla 24.Diseño de caso de prueba. CU Gestionar Imagen.	89
Tabla 12.Diseño de caso de prueba. CU Interactuar con el modelo 3D.	90
Tabla 13.Diseño de caso de prueba. CU Administrar comentario sobre modelo 3D.	91

Índice de figuras:

Ilustración 1. Modelo del dominio.....	35
Ilustración 2. Interfaz inicial de la aplicación.	37
Ilustración 3. Interfaz donde se gestionan los proyectos dentro de la aplicación.....	37
Ilustración 4. Interfaz destinada para conformar la escena dentro de la aplicación.	38
Ilustración 5. Interfaz donde se muestran los objetos tridimensionales dentro de la aplicación.....	39
Ilustración 6. Interfaz donde se muestran imágenes de proyectos realizados.....	39
Ilustración 7. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación.....	40
Ilustración 8. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación.....	41
Ilustración 9. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación.....	41
Ilustración 10. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación.....	42
Ilustración 11. Diagrama de Casos de Uso.....	49
Ilustración 12. Modelo Vista Controlador	56
Ilustración 13. Diagrama de Componentes.	68
Ilustración 14. Modelo de Despliegue.	69

Introducción

Desde sus inicios hasta el presente día, la Web ha permitido el flujo de comunicación global a una escala sin precedentes en la historia humana, todo puede ser compartido y diseminado, desde pensamiento y actitudes, hasta cultura, idioma, arte, literatura y negocio, lo que se desee con el menor esfuerzo y de una forma sorprendentemente rápida a cualquier rincón del planeta. Gracias a su carácter virtual, carece de las desventajas físicas como ocurre con la prensa escrita o con los libros en formato duro, la información puede ser buscada generalmente de forma sencilla y se puede obtener más rápido de lo que una persona pudiera acumular a través de viajes, correo, teléfono o cualquier otro medio de comunicación, incluso la combinación de los mismos. El hipertexto ha hecho posible el sueño de la humanidad completa, la concentración de todo el saber almacenado a través de los siglos en un mismo tiempo y lugar, al alcance de cualquier ser humano.

Específicamente para una empresa u organización, la Web representa una parte esencial, ya que constituye la primera imagen que los clientes van a conocer. De esto se derivan un conjunto de ventajas que se listan a continuación:

- Provee a clientes la información acerca de sus productos y servicios, y los mantiene actualizados a medida que se van desarrollando nuevos aspectos de ellos.
- Mediante encuestas publicadas dentro de su página web, puede obtener una respuesta de sus clientes actuales, para así abordar mejor sus inquietudes y sugerencias respecto al servicio que están recibiendo de su empresa u organización.
- Otorga información detallada y específica acerca de sus productos. Si su público desea conocer más sobre su organización, la página web es el mejor vehículo para lograrlo.

En la Universidad de las Ciencias Informáticas (UCI), particularmente en el Centro de Informática Industrial (CEDIN), en la línea de Imágenes y Videos Basados en Render¹, se está desarrollando una herramienta para la creación de Centros Expositivos Virtuales (CEV) que permitirá a un usuario, sin tener conocimientos previos de programación, diseñar un paseo virtual, pudiendo incorporar modelos diseñados en herramientas de diseño en tres dimensiones (3D) y editar el escenario a su gusto. Por tanto, si se desea ampliar el alcance de esta herramienta, con objetivo de entregar un producto final de calidad

¹ La generación de la imagen o imágenes en movimiento de modelos virtuales en 3D.

minimizando cualquier inconformidad del cliente, tanto en relación a los modelos 3D como a la distribución de los mismos dentro de la escena, resulta necesario proveer a este proyecto de una aplicación Web que solucione dicha situación planteándose como **problema a resolver** ¿cómo lograr que los clientes que usen la herramienta CEV puedan manipular o gestionar cada elemento de las escenas de sus paseos virtuales de forma interactiva?

Dado el problema científico detectado se traza como **objetivo general**, desarrollar una aplicación Web para la interacción entre los usuarios y el equipo de desarrollo de un Centro Expositivo Virtual. Como **objeto de estudio** se tiene a las aplicaciones Web para la manipulación de información y el intercambio de criterios entre desarrolladores e interesados, derivándose como **campo de acción** las aplicaciones Web para el manejo y presentación de elementos gráficos conjunto a la interacción entre desarrolladores e interesados. Dado el problema científico detectado se trazan como **tareas a desarrollar**:

- Elaboración del marco teórico relacionado con el desarrollo de aplicaciones Web con objetivos similares al de la presente investigación.
- Caracterización de herramientas de desarrollo de aplicaciones Web con objetivos similares.
- Identificación de las herramientas y lenguajes para el desarrollo de la aplicación.
- Realización de la captura de los requisitos del software.
- Realización de los modelos del Diseño del software.
- Realización del diseño gráfico.
- Realización del Diagrama de despliegue.
- Realización del Diagrama de Componentes.
- Realización de pruebas de Caja Negra.
- Desarrollar un servicio en la Web donde los interesados en la herramienta CEV puedan prediseñar la distribución espacial de sus elementos dentro de la escena.
- Desarrollar un servicio el cual muestre elementos gráficos de su paseo virtual final, permitiendo evaluarlos, es decir, que permita la interacción entre los usuarios y el equipo de desarrollo.

Esperándose como **posibles resultados**:

Una aplicación Web capaz de brindar el entorno propicio a todo aquel interesado en la herramienta CEV, con la cual de forma visual y sencilla puedan interactuar tanto con la aplicación, como con el equipo de desarrollo, donde puedan establecer los principales requisitos que debe poseer su herramienta.

Para el cumplimiento de estos objetivos se ponen en práctica varios **métodos y técnicas en la búsqueda y procesamiento de la información** como son:

Métodos teóricos:

- **Histórico lógico:** en la primera parte de la investigación se desarrollará un estudio del estado del arte de la problemática, además se analizarán las ventajas y desventajas del uso de una aplicación Web para dar cumplimiento a la misma.
- **Analítico – sintético:** permitirá la extracción de información de los elementos más importantes de cada documento que se analice. La inducción-deducción se utilizará durante toda la investigación, para llegar a conclusiones y hacer generalizaciones.

Métodos empíricos:

- **Observación:** Para observar y verificar los problemas que existen en la universidad por falta de una aplicación que permita crear escenarios bidimensionales en un entorno web.

El presente documento está compuesto por tres capítulos, a continuación se brinda un resumen de los temas tratados dentro de los mismos.

Capítulo 1. Fundamentación Teórica.

En el presente capítulo se brindará un panorama general asociado a las aplicaciones Web y las tendencias existentes para el desarrollo de las mismas en entornos virtuales. Se investigarán y analizarán las tecnologías, características y técnicas más usadas en la actualidad destinadas al desarrollo de dichas aplicaciones. Paralelamente se realizará un análisis de la metodología y lenguajes de programación a emplear, los sistemas de bases de datos, los servidores Web, así como del sistema de gestión de contenido que se empleará, brindando de esta forma una solución al problema científico de esta investigación.

Capítulo 2. Características del sistema. Análisis y Diseño.

En este capítulo se realizará la descripción de la solución propuesta con el objetivo de dar respuesta al problema planteado en el anterior capítulo. Se opta por la realización de un modelo de dominio al no presentarse claramente los procesos de negocio, con el único objetivo de lograr un mejor entendimiento del sistema a desarrollar. Se especificarán tanto los requisitos funcionales como no funcionales de la aplicación, así como también se mostrará el diagrama de casos de usos del sistema y se realizará la descripción de cada uno de estos, puesto que son los que guían el proceso de desarrollo de software con el fin de dar solución a cada uno de los requerimientos planteados. Se tratarán las diferentes arquitecturas de software existentes, haciendo énfasis en la arquitectura seleccionada para este sistema, de igual modo se hará con los patrones de arquitectura. Se confeccionará el diagrama de clases del diseño, los diagramas de secuencia y se mostrará una vista del diseño de la base de datos.

Capítulo 3. Implementación y Validación del Sistema.

En la fase de Implementación se traduce el diseño de clases, elaborado en la fase de Análisis y Diseño, en componentes físicos, los cuales se convierten en los ficheros correspondientes a la implementación de la aplicación. En este también se harán las pruebas correspondientes a los casos de uso descritos en el flujo de trabajo anterior.

Capítulo 1. Fundamentación Teórica.

1.1 Tendencias y tecnologías para el desarrollo web.

Aplicaciones Web.

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet. Son populares debido a la practicidad del navegador Web como cliente ligero, además de la facilidad de actualización sin distribuir e instalar software en miles de potenciales clientes. Una de las ventajas principales es que deberían funcionar igual, independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux y otros sistemas operativos, la aplicación es escrita una vez y generalmente es mostrada de la misma forma en todos. (DOCFORGE, 2010)

Ventajas:

- Desarrollo barato, sencillo y rápido.
- Acceso ubicuo, sin necesidad de distribución e idealmente, con pocos requerimientos técnicos.
- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Permiten el desarrollo de comunidades que dan valor a las aplicaciones (software social).
- Una empresa puede migrar de sistema operativo o cambiar el hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor.
- No se requieren complicadas combinaciones de hardware/software para utilizar estas aplicaciones. Solo un computador con un buen navegador Web.
- Actualizar o hacer cambios en el software es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión a Internet o a una red interna o privada. (IBM, 2006)

Aplicaciones de Internet Enriquecidas (RIA)

RIA son la nueva generación de las aplicaciones. Surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales. En los entornos RIA no se producen recargas de página, ya que desde el principio se carga toda la aplicación y sólo se produce comunicación con el servidor cuando se necesitan datos externos como: datos de una Base de Datos o de otros ficheros externos.

Consisten en el aprovechamiento de la experiencia del usuario en herramientas y funciones de escritorio tan naturales como copiar, cortar, pegar, redimensionar columnas y ordenar, con el alcance, flexibilidad de presentación y despliegue que ofrecen las aplicaciones o páginas Web junto con lo mejor de la multimedia (voz, vídeo, etc.).

Entre los beneficios principales de las aplicaciones RIA están:

- Importante mejora en la experiencia visual, permitiendo el fácil uso de las mismas.
- Mejoras en la conectividad y despliegue instantáneo, agilizando su acceso.
- Desvinculación de la capa de presentación; es decir, acceso a la aplicación desde un computador ubicado en cualquier lugar del mundo. (ADOBE, 2009)

Estándares de programación WEB.

Para la conformación de un sitio Web no basta con la simple utilización de un sistema de gestión de contenidos (CMS). En la actualidad existen disímiles tecnologías y lenguajes, los cuales han sufrido un largo período de evolución y desarrollo con el objetivo de permitir un óptimo funcionamiento y brindarle una agradable imagen al usuario a la hora de visualizar una página Web en un navegador determinado. A continuación se mostrarán algunos de los comúnmente más usados.

HTML (Lenguaje etiquetado de Hipertexto)

HTML es un lenguaje utilizado normalmente en la World Wide Web. Es muy sencillo y permite describir textos de forma estructurada y agradable, con enlaces, los cuales permiten conectar dos elementos entre sí o fuentes de información relacionadas. El mismo es un sistema de etiquetas basado en especificar la estructura lógica del contenido así como los diferentes efectos que se quieren dar, indicando cómo debe verse la información. En resumen, este lenguaje tiene como propósito fundamental definir la estructura y

aparición básica de documentos, de tal manera que puedan ser manejados de forma rápida y fácil por un usuario en red para verlos en distintos dispositivos. (W3C, 1995)

XML (Lenguaje de Marcas extensible)

XML es utilizado para definir lenguajes de estructuración de texto basados en marcas explícitas o tags, que son entendibles por una persona y que pueden ser interpretadas por un computador.

Los documentos XML se componen de unidades de almacenamiento llamadas entidades las cuales contienen datos. Este marcado codifica una descripción del diseño de almacenamiento del documento y la estructura lógica. Además soporta una amplia variedad de aplicaciones que se pueden generar de una forma rápida y su mantenimiento es mucho más sencillo. El mismo facilita la integración desde fuentes de datos heterogéneas, por ejemplo: páginas WEB, bases de datos y propone un estándar para el intercambio de información estructurada entre diferentes plataformas. (W3C, 2008)

CSS (Hojas de estilos en cascadas)

Las CSS se utilizan para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. Estas constituyen un lenguaje sencillo o plantillas que complementa el de HTML cuando se trata de modificar algunas etiquetas de las páginas Web ya que permiten una mayor precisión en el ajuste de los elementos del diseño. Esta técnica consiste en separar el diseño del contenido, de manera que las indicaciones para conformar el diseño se agrupan en una hoja de estilo o archivo que está fuera del contenido del documento de la página HTML. (W3C, 2010)

Lenguajes de programación Web.

Para el desarrollo de un sitio Web en la actualidad existen varios lenguajes de programación orientados a este fin, entre los más usados se encuentran:

Java

El lenguaje para la programación en Java, es un lenguaje orientado a objeto, de una plataforma independiente. Fue desarrollado por la compañía Sun Microsystems², con la idea original de implementar

² Empresa informática fabricante de semiconductores y software.

una máquina virtual y un lenguaje con una estructura y sintaxis similar a C++. El equipo reorientó la plataforma hacia la Web al percatarse de que Internet se convertiría en un medio interactivo.

El código generado por el compilador Java es independiente de la arquitectura; podría ejecutarse en un entorno UNIX, Mac o Windows. Esto es posible ya que el código generado por el compilador se ejecuta mediante una máquina virtual y por el procesador del ordenador directamente. Esto permite que los Applets³ de una Web pueda ejecutarlos cualquier máquina que se conecte a ella independientemente de qué sistema operativo emplee, siempre y cuando el ordenador en cuestión tenga instalada una máquina virtual de Java. (James Gosling, 2005)

PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Este es un acrónimo recursivo que significa *PHP Hypertext Pre-processor*. Es usado principalmente en interpretación del lado del servidor. Actualmente está siendo utilizado en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. (PHP, 2009)

Lenguaje seleccionado (PHP)

Para dar cumplimiento al objetivo planteado a inicio de esta investigación se optó por seleccionar PHP como uno de los lenguajes de programación a emplear durante el desarrollo de esta aplicación, teniendo en cuenta que los módulos de Joomla están desarrollados principalmente en dicho lenguaje, además de permitir la creación de aplicaciones Web muy robustas.

Principales ventajas de PHP:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.

³ Componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

Frameworks

Producto que sirve como base para la programación avanzada de aplicaciones, el cual aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar. Estos son utilizados por los programadores para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en éste ya existen implementaciones que están probadas, funcionan y no se necesitan volver a programar. (FRAMEWORKS, 2003)

jQuery

jQuery es una biblioteca o framework de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM⁴, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX⁵ a páginas Web. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

jQuery presenta un serie de características que hacen de este framework una herramienta poderosa (Resig, 2010), entre ellas destacan:

- Optimizado para la manipulación del DOM.
- Opera con varios elementos sin una sintaxis especial.

⁴ Interfaz de programación de aplicaciones para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos.

⁵ (JavaScript asíncrono y XML), técnica de desarrollo web para crear aplicaciones interactivas.

- Dirigido por selectores CSS.
- Método de encadenamiento que hace fácil escribir código.
- API⁶ y sintaxis intuitiva.
- Eventos.
- Efectos y animaciones.
- Animaciones personalizadas.
- Compatible con los navegadores Mozilla Firefox, Internet Explorer, Safari, Opera y Google Chrome.

Desventajas:

- Orientado a DOM en lugar de a JavaScript.
- Algunos desarrolladores tienen problemas en torno a "this" en las funciones de callback.

MooTools

MooTools (My oriented object tools) es un framework Web orientado a objetos para JavaScript, de código abierto, compacto y modular. El objetivo de del mismo es aportar una manera de desarrollar JavaScript sin importar en qué navegador se ejecute de una manera elegante. MooTools aporta una API documentada más enfocada a la orientación de objetos que la implementación estándar soportada por los navegadores web. (MooToolsAPI, 2009)

Éste aporta al usuario ventajas tales como:

- Framework modular y extensible, el desarrollador puede elegir que componentes usar y cuáles no.
- Orientado a objetos y sigue los principios DRY⁷, que hacen de él un framework rico, potente y eficiente.

⁶ Interfaz de programación de aplicaciones

⁷ Evita duplicar código abstrayendo las cosas que son comunes y poniéndolas en un solo sitio.

- Componente avanzado de efectos, con transiciones, de función parabólica, optimizadas y utilizadas por una gran cantidad de desarrolladores Flash.
- Framework desarrollado por programadores para programadores.

Desventajas:

- Sintaxis y API más pesadas que jQuery
- No es tan fácil como jQuery para actualizar el DOM

ExtJS

Es una biblioteca de JavaScript para el desarrollo de aplicaciones Web interactivas usando tecnologías y técnicas como AJAX, DHTML y DOM. Además de permitir la construcción de aplicaciones enriquecidas en Internet o RIAs. (SENCHA, 2011) Esta librería incluye:

- Componentes de interfaz de usuario (UI) del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias de código abierto y comercial.

Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de paneles, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno de ellos.

Desventajas:

- Descargas lentas. Al ser aplicaciones grandes, especialmente porque cargan todo al inicio, hace que el tiempo de descarga sea mayor al de una aplicación Web tradicional.
- Estas aplicaciones tienen problemas con los programas de accesibilidad pues, al igual que los motores de búsqueda, no trabajan bien con texto cargado dinámicamente.

- Su sistema de licenciamiento no contempla la licencia LGPL⁸, o tu código es 100% GPL⁹ o debes pagar por su licencia de desarrollo.

Frameworks seleccionado (jQuery)

Se opta por la selección de JQuery como framework de desarrollo ya que es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de su mejora y actualización. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins¹⁰ o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos y otros. Además de presentar licencia GPL para uso en cualquier tipo de plataforma, personal o comercial. Para ello simplemente tendremos que incluir en nuestras páginas un script JavaScript que contiene el código de jQuery, el cual podemos descargar de la propia página Web del producto y comenzar a utilizarlo. (DIGITTA, 2011)

Flex y Flash Builder 4.

Flex es un framework libre, de código abierto, para construir aplicaciones Web altamente interactivas, su principal objetivo consiste en permitir a los desarrolladores de aplicaciones Web construir RIA de manera rápida y fácil. En un modelo multi-capa, las aplicaciones Flex son el nivel de presentación. Este proporciona un lenguaje declarativo, servicios de aplicaciones y componentes de conectividad de datos. El mismo fue inicialmente liberado como una biblioteca de etiquetas JSP¹¹ que compilaba el lenguaje de marcas Flex (MXML) y ejecutaba mediante ActionScript aplicaciones Flash (archivos SWF binarios). Versiones posteriores de Flex soportan la creación de archivos estáticos que son compilados, los cuales pueden ser distribuidos en línea sin la necesidad de tener una licencia de servidor (FLEX, 2009). A inicios del presente año Adobe publica la nueva generación de sus exitosas herramientas: Flex 4 y el IDE¹² Flash Builder 4, las cuales presentan como objetivo fundamental la creación de RIA usando la plataforma Flash.

