UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

SISTEMA DE GESTIÓN DEL PORTAFOLIO DE LOS RECURSOS HUMANOS Y TAREAS DE ESCENARIOS 3D.

Trabajo de Diploma para optar por el
Título de Ingeniero en Ciencias Informáticas

Autor
Julio Cesar Del Castillo Arias

Tutor
Ing. Gadied A. Carrero Sotolongo

Co-tutor
Ing. Loyda Cárdenas Rey

Ciudad de La Habana, junio 2011
"Año 53 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ___ días del mes de _____del año _____

_____
Julio Cesar Del Castillo Arias
Autor

_____
Ing. Gadied A. Carrero Sotolongo
Tutor

_____
Ing. Loyda Cárdenas Rey
Co-Tutor

Datos de Contacto

Tutor: Ing. Gadied A. Carrero Sotolongo.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas.

Título: Ingeniero en Telecomunicaciones y Electrónica.

Categoría Docente: Asistente.

E-mail: gadied@uci.cu

Graduado de la Universidad Central Marta Abreu de las Villas.

Co-Tutor: Ing. Loyda Cárdenas Rey.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas.

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente:

E-mail: lrey@uci.cu

Graduado de la Universidad de Ciencias Informáticas.

Dedicatoria y Agradecimiento:

Quiero dedicar esta tesis en especial a mis padres que me han apoyado a lo largo de estos años motivándome a seguir adelante, a ellos que me han enseñado que no importan las veces que tropiezas y te caes en la vida, sino las veces que te levantas.

También quiero dedicar esta tesis a todos aquellos que de una forma u otra han estado ahí ayudándome a seguir adelante…

Quiero agradecer a todos los que de una forma u otra me han me han brindado su apoyo, a mi familia a mis amigos, a la uci, a Fidel por haber sido el creador de esta majestuosa Universidad y sobre todo a mis mismo…

Resumen.

Este trabajo de diploma se realizó con el objetivo de gestionar el portafolio de los recursos humanos y tareas de Escenarios 3D.

Para darle solución al objetivo de la investigación se analizan las metodologías de desarrollo, tecnologías, técnicas, lenguajes y herramientas más utilizadas, a partir de las cuales se hace una propuesta sobre las que deben ser empleadas en la realización del Sistema de gestión del portafolio de los recursos humanos y tareas de Escenarios 3d.

Se realizó la fase de modelación del negocio de la metodología OpenUp y posteriormente el diseño e implementación del sistema a partir de la captura de requisitos.

La aplicación se desarrolló usando el framework de desarrollo ExtJS para realizar la interfaz, como lenguaje de programación se usó el lenguaje libre PHP y como servidor web el Apache.

El Sistema de gestión del portafolio de los recursos humanos y tareas de Escenarios 3d brinda la posibilidad de manejar de manera sencilla y eficiente la información que se gestiona en Escenarios 3D. Permite la visualización de modelos 3d en la web permitiendo que varios usuarios puedan observar estos modelos desde cualquier parte de la universidad con solo un navegador web.

Este sistema brinda una solicitud de servicios para posibles clientes, los cuales pueden ver los portafolios de trabajo de los realizadores, revisor. Además de ver el estado en que se está realizando su solicitud a través del visor 3d.

Palabras Clave: Portafolio, framework, interfaz, modelos 3d, visualización de modelos 3d.

Indicé de contenido.

Índice de figuras.

Índice de Tablas.

Introducción

La web nació bajo la idea de libre flujo de información, de modo que todos pudieran acceder. Su padre fue Tim Berners-Lee y su lugar de nacimiento el Consejo Europeo para la Investigación Nuclear (CERN). El CERN tenía un ambiente parecido a la web, en el cual muchas personas coexistían y trabajaban en diferentes proyectos.

Tim creó un sitio que describía como funcionaba la web y lo publicó en el primer servidor: “info.cern.ch”. Europa demostró muy poco interés en la web en aquel entonces, es por eso que los estadounidenses fueron los primeros en publicar servidores. Tim añadió enlaces a estos servidores desde su directorio conocido como "*Virtual Library*" o "Biblioteca Virtual" y así nació la *World Wide Web* (Red Mundial, internet, en adelante WWW).

Internet se ha convertido en los últimos 15 años en la herramienta tecnológica más revolucionaria y poderosa de todas, influyendo en prácticamente todos los niveles de la actividad humana. Paralelamente al surgimiento de internet se fue buscando una forma para sacarle provecho a este gran sistema que ocupa todo el globo terráqueo y por lo cual vinieron las famosas aplicaciones online o aplicación basada en navegadores.[1]

Las aplicaciones online son programas que se diseñan para funcionar a través de un navegador de internet. Estas residen en un servidor y su ejecución requiere disponer de un PC con conexión a internet, un navegador como Internet Explorer, Mozilla Firefox, Opera y por supuesto que esté funcionando en el servidor que la hospeda.

Estas aplicaciones proporcionan gran movilidad, dado que pueden ejecutarse desde cualquier ordenador con conexión a internet. La información que manejan se accede a través de la web, motivo por el cual son especialmente interesantes para desarrollar aplicaciones multiusuario basadas en la compartición de información. El cliente o usuario que utiliza la aplicación no necesita tener un ordenador de grandes prestaciones para poder ejecutarla.

Con el paso de los años estas aplicaciones se han ido perfeccionando y han surgido aplicaciones online que bien pueden sustituir o al menos ir de la mano con los programas de escritorio existentes. Su uso se hace cada vez más imprescindible en la gestión de contenidos de las empresas, solicitudes de servicios y sus intercambios con los clientes.

La Universidad de las Ciencias Informáticas (UCI) surgida al calor de la batalla de ideas, es un proyecto creado por la Revolución con el objetivo de formar ingenieros informáticos comprometidos con la patria, este centro no está exento al gran avance de la informática; es una de las instituciones con más avances tecnológicos en todo el país y que produce software para su posterior comercialización.

La UCI reconoce las ventajas que ofrece una aplicación web y dentro de la universidad el proyecto Escenarios 3D de la Facultad 5. Este proyecto se encarga entre otras cosas de crear escenarios tridimensionales y plantean la necesidad de tener una aplicación que les permita gestionar las tareas y productos que aquí se realizan. Actualmente la UCI cuenta con una aplicación que gestiona las tareas que se realizan en los diferentes proyectos que la componen, el "redmine" pero esta aplicación no es eficiente para los proyectos de la línea de producción denominada "Imágenes y Videos Basados en Rendering" de la Facultad 5 a la cual pertenece este proyecto, ya que no permite visualizar objetos tridimensionales en la web, lo que provoca que el procesos de gestión de las tareas del área de Escenarios 3D sea incompleto además se vea afectado en tiempo y forma, ya que el evaluador tiene que tener los software en los que se desarrollaron dicha tarea instalados en su PC para poder revisarlas.

Dada la situación problémica anteriormente presentada se define el siguiente **problema científico:** ¿Cómo gestionar los artefactos que se realizan en Escenarios 3D y el portafolio de trabajo de sus integrantes, además de proporcionar una solicitud de servicio para los clientes?

El **objeto de la investigación** se centra en el proceso de gestión de la información y las solicitudes de servicio enfocándose en el **campo de acción** El proceso de gestión de los artefactos, las tareas, el portafolio y las solicitudes generados en el área de Escenarios 3D.

El **objetivo** Implementar una aplicación web que gestione los artefactos generados en Escenarios 3D, el portafolio de trabajo de sus integrantes y un sistema de solicitud del servicio que presta el proyecto.

Como **idea a defender,** si se implementa la aplicación web para gestionar los artefactos de Escenarios 3D, el portafolio de trabajo de la comunidad y las solicitudes de los clientes se obtendrá un mayor control de los productos y se agilizará el proceso de gestión de las tareas.

Tareas de investigación.

- Definición del marco teórico a partir del estado del arte.
- Descripción de los antecedentes de los sistemas de gestión.
- Caracterización de los sistemas de gestión.
- Definición de la metodología del software a utilizar.
- Identificación los lenguajes de programación más adecuado para la investigación.
- Definición de las funcionalidades del sistema.
- Documentación sobre la arquitectura de software y patrones de diseño.
- Definición de la arquitectura sobre la cual estará basado el producto.
- Descripción de la estructura del sistema de gestión y los servicios de mensajería.
- Implementación del sistema.

Para el desarrollo de la investigación y elaboración de este trabajo se tomará en cuenta la utilización de varios métodos científicos de investigación como:

Analítico-Sintético: permitirá analizar documentación científica sobre los sistemas de gestión, el uso de solicitudes, portafolios y sintetizar la información construyendo el marco teórico de este trabajo de diploma

Histórico-Lógico: Se utilizará para recorrer una secuencia de hechos que describen el surgimiento y evolución de la gestión, las solicitudes, portafolio, y servicio de mensajería.

La tesis estará estructurada de la siguiente manera:

Capítulo 1. Fundamentación teórica.

Este capítulo se centra en las tecnologías que se utilizaran para el desarrollo de esta investigación y se abordará la fundamentación teórica para el desarrollo de la aplicación.

Capítulo 2. Características de la solución propuesta.

En el Capítulo 2 se describen las particularidades que definirán el sistema, teniendo en cuenta las tecnologías a emplear para su elaboración, los estándares a utilizar, los Patrones de Diseño y Arquitectura más convenientes de acuerdo a los resultados que se esperan entre otros aspectos, siempre respetando las necesidades del cliente.

Capítulo 3. Diseño del sistema.

Aquí se describe la puesta en práctica de la construcción de la solución propuesta y se desarrolla el diseño del sistema definiéndose los diagramas de clases de diseño y los diagramas de interacción siguiendo un flujo de proceso. El diagrama de secuencia y el modelo de bases de datos.

Capítulo 4. Implementación.

En el siguiente capítulo se representarán un conjunto de diagramas que reflejan los componentes por los que está constituido el sistema, así como la distribución física de los mismos facilitando la comprensión de la estructura y composición de la aplicación.

1

Capítulo 1. Fundamentación teórica.

1.1. Introducción.

Este capítulo se centra en las tecnologías que se utilizaran para el desarrollo de esta investigación y se abordará la fundamentación teórica para el desarrollo de la aplicación. Para ello se realizará un meticuloso análisis de las aplicaciones similares existentes relacionadas con el campo de acción, un estudio detallado de las metodologías de desarrollo de software, de los lenguajes de modelado y programación, de las herramientas y de las plataformas de desarrollo que las soportan, de los diferentes tipos de servidores web y de los gestores de base de datos, con el objetivo de proponer la más adecuada para la solución del problema.

1.2. ¿Qué son los sistemas de gestión?

Los sistemas de gestión son una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización. Ayuda a alcanzar los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado. [2]

1.3. ¿Por qué los sistemas de gestión son necesarios?

En la actualidad las empresas se enfrentan a muchos retos, significativos, entre ellos:

- Rentabilidad
- Competitividad
- Globalización
- Velocidad de los cambios
- Capacidad de adaptación
- Crecimiento
- Tecnología

Equilibrar estos y otros requisitos empresariales puede constituir un proceso difícil y desalentador. Es aquí donde entran en juego los sistemas de gestión, al permitir aprovechar y desarrollar el potencial existente en la organización.

La implementación de un sistema de gestión eficaz puede ayudar a:

- Gestionar los riesgos sociales, medioambientales y financieros
- Mejorar la efectividad operativa
- Reducir costos
- Aumentar la satisfacción de clientes y partes interesadas
- Proteger la marca y la reputación
- Lograr mejoras continuas
- Potenciar la innovación
- Eliminar las barreras al comercio
- Aportar claridad al mercado

El uso de un sistema de gestión le permite renovar constantemente su objetivo, sus estrategias, sus operaciones y niveles de servicio [3]

1.4. Metodologías de desarrollo de Software estudiadas.

Una **metodología de desarrollo de software** es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software. Una metodología está compuesta por:

- Cómo dividir un proyecto en etapas.
- Qué tareas se llevan a cabo en cada etapa.
- Qué restricciones deben aplicarse.
- Qué técnicas y herramientas se emplean.
- Cómo se controla y gestiona un proyecto.

En lo que a proceso de desarrollo respecta se conocen hoy los llamados métodos pesados (conocidos también como tradicionales, robustos o no ágiles) y los llamados métodos ligeros (conocidos como ágiles). Los métodos pesados están dirigidos al control del proceso en toda su magnitud demostrando ser efectivos y necesarios en súper-proyectos (respecto

a tiempo y recurso). Mientras que los ágiles tratan de adaptarse a la realidad del propio desarrollo o sea son menos orientados al documento y más al código.[4]

1.5. 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; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

OpenUp está caracterizado por cuatro principios básicos interrelacionados, a saber:

1. Colaboración para unificar intereses y compartir conocimientos.

2. Equilibrio de prioridades competentes a maximizar el valor de los involucrados con el resultado del proyecto.

3. Enfoque en la articulación de la arquitectura.

4. Desarrollo continuo para obtener realimentación y realizar las mejoras respectivas. OpenUp/Basic se centra en articular la arquitectura para facilitar la colaboración técnica, reducir el riesgo y minimizar el sobreesfuerzo 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. [5]