⁸ Licencia pública general reducida.

⁹ Licencia pública general.

¹⁰ Conjunto de componentes de software que agregan capacidades específicas a aplicaciones de software.

¹¹ **JavaServer Pages (JSP)** tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

¹² Entorno Integrado de Desarrollo (del inglés "Integrated Development Environment").

Flex como framework se podría asumir como el lenguaje y Flash Builder 4 como su editor, entre las novedades que promueve esta nueva versión se encuentra principalmente la arquitectura de componentes Spark, la cual promueve la separación completa de los elementos visuales de un componente de su lógica, además de las mejoras implementadas en la velocidad del compilador. Las aplicaciones desarrolladas sobre la plataforma Flex pueden interactuar con otras tecnologías del lado servidor a través de la invocación de un servicio Web, REST¹³ o bajo el formato binario AMF¹⁴. La interacción a través de AMF es bastante eficiente y de éste se han realizado implementaciones para diferentes lenguajes. La principal desventaja que presenta radica en el lenguaje de implementación ActionScript 3, el cual carece de constructores privados y enumeraciones, además de presentar sobrecarga de métodos.

Características de Flash Builder 4:

- Mejoras en el depurador: Permite establecer puntos de interrupción, evaluar expresiones así como conjuntos de variables, y detener la ejecución en el cambio de un valor.
- Monitor de red: Generar una pista de auditoría detallada de todos los datos que se pasan entre la aplicación local de Flex y la parte de atrás, lo cual ayuda con la depuración y optimización del rendimiento.
- Diseño visual rico: El diseño visual y la disposición previa de interfaz de usuario, la apariencia y el comportamiento cuenta con una amplia biblioteca de componentes integrados.
- ASDoc apoyo: Muestra comentarios en MXML y ActionScript mediante editores ASDoc.
- Mejora de aspectos y estilos: Personaliza la apariencia de una aplicación usando CSS y editores gráficos de propiedad. Estableciendo las propiedades de los más utilizados y pre visualizando los resultados en la vista Diseño. (BUILDER, 2010)

ActionScript 3.0

¹³ Transferencia de Estado Representacional (del inglés "Representational State Transfer"), técnica de arquitectura software para sistemas hipermedia.

¹⁴ Formato de mensaje de acción (del inglés "Action Message Format"), formato binario utilizado para serializar los objetos de ActionScript.

Como complemento de la propuesta de solución se utilizará ActionScript 3.0 (AS3), el cual constituye fundamentalmente el lenguaje de programación de Flex y de Away3D. A continuación se describen las características fundamentales del mismo, las cuales se tuvieron en cuenta a la hora de proponer la solución.

AS3 ofrece un modelo de programación robusto orientado a objetos. Algunas de las principales funciones que brinda son:

- Una nueva máquina virtual ActionScript, denominada AVM2, que utiliza un nuevo conjunto de instrucciones de código de bytes y proporciona importantes mejoras de rendimiento.
- Una base de código de compilador más moderna, que se ajusta mejor al estándar ECMAScript¹⁵ (ECMA 262) y que realiza mejores optimizaciones que las versiones anteriores del compilador.
- Una interfaz de programación de aplicaciones (API) ampliada y mejorada, con un control de bajo nivel de los objetos y un auténtico modelo orientado a objetos.
- Una API XML basada en la especificación de ECMAScript para XML (E4X) (ECMA-357 edición 2). E4X es una extensión del lenguaje ECMAScript que añade XML como un tipo de datos nativo del lenguaje.
- Un modelo de eventos basado en la especificación de eventos DOM de nivel 3. (ACTIONSRIPT, 2010)

Ventajas:

Diseñado para facilitar la creación de aplicaciones muy complejas con conjuntos de datos voluminosos y bases de código reutilizables y orientadas a objetos. Aunque no se requiere para el contenido que se ejecuta en Adobe Flash Player 9¹⁶, ActionScript 3.0 permite introducir unas mejoras de rendimiento que

¹⁵ Especificación de lenguaje de programación publicada por ECMA International, define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C.

¹⁶ Aplicación en forma de reproductor multimedia creado inicialmente por Macromedia y actualmente distribuido por Adobe Systems Incorporated.

sólo están disponibles con AVM2, la nueva máquina virtual. El código de este lenguaje puede ejecutarse con una velocidad diez veces mayor que el código ActionScript heredado. (GROSSMAN, 2006)

Amfphp

Amfphp es un software libre y de código abierto, el cual crea una puerta de enlace permitiendo la comunicación de datos entre el cliente (Flash) y el servidor (PHP). El mismo funciona como punto de partida para su cliente. Si este envía una petición a un script PHP en el servidor donde se carga Amfphp, se analiza la solicitud, las cargas del servicio solicitado, lo llama y devuelve la respuesta en consecuencia, para ello el cliente y el servidor (Flash y PHP) están sincronizados. Al ser capaz de brindar esta posibilidad, los desarrolladores pueden centrarse en las características únicas de sus proyectos, independientemente de la comunicación entre el cliente y el servidor. Amfphp permite conectar PHP con: Flash y Flex con Remoting, JavaScript y Ajax con JSON y clientes XML con XML-RPC. (AMFPHP, 2011)

Algunas de sus principales características son:

- Compatible con PHP4 y PHP5, sin necesidad de extensiones.
- Rápido y ligero.
- Posee un navegador integrado de servicios, con la generación de código y pruebas en línea.
- Soporte para varios tipos de Base de Datos, destacando Postgres, Mysql y Oracle.
- Software activo en la comunidad de desarrolladores, con miles de usuarios trabajando en su mejoramiento.

Sistemas de Gestión de Bases de Datos

Los Sistemas Gestores de Bases de Datos son software dedicados a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Estos sistemas permiten el acceso directo a la información y son capaces de manipularla ya que funcionan como una interfaz entre el usuario, la base de datos y las aplicaciones que la utilizan. Estos brindan un independiente y eficiente acceso a los datos, reduciendo el tiempo de ejecución, integridad y seguridad de los mismos. (Kroenke, 2007)

PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Sus inicios datan alrededor de 1986, como un sucesor de Ingres y es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista y libre, apoyados por organizaciones comerciales. Soporta casi todas la sintaxis SQL, incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario, contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, tcl y Python). Es un sistema robusto, por lo que consume gran cantidad de memoria y esto lo hace más lento, además de ser complejo. (POSTGRE, 2006)

MySQL

MySQL surge con el objetivo principal de manipular bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, brindando velocidad y flexibilidad, cuenta con un sistema de privilegios y contraseñas muy seguro que permite la autenticación básica para el acceso al servidor.

MySQL, es uno de los gestores más usado en el mundo del software libre, debido a la rapidez y facilidad de uso del mismo. En gran medida la aceptación que goza este gestor viene dada a partir de que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Es un software de código abierto, el cual puede descargarse desde Internet y usarlo sin costo, de esta manera el interesado puede inclinarse a estudiar el código fuente y cambiarlo de acuerdo a sus necesidades. (MYSQL, 2010).

Servidor de Base de Datos seleccionado (MySQL)

Después de realizar un profundo estudio sobre los servidores de base de datos, se selecciona para el desarrollo del entorno MySQL debido a que:

- Es un sistema gestor de bases de datos de código abierto.
- Presenta múltiples motores de almacenamiento, permitiendo al usuario escoger el que sea más adecuado para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Servidores WEB

Los servidores WEB son programas que envían contenidos tales como: páginas Web, hipertextos, formularios, imágenes y botones a través del protocolo HTTP. Estos manejan la entrega de los componentes de las páginas Web como respuesta a peticiones de los navegadores de los clientes, pues se ejecutan continuamente en un ordenador y están manteniéndose siempre a la espera de peticiones por parte de un cliente o navegador Web. (SERVER, 2010)

Apache

El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. El mismo tiene amplia aceptación en la red: en el 2005, fue el servidor HTTP más usado empleado en el 70% de los sitios Web en el mundo. En gran medida el éxito de este servidor se debe a que es multiplataforma y a su estructura modular, lo cual permite emplear diversos lenguajes en el lado del servidor (PHP, Python y Perl principalmente), así como incorporar características como la compresión de datos, las conexiones seguras y la utilización de URLs amigables. (APACHE, 2011)

Microsoft IIS

Es la solución de Microsoft a las necesidades de las empresas y usuarios de enviar y recibir la información no sólo de sus clientes sino también en el entorno de la empresa. Este servidor Web engloba una serie de herramientas administrativas que le permitirán controlar sitios Web, *FTP*, *SMTP* (correo saliente) y servicio de noticias. Dispone también del soporte necesario para crear páginas dinámicas (*ASP* y *ASP.NET*), lenguaje de aplicaciones para Internet bastante extendido permitiendo la conexión y acceso a bases de datos, consiguiendo realizar aplicaciones Web dinámicas y escalables. (IIS, 2011)

Servidor WEB seleccionado (Apache)

Luego de concluida la investigación respecto a los servidores Web se decide optar por la utilización del servidor Apache para el desarrollo del entorno, de acuerdo a una serie de características que a continuación se ponen de manifiesto:

- Modular.
- Open source.
- Multiplataforma.

- Extensible.
- Popular (fácil conseguir ayuda/soporte en Internet y otros sitios).
- Mensajes de error altamente configurables.
- Amplia aceptación en la red.
- El CMS usado en la aplicación a desarrollar, propone principalmente como servidor Web al Apache.

Metodologías de desarrollo de software

El proceso de desarrollo de software es el conjunto de actividades destinadas a guiar los esfuerzos de las personas implicadas en un proyecto de desarrollo. En dicho proyecto la metodología define ¿quién debe hacer qué? y ¿cuándo y cómo debe hacerlo? No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

RUP (Rational Unified Process)

RUP es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, este es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. RUP está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. RUP está definido por sus tres características principales: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto. (Krebs, 2007)

OpenUP

OpenUP/Basic es un framework de procesos de desarrollo de software de código abierto. Es un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental; el mismo es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

Este proceso de desarrollo unificado está basado en Rational Unified Process (RUP), desarrollado por IBM y reconocido mundialmente como uno de los procesos de desarrollo de software de mayor calidad, basándose en los principios de adaptación, importancia a los involucrados e interesados en los resultados del proyecto; colaboración, valor a la iteración y calidad continua.

OpenUP/Basic permite un abordaje ágil al proceso de desarrollo de software, con sólo proveer un conjunto simplificado de contenidos, fundamentalmente relacionados con orientación, productos de trabajo, roles, y tareas. (BALDUINO, 2007)

Este es un proceso interactivo de desarrollo de software simplificado, completo y extensible el cual está caracterizado por cuatro principios básicos interrelacionados, a saber:

- Colaboración para unificar intereses y compartir conocimientos.
- Equilibrio de prioridades competentes a maximizar el valor de los involucrados con el resultado del proyecto.
- Enfoque en la articulación de la arquitectura.
- Desarrollo continuo para obtener realimentación y realizar las mejoras respectivas.

Metodología seleccionada (OpenUP)

Después de concluir un estudio sobre las principales metodologías, se opta por la utilización de OpenUP para el desarrollo del sistema de acuerdo a:

- Se centra en articular la arquitectura para facilitar la colaboración técnica, reducir el riesgo y minimizar el sobreesfuerzo de desarrollo.
- Procura un equilibrio entre las necesidades de los involucrados con los resultados del proyecto y los costos técnicos, con el fin de maximizar el valor de los involucrados y las guías del proceso de desarrollo.
- Desarrolla un ciclo de vida interactivo que mitiga el riesgo a tiempo y ofrece demostrar resultados en curso al cliente del proyecto.

1.2 Tecnologías de visualización 3D en la Web

WebGL

WebGL es una de las más recientes tecnologías de visualización 3D sobre plataforma Web, está diseñada para presentar gráficos en tres dimensiones en las páginas Web que son acelerados por hardware sin la necesidad de plugins para el navegador. El sistema WebGL basa su tecnología combinando JavaScript, OpenGL y la etiqueta canvas, incorporada a HTML5. Además, debido a que está plenamente integrado en el navegador, una aplicación de este tipo puede aprovechar la infraestructura de JavaScript y DOM, fundamental para cualquier documento HTML. Entre sus principales ventajas se encuentran la carga de imágenes, gestión de eventos y gestión de memoria automática, pero su principal desventaja radica en que se encuentra actualmente en desarrollo, por lo que su utilización es muy limitada. (GUTIÉRREZ, 2010)

Alternativa3D

Es un motor 3D basado en la plataforma Adobe Flash que posibilita mostrar mundos tridimensionales, juegos, visitas virtuales o simplemente objetos en el navegador. Una de las mayores ventajas es que utiliza Flash, que se encuentran en la mayoría de los navegadores. Recientemente, exactamente en su versión 8 Alternativa3D brinda una serie de características impresionantes (ALTERNATIVA3D, 2009), entre ellas:

- **Alto rendimiento**
Prestación de GPU. Permite la visualización de objetos sobre los 3 millones de polígonos.
- **Sistema de iluminación**
Punto, dirección y lugar de fuentes de luz. Cada objeto puede ser iluminado por seis fuentes de luz al mismo tiempo.
- **Materiales avanzados**
Permite el uso de iluminación dinámica completa con el mapa normal, el mapa especular, hoja brillante o más materiales ligeros.
- **Carácter de dibujo**
Potente sistema de mezcla de animación jerárquica. Modelos de los personajes de una complejidad sin límite de cantidad de hueso.

- **Alta capacidad de formato de datos**

Formato binario especial, reduce la cantidad de datos necesarios para el envío por la red y acelera la carga de la escena en el motor.

Su principal desventaja radica en que es un software propietario y se necesita de licencia para su utilización.

Away3D

Away3D es un motor gráfico libre para el desarrollo de aplicaciones embebidas en los navegadores en tiempo real para Flash en ActionScript 3, originalmente derivado de Papervision3D. Away3D motor está diseñado para ser rápido y extensible, además de permitir visualizar elementos, ya sea de tipo tridimensional, bidimensional o simplemente un componente visual. (AWAY3D, 2010) Entre sus características de más relevantes se encuentran:

Visualización de objetos 3D: presenta un amplio directorio de funcionalidades las cuales permiten recrear objetos tridimensionales en el espacio con elevados números de polígonos. Además de permitir cargar archivos con el formato 3ds, Collada, AS, md2, etc. (AWAY3D, 2010)

Robustez del núcleo gráfico: provee alta capacidad para el manejo de objetos 3D que van desde primitivas básicas y la posición básica de la cámara hasta objetos de alta complejidad como personajes en movimiento, con altos niveles de polígonos. (AWAY3D, 2010)

Tecnología seleccionada (Away3D)

La integración de Flex y Away3D se presenta como la mejor alternativa para resolver una de las funcionalidades del problema en cuestión. El uso combinado de los mismos está respaldado con una documentación organizada que lleva Adobe para el desarrollo de su plataforma. Las distintas publicaciones en línea sobre las capacidades, flexibilidad y escalabilidad de los proyectos basados en Flex y Away3D son enormes.

1.3 Sistema de gestión de contenidos (CMS)

“Los sistemas de gestión de contenidos (Content Management System o CMS) son software que se utilizan principalmente para facilitar la gestión de Webs, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido Web (Web Content Management o WCM). Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a las Web.” La principal

ventaja que proporcionan estos sistemas es que brindan la posibilidad de actualizar su sitio de manera fácil y segura desde cualquier host conectado a Internet. (CMS, 2010)

Joomla

Joomla es un sistema de administración de contenido para sitios Web. Es un CMS robusto, personalizable y escalable. Este permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios. Joomla es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos, las cuales pueden ser editadas utilizando un entorno Web incluido en el producto.

Este CMS se compone de una infraestructura base y un conjunto de módulos que ofrecen un amplio conjunto de funciones, incluyendo chat, foros, búsquedas, entre otros, además de ofrecer una arquitectura de aplicación potente que permite crear sofisticadas extensiones que amplían su capacidad en múltiples e ilimitadas direcciones.

Es importante resaltar de Joomla que se puede utilizar la estructura e interfaz de usuario para suministrar aplicaciones que presenten un entorno potente y familiar, además de que es software libre, abierto y está disponible para cualquier usuario bajo licencia GPL. (Joomla, 2011)

Drupal

Drupal es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. El mismo destaca por la calidad de su código, las páginas generadas y el respeto de los estándares de la web con especial énfasis en la usabilidad y consistencia de todo el sistema. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio Web. (Buytaert, 2005)

CMS seleccionado (Joomla)

Como parte de la solución es seleccionado Joomla de acuerdo a diversos criterios acumulados durante el transcurso de la investigación. Partiendo de que los CMS propuestos tienen ventajas y desventajas similares, se optó por el antes expuesto teniendo en cuenta principalmente la documentación que

presenta el mismo. Joomla cuenta con una comunidad y centro de extensiones con más de 4000 extensiones, prácticamente para cualquier fin. Además presenta una curva de aprendizaje muy sencilla, lo cual debe tenerse en cuenta de acuerdo a que el tiempo para el desarrollo de la aplicación es relativamente corto.

1.4 Herramientas de desarrollo

Rational Rose

Rational Rose es una poderosa herramienta software para el modelado visual de sistemas software. Utiliza UML como lenguaje de modelado. Permite especificar, analizar, diseñar el sistema antes de codificarlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Rational Rose permite que se mantenga la consistencia de los modelos del sistema software, el chequeo de la sintaxis UML, la generación de documentación automática, generación de código a partir de los modelos, realizar la ingeniería inversa (crear modelo a partir código) y trabajo en grupo. Propone la utilización de cuatro tipos de modelo para realizar el diseño de un sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (RATIONAL, 2010)

Aptana Studio

Aptana Studio es un IDE de desarrollo para aplicaciones de la WEB 2.0, gratuito, código libre, con soporte Ajax, PHP, Ruby on Rails, Adobe Air, iPhone y otros. Con Aptana se facilita en desarrollo integrado de Ajax con las tecnologías emergentes. Aptana está basado en el conocido entorno de desarrollo Eclipse (IDE), también Open Source. Pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo WEB, con soporte HTML, CSS y JavaScript, así como opcionalmente a otras tecnologías mencionadas como PHP, Adobe Air o Ruby on Rails. (VERA, 2009)

1.5 Consideraciones del capítulo.

En el presente capítulo se realizó un estudio bibliográfico de las principales tendencias y tecnologías existentes para el desarrollo de aplicaciones Web interactivas, las metodologías con el fin de proveernos un marco de trabajo más limpio y organizado así como los estándares y lenguajes que brinden robustez e interacción entre los mismos, con el único propósito de solucionar el problema propuesto a inicios del capítulo.