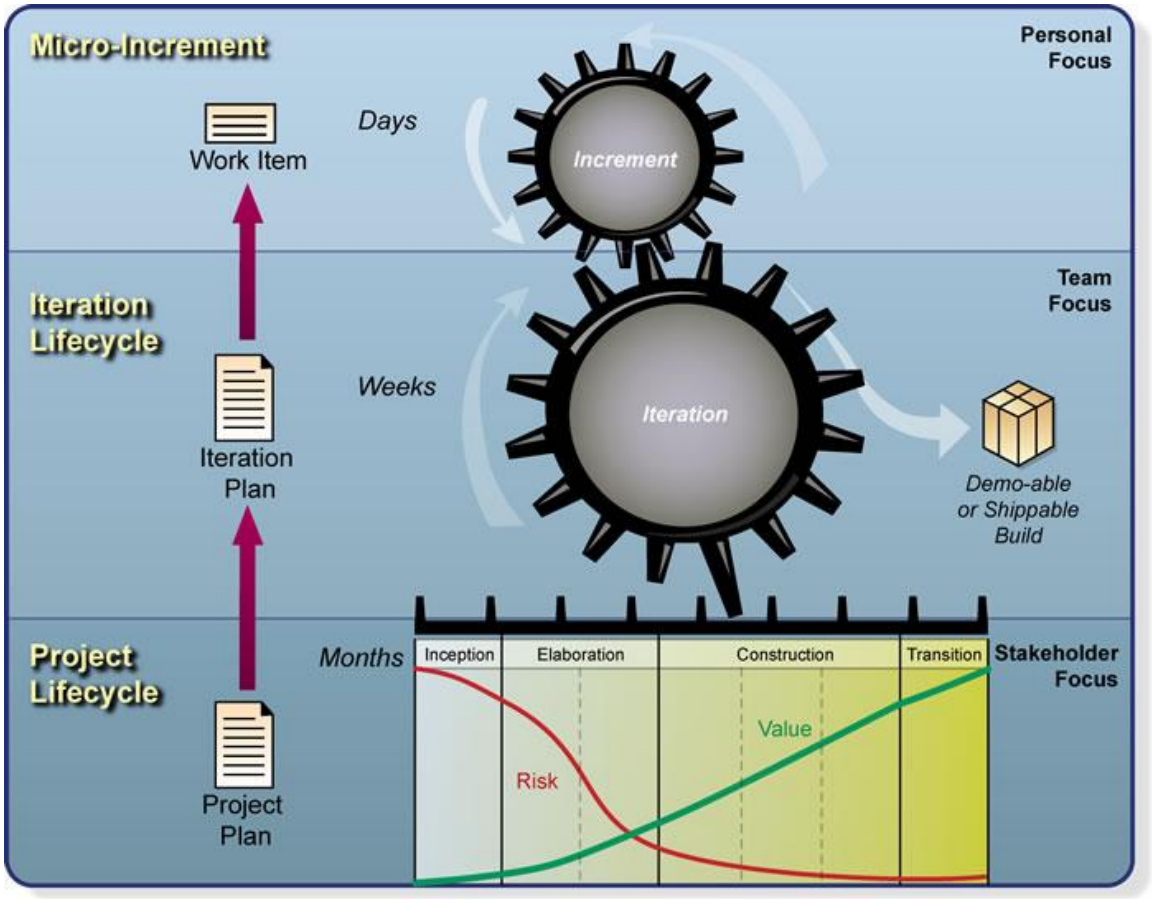

FIGURA 1: FASES DE OPENUP.

1.6. Rational Unified Process (RUP):

El Proceso Unificado Racional, *Rational Unified Process* en inglés, y sus siglas RUP, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí.[6]

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto.

RUP como proceso de desarrollo:

Es explícito en la definición de software y su trazabilidad, es decir, contempla en relación causal de los programas creados desde los requerimientos hasta la implementación y pruebas.

RUP identifica claramente a los profesionales (actores) involucrados en el desarrollo del software y sus responsabilidades en cada una de las actividades.

Fases de desarrollo del software:

- Inicio.
- Elaboración.
- Construcción.
- Transición.

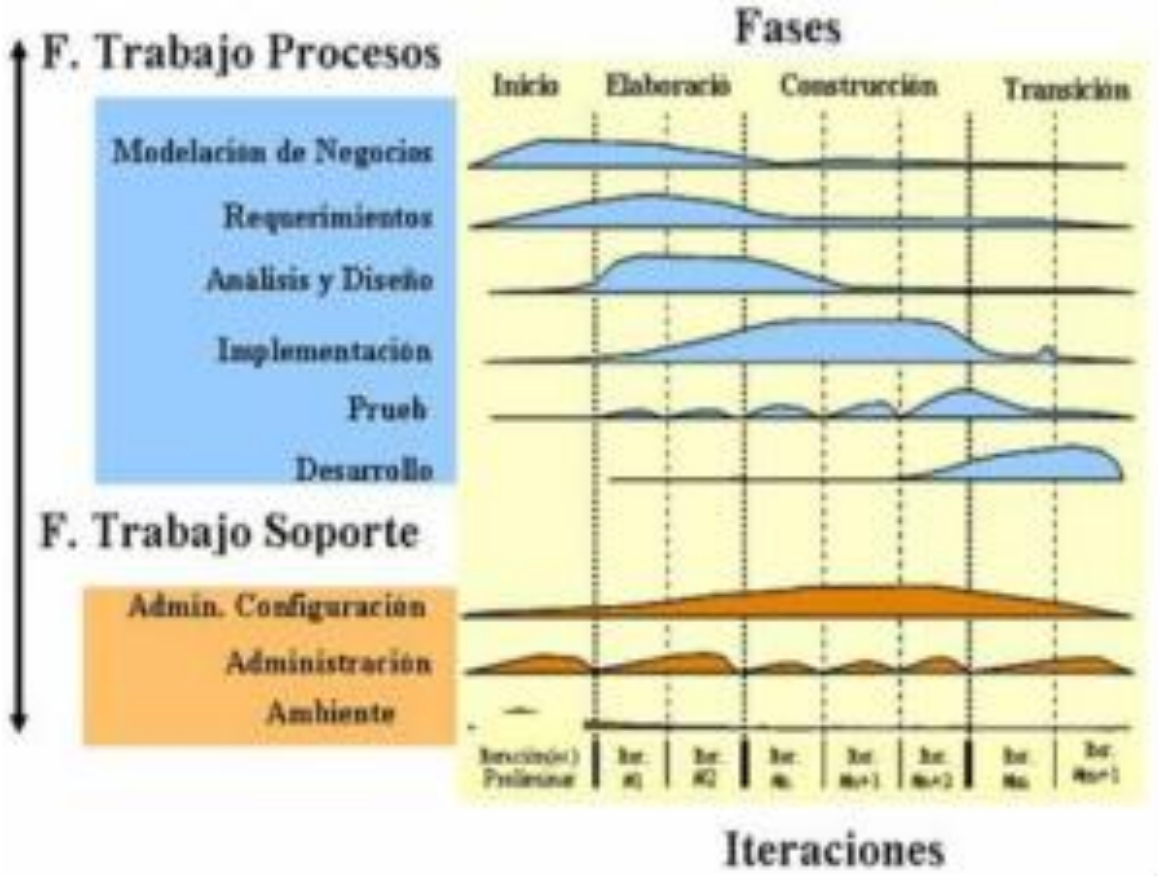

FIGURA 2: FLUJOS DE TRABAJO DE RUP.

1.7. ¿Por qué OpenUp/Basic?