Capítulo 2. Características del sistema. Análisis y Diseño.

2.1 Modelo del dominio.

El modelo de dominio permite comprender el funcionamiento del negocio a través de la modelación de los procesos identificados, conceptos y sus relaciones. Una manera de entender los términos empleados en el negocio es mediante un modelo conceptual, el cual explica los conceptos significativos en un dominio del problema. El modelo conceptual es una representación de conceptos u objetos que tiene como objetivo lograr un conocimiento básico del vocabulario y de los conceptos que se incluyen en los requerimientos.

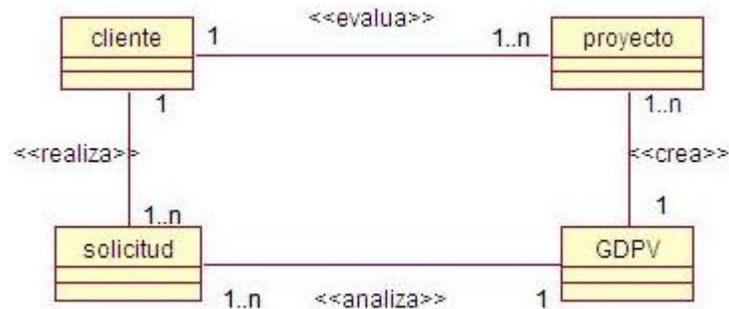


Ilustración 1. Modelo del dominio.

2.2 Soluciones Técnicas.

La fundamentación teórica plasmada en el capítulo anterior sirvió de base para seleccionar las soluciones técnicas. La aplicación Web para Centros Expositivos Virtuales (CEV) cuenta fundamentalmente de dos partes, la primera es el sitio web con sus funcionalidades y una base de datos implícita. La segunda es una aplicación Flex, la cual consta de varios componentes visuales para escoger los modelos 3D e interactuar con ellos; esta interacción se basa principalmente en la escalabilidad y rotación de los mismos. Como es un componente separado del resto de la aplicación, luego de implementado, será embebido dentro del sitio convirtiéndolo en parte de este y permitiendo a los clientes interactuar con el mismo. Se optó por el CMS Joomla para la gestión de usuarios y la gestión del sitio en general. Como gestor de base de datos se optó por Mysql y como servidor web se escogió el Apache. Se eligieron los frameworks Flex 4 y jQuery para el desarrollo de la aplicación pues ambos brindan amplias funcionalidades para la creación de Aplicaciones Enriquecidas de Internet (RIA). La aplicación está separada por componentes

fundamentalmente, cada uno de ellos representan las distintas interfaces a las cuales el usuario va a acceder, estas están compuestas por la integración del lenguaje PHP y JavaScript; además de la vinculación de estos dos lenguajes con ActionScript 3.0 para un componente específico dentro de la misma, el mismo para establecer la conexión a datos con el servidor, utiliza la puerta de enlace Amfphp. Para la carga y dibujo de modelos 3D dentro de la aplicación se escogió el motor Away3D, de acuerdo a la capacidad del mismo para la manipulación de modelos complejos y por la facilidad de trabajo de esta biblioteca. Como formato de los ficheros 3D a visualizar se eligió 3DS por ser común para todos los programas de Creación de Contenido Digital (DCC por sus siglas en inglés) como 3D Max, Maya y Blender.

Para el proceso de desarrollo de la aplicación se seleccionó la metodología de desarrollo de software OpenUP, el lenguaje de modelado UML y la herramienta Rational Rose.

Lo anteriormente expuesto nos dotó de las herramientas necesarias para comenzar el desarrollo de la aplicación Web para CEV.

2.3 Prototipo de interfaz.

En las figuras a continuación se muestra la interfaz de usuario de la aplicación Web, se describen a continuación los componentes de la interfaz.

Ilustración 2: Página principal.

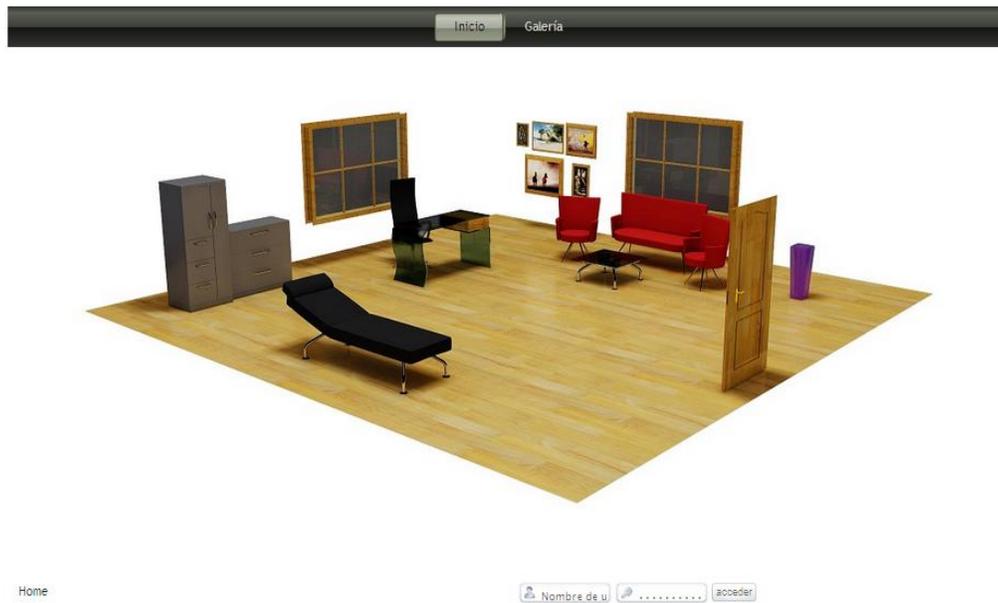


Ilustración 2. Interfaz inicial de la aplicación.

Ilustración 3-A: Componente Nuevo Proyecto. Permite al usuario crear un nuevo proyecto.

Ilustración 3-B: Componente Cargar y Eliminar. Permite al usuario cargar o eliminar un proyecto anteriormente creado.

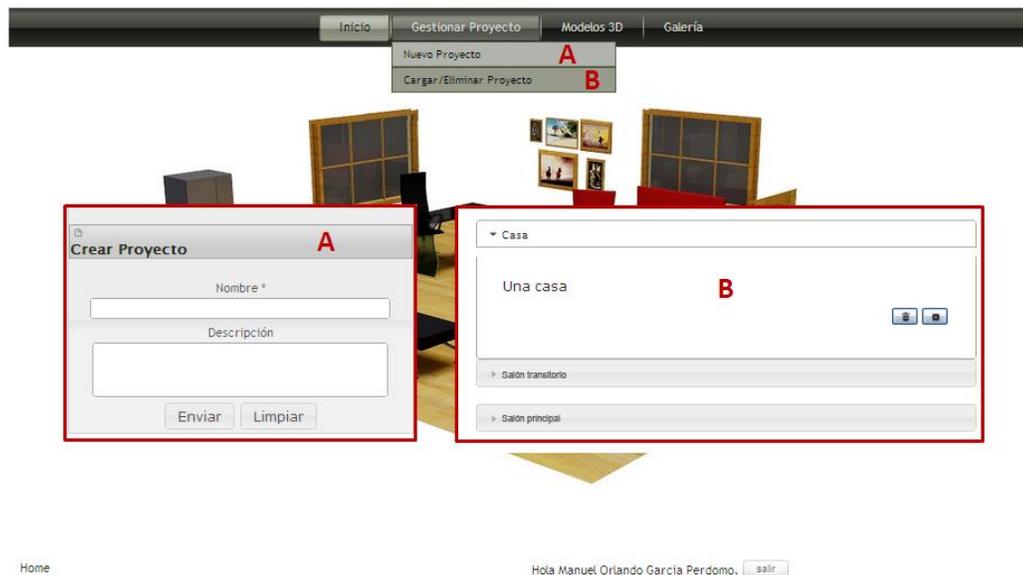


Ilustración 3. Interfaz donde se gestionan los proyectos dentro de la aplicación.

Ilustración 4-A: Componente editor de escenario. Permite al usuario crear la región (largo x ancho) donde ubicara los elementos que conformen su escena.

Ilustración 4-B: Componente editor de escenario. Permite al usuario añadir una nueva imagen y una breve descripción de la misma, esto ocurre en caso de que el usuario quiera personalizar su escenario, es válido aclarar que cada imagen es una referencia a un modelo 3D

Ilustración 4-C: Componente editor de escenario. Permite al usuario crear un área o espacio que corresponde a la ubicación exacta donde se situara el modelo 3D, es válido aclarar que sobre este espacio serán arrastradas las imágenes B.

Ilustración 4-D: Componente editor de escenario. Permite al usuario guardar su proyecto.

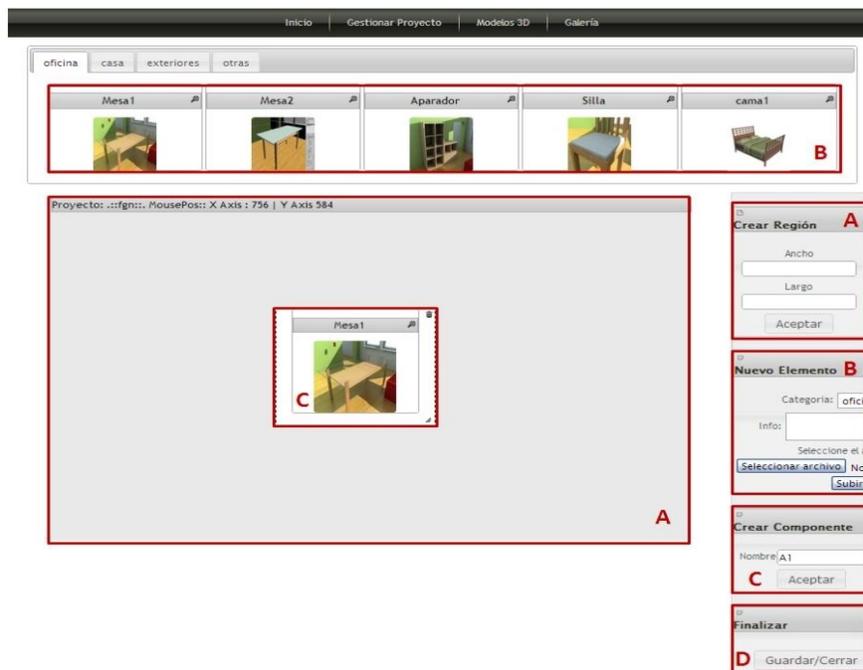


Ilustración 4. Interfaz destinada para conformar la escena dentro de la aplicación.

Ilustración 5-A: Componente visor de modelo 3D. Permite al usuario realizar comentarios sobre el modelo visualizado B.

Ilustración 5-B: Componente visor de modelo 3D. Permite al usuario interactuar con los modelos 3D (visualizarlos, escalarlos y rotarlos en cualquier dirección).

Ilustración 5-C: Componente visor de modelo 3D. Permite al usuario emitir una votación acerca del modelo 3D visualizado B.

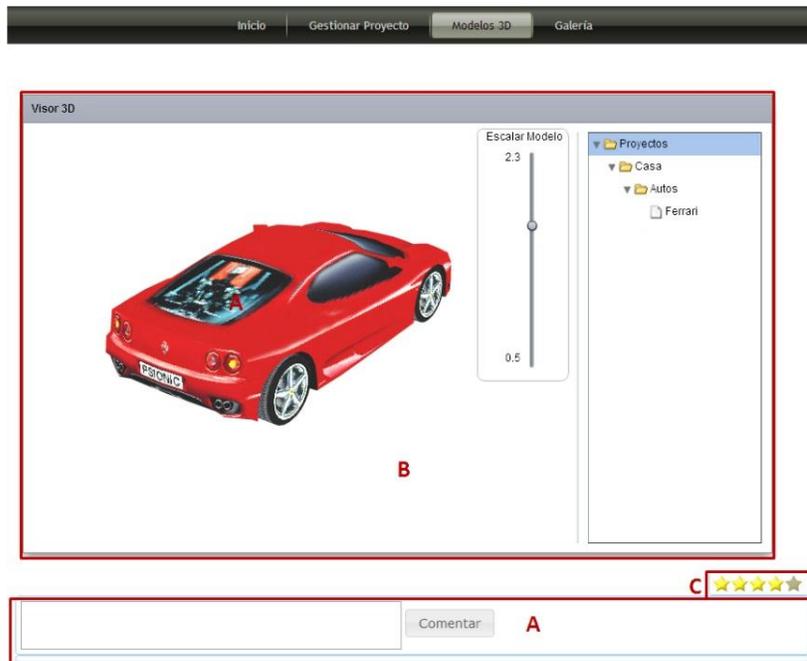


Ilustración 5. Interfaz donde se muestran los objetos tridimensionales dentro de la aplicación.

Ilustración 6-A: Módulo galería. Permite al usuario seleccionar una galería de imágenes.

Ilustración 6-B: Módulo galería. Cada imagen reflejada en la galería corresponderá a los proyectos más relevantes creados para un usuario específico.

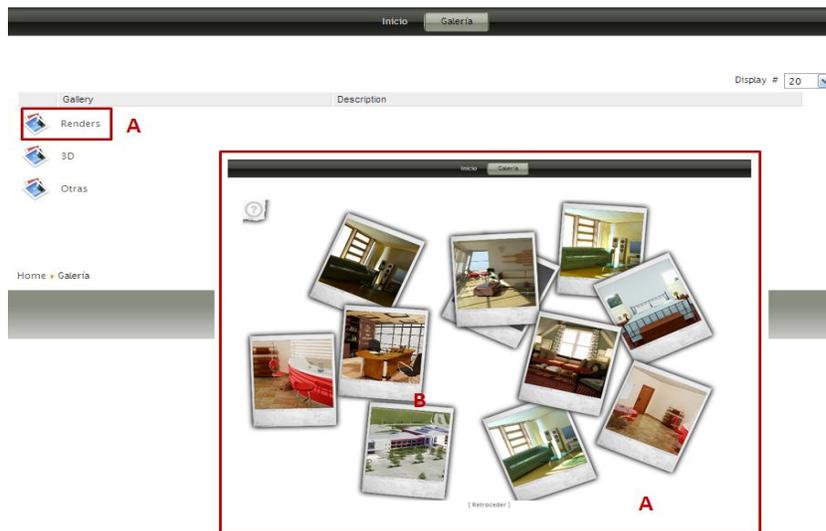


Ilustración 6. Interfaz donde se muestran imágenes de proyectos realizados.

Ilustración 7-A: Componente administración. El administrador tendrá una lista con todos los usuarios y con todos los proyectos asociados a los mismos ej. A.

Ilustración 7-B: Componente administración. El administrador tendrá la posibilidad de responder los comentarios emitidos por los usuarios con respecto a algún modelo 3D en específico que contenga el proyecto A.

Ilustración 7-C: Componente administración. El administrador tendrá la posibilidad de borrar todos los comentarios emitidos por los usuarios con respecto a algún modelo 3D del proyecto A.

Ilustración 7-D: Componente administración. El administrador tendrá la posibilidad de borrar los modelos 3D relacionados con algún proyecto ej. A.

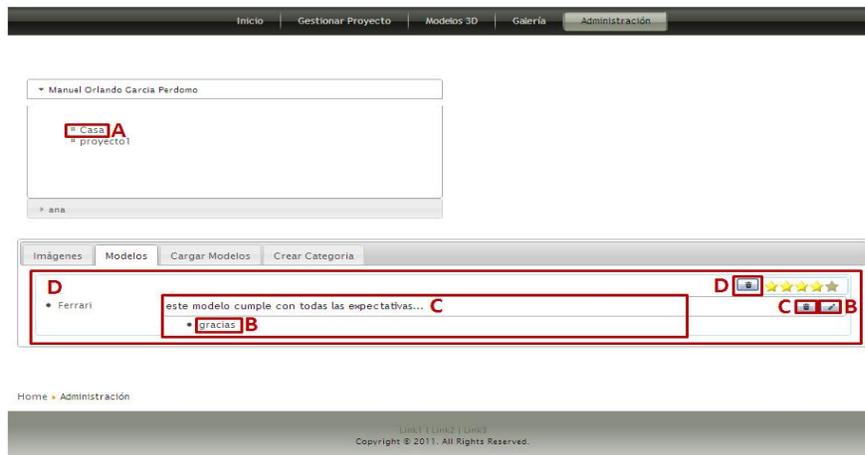


Ilustración 7. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación.

Ilustración 8-A: Componente administración. El administrador selecciona el proyecto A.

Ilustración 8-B: Componente administración. El administrador tendrá la posibilidad de subir un modelo 3D al proyecto A.

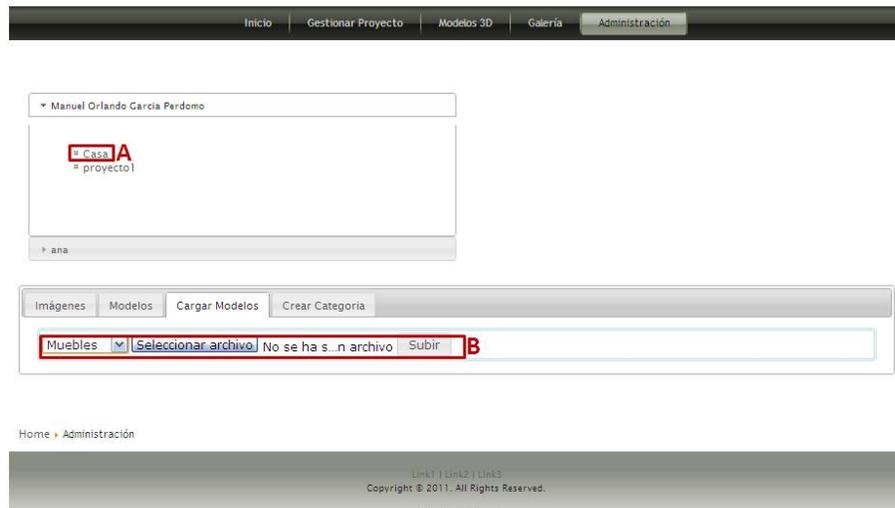


Ilustración 8. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación

Ilustración 9-A: Componente administración. El administrador selecciona el proyecto A.

Ilustración 9-B: Componente administración. El administrador tendrá la posibilidad de crear una nueva categoría de imagen para el proyecto A.

Ilustración 9-C: Componente administración. El administrador tendrá la posibilidad de crear una nueva categoría de modelo para el proyecto A.

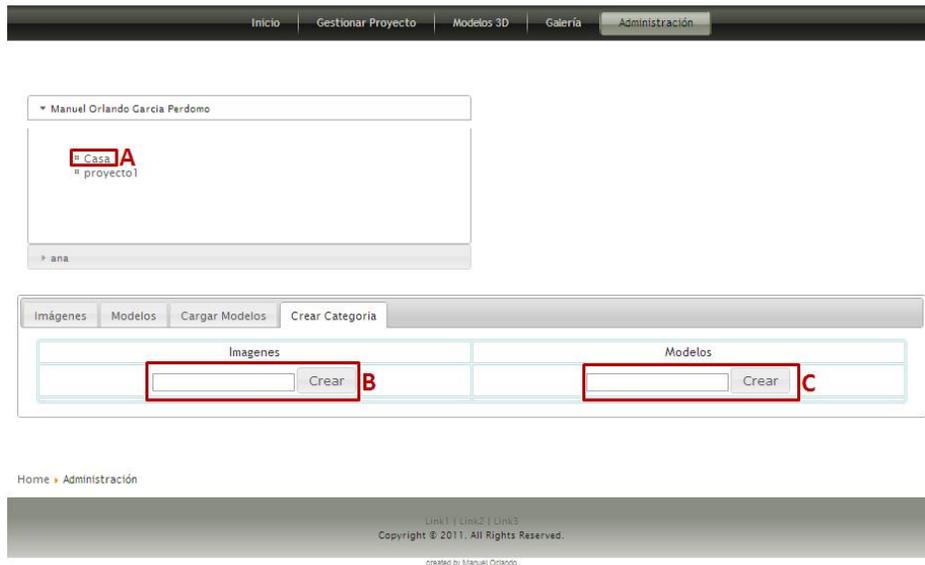


Ilustración 9. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación

Ilustración 10-A: Componente administración. El administrador selecciona el proyecto A y a continuación se le mostrarán todas las imágenes asociadas a dicho proyecto, en caso de ser imágenes cargadas por el usuario se le mostrara junto con esta la información adicional.

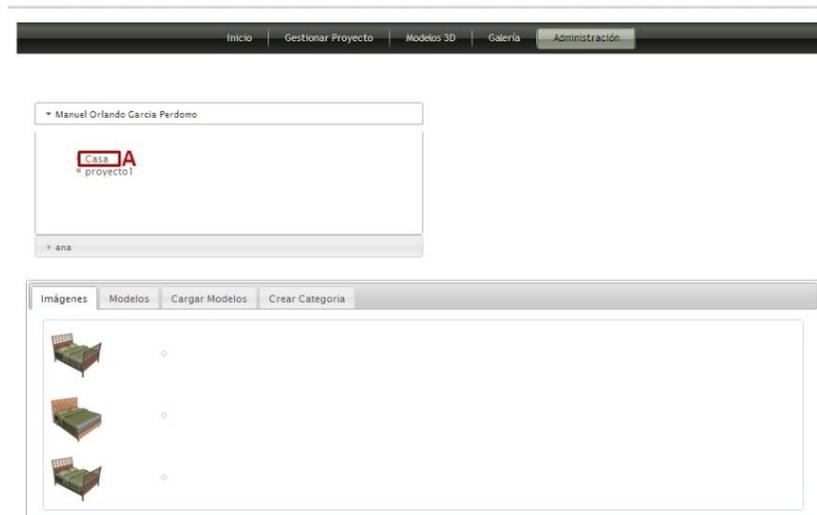


Ilustración 10. Interfaz gráfica destinada a la administración de los proyectos existentes dentro de la aplicación.

A continuación se mencionan los componentes de la propuesta de interfaz de usuario que dan respuesta a los casos de uso arquitectónicamente significativos de la aplicación:

- CU-1. Autenticar Usuario.

La aplicación da respuesta mediante el módulo de autenticación que se encuentra en la página principal. Ver Ilustración 2.

- CU-3. Administrar modelo 3D.

La aplicación da respuesta mediante los componentes que se encuentra en el área B y D (Componente administración). Ver Ilustración 8-B. Ver Ilustración 7-D.

- CU-4. Visualizar modelo 3D.

La aplicación da respuesta mediante los componentes que se encuentra en el área B (Componente visor de modelo 3D). Ver Ilustración 5-B.

- CU-5. Interactuar con modelo 3D.

La aplicación da respuesta mediante los componentes que se encuentra en el área B (Componente visor de modelo 3D). Ver Ilustración 5-B.

- CU-5. Administrar comentario sobre modelo 3D.

La aplicación da respuesta mediante los componentes que se encuentra en el área A (Componente visor de modelo 3D) y en el área C (Componente administración). Ver Ilustración 5-A. Ver Ilustración 7-C.

- CU-6. Visualizar galería de imágenes.

La aplicación da respuesta mediante los componentes que se encuentra en el área A (Modulo galería). Ver Ilustración 6-A.

- CU-9. Crear escena.

La aplicación da respuesta mediante los componentes que se encuentra en el área A, B, C (Componente editor de escenario). Ver Ilustración 4-A. Ver Ilustración 4-B. Ver Ilustración 4-C

- CU-10. Gestionar Proyecto.

La aplicación da respuesta mediante los componentes que se encuentra en el área A (Componente Nuevo Proyecto), B (Componente Cargar y Eliminar) y D (Componente editor de escenario). Ver Ilustración 3-A. Ver Ilustración 3-B. Ver Ilustración 4-D.

- CU-11. Realizar votación sobre modelo 3D.

La aplicación da respuesta mediante los componentes que se encuentra en el área C (Componente visor de modelo 3D). Ver Ilustración 5-C.

2.4 Especificación de los requisitos de software.

Los requisitos son una especificación de lo que el sistema debe cumplir para satisfacer al cliente. Estos pueden dividirse en requisitos funcionales y requisitos no funcionales.

Requisitos funcionales:

Son capacidades o condiciones que el sistema debe cumplir. A continuación se listan los requisitos funcionales del sistema:

Tabla 1.Requisitos Funcionales.

RF	Nombre	Descripción
RF1	Autenticar Usuario.	Se encarga de la autenticación de los usuarios al sistema.
RF2	Gestionar Usuario.	Maneja las funcionalidades de insertar, modificar o eliminar un usuario.
RF2.1	Insertar Usuario.	

RF2.2	Modificar Usuario.	
RF2.3	Eliminar Usuario.	
RF3	Administrar modelo 3D.	Maneja las funcionalidades de insertar o eliminar un modelo 3D.
RF3.1	Insertar modelo 3D.	
RF3.2	Eliminar modelo 3D.	
RF4	Visualizar modelo 3D.	Se visualizan los modelos 3D cargados previamente.
RF5	Interactuar con modelo 3D	Permite rotar y escalar un modelo 3D.
RF5.1	Rotar modelo 3D.	
RF5.2	Escalar modelo 3D.	
RF6	Administrar comentario sobre modelo 3D.	Maneja las funcionalidades de insertar o eliminar un comentario sobre un modelo 3D.
RF6.1	Insertar comentarios sobre modelo 3D.	
RF6.2	Eliminar comentarios sobre modelo 3D.	
RF7	Visualizar galería de imágenes.	Se visualizan las imágenes de la galería.
RF8	Gestionar imagen.	Maneja las funcionalidades de insertar, publicar, despublicar o eliminar una imagen.
RF8.1	Insertar imagen.	
RF8.2	Publicar imagen.	
RF8.3	Despublicar imagen.	

RF8.4	Eliminar imagen.	
RF9	Crear escena.	Permite añadir los elementos que componen una escena.
RF9.1	Cargar imagen.	
RF9.2	Crear Región.	
RF9.3	Insertar Área.	
RF9.4	Insertar imagen en el área.	
RF 10	Gestionar Proyecto.	Maneja las funcionalidades de insertar, cargar, guardar o eliminar un proyecto determinado.
RF10.1	Insertar Proyecto.	
RF10.2	Cargar Proyecto.	
RF10.3	Guardar Proyecto.	
RF10.4	Eliminar Proyecto.	
RF11	Realizar votación sobre modelo 3D	Permite emitir un criterio cuantitativo acerca de un modelo 3D.

Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe cumplir. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Para la herramienta a desarrollar se especifican los siguientes requisitos no funcionales:

- **Usabilidad**

Debe cumplir como meta principal la fácil manipulación aún para personas que no presenten amplios conocimientos de trabajo en ambientes Web.

- **Seguridad**

Confidencialidad:

La información solo será editada por las personas autorizadas de acuerdo al nivel de autorizo que les sea otorgado por el administrador del sistema.

Garantizar un mecanismo de seguridad basado en el modelo (Autenticación, Autorización y Auditoría).

Mostrar mensajes de error en caso de una autenticación por parte del usuario con datos incorrectos.

Integridad:

Prevenir de posibles fallos. Deberá existir un mecanismo capaz de proporcionar la recuperación de los datos, se deben hacer tanto salvas automáticas de la base de datos como copias de respaldo, las cuales contribuyan al restablecimiento ante pérdida total de información.

Protección contra acciones no autorizadas que afecten la integridad del sistema.

Disponibilidad:

Sistema con tiempo de respuesta rápida.

Garantizar el acceso a la información de acuerdo a los niveles de autorización implementados para cada usuario de manera ágil, es decir, la seguridad implantada no podrá afectar el tiempo de respuesta a las solicitudes de los usuarios.

Garantizar la rápida visualización de modelos 3D e imágenes, de forma tal que no sobrecargue el procesamiento de computadoras sin muchas prestaciones.

- **Soporte**

Sistema de fácil mantenimiento.

Sistema capaz de incorporar nuevos recursos que amplíen y enriquezcan su contenido.

- **Interfaz**

La aplicación presentará una interfaz amigable la cual debe permitir una rápida comprensión por parte de los usuarios.

Las funcionalidades deberán estar visibles en todo momento facilitando la navegación a través de la aplicación.

- **Requerimientos de software**

Requisitos óptimos del software del servidor:

Sistema operativo Windows.

Servidor WEB Apache (2.0 o superior).

Servidor de base de datos MySQL 5.x.

PHP 5 o superior.

Requisitos óptimos del software del cliente:

Navegador Web con capacidad de interpretación de JavaScript (Mozilla Firefox, Google Chrome).

Instalación del Flash Player 10.

- **Requerimientos de hardware**

Requisitos mínimos de la maquina servidora:

Procesador: Igual o superior a 500 MHz de velocidad.

Memoria RAM de: 512 MB.

Capacidad mínima de disco duro de 20 GB.

Periféricos: tarjeta red.

Requisitos mínimos de la estación de trabajo o cliente:

Procesador: Igual o superior a 500 MHz de velocidad.

Memoria RAM de 512 MB.

Capacidad mínima de disco duro de 500 MB.

- **Portabilidad**

El sistema debe ser de fácil instalación sobre la plataforma Windows.

2.5 Definición de los Casos de Uso.

En general, un caso de uso es simplemente, un texto escrito que describe el papel de un actor que interactúa con el acontecer del sistema.

Definición de los actores.

Un actor es algo que se comunica con el sistema o producto y que es externo al sistema en sí mismo, representan papeles que las personas (o dispositivos) juegan como impulsores del sistema. Existe actores que trabajan directa y frecuentemente con el software, interactúan para conseguir funciones del sistema y derivar el beneficio deseado del mismo. Otros existen para dar soporte al sistema.

A continuación se describen los actores que formarán parte de sistema implementado.

Tabla 2. Actores del Sistema.

Actor	Descripción
Administrador	Es la persona que tiene permisos y privilegios de edición o modificación de los recursos con que cuenta el sistema o sea modifica su contenido.
Cliente	Es la persona que tiene acceso a las funcionalidades tanto públicas como las de gestión de proyectos que lo involucren.
Usuario Sistema	Es la persona que accede al sistema mediante el proceso de autenticación.

2.6 Diagrama de Casos de Uso.

La representación de los diagramas de casos de uso permite especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los actores del mismo.

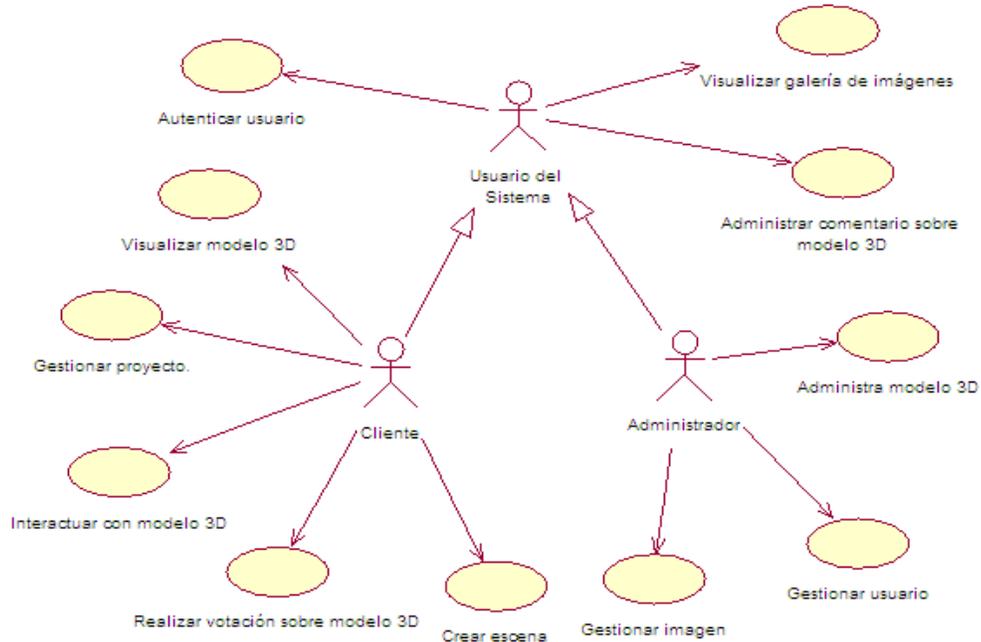


Ilustración 11. Diagrama de Casos de Uso

2.7 Descripción textual de los Casos de Uso.

Las descripciones textuales de los casos de uso pueden verse en el Anexo 1, las descripciones textuales de todos los casos de uso definidos para el sistema.

Tabla 3. CU Administrar Modelo.

Caso de uso	
CU-3	Administrar modelo 3D.
Propósito	El caso de uso se encargará de permitirle al actor la administración de modelos 3D en la aplicación.
Prioridad	Crítica.
Actores: Administrador.	

Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de administrar (insertar o eliminar) algún modelo del sistema.	
Referencias	RF3.
Precondiciones	El administrador deberá estar autenticado en el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor ya autenticado selecciona la opción administrar modelo 3D.	2. El sistema muestra la interfaz donde pide el tipo de modificación a realizar. <ul style="list-style-type: none"> • Si el actor desea insertar un modelo ir a la <u>Acción 1</u>. • Si el actor desea eliminar un modelo ir a la <u>Acción 2</u>.
3. El actor escoge una opción y hace clic en el botón aceptar.	4. El sistema muestra la información relacionada con la opción.
Acción 1 Insertar Modelo	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona la opción “Subir”.	2. El sistema muestra el formulario con todos los campos a llenar.
3. El actor presiona la opción “Guardar”.	4. Termina el caso de uso.
Acción 2 Eliminar Modelo	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona el modelo que desea eliminar y presiona la opción “Eliminar”.	2. El sistema elimina el modelo deseado.

Tabla 4.CU Visualizar Modelo 3D.

Caso de uso	
CU-4	Visualizar modelo 3D
Propósito	El caso de uso se encargará de permitirle al actor la visualización del modelo 3D.
Prioridad	Crítica.
Actores: Cliente.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de visualizar los modelos 3D.	
Referencias	RF4.
Precondiciones	El actor deberá estar autenticado en el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor ya autenticado selecciona la opción “Modelos 3D”.	2. El sistema muestra la interfaz donde se visualizan los modelos 3D.

Tabla 5.CU Crear Escena.

Caso de uso	
CU-9	Crear Escena.
Propósito	El caso de uso se encargará de permitirle al actor la creación de la escena.
Prioridad	Crítica.
Actores: Cliente.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de crear la escena.	
Referencias	RF 9
Precondiciones	Debe estar autenticado en la aplicación.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema

<p>1. El actor accede la página de edición del escenario.</p>	<p>2. El sistema muestra la interfaz donde pide el tipo de modificación a realizar.</p> <ul style="list-style-type: none"> • Si el actor desea cargar una imagen ir a la <u>Acción 1</u>. • Si el actor desea crear una región ir a la <u>Acción 2</u>. • Si el actor desea insertar un área ir a la <u>Acción 3</u>. • Si el actor desea insertar una imagen en el área ir a la <u>Acción 4</u>.
Acción 1 Cargar Imagen	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
<p>1. El actor escoge en las categorías existentes el elemento que desee insertar en la escena.</p>	<p>2. El sistema muestra todos los elementos existentes por categorías.</p>
<p>3. El actor arrastra el elemento esta que este dentro del componente creado previamente.</p>	<p>4. el sistema muestra el elemento dentro del componente.</p>
Acción 2 Crear Región	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
	<p>1. El sistema muestra los campos a llenar para crear la región.</p>
<p>2. El actor llena los campos</p>	<p>3. El sistema muestra la región creada.</p>
	<p>4. El sistema muestra los campos a llenar para crear la región.</p>
Acción 3 Insertar Área.	

Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra el campo para insertar un área dentro de la región creada.
2. El actor llena los campos.	3. El sistema muestra el área insertada dentro de la región.
Acción 4 Insertar imagen en la escena.	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor escoge la categoría que le corresponde y describe las características del mismo.	
2. El actor hace clic en el botón “Seleccionar Archivo”.	3. El sistema muestra la ventana para buscar el elemento a insertar.
4. El actor escoge el elemento y presiona el botón Aceptar.	5. El sistema muestra la ruta donde se encuentra el elemento.
6. El actor presiona el botón Subir.	7. El sistema muestra el elemento añadido.

Tabla 6.CU Gestionar Proyecto.