Para el desarrollo de la investigación se decidió utilizar como metodología de desarrollo de software OpenUp/Basic porque es un proceso interactivo de desarrollo de software simplificado, completo y extensible. Es un proceso para pequeños equipos de desarrollo, que valoran los beneficios de la colaboración y de los involucrados con el resultado del proyecto, por encima de formalidades innecesarias.

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.

1.8. Herramientas Case (CASE).

Las **herramientas CASE** (***C**omputer **A**ided **S**oftware **E**ngineering*, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. [7]

1.9. Unified Modeling Language (UML).

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Hoy en día, está considerado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. [8]

Los principales beneficios de UML son:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.

- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

1.10. Visual Paradigm.

Visual Paradigm es una herramienta CASE que utiliza "UML": como lenguaje de modelado. Se caracteriza principalmente por su robustez, usabilidad y portabilidad. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Esta herramienta permite realizar ingeniería tanto directa como inversa, a partir de un modelo relacional en *SQL Server*, *MySQL*, *PostgresSQL*, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Soporta múltiples usuarios trabajando sobre el mismo proyecto, permite control de versiones y genera la documentación automáticamente en formatos como web o .PDF. Es libre y multiplataforma.[9]

1.11. Rational Rose.

E *Rational Rose* es una herramienta de modelado visual con plataforma independiente, que ayuda a la comunicación entre los miembros del equipo. Utiliza UML como lenguaje de modelado. Permite Especificar, Analizar y 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. Mantiene la consistencia de los modelos del sistema de software. Permite el chequeo de la sintaxis UML y realizar ingeniería inversa. Posibilita la generación de documentación automáticamente y la generación de código a partir de los modelos.[10]

1.12. Fundamentación de la herramienta CASE a utilizar.

Tanto Visual Paradigm como Rational Rose son herramientas CASE que utilizan el lenguaje UML para el modelado visual de sistemas. Para el modelado de nuestra aplicación se decidió utilizar la herramienta CASE *Visual Paradigm* debido a que: cuenta con un amigable entorno de desarrollo, es libre y multiplataforma características de suma importancia ya que la universidad cuenta con su licencia y que en un por cierto elevado de computadoras se encuentran instalados los sistemas operativos Windows y Linux incluso diferentes versiones de éstos.

1.13. Tipos de aplicaciones estudiadas:

Entre los tipos de aplicaciones que se estudiaron para dar solución al problema científico planteado se encuentran las aplicaciones web y las de escritorio.

1.14. Aplicaciones Web.

En la ingeniería de software se denomina **aplicación web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los *webmails*, *wikis*, *weblogs*, tiendas en línea y la propia Wikipedia que son ejemplos bien conocidos de aplicaciones web.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.[11]

1.15. Aplicaciones de Escritorio.

Toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows, GNU/Linux o Mac se le denomina: **Aplicación de escritorio**. Este tipo de aplicación se desarrolla generalmente para cubrir necesidades específicas de las empresas como la contabilidad o gestión de personal. [12]

Entre las ventajas de las aplicaciones de escritorio se destacan:

- Mayor capacidad gráfica visual.
- Menor tiempo de respuesta (aplicación más rápida).
- Mayor personalización.

El problema principal radica en que el desarrollo de este tipo de aplicación sobre una plataforma, normalmente, implica que "no" pueda ser ejecutada en otras. El duplicado de los datos por la falta de centralización de los mismos. Implica su instalación en cada uno de los clientes. Trae problemas a la hora de realizar alguna actualización o corrección al programa debido a que las instalaciones están distribuidas. Dificultad para configurar las estaciones ya que cada usuario tiene necesidades diferentes.

Existen aplicaciones de escritorio con tecnología web y GNU Consultores se ha especializado en su desarrollo basadas en tecnologías web como: AJAX, DHTML y CSS.

1.16. Fundamentación del tipo de aplicación a utilizar.

Por parte del equipo de desarrollo se decidió desarrollar una aplicación web puesto que:

- Se reducen los gastos en gran medida: al ser una aplicación web no existe la necesidad de distribuirlo ni instalarlo.
- Solo se necesita de un navegar Web para acceder a la aplicación: solo bastará con tener Internet Explorer o Mozilla Firefox instalado en su PC.
- Su mantenimiento se concentra en el servidor: Solo tendrá un administrador. Lo que posibilitará que el cliente cuente con las actualizaciones en el instante en que se realizan y que no haya un gran número de personas dedicadas a su mantenimiento.
- Permite la navegación desde cualquier punto: lo que facilita el acceso desde cualquier sitio de la Universidad.

1.17. Servidores Web.

Básicamente, un servidor web es un programa que sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. [13]

Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto.

1.18. Servidor Apache.

El **servidor HTTP Apache** es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (HTTPd) de la Apache *Software Foundation*.

Apache presenta entre otras características 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. [14]

1.19. ¿Por qué Apache?

Se decidió utilizar como servidor web Apache porque es sin dudas, uno de los proyectos de Software Libre más exitosos del momento, por tanto cuenta con muy buena documentación y es bastante popular tanto en la comunidad informática internacional como en la UCI y es muy fácil conseguir soporte y ayuda en caso que sea necesario, es gratuito por lo que no se necesita licencia alguna para su instalación y utilización y multiplataforma, lo que posibilita su utilización en cualquier PC de la universidad en las que se encuentran instalados sistemas operativos como Linux y Windows incluyendo también varias de sus versiones siendo también compatible con distintas alternativas Hardware: prácticamente todas las arquitecturas actuales.

1.20. Framework de Desarrollo a utilizar.

1.21. ExtJS.

ExtJS es una librería JavaScript que permite construir aplicaciones complejas en internet. Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelos de componentes extensibles.
- Un API fácil de usar.
- Licencia Open Source y comerciales.

Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de *layouts* similar al que provee *Java Swing*, 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 (Firefox, IE, Safari, etc.).

Además la ventana flotante que provee ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja tremenda frente a otros.

Usar un motor de render como ExtJS nos permite tener además estos beneficios:

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico. [15]

1.22. ¿Por qué este Framework?

Para el desarrollo de la Aplicación se decidió usar este Framework debido a sus ventajas:

- ✓ Permite agilizar el desarrollo**,** sobretodo en sus momentos iniciales**.**
- ✓ Te ahorra las habituales batallitas entre navegadores para conseguir que tus *layouts* sean "*Cross-browser*".
- ✓ Partes de una base normalizada / homogeneizada sobre la que desarrollar un trabajo adicional.
- ✓ Agiliza el trabajo en un equipo relativamente grande.

1.23. Lenguaje de programación.

Un lenguaje de programación es un dialecto artificial trazado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden utilizarse para crear sistemas que controlen el comportamiento físico y lógico de una máquina, para

expresar algoritmos con precisión, o como modo de comunicación humana. Constituyen además una técnica de comunicación mediante el cual el programador le entrega las instrucciones al computador. Pueden clasificarse teniendo en cuenta distintos criterios: lenguajes interpretados, lenguajes compilados, lenguajes de script entre otros, estas características dividen a los lenguajes de programación en dos grupos, declarativos e imperativos.

1.24. PHP.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor. [16]

Algunas de sus principales características son:

- Es un lenguaje libre.
- Soporta gran cantidad de bases de datos: MySQL, PostgresSQL, Oracle, entre otras.
- Ofrece una solución simple y universal para las paginaciones dinámicas del web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- Permite el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Multiplataforma (Windows, Unix, Linux, Mac, etc.).
- Perfecta integración del Apache-PHP-MySQL, Oracle, MS-SQL, PostgresSQL.
- Bastante sencillo de aprender y utilizar.

1.25. ¿Por qué PHP?

Como lenguaje de programación para el desarrollo de la aplicación se decidió utilizar PHP: Ya que es un lenguaje muy sencillo de aprender, es multiplataforma, soporta la programación orientada a objetos (POO), posee una amplia documentación y es libre por lo que se presenta como una alternativa de fácil acceso para todos.

1.26. Away3D:

En la actualidad existen muchos motores 3D para la web. Las alternativas libres y gratuitas no escasean, pero es engorroso encontrar las más factibles. Esta es la razón por la que ha sido necesario acudir a soluciones alternativas.

Durante el desarrollo de la aplicación se decidió tomar como motor 3D para la misma el Away3D, este motor 3D es bastante rápido, Away3D es uno de los motores 3D más populares en tiempo real para Flash. Además de crear varios entornos 3D detallados también puede crear escenas animadas en 3D, utiliza diversos efectos especiales, integrar las bibliotecas de terceros, y mucho más. Las posibilidades de utilizar este motor son infinitas.[17]

1.27. Lenguaje de Consulta Estructurado (SQL).

Structured Query Language (SQL): Es un lenguaje de acceso a bases de datos que permite la especificación de diversos tipos de operaciones sobre las mismas. Gracias a base teórica y su orientación al manejo de conjuntos de registros y no precisamente a registros individuales, es considerado un lenguaje de alto nivel permitiendo una alta productividad en el código.

Hoy, en el mundo de la informática existe una diversidad de lenguajes y de bases de datos. Lo que ha traído como consecuencia el surgimiento de estándares que permitan realizar las operaciones básicas de forma universal, de lo contrario la gestión se tornaría realmente complicada. *Structured Query Language* es precisamente eso: un lenguaje estándar de comunicación con bases de datos, no queriendo decir que sea el mismo para cada una. Se habla de un normalizado lenguaje que permite trabajar con cualquier tipo de lenguaje (PHP o ASP) así como con cualquier tipo de base dato (MS Access, SQL Server, MySQL). Aparte de esta universalidad, el SQL posee otras dos características muy apreciadas. Por una

parte, presenta una potencia y versatilidad notable que contrasta, por otra, con su accesibilidad de aprendizaje.[18]

1.28. Sistemas Gestores de Bases de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (*DataBase Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, las aplicaciones usuario y se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.[19]

Algunos ejemplos de SGBD son Oracle, DB2, PostgresSQL, MySQL, MS SQL Server, etc.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Las características de un Sistema Gestor de Base de Datos SGBD son:

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una

redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

- **Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

- **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

- **Control de la concurrencia.** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

1.29. MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente.

MySQL fue creada por la empresa sueca MySQL AB, que mantiene el *copyright* del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a 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. [20]

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y *passwords*, manteniendo un muy buen nivel de seguridad en los datos.

1.30. PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker, y fue patrocinado por *Defense Advanced Research Projects Agency* (DARPA), el Army Research Office (ARO), el *National Science Foundation* (NSF), y ESL, Inc.[21]

PostgreSQL es una derivación libre (*OpenSource*) de este proyecto, y utiliza el lenguaje SQL92/SQL99.

A continuación se muestran las principales características de este gestor de bases de datos:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos *array*.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Ofrece ventajas como:

- Instalación ilimitada.
- Mejor soporte que los proveedores comerciales.
- Ahorros considerables en costos de operación.
- Estabilidad y confiabilidad legendarias
- Extensible.
- Multiplataforma.

- Diseñado para ambientes de alto volumen.
- Herramientas gráficas de diseño y administración de bases de datos.

1.31. Fundamentación de la solución propuesta.

Después de haber leído diversos artículos sobre estos gestores de bases de datos, la conclusión que se puede sacar es que en realidad no hay que sacar ninguna conclusión sobre el tema. Cada uno de estos gestores es idóneo para ciertos campos, e intentar utilizar el otro acarrearía una pérdida de productividad del programa, como también grandes quebraderos de cabeza.

Ninguno de estos dos gestores son totalmente perfectos, por lo que no hay que obcecarse en la elección única y fanática, como se suele hacer en muchos casos de alguno de ellos. Simplemente se trata de escoger el más conveniente en cada caso.

Éstos son los grandes inconvenientes y a la vez las grandes maravillas que conlleva el mundo *OpenSource*.

Como reflexión final, decir que el gestor que se decidió tomar para la realización de este trabajo es MySQL por tener una mayor documentación y contar con la ayuda de un personal experimentado en el uso de este gestor.

1.32. Conclusiones.

En este Capítulo se realizó un estudio sobre los Sistemas de Gestión. Se realizó un estudio minucioso de las aplicaciones, metodologías, lenguajes de programación, servidores web y sistemas gestores de bases de datos con el objetivo de seleccionar lo más apropiado para el desarrollo del mismo.
Los objetivos propuestos antes del desarrollo de este capítulo se cumplieron satisfactoriamente.

2

Capítulo 2. Características de la solución propuesta.

2.1. Introducción.

En el siguiente capítulo se describen las particularidades que definirán el sistema, teniendo en cuenta las tecnologías a emplear para su elaboración, los estándares a utilizar, los Patrones de Diseño y Arquitectura más convenientes de acuerdo a los resultados que se esperan entre otros aspectos, siempre respetando las necesidades del cliente.

2.2. Objeto de automatización.

En el proyecto Escenario 3D de la facultad 5 se llevan a cabo el proceso de gestión de las tareas que se les asignan a sus estudiantes en un ciclo de 6 pasos fundamentales en el caso en el que estas tareas son modelos 3D.

1- Asignación de tarea.

2- Aceptar la tarea.

3- Realizar la tarea.

4- Actualizar el estado de la tarea.

5- Revisión de la tarea.

6- Dar evaluación.

Este ciclo se puede acortar si automatizamos la revisión de esta tarea. Actualmente los profesores descargan las tareas o productos ya terminados en sus ordenadores y de aquí dan su evaluación o criterio del resultado final, en algunas ocasiones estos productos no cumplen con las expectativas del cliente y tienen que volver a realizarse atrasando de forma gradual la fecha de entrega de dicha tarea o producto.

2.3. Propuesta del Sistema.

Con el objetivo de automatizar el proceso de gestión de las tareas o productos que se realizan en el proyecto Escenarios 3D se decidió utilizar un motor gráfico web para visualizar el estado del producto en la web. De esta forma se crea un intercambio cliente-diseñador donde el cliente puede ir puede ir valorando el estado y evolución del producto a medida que se vaya desarrollando. El sistema también cuenta con el portafolio de trabaja de cada uno de sus integrantes, donde estos pueden publicar sus trabajos y compartirlos con el resto de la comunidad.