Caso de uso	
CU-10	Gestionar Proyecto
Propósito	El caso de uso se encargará de permitirle al actor la gestión de proyecto.
Prioridad	Crítica.
Actores: Cliente.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de insertar, cargar, guardar o eliminar un proyecto determinado.	
Referencias	RF10.
Precondiciones	Debe estar autenticado en la aplicación.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema

<p>1. El actor accede a la opción Gestionar Escenario.</p>	<p>2. El sistema muestra la interfaz donde pide el tipo de modificación a realizar.</p> <ul style="list-style-type: none"> • Si el actor desea insertar un proyecto ir a la <u>Acción 1</u>. • Si el actor desea cargar un proyecto ir a la <u>Acción 2</u>. • Si el actor desea guardar un proyecto ir a la <u>Acción 3</u>. • Si el cliente desea eliminar un proyecto ir a la <u>Acción 4</u>.
Acción 1 Insertar Proyecto	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor hace clic en la opción Nuevo Proyecto	2. El sistema muestra los campos a llenar.
3. El actor llena los campos correspondientes.	
4. Hace clic en el botón Aceptar.	5. El sistema muestra el proyecto creado.
Curso Alternativo de Eventos "Campos Vacíos"	
	1. El sistema muestra un mensaje de error: "Campos Obligatorios".
Acción 2 Cargar Proyecto	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor hace clic en la opción Cargar/Eliminar Proyecto.	2. El sistema muestra todos los proyectos existentes.
3. El actor selecciona el proyecto que desee.	4. El sistema muestra el proyecto cargado

Acción 3 Guardar Proyecto	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona el botón Guardar/Cerrar.	2. El sistema muestra todos los proyectos guardados hasta el momento.
Acción 4 Eliminar Proyecto	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona el botón 	2. El sistema muestra el proyecto eliminado.

Tabla 7. Realización Votación sobre modelo 3D.

Caso de uso	
CU-11	Realizar Votación sobre modelo 3D.
Propósito	El caso de uso se encargará de permitirle al actor el gestionar región.
Prioridad	Secundario.
Actores: Usuario del sistema.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de realizar su votación respecto a un modelo determinado.	
Referencias	RF11
Precondiciones	Debe estar autenticado en la aplicación. Existir un proyecto cargado.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Modelo 3D.	2. El sistema muestra la interfaz donde se encuentran todos los modelos cargados.
3. El actor escoge el modelo al que le quiere dar su voto.	4. El sistema muestra cinco estrellas mediante las cuales se realizará la votación.
5. El actor dando clic en las estrellas realiza su voto.	6. El sistema muestra las estrellas de color amarillo con el voto correspondiente al

	modelo seleccionado.
--	----------------------

2.8 Patrón "Modelo-Vista-Controlador".

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón Modelo-Vista-Controlador (MVC) se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

Modelo: Esta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos; por ejemplo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente (como puede ser una base de datos) para almacenar los datos. MVC no menciona específicamente esta capa de acceso a datos. (Gestiona las entradas del usuario). (Lago, 2007)

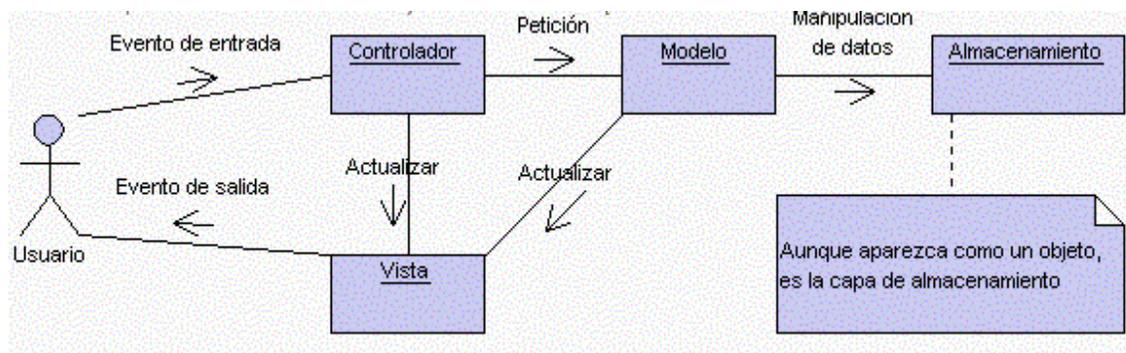


Ilustración 12. Modelo Vista Controlador

2.9 Diseño.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción, lo cual contribuye a una arquitectura estable y sólida. Durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación.

El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software, éste tiene como propósitos:

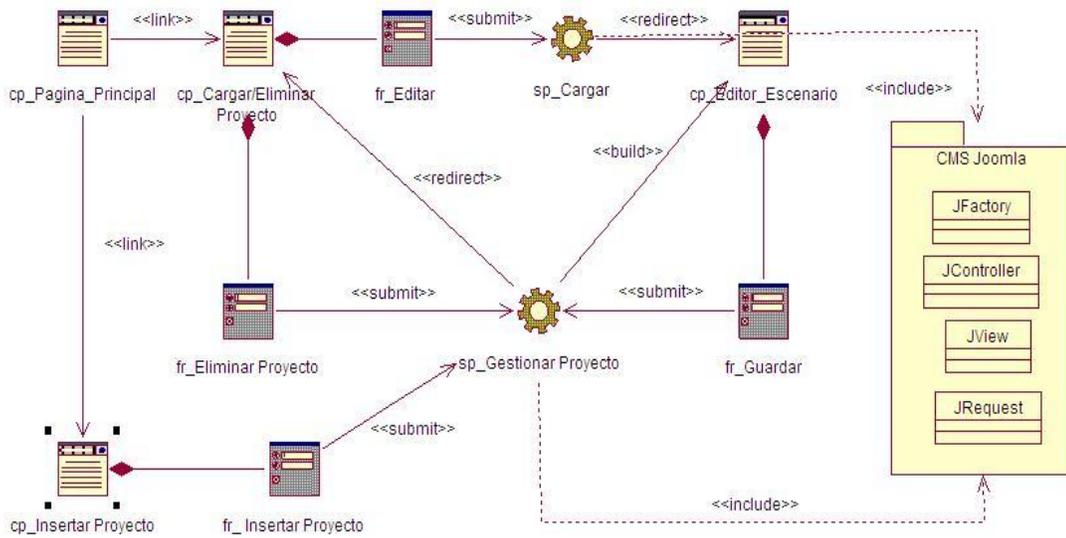
- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

2.10 Diagramas de clases del diseño.

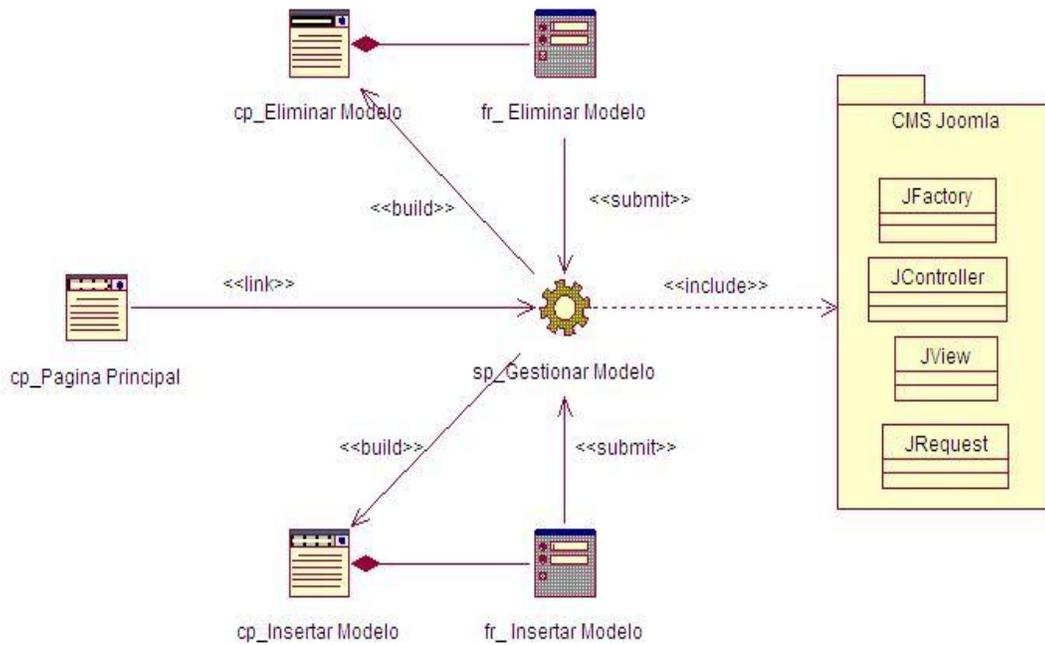
Es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema, las clases y sus relaciones.

Diagrama de clases del diseño (Autenticar Usuario) ver Anexo 3.

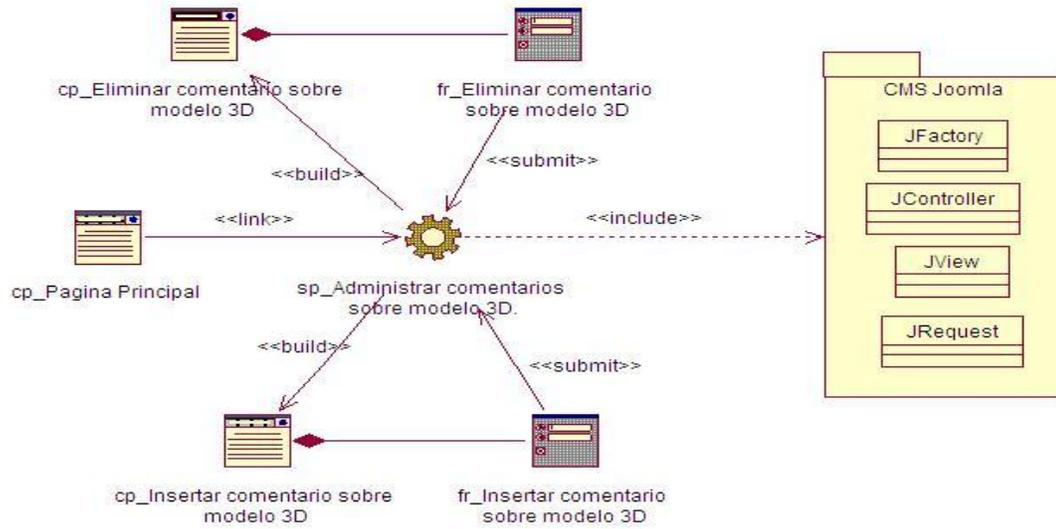
2.10.1 Diagrama de clases del diseño (Gestionar Proyecto).



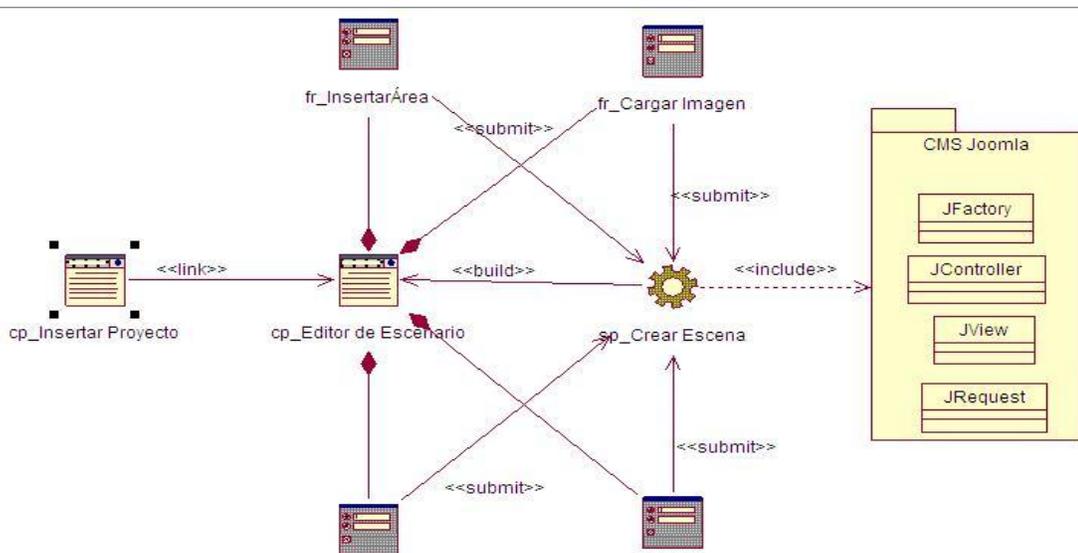
2.10.2 Diagrama de clases del diseño (Administrar Modelo 3D).



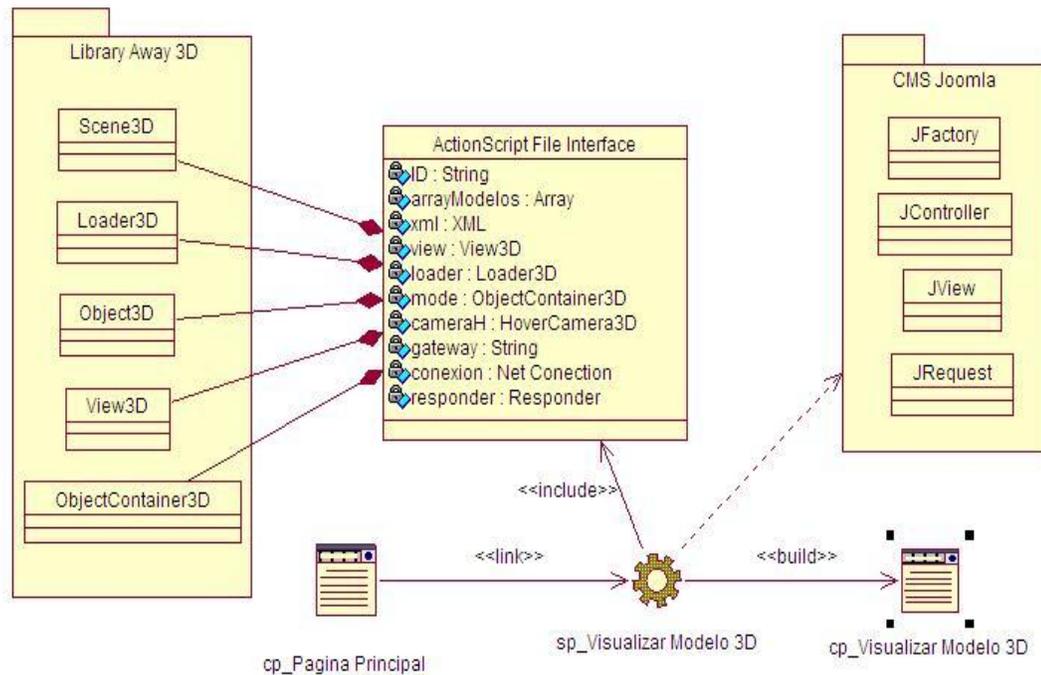
2.10.3 Diagrama de clases del diseño (Administrar Comentario sobre modelo 3D).



2.10.4 Diagrama de clases del diseño (Crear Escena).



2.10.5 Diagrama de clases del diseño (Visualizar modelo 3D).



2.11 Diagramas de secuencia.

Un diagrama de secuencia muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. Si los casos de uso tienen varios flujos o sub-flujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos. Los elementos que lo componen son:

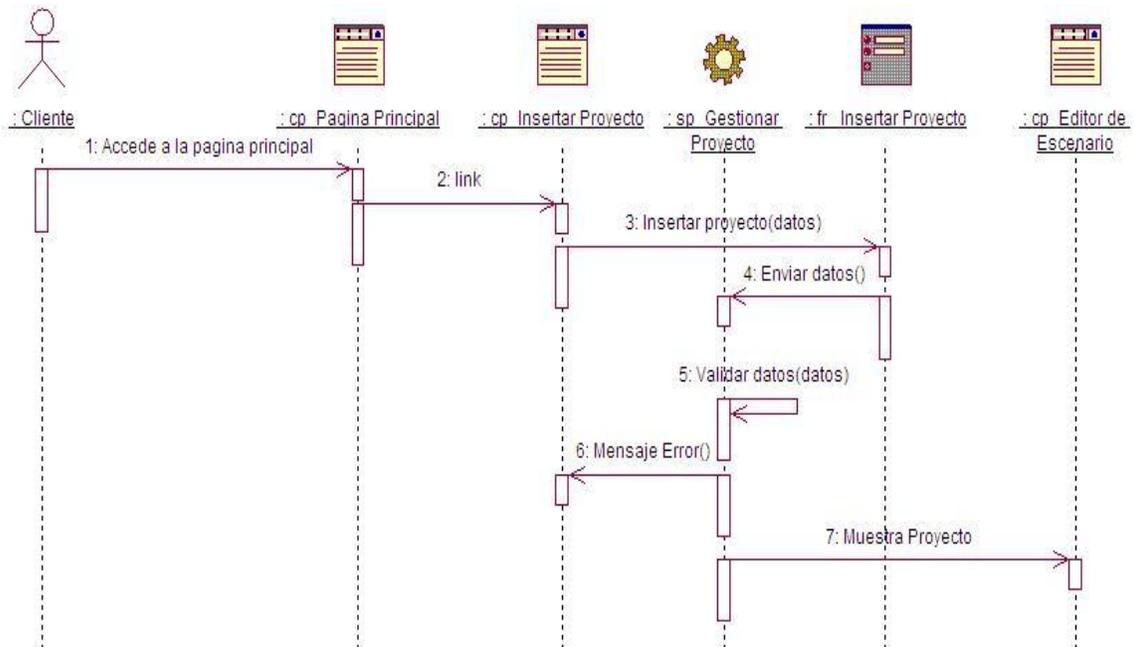
Activación: Período de tiempo en el cual el objeto se encuentra desarrollando una acción.

Línea de vida de un objeto: Representa la vida del objeto durante la interacción.

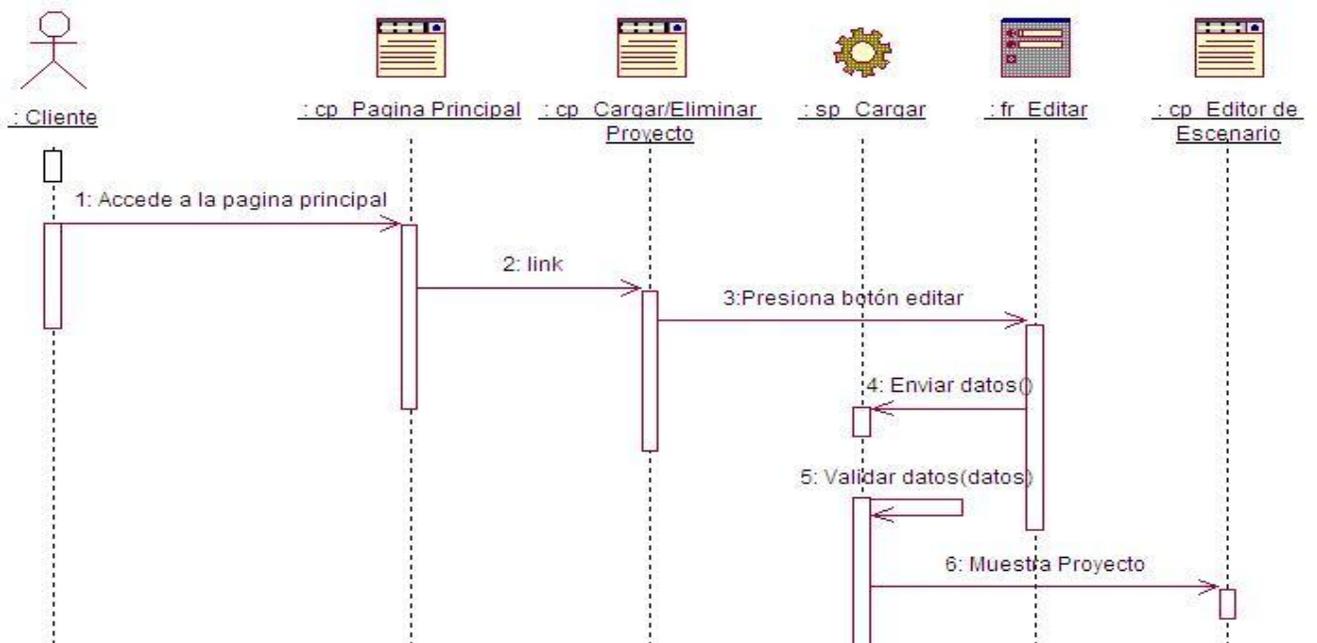
Mensaje: El envío de mensajes se denota mediante una línea sólida dirigida desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.

Diagrama de secuencia: Autenticar Usuario ver Anexo 4.

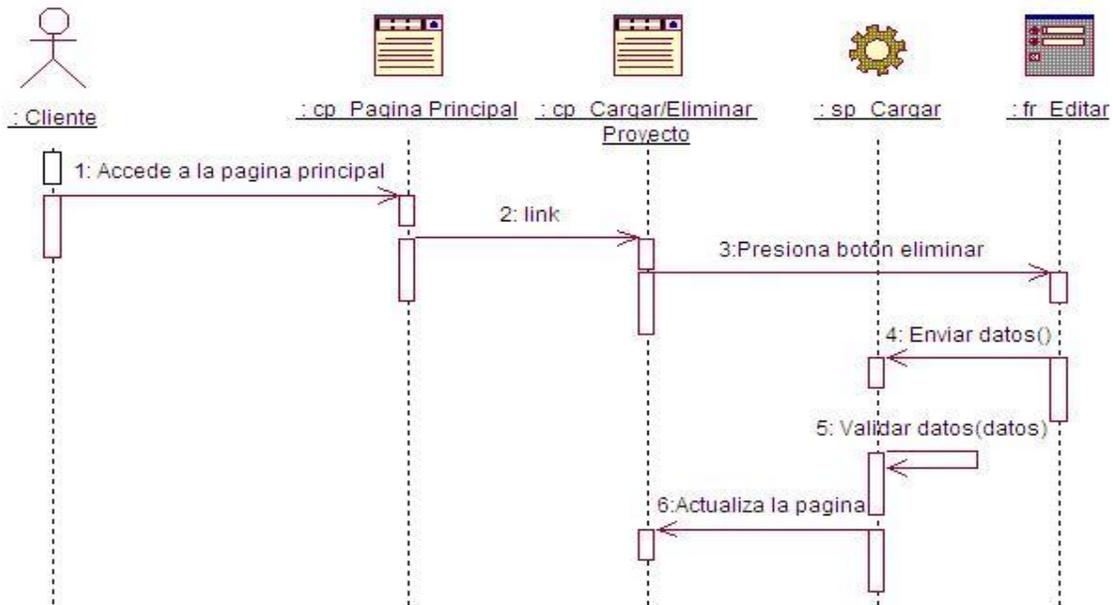
2.11.1 Diagrama de secuencia: Gestionar Proyecto (Insertar Proyecto).



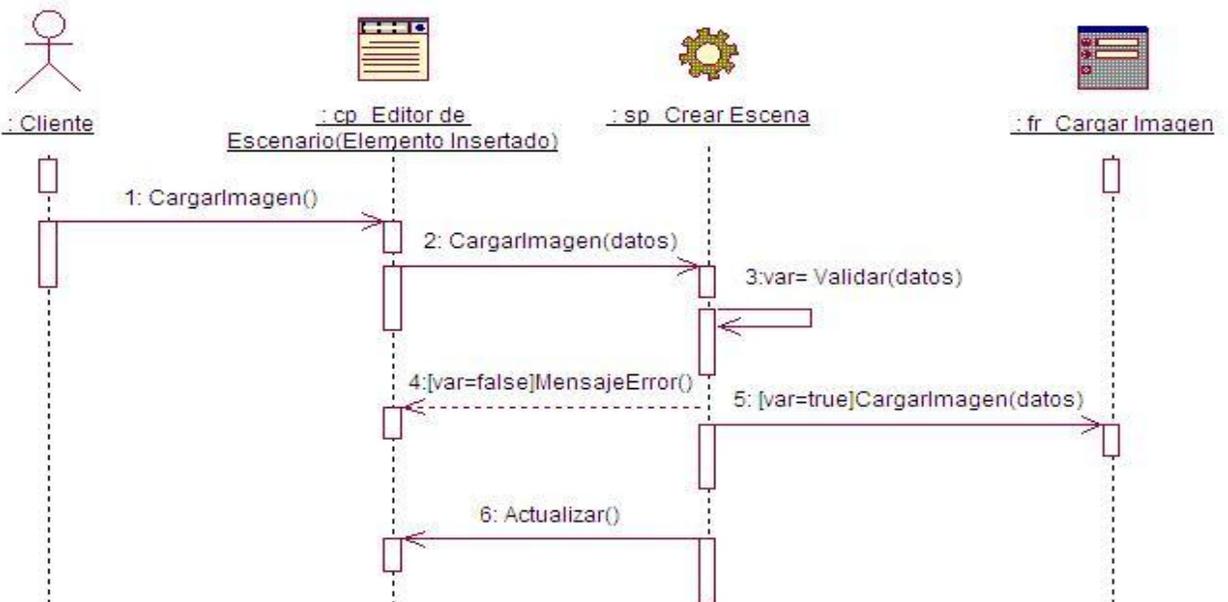
2.11.2 Diagrama de secuencia: Gestionar Proyecto (Cargar Proyecto).



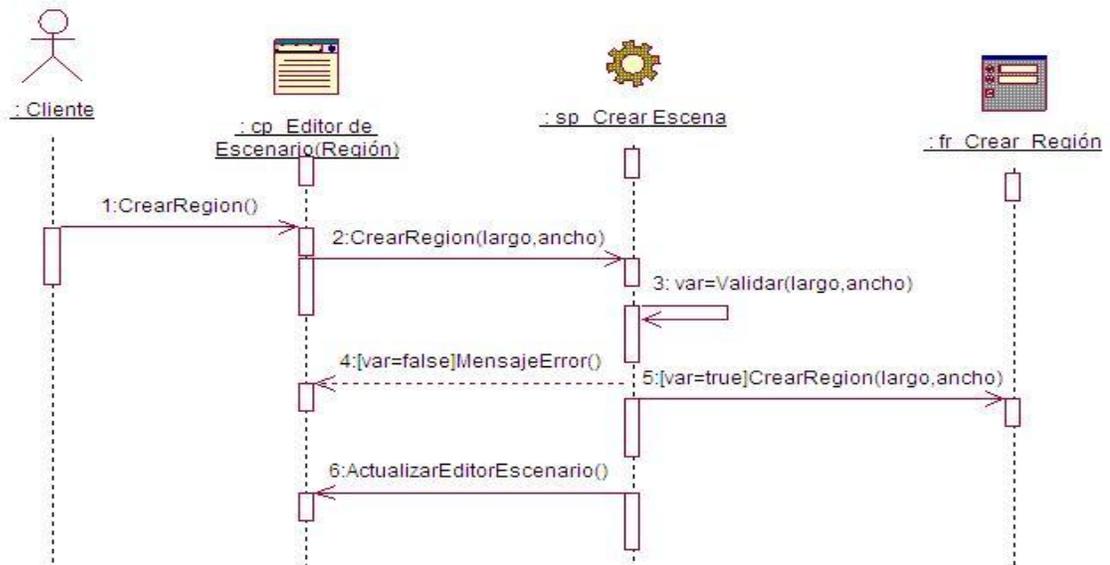
2.11.3 Diagrama de secuencia: Gestionar Proyecto (Eliminar Proyecto).



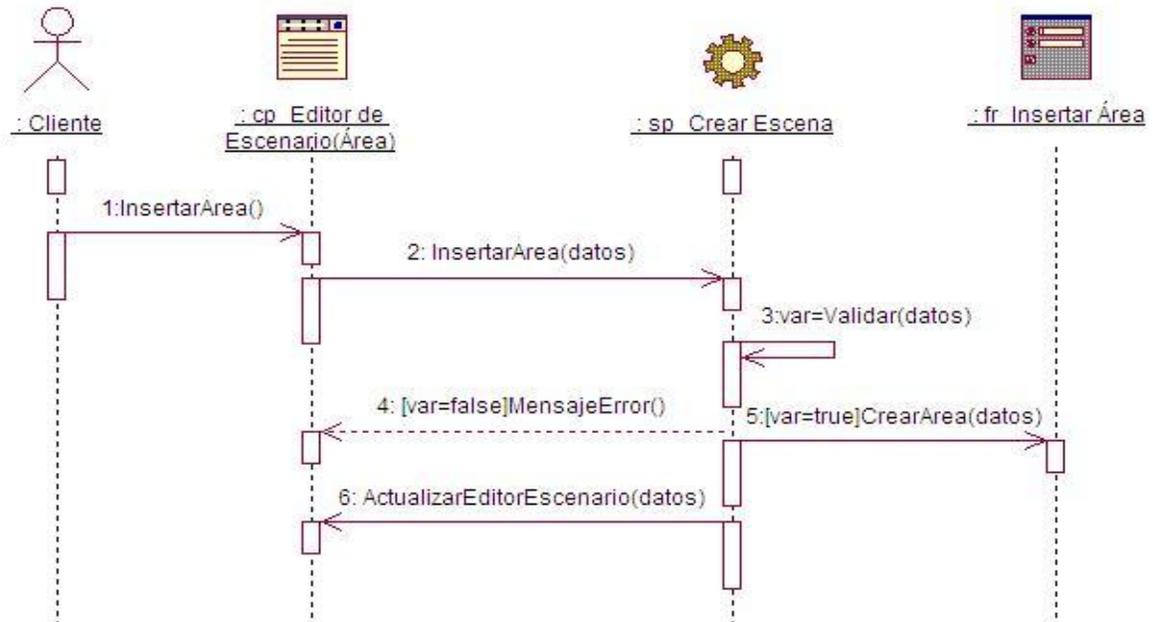
2.11.4 Diagrama de secuencia: Crear Escena (Cargar Imagen).



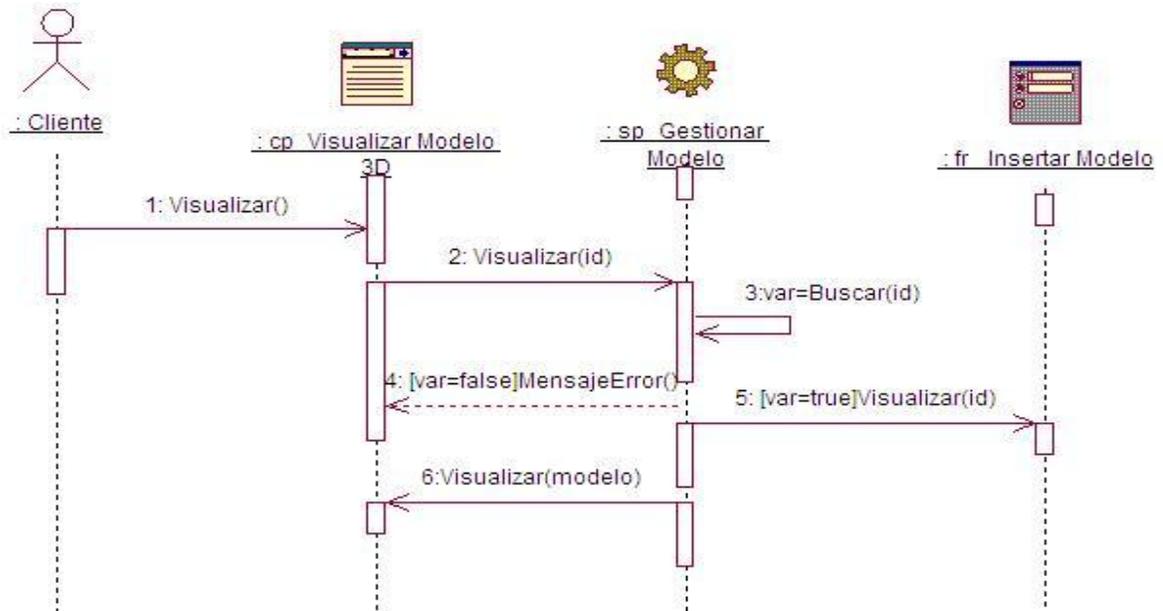
2.11.5 Diagrama de secuencia: Crear Escena (Crear Región).



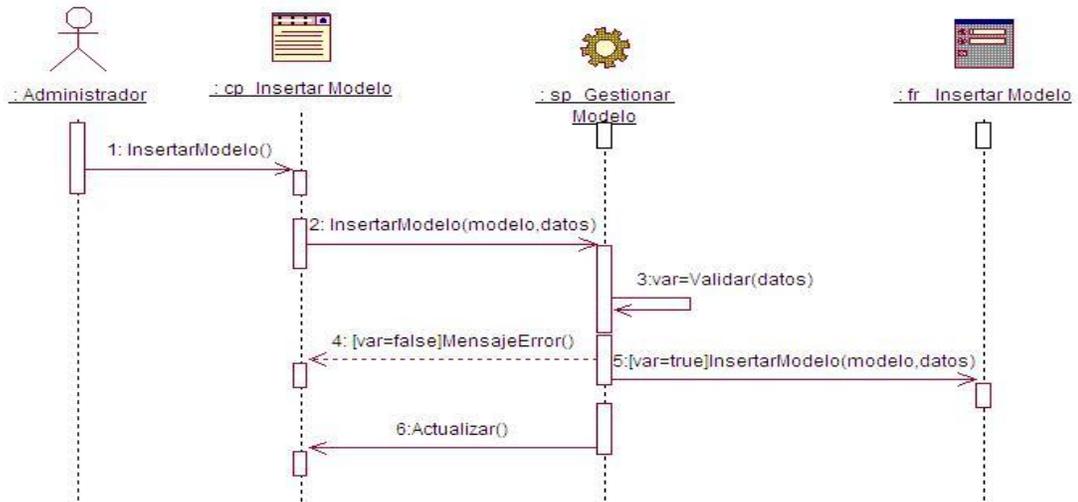
2.11.6 Diagrama de secuencia: Crear Escena (Insertar Área).



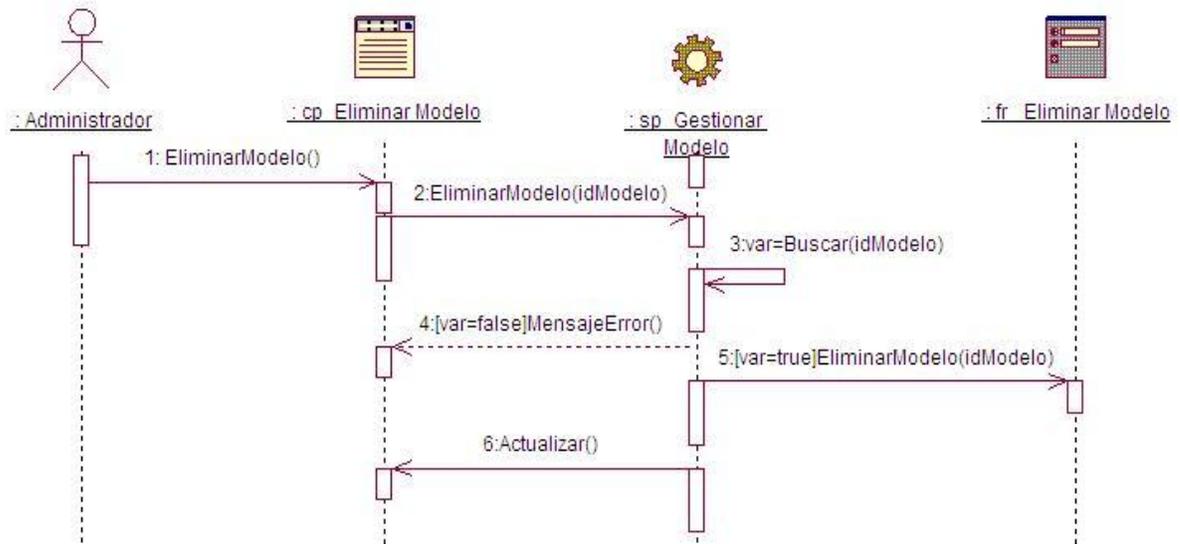
2.11.7 Diagrama de secuencia: Crear Escena (Visualizar Modelo).



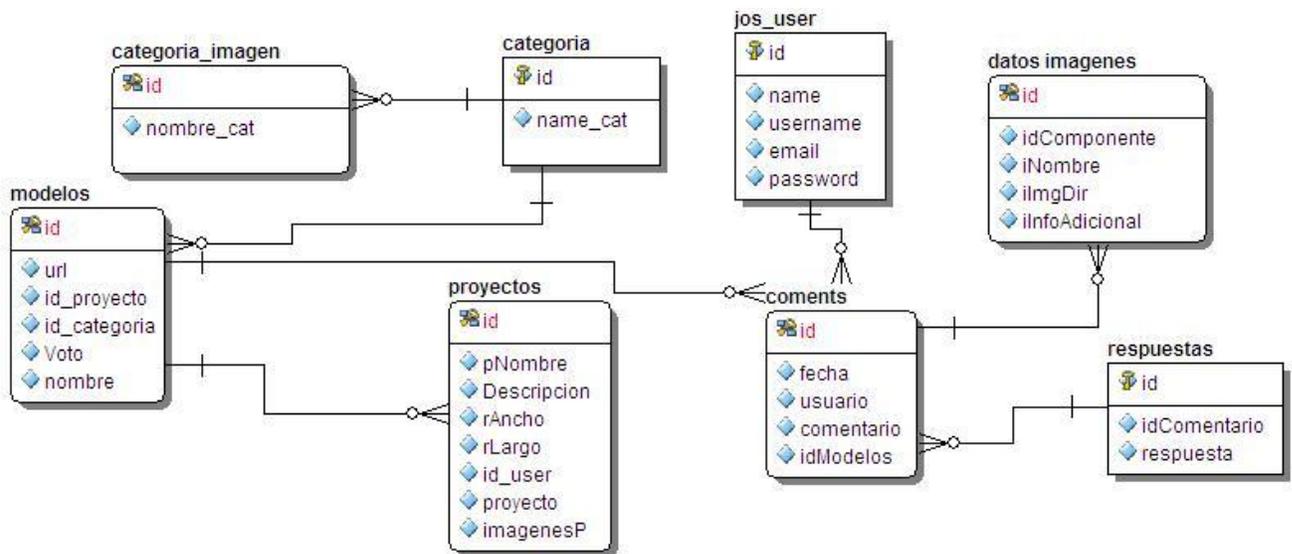
2.11.8 Diagrama de secuencia: Administrar Modelo (Insertar Modelo).