El sistema cuenta con un menú de tipo árbol donde se muestra la información del usuario y el portafolio de trabajo del mismo. Este portafolio esta compuestos por modelos, videos imágenes y documentos.

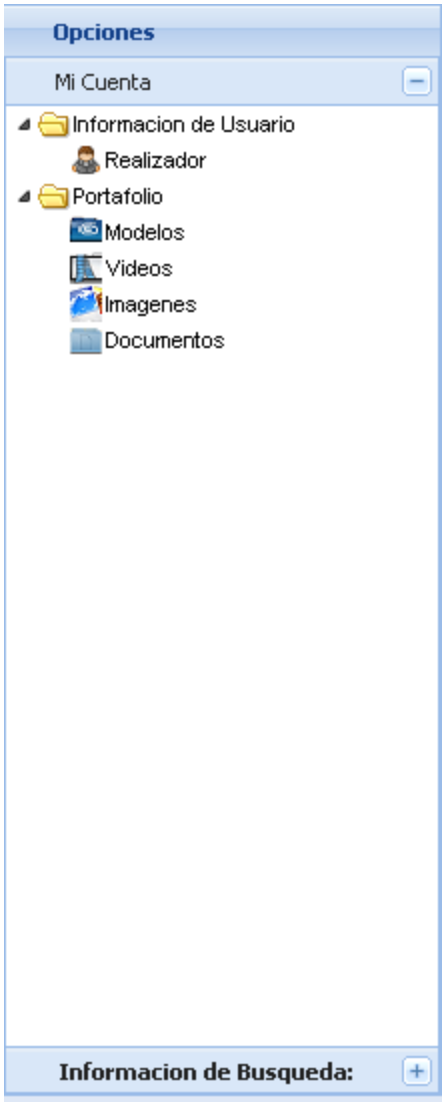

FIGURA 3: MENU TIPO TREE YLOGO AWAY3D.

2.4. Modelo del Dominio.

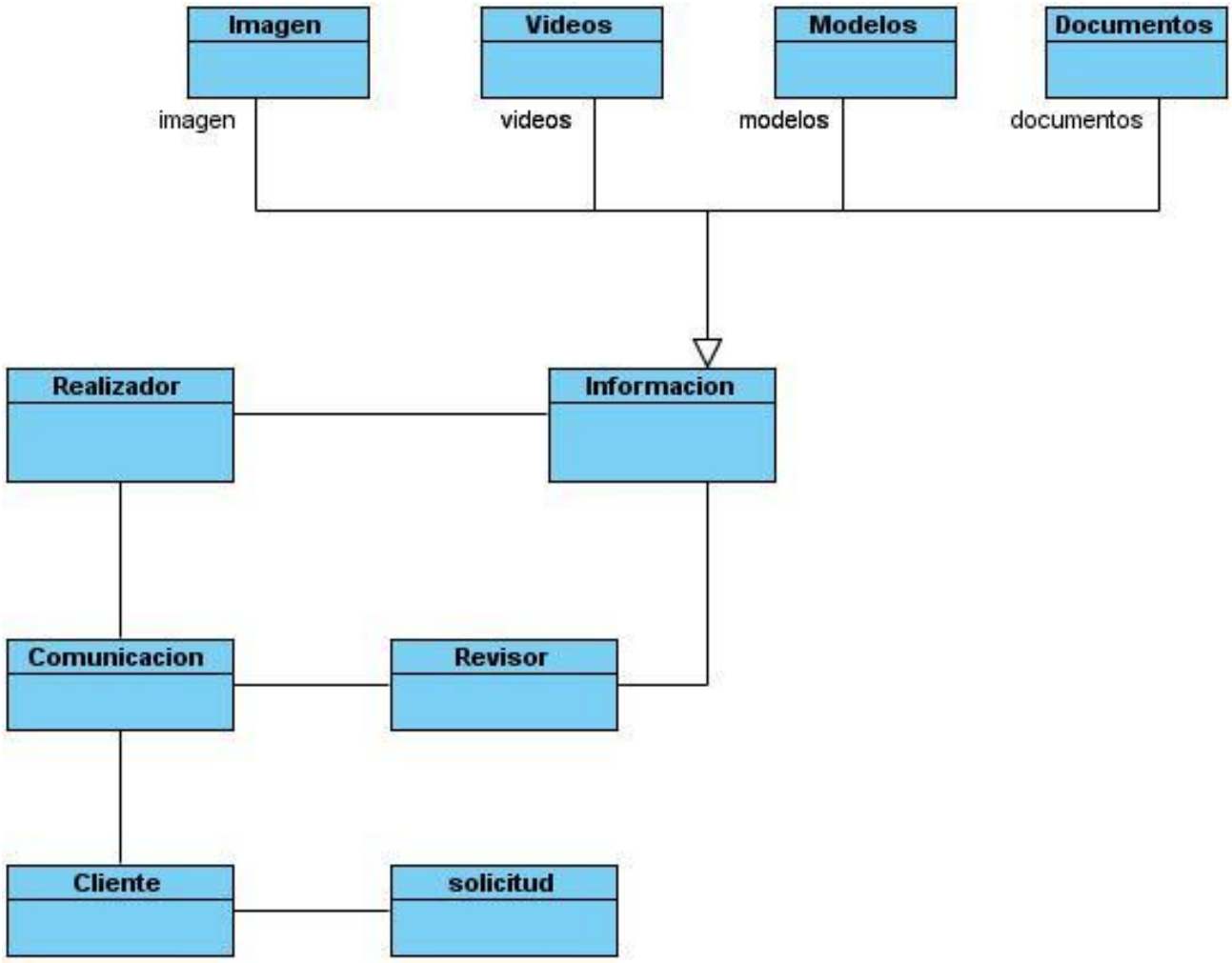

FIGURA 4: MODELO DE DOMINIO.

2.5. Especificación de los Requisitos de Software.

Con la especificación de los requisitos de software se obtiene una descripción detallada de las necesidades de un producto informático. Estos se dividen en dos grupos para una mejor especificación, los requisitos funcionales y los no funcionales. Los funcionales definen las capacidades que el sistema debe cumplir, o el qué debe hacer el sistema, mientras que los no funcionales son las propiedades que el mismo presenta, o el cómo debe hacerse.