2.11.9 Diagrama de secuencia: Administrar Modelo (Eliminar Modelo).



2.12 Diseño de la Base de Datos.



2.13 Consideraciones del capítulo.

Se describió el proceso que actualmente se realiza para la creación de aplicaciones Web, para identificar los problemas existentes y precisar los subprocesos que serán objeto de automatización. Se realizó el modelo de dominio para obtener una mejor comprensión de los conceptos y procesos que se manejan dentro del negocio a automatizar. Se presentó la propuesta del sistema a desarrollar como parte de la

solución al problema planteado. Se especificaron y describieron los requisitos que la aplicación debe cumplir para obtener su comprensión. Se realizó la definición, representación y descripción textual de los casos de uso del sistema con el objetivo de especificar las funcionalidades del sistema y los actores que interactuarán con el mismo. Se describieron paso a paso la secuencia de eventos que los actores realizarán para inicializar un caso de uso y la respuesta del sistema ante cada acción del actor. De forma general con la conclusión del presente capítulo se logró describir el sistema que se desea construir lo cual dará paso a una nueva etapa del desarrollo del trabajo para darle cumplimiento a los objetivos trazados.

Capítulo 3. Implementación y Validación del Sistema.

3.1 Modelo de implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, ficheros de código fuente y ejecutables. Además describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, así como la dependencia existentes entre los mismos. Los diagramas de despliegue y componentes conforman lo que se conoce como un **modelo de implementación** al describir los componentes a construir, su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

El modelo de implementación tiene como objetivos principales:

- Definir la organización del sistema en términos de Subsistemas de Implementación organizados en capas.
- Implementar los elementos de diseño en términos de “Elementos de Implementación” (ficheros Fuentes, binarios, ejecutables y otros).
- Probar los componentes desarrollados independientemente como unidades.
- Integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable.

3.2 Diagrama de Componentes.

Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones, el cual se utilizan para modelar la vista estática de un sistema. En éste se muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Un ejemplo típico

de una jerarquía en capas de este tipo es: Interfaz de usuario; Paquetes específicos de la aplicación; Paquetes reusables; Mecanismos claves; y Paquetes hardware y del sistema operativo.

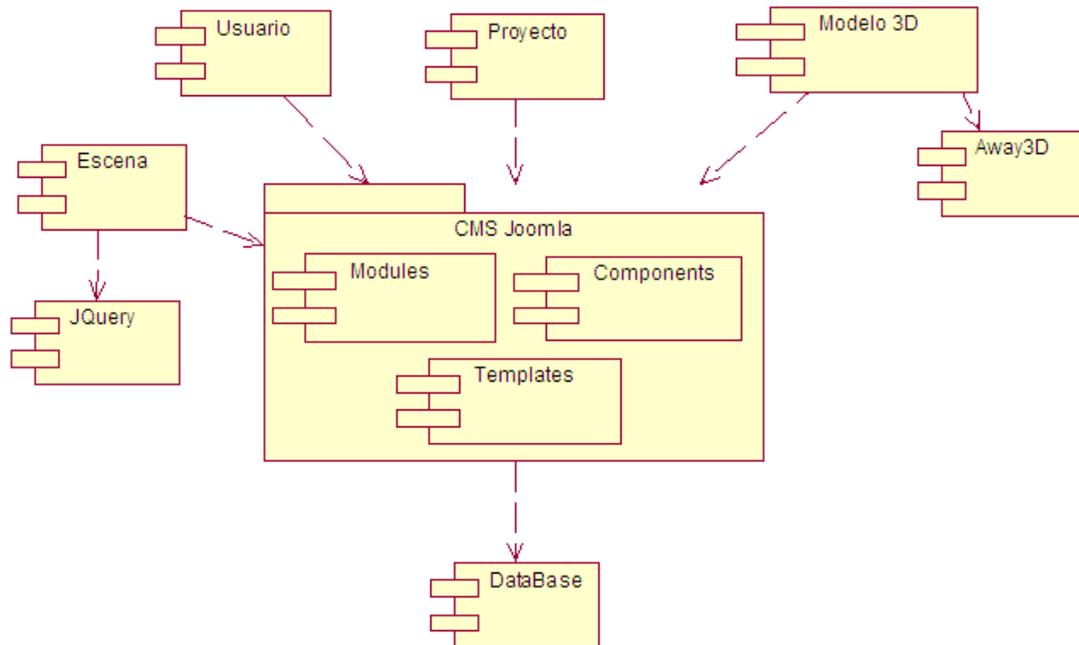


Ilustración 13. Diagrama de Componentes.

3.3 Modelo de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de computo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. Algunos de los elementos fundamentales dentro del modelo de despliegue son:

Procesadores: Nodos que tienen capacidad de procesamiento, computadoras por lo general. Ej. Maquina Cliente, Servidor de Datos, Servidor Web, Servidor de Aplicaciones, Servidor de Correo.

Dispositivos: Nodos que no tienen capacidad de procesamiento. Ej. Impresora, Scanner, Webcam, Lector de Tarjeta.

Protocolos: Estándares que deben existir implementados en la red entre máquinas, para efectuar cierta comunicación. Ej. SMTP, POP3 para correo, RMI para componentes distribuidos sobre Java, DCOM para componentes distribuidos de Microsoft, ADO, JDBC, OLE-DB para encuestar bases de datos.

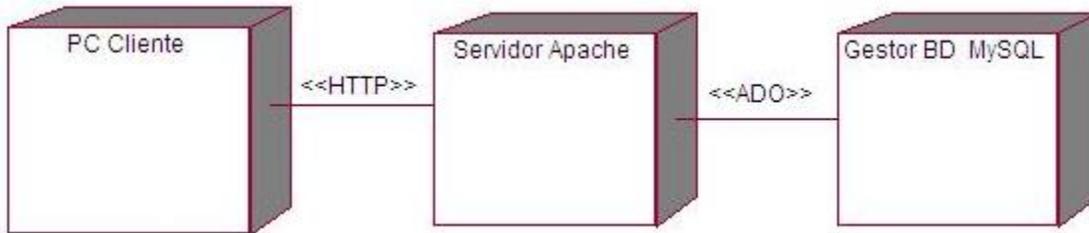


Ilustración 14. Modelo de Despliegue.

3.4 Modelo de Prueba.

Las pruebas es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados. Luego se hace una evaluación sobre algún aspecto del sistema o componente.

Objetivos

Encontrar y documentar los defectos que puedan afectar la calidad del software.

Validar que el software trabaje como fue diseñado.

Validar y probar los requisitos que debe cumplir el software.

Validar que los requisitos fueron implementados correctamente.

3.5 Descripción de los casos de prueba de integración.

Las descripciones de los casos de prueba de integración pueden verse en el Anexo 2.

3.5.1 Diseño de caso de prueba. CU Administrar modelo 3D.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de administrar los modelos, insertar o eliminar algún modelo del sistema.

Condiciones de ejecución:

El actor deberá estar autenticado en el sistema.

Tabla 8. Diseño de caso de prueba. CU Gestionar modelo 3D.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
----------------------	--------------------------	---------------------------------	---------------

SC1: Administrar modelo 3D.	EC 1.1: Insertar Modelo	Insertar Modelo	El actor presiona la opción “Subir”. El sistema muestra el formulario con todos los campos a llenar. El actor presiona la opción “Guardar”.
	EC 1.2: Eliminar Modelo	Eliminar Modelo	El actor selecciona el modelo que desea eliminar y presiona la opción “Eliminar”. El sistema elimina el modelo deseado.

3.5.2 Diseño de caso de prueba. CU Visualizar modelo 3D.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de visualizar los modelos 3D.

Condiciones de ejecución:

El actor deberá estar autenticado en el sistema.

Tabla 9. Diseño de caso de prueba. CU Visualizar modelo 3D.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Visualizar modelo 3D.	EC 1.1: Visualizar modelo 3D	Visualizar modelo 3D	El actor ya autenticado selecciona la opción “Modelos 3D”. El sistema muestra la interfaz donde se visualizan los modelos 3D.

3.5.3 Diseño de caso de prueba. CU Crear Escena.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de crear la escena.

Condiciones de ejecución:

Debe estar autenticado en la aplicación.

Capítulo 3. Implementación y Validación del Sistema

Tabla 10. Diseño de caso de prueba. CU Crear Escena.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Crear Escena	EC 1.1 Cargar Imagen	Cargar Imagen	<p>El actor escoge en las categorías existentes el elemento que desee insertar en la escena.</p> <p>El sistema muestra todos los elementos existentes por categorías.</p> <p>El actor arrastra el elemento esta que este dentro del componente creado previamente.</p> <p>El sistema muestra el elemento dentro del componente.</p>
	EC 1.2 Crear Región	Crear Región	<p>El sistema muestra los campos a llenar para crear la región.</p> <p>El actor llena los campos.</p> <p>El sistema muestra la región creada.</p> <p>El sistema muestra los campos a llenar para crear la región.</p>
	EC 1.3 Insertar Área.	Insertar Área.	<p>El sistema muestra el campo para insertar un área dentro de la región creada.</p> <p>El actor llena los campos.</p> <p>El sistema muestra el área insertada dentro de la región.</p>
	EC 1.4 Insertar imagen en la escena.	Insertar imagen en la escena.	<p>El actor escoge la categoría que le corresponde y describe las características del mismo.</p> <p>El actor hace clic en el botón “Seleccionar Archivo”.</p> <p>El sistema muestra la ventana para buscar el elemento a insertar.</p> <p>El actor escoge el elemento y presiona el botón Aceptar.</p> <p>El sistema muestra la ruta donde se encuentra el elemento.</p>

			<p>El actor presiona el botón Subir.</p> <p>El sistema muestra el elemento añadido.</p>
--	--	--	---

3.5.4 Diseño de caso de prueba. CU Gestionar Proyecto.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de insertar, cargar, guardar o eliminar un proyecto determinado.

Condiciones de ejecución:

Debe estar autenticado en la aplicación.

Tabla 11. Diseño de caso de prueba. CU Gestionar Proyecto.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Gestionar Proyecto	EC 1.1 Insertar Proyecto	Insertar Proyecto	<p>El actor hace clic en la opción Nuevo Proyecto.</p> <p>El sistema muestra los campos a llenar.</p> <p>El actor llena los campos correspondientes.</p> <p>Hace clic en el botón Aceptar.</p> <p>El sistema muestra el proyecto creado</p>
	Campos Vacíos	Campos Vacíos	El sistema muestra un mensaje de error:” Campos Obligatorios”.
	EC 1.2 Cargar Proyecto	Cargar Proyecto	<p>El actor hace clic en la opción Cargar/Eliminar Proyecto.</p> <p>El sistema muestra todos los proyectos existentes.</p> <p>El actor selecciona el proyecto que desee.</p> <p>El sistema muestra el proyecto cargado</p>
	EC 1.3 Guardar Proyecto	Guardar Proyecto	<p>El actor presiona el botón Guardar/Cerrar.</p> <p>El sistema muestra todos los proyectos guardados hasta el momento.</p>

	EC 1.4 Eliminar Proyecto	Eliminar Proyecto	El actor presiona el botón  El sistema muestra el proyecto eliminado.
--	---------------------------------	-------------------	---

3.5.5 Diseño de caso de prueba. CU Realizar votación sobre modelo 3D.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de realizar su votación respecto a un modelo determinado.

Condiciones de ejecución:

Debe estar autenticado en la aplicación.

Existir un proyecto cargado.

Tabla 12. Diseño de caso de prueba. CU Realizar votación sobre modelo 3D.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Realizar votación sobre modelo 3D	EC 1.1 Realizar votación sobre modelo 3D	Realizar votación sobre modelo 3D	El actor accede a la opción Modelo 3D. El sistema muestra la interfaz donde se encuentran todos los modelos cargados. El actor escoge el modelo al que le quiere dar su voto. El sistema muestra las cinco estrellas para que vote. El actor dando clic en las estrellas realiza su voto. El sistema muestra las estrellas de color amarillo con el voto correspondiente al modelo seleccionado.

3.6 Consideraciones del capítulo.

En este capítulo se describió el proceso de implementación de acuerdo a los resultados obtenidos una vez que concluyó la fase de Análisis y Diseño. Se mostró la manera de organizar los componentes en el entorno de implementación y las dependencias entre estos. Además se realizaron los casos de pruebas diseñados para realizar el flujo de las mismas, asegurando que las deficiencias puedan ser solucionadas rápidamente por el desarrollador.

Conclusiones.

Referente al trabajo concluido, se puede decir que los objetivos trazados a inicios del mismo han sido cumplidos exitosamente, creando una aplicación Web que cumple estrictamente con los requisitos y ventajas propuestas por el cliente. Dicha aplicación es capaz de proveerle al mismo una interfaz mediante la cual puedan recrear escenas de sus proyectos en forma de maqueta 2D e interactuar con los modelos 3D que van a formar parte de su paseo virtual. Además pretende resolver el problema de interacción y comunicación con el grupo de desarrollo de Paseos Virtuales, convirtiendo al mismo cliente en un integrante más del equipo de desarrollo.

Con el uso de la metodología OpenUp se obtuvo una mayor eficiencia del equipo de desarrollo y a su vez un producto funcional por su condición de metodología ágil, centrada principalmente en el cliente y que presenta iteraciones cortas, además evitó la elaboración de documentos, diagramas e iteraciones innecesarias requeridos por la metodología RUP.

La utilización del lenguaje PHP vinculado al framework jQuery, ambos integrados dentro del CMS Joomla permitieron la creación de un ambiente que agrupa las principales ventajas de los mismos, logrando una aplicación programada sobre un lenguaje robusto y orientado a objetos. A su vez se brinda dinamismo e interacción con los elementos presentes dentro de la interfaz; ejemplo de esto son los efectos de arrastrar y soltar diseños con componentes presentes dentro de la misma.

De igual forma el lenguaje AS3 y el framework Flex integrado con Away3D demostraron la posibilidad de visualizar en los navegadores elementos tridimensionales en tiempo real e interactuar con ellos.

Recomendaciones.

Es importante para futuros desarrollos:

- Crear un componente que permita delimitar los espacios dentro de la escena.
- Realizar actualizaciones en la propuesta de solución actual proponiendo que se utilicen herramientas libres, el autor propone la utilización de Visual Paradigm por Rational Rose y el IDE de desarrollo Eclipse por Flash Builder 4.
- Identificar nuevos requisitos que van surgiendo con el desarrollo de nuevas aplicaciones y que permitirán ir actualizando la propuesta actual.

Referencias Bibliográficas.

ACTIONSCRIPT. 2010. actionscript.org. [Online] 2010.

<http://www.actionscript.org/resources/categories/Articles/Product-Reviews/>.

ADOBE. 2009. Adobe.com. [Online] 07 14, 2009.

http://www.adobe.com/resources/business/rich_internet_apps/.

ALTERNATIVA3D. 2009. alternativaplatform.com. [Online] Alternativa3D, 2009.

<http://alternativaplatform.com/en/about/>.

AMFPHP. 2011. amfphp CODEX. [Online] What is amfPHP?, 2011.

<http://community.silexlabs.org/amfphp/documentation/>.

APACHE. 2011. Apache HTTP Server Project. [Online] Documentation, 2011.

<http://httpd.apache.org/>.

AWAY3D. 2010. away3d.com. [Online] Away3D Flash Engine, 2010.

<http://www.away3d.com/documentation>.

BALDUINO, Ricardo. 2007. eclipse.org. [Online] 09 2007.

<http://www.eclipse.org/epf/general/OpenUP.pdf>.

BUILDER. 2010. Adobe.com. [Online] Adobe Flash Builder 4, 2010.

<http://www.adobe.com/products/flashbuilder/>.

CMS. 2010. plone.org. [Online] What is a CMS?, 2010.

<http://plone.org/documentation/faq/what-is-a-cms>.

DIGITTA. 2011. digitta.com. [Online] 2 2011.

<http://digitta.com/2011/02/vuelvo-deconstruyendo-jquery-mootools-y.html>.

DocForge. 2010. DocForge.com. *DocForge.com*. [Online] 11 25, 2010. [Cited: 1 05, 2011.]

http://docforge.com/wiki/Web_application_framework.

DOCFORGE. 2010. DocForge.com. [Online] Web application framework, 2010.

http://docforge.com/wiki/Web_application_framework.

DOCUMENTATION, MooTools API. 2009. mootools.net. [Online] 2009.

<http://mootools.net/docs/core>.

FLEX. 2009. Adobe.com. [Online] Adobe Flex 4, 2009.

<http://www.adobe.com/products/flex/>.

FRAMEWORKS. 2003. codeproject.com. [Online] Marc Clifton, 11 03, 2003.

<http://www.codeproject.com/KB/architecture/WhatIsAFramework.aspx>.

GROSSMAN, H. 2006. Adobe.com. [Online] ActionScript 3.0 overview, 2006.

<http://www.adobe.com/devnet/actionsript/articles/actionsript3overview.html>.

GUTIÉRREZ, C. 2010. fayerwayer.com. [Online] WebGL: 3D para tu browser, 2010.

<http://www.fayerwayer.com/2009/09/webgl-3d-para-tu-browser/>.

IBM. 2006. IBM.com. [Online] 09 05, 2006. [Cited: 01 05, 2011.]

<http://www.ibm.com/developerworks/library/wa-abilities/index.html>.

IIS. 2011. iis.net. [Online] Overview. What is IIS?, 2011.

<http://www.iis.net/overview>.

James Gosling, Bill Joy. 2005. java.sun.com. [Online] Addison-Wesley, 2005.

http://java.sun.com/docs/books/jls/third_edition/html/j3TOC.html.

JOOMLA. 2011. joomla.org. [Online] Joomla! Official Documentation, 2011.

<http://docs.joomla.org/>.

Krebs, Jochen. 2007. ibm.com. [Online] "The value of RUP certification". IBM. , 01 15, 2007.

<http://www.ibm.com/developerworks/rational/library/jan07/krebs/index.html>.

Kroenke, David M. and David J. Auer. 2007. [Online] Database Concepts 3rd ed New York: Prentice, 2007. http://www.upload.name/Data_Architecture.php.

Lago, Ramiro. 2007. proactiva-calidad.com. [Online] Patrones de diseño software, 04 2007.

http://www.proactiva-calidad.com/java/patrones/index.html#algunos_patrones.

MYSQL. 2010. mysql.com. [Online] MySQL Documentation: MySQL Reference Manuals, 2010.

<http://dev.mysql.com/doc/>.

- PHP. 2009.** PHP Documentation HOWTO. [Online] Stig Sæther Bakken, Gábor Hojtsy , 2009.
<http://doc.php.net/php/dochowto/>.
- POSTGRE. 2006.** postgresql.org. [Online] Documentation, 2006.
<http://www.postgresql.org/docs/>.
- RATIONAL. 2010.** ibm.com. [Online] Rational Software, 2010.
<http://www-01.ibm.com/software/rational/>.
- Resig, John. 2010.** jquery.com. [Online] How jQuery Works, 2010.
http://docs.jquery.com/How_jQuery_Works.
- SENCHA. 2011.** sencha.com. [Online] Ext JS API Documentation, 2011.
<http://docs.sencha.com/ext-js/4-0/>.
- SERVER. 2010.** webdevelopersnotes.com. [Online] What is web server - a computer of a program?, 2010.
http://www.webdevelopersnotes.com/basics/what_is_web_server.php.
- VERA, P. 2009.** adictosalared.com. [Online] Aptana Studio: excelente entorno gratuito para desarrollo web, 09 2009.
<http://www.adictosalared.com/aptana-studio-excelente-entorno-gratuito-para-desarrollo-web/>.
- W3C. 2008.** w3.org. [Online] Extensible Markup Language (XML)Tim Bray, 11 26, 2008.
<http://www.w3.org/TR/REC-xml/>.
- . **2010.** w3.org. [Online] Bert Bos, 12 2010.
<http://www.w3.org/Style/CSS/>.
- . **1995.** W3C HTML. *What is HTML?* [Online] Tim Berners Lee, Dan Connolly , 11 1995.
<http://www.w3.org/wiki/HTML/Specifications>.

Glosario de Términos

RIA: Aplicaciones Enriquecidas de Internet (del inglés “Rich Internet Applications”) son un nuevo tipo de aplicaciones con más ventajas que las tradicionales aplicaciones Web. Esta surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales.

3D: En computación, las tres dimensiones son el largo, el ancho y la profundidad de una imagen. Técnicamente hablando el único mundo en 3D es el real, la computadora sólo simula gráficos en 3D, pues, en definitiva toda imagen de computadora sólo tiene dos dimensiones, alto y ancho (resolución).

CASE (Computer Aided Software Engineering): Bajo el término de Ingeniería de Software Asistida por Ordenador se incluyen una serie de herramientas, lenguajes y técnicas de programación que permiten la generación de aplicaciones de manera semiautomática. Las herramientas CASE liberan al programador de parte de su trabajo y aumentan la calidad del programa a la vez que disminuyen sus posibles errores.

UML: El Lenguaje Unificado de Modelado (“Unified Modeling Language” por sus siglas en inglés) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Framework: En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Away3D: es un motor gráfico 3D de código abierto, escrito para la plataforma Adobe Flash en ActionScript 3 el cual se ejecuta en los navegadores modernos que utilizan Adobe Flash Player. Una de sus principales características es la compatibilidad con la transformación de objetos jerárquicos con funciones tales como posición, rotación, escala e iluminación en tiempo real.

Anexos.

Anexo1. Descripciones textuales de los casos de uso.

Tabla 13.CU Autenticar Usuario.

Caso de uso	
CU-1	Autenticar Usuario.
Propósito	El caso de uso se encargará de permitirle al actor autenticarse en la aplicación.
Prioridad	Crítica.
Actores: Usuario del Sistema.	
Resumen: El caso de uso se inicializa cuando el actor introduzca sus datos para así acceder al sistema.	
Referencias	RF1.
Precondiciones	El actor deberá autenticarse en la aplicación
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
El actor acciona la página principal del sistema.	El sistema muestra la interfaz visible para todos los usuarios, además muestra un apartado para la autenticación.
2. El actor introduce su usuario y contraseñas y hace clic en el botón aceptar.	2.1 El sistema muestra la interfaz de usuario correspondiente al actor autenticado.

Tabla 14.CU Gestionar Usuario.

Caso de uso	
CU-2	Gestionar Usuario.
Propósito	El caso de uso se encargará de permitirle al actor la gestión de usuarios en la aplicación.
Prioridad	Crítica.
Actores: Administrador.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de gestionar (insertar, modificar o eliminar) uno o varios usuarios del sistema.	

Referencias	RF2.
Precondiciones	El administrador deberá estar autenticado en el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor ya autenticado selecciona la opción Gestor de usuarios.	1.1 El sistema muestra la interfaz donde pide el tipo de modificación a realizar. Si el actor desea insertar un usuario ir a la <u>Acción1</u> . Si el actor desea modificar un usuario ir a la <u>Acción2</u> . Si el actor desea eliminar un usuario ir a la <u>Acción3</u> .
2. El actor escoge una opción y hace clic en el botón aceptar.	2.1 El sistema muestra la información relacionada con la opción.
Acción 1 Insertar Usuario	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona la opción "Nuevo".	2. El sistema muestra el formulario con todos los campos a llenar.
3. El actor presiona la opción "Guardar".	4. Termina el caso de uso.
Acción 2 Modificar Usuario	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona el usuario que desea modificar y presiona la opción "Editar".	2. El sistema muestra el formulario con todos los datos del usuario seleccionado.
3. El actor modifica el campo que desee y presiona la opción "Guardar".	4. Termina el caso de uso.
Acción 3 Eliminar Usuario	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
	1. El sistema muestra los usuarios existentes.
2. El actor presiona la opción "Borrar".	3. Termina el caso de uso.

Tabla 15.CU Visualizar galería de imágenes.

Caso de uso	
CU-7	Visualizar galería de imágenes.
Propósito	El caso de uso se encargará de permitirle al actor la visualización de la galería de imágenes.
Prioridad	Secundario.
Actores: Usuario del sistema.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de visualizar la galería de imágenes.	
Referencias	RF7.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor accede a la opción Galería.	2. El sistema muestra la galería de imágenes existentes en la aplicación.
3. El actor hacer clic en cualquiera de las carpetas.	4. El sistema muestra las imágenes que contiene la misma.
5.	6. Termina el caso de uso.

Tabla 16.Gestionar imagen.

Caso de uso	
CU-8	Gestionar Imagen.
Propósito	El caso de uso se encargará de permitirle al actor la gestión de imagen.
Prioridad	Crítica.
Actores: Administrador.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de gestionar las imágenes de la galería.	
Referencias	RF 8.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema

<ol style="list-style-type: none"> 1. El actor accede la página de administración de Joomla. 2. El actor accede al menú Componentes, luego al submenú MorfeoShow y presiona clic en la opción Galleries. 	<ol style="list-style-type: none"> 3. El sistema muestra la interfaz donde pide el tipo de modificación a realizar. <ul style="list-style-type: none"> • Si el actor desea insertar una imagen ir a la <u>Acción 1</u>. • Si el actor desea publicar una imagen ir a la <u>Acción 2</u>. • Si el actor desea despublicar una imagen ir a la <u>Acción 3</u>. • Si el actor desea eliminar una imagen ir a la <u>Acción 4</u>.
Acción 1 Insertar Imagen	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor selecciona la carpeta donde insertar la imagen y presiona clic sobre la misma	2. El sistema muestra la interfaz para insertar la imagen.
3. El actor presiona el botón seleccionar archivo y busca la imagen.	4. El sistema muestra la imagen seleccionada.
5. El actor presiona clic en el botón Aplicar.	
Acción 2 Publicar Imagen	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona clic en la opción Imágenes 	2. El sistema muestra todas las imágenes pertenecientes a la carpeta seleccionada.
3. El actor presiona clic sobre el botón publicar.	4. El sistema muestra la imagen publicada.
Acción 3 Despublicar Imagen.	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona clic en la opción	2. El sistema muestra todas las imágenes

Imágenes 	pertenecientes a la carpeta seleccionada.
3. El actor presiona clic sobre el botón despublicar.	4. El sistema muestra la imagen despublicada.
Acción 4 Eliminar Imagen.	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor presiona clic en la opción Imágenes 	2. El sistema muestra todas las imágenes pertenecientes a la carpeta seleccionada
3. El actor hace clic en el botón “Borrar”  .	4. El sistema las imágenes restantes.

Tabla 17.CU Administrar comentarios sobre modelo 3D

Caso de uso	
CU-6	Administrar comentarios sobre modelo 3D.
Propósito	El caso de uso se encargará de permitirle al actor la administración de comentarios de los modelos 3D.
Prioridad	Crítica.
Actores: Usuario del sistema.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de administrar los comentarios de los modelos 3D.	
Referencias	RF6
Precondiciones	El actor deberá estar autenticado en el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor ya autenticado selecciona la opción “Modelos 3D”.	2. El sistema muestra la interfaz donde pide el tipo de modificación a realizar. <ul style="list-style-type: none"> • Si el actor desea insertar un comentario ir a la <u>Acción 1</u>.

	<ul style="list-style-type: none"> • Si el actor desea eliminar un comentario ir a la <u>Acción 2</u>.
Acción 1 Insertar comentario sobre modelo 3D	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor inserta el comentario y presiona el botón Aceptar.	2. El sistema muestra la interfaz con el comentario insertado.
Acción 2 Eliminar comentario sobre modelo 3D	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor elimina el comentario y presiona el botón Eliminar.	2. El sistema muestra la interfaz luego de eliminar el comentario.

Tabla 18.CU Interactuar con modelo 3D.

Caso de uso	
CU-5	Interactuar con modelo 3D.
Propósito	El caso de uso se encargará de permitirle al actor la interacción con los modelos 3D.
Prioridad	Crítica.
Actores: Cliente.	
Resumen: El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de interactuar con los modelos 3D.	
Referencias	RF5.
Precondiciones	El actor deberá estar autenticado en el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor ya autenticado selecciona la opción "Modelos 3D".	2. El sistema muestra la interfaz donde pide el tipo de modificación a realizar.

	<ul style="list-style-type: none"> • Si el actor desea rotar un modelo ir a la <u>Acción 1</u>. • Si el actor desea escalar un modelo ir a la <u>Acción 2</u>.
Acción 1 Rotar modelo 3D	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor rota el modelo que seleccione con el movimiento del mouse.	2. El sistema muestra la interfaz con el modelo rotado
Acción 2 Escalar modelo 3D	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El actor escala el modelo seleccionado dando mediante el componente slider.	2. El sistema muestra la interfaz con el modelo escalado.

Anexo2. Diseños de prueba.

Diseño de caso de prueba. CU Autenticar Usuario.

Descripción general:

El caso de uso se inicializa cuando el actor introduzca sus datos para así acceder al sistema.

Condiciones de ejecución:

El actor deberá autenticarse en la aplicación.

Tabla 19. Diseño de caso de prueba. CU Autenticar Usuario.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central

SC1. Autenticar Usuario.	EC 1.1: Autenticar Usuario.	El actor acciona la página principal para introducir sus datos.	<ol style="list-style-type: none"> 1. El actor acciona la página principal del sistema. 2. El sistema muestra la interfaz visible para todos los usuarios, además muestra un apartado para la autenticación. 3. El actor introduce usuario, contraseña y hace clic en el botón aceptar. 4. El sistema muestra la interfaz de usuario correspondiente al actor autenticado.
---------------------------------------	--	---	--

Diseño de caso de prueba. CU Gestionar Usuario.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de gestionar uno o varios usuarios, llámese insertar, modificar o eliminar algún usuario del sistema.

Condiciones de ejecución:

El actor deberá estar autenticado en el sistema.

Tabla 20. Diseño de caso de prueba. CU Gestionar usuario.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Gestionar usuario.	EC 1.1: Insertar usuario.	Insertar usuario.	<ol style="list-style-type: none"> 1. El actor presiona la opción "Nuevo". 2. El sistema muestra el formulario con todos los campos a llenar. 3. El actor presiona la opción "Guardar".
	EC 1.2: Modificar Usuario.	Modificar Usuario.	<ol style="list-style-type: none"> 1. El actor selecciona el usuario que desea modificar y presiona la opción "Editar". 2. El sistema muestra el formulario con todos los datos del usuario seleccionado. 3. El actor modifica el campo que desee y presiona la opción "Guardar".

	EC 1.3: Eliminar Usuario.	Eliminar Usuario	<ol style="list-style-type: none"> 1. El sistema muestra los usuarios existentes. 2. El actor presiona la opción "Borrar".
--	-------------------------------------	------------------	--

Diseño de caso de prueba. CU Visualizar galería de imágenes.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de visualizar la galería de imágenes.

Condiciones de ejecución:

Tabla 21. Diseño de caso de prueba. CU Visualizar galería de imágenes.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Visualizar galería de imágenes	EC 1.1 Visualizar galería de imágenes	Visualizar galería de imágenes	<ol style="list-style-type: none"> 1. El actor accede a la opción Galería. 2. El sistema muestra la galería de imágenes existentes en la aplicación. 3. El actor hacer clic en cualquiera de las carpetas. 4. El sistema muestra las imágenes que contiene la misma.

Diseño de caso de prueba. CU Gestionar Imagen.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de gestionar las imágenes de la galería.

Condiciones de ejecución:

Tabla 22. Diseño de caso de prueba. CU Gestionar Imagen.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Gestionar Imagen	EC 1.1 Insertar Imagen	Insertar Imagen	<ol style="list-style-type: none"> 1. El actor selecciona la carpeta donde insertar la imagen y presiona clic sobre la misma. 2. El sistema muestra la interfaz para insertar la imagen. 3. El actor presiona el botón seleccionar archivo y busca la imagen. 4. El sistema muestra la imagen seleccionada. 5. El actor presiona clic en el botón Aplicar.
	EC 1.2 Publicar Imagen	Publicar Imagen	<ol style="list-style-type: none"> 1. El actor presiona clic en la opción Imágenes . 2. El sistema muestra todas las imágenes pertenecientes a la carpeta seleccionada. 3. El actor presiona clic sobre el botón publicar. 4. El sistema muestra la imagen publicada.
	EC 1.3 Despublicar Imagen	Despublicar Imagen	<ol style="list-style-type: none"> 1. El actor presiona clic en la opción Imágenes 2. El sistema muestra todas las imágenes pertenecientes a la carpeta seleccionada. . 3. El actor presiona clic sobre el botón despublicar. 4. El sistema muestra la imagen despublicada.

	<p>EC 1.4 Eliminar Imagen</p>	<p>Eliminar Imagen</p>	<ol style="list-style-type: none"> 1. El actor presiona clic en la opción  Imágenes. 2. El sistema muestra todas las imágenes pertenecientes a la carpeta seleccionada 3. El actor hace clic en el botón "Borrar" . 4. El sistema las imágenes restantes.
--	--	------------------------	---

Diseño de caso de prueba. CU Interactuar con el modelo 3D.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de interactuar con los modelos 3D.

Condiciones de ejecución:

El actor deberá estar autenticado en el sistema.

Tabla 23. Diseño de caso de prueba. CU Interactuar con el modelo 3D.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
<p>SC1: Interactuar con el modelo 3D</p>	<p>EC 1.1: Rotar modelo</p>	<p>Rotar modelo</p>	<p>El actor rota el modelo que seleccione con el movimiento del mouse. El sistema muestra la interfaz con el modelo rotado.</p>
	<p>EC 1.2: Escalar modelo</p>	<p>Escalar modelo</p>	<p>El actor escala el modelo seleccionado mediante el componente slider. El sistema muestra la interfaz con el modelo escalado.</p>

Diseño de caso de prueba. CU Administrar comentario sobre modelo 3D.

Descripción general:

El caso de uso se inicializa cuando el actor accede al sistema con el objetivo de administrar los comentarios sobre los modelos 3D.

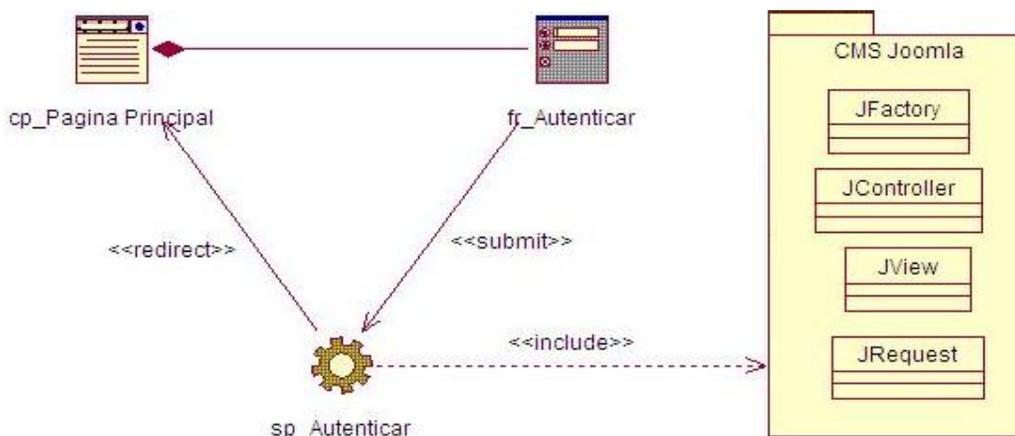
Condiciones de ejecución:

El actor deberá estar autenticado en el sistema.

Tabla 24. Diseño de caso de prueba. CU Administrar comentario sobre modelo 3D.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
SC1: Administrar comentarios sobre modelo 3D	EC 1.1: Insertar comentario sobre modelo 3D	Insertar comentario sobre modelo 3D	El actor inserta el comentario y presiona el botón Aceptar. El sistema muestra la interfaz con el comentario insertado.
	EC 1.2: Eliminar comentario sobre modelo 3D	Eliminar comentario sobre modelo 3D	El actor elimina el comentario y presiona el botón Eliminar. El sistema muestra la interfaz luego de eliminar el comentario.

Anexo 3. Diagrama de clases del diseño (Autenticar Usuario).



Anexo 4. Diagrama de secuencia: Autenticar Usuario.