2.6. Requerimientos Funcionales.

A continuación se muestran los requisitos funcionales de la aplicación:

Alias	Requerimientos	Descripción
RF 1	**Gestionar Modelo**	**Gestiona los modelos que se almacenan.**
RF 1.1	Insertar Modelo	Se inserta el modelo
RF 1.2	Eliminar Modelo	Se elimina el modelo
RF 1.3	Buscar Modelo	Se busca un modelo
RF 1.4	Visualizar Modelo	Se visualiza el modelo
RF 1.5	Salvar Modelo	Se descarga un modelo
RF 2	**Gestionar Documentos**	**Gestiona los documentos que se almacenan**
RF 2.1	Insertar Documentos	Se inserta el documentos
RF 2.2	Eliminar Documento	Se elimina el documento
RF 2.3	Buscar Documento	Se busca un documento
RF 2.4	Salvar Documento	Se descarga un documento
Rf 3	**Gestionar imagen**	**Gestiona las imágenes que se almacenan**
RF 3.1	Insertar imagen	Se inserta la imagen
RF 3.2	Eliminar Imagen	Se elimina la imagen
RF 3.3	Buscar Imagen	Se busca una imagen
RF 3.4	Salvar Imagen	Se descarga una imagen
RF 4	**Gestionar Videos**	**Gestiona los videos que se almacenan**
Rf 4.1	Insertar Video	Se inserta el video
RF 4.2	Eliminar Video	Se elimina el video
RF 4.3	Buscar Video	Se busca un video
RF 4.4	Salvar Video	Se descarga un video
RF 5	**Gestionar Usuario**	**Gestiona la información de los usuarios**
RF 5.1	Insertar Usuario	Se Inserta la información de un usuario
RF 5.2	Buscar Usuario	Se busca la información de un usuario

RF 5.3	Eliminar Usuario	Se elimina el usuario
RF 5.4	Modificar usuario	Se modifica el usuario
RF 6	**Autenticar**	**El usuario se logea.**
RF 7	**Gestionar Solicitud**	**Gestiona las solicitudes**
RF 7.1	Realizar Solicitud	Un cliente realiza un solicitud de servicio
RF 7.2	Eliminar Solicitud	Se elimina una solicitud
RF 7.3	Buscar Solicitud	Se busca una solicitud

TABLA 1: REQUERIMIENTOS FUNCIONALES.

2.7. Requerimientos No Funcionales.

Categoría	Requerimiento	Descripción
Usabilidad	Fácil empleo para usuarios sin experiencia	El sistema debe tener una interfaz que le sea familiar al usuario para aprovechar sus conocimientos en el manejo de herramientas de software. Además debe ser de fácil aprendizaje para que usuarios inexpertos puedan familiarizarse rápidamente.
Seguridad	Protección de la Base de Datos	La base de datos debe tener varios esquemas, dividiendo así de una forma lógica las funcionalidades, evitando la pérdida total de la información en caso de algún accidente o ataque.
Rendimiento	Escalabilidad	El sistema debe ser capaz de mantener el mismo rendimiento y estabilidad a medida que aumenta la cantidad de datos a gestionar.

TABLA 2: REQUERIMIENTOS NO FUNCIONALES.

2.8. Definición de los Actores del Sistema.

Actor	Descripción
Administrador	Es el actor que se encarga de la administración de todos los permisos dentro de la aplicación.
Realizador	Es el principal actor del sistema, controla su portafolio de trabajo.
Revisor	El revisor es el actor del sistema que se encarga de la revisión de las tares y los productos que se realizan en Escenarios 3D, Es el actor más calificado del sistema.
Cliente	Es el actor externo de la organización, el cual puede presentar una solicitud de servicio.

TABLA 3: DEFINICION DE LOS ACTORES DEL SISTEMA.

2.9. Diagrama de Caso de Uso del Sistema.

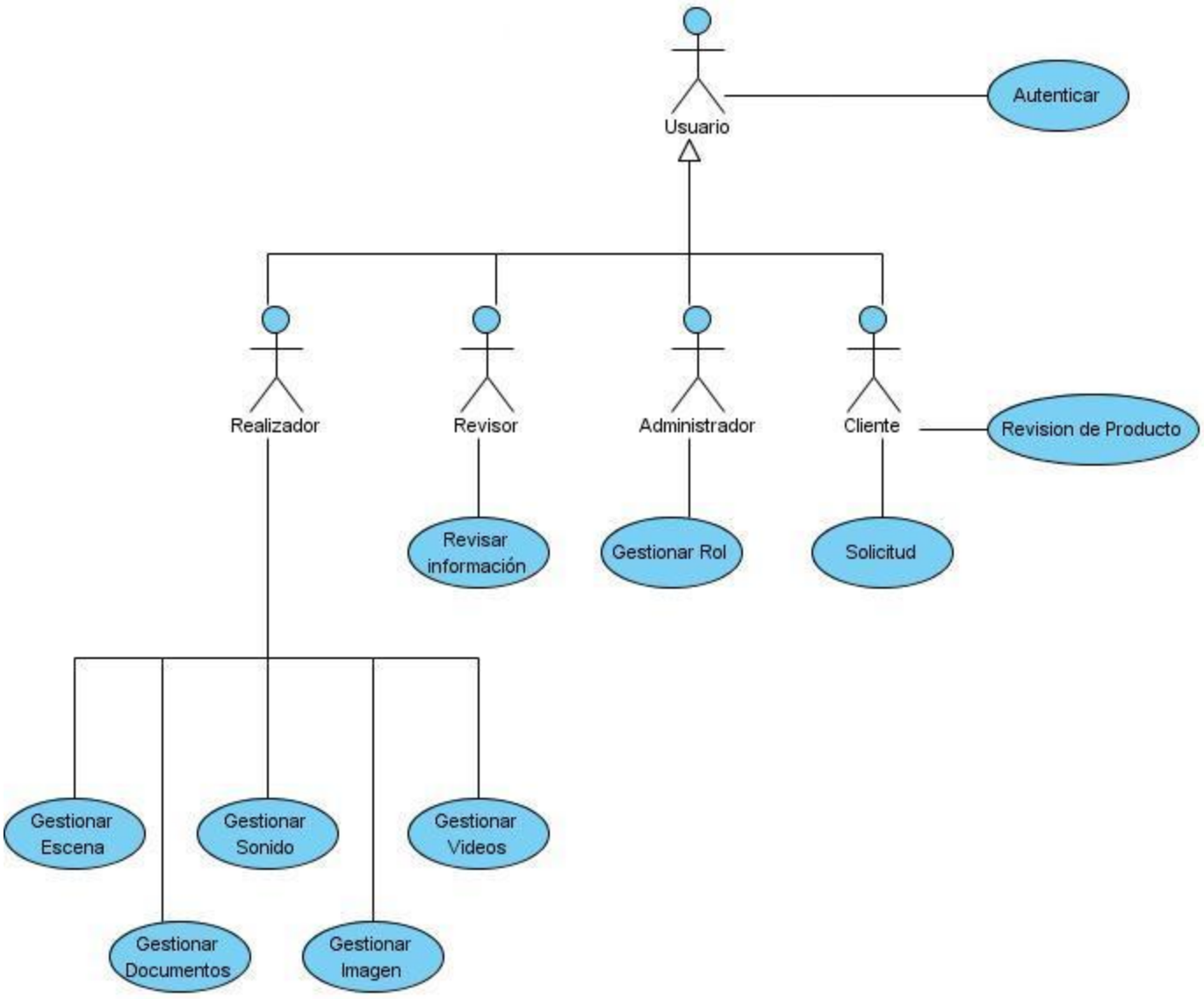

FIGURA 5: DIAGRAMA DE CASOS DE USO DEL SISTEMA.

2.10. Descripción de Casos de Usos del Sistema.

Al describir los casos de uso del sistema obtenemos mayor información y entendimiento de la aplicación a implementar. También se refleja las acciones de los actores y como debe de responder el sistema.

Caso de Uso Autenticarse:

Caso de Uso:	**Autenticarse**
Actores:	Usuarios
Resumen:	El caso de uso se inicia cuando el usuario del sistema intenta acceder a la aplicación
Precondiciones:	El sistema debe estar funcionando y activo de forma tal que ofrezca la interfaz para la autenticación, además de estar instalado correctamente para poder realizar la verificación y validación necesaria.
Referencias	RF6
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1- Accede a la aplicación.	2- Muestra la página inicial con la opción para autenticarse en el sistema.
3- Introduce los datos solicitados. ✓ Usuario. ✓ Contraseña.	4- Verifica la existencia del usuario.
	5- Verifica que la contraseña introducida es correcta.
	6- Carga el sistema con los menús, las opciones y funcionalidades a las que el usuario tiene acceso y permisos.
Prototipo de Interfaz	

Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	4.1 Si no encuentra al usuario en la base datos y la contraseña es correcta lo inserta en la BD.
	6.1 Carga el sistema con los menús, las opciones y funcionalidades a las que el Usuario tiene acceso y permisos(Cliente).
Prototipo de Interfaz	
Poscondiciones	1- Se le permite el acceso al usuario. 2- Quedan habilitadas las opciones a las que el usuario tiene acceso.

TABLA 4: CASO DE USO AUTENTICARSE.

Caso de Uso Gestionar Modelo:

Caso de Uso:	**Gestionar Modelo**
Actores:	Realizador, Revisor
Resumen:	El caso de uso comienza cuando el realizador o el revisor decide realizar una acción en la sección de Modelos de su portafolio de trabajo.
Precondiciones:	Autenticación del usuario como realizador o revisor
Referencias	RF 1, RF 1.1, RF 1.2, RF 1.3, RF 1.4, RF 1.5
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1- El actor solicita los datos de la sección Modelos de su portafolio de trabajo.	2- El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar un modelo ir a la sección “nuevo” b) Para eliminar un modelo ir a la sección “Eliminar ” c) Para visualizar un modelo ir a la sección “Visualizar”

Sección: "Nuevo"	
Acción del Actor	**Respuesta del Sistema**
1- Selecciona la opción Nuevo	2- Muestra un campo para introducir una descripción del nuevo modelo y otro para cargar el modelo.
3- Introduce la información y acepta.	4- Se visualiza el tema insertado en el Grid de la sección.
Prototipo de Interfaz	
Sección: "Eliminar "	
Acción del Actor	**Respuesta del Sistema**
1- Selecciona el modelo a eliminar.	2- El sistema marca el modelo a eliminar.
3- Selecciona eliminar	4- Elimina el modelo y actualiza la interfaz
Prototipo de Interfaz	
Sección: "Visualizar"	
Acción del Actor	**Respuesta del Sistema**
1- Selecciona un modelo	2- Marca el modelo
3- Selecciona visualizar	4- El sistema visualiza el modelo
Prototipo de Interfaz	
Poscondiciones	1- Se adiciona un modelo 2- Se elimina un modelo 3- visualiza un modelo

TABLA 5: CASO DE USO GESTIONAR MODELO.

Caso de Uso Realizar Solicitud:

Caso de Uso:	**Realizar Solicitud.**
Actores:	Cliente, Administrador
Resumen:	El caso de uso se inicia cuando el cliente o el administrador decide realizar una operación sobre la sección solicitud de servicios
Precondiciones:	Autenticación del usuario como Cliente o Administrador
Referencias	RF 7, RF 7.1, RF 7.2, RF 7.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1- El actor solicita los datos de la sección Solicitud de servicios.	2- El sistema muestra un Grid con un modelo de solicitud a para descargar y las solicitudes realizadas.
	3- El sistema ejecuta alguna de las siguientes acciones. a) Para adicionar una solicitud ir a la sección "nuevo" b) Para eliminar una solicitud ir a la sección "Eliminar " c) Para descargar el modelo de solicitud ir a la opción descargar.
Sección: "Nuevo"	
Acción del Actor	**Respuesta del Sistema**
1- Selecciona la opción Nuevo	2- Muestra un campo para introducir una descripción de la nueva solicitud o modelo de solicitud y otro para cargar el fichero.

3- Introduce la información y acepta.	4- Se visualiza la solicitud insertado en el Grid de la sección.
Prototipo de Interfaz	
Sección: "Eliminar "	
Acción del Actor	**Respuesta del Sistema**
1- Selecciona la solicitud a eliminar.	2- El sistema marca la solicitud a eliminar.
3- Selecciona eliminar	4- Elimina la solicitud y actualiza la interfaz
Prototipo de Interfaz	
Sección: "Descargar"	
Acción del Actor	**Respuesta del Sistema**
1- Selecciona una solicitud o el modelo de solicitud	2- Marca la solicitud o modelo.
3- Selecciona descargar	4- El sistema le pregunta donde desea descargar el fichero
3- El usuario le dice dónde y acepta	4- El sistema descarga el fichero y termina
Prototipo de Interfaz	
Poscondiciones	1- Se adiciona una solicitud o modelo 2- Se elimina una solicitud. 3- Se descarga una solicitud o modelo.

TABLA 6: CASO DE USO GESTIONAR SOLICITUD.

2.11. Conclusiones.

En este capítulo se describieron las particularidades que definen el sistema teniendo en cuenta las tecnologías a emplear para su elaboración, se realizó un modelo de dominio, se establecieron los requisitos y los casos de uso del sistema alcanzando los objetivos propuestos antes de comenzar este capítulo.

3

Capítulo 3. Diseño del Sistema.

3.1. Introducción.

En este capítulo se describe la puesta en práctica de la construcción de la solución propuesta y se desarrolla el diseño del sistema definiéndose los diagramas de clases de diseño y los diagramas de interacción siguiendo un flujo de proceso. El diagrama de secuencia y el modelo de bases de datos.

3.2. Arquitectura del Software.

La Arquitectura de Software, también dominada Arquitectura lógica está integrada por un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencias necesarias para guiar el desarrollo de un software. Establece los fundamentos para que un equipo de desarrollo trabaje sobre una misma línea común que permita alcanzar los objetivos específicos de un sistema informático. A la hora de crear un software informático se selecciona o se crea una arquitectura que rige el desarrollo del mismo. Algunas de las diferentes arquitecturas de software que existen son:

- **Monolítica:** Donde el software se estructura en grupos funcionales muy acoplados.
- **Cliente-Servidor**: Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.
- **Arquitectura de tres niveles:** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

De las arquitecturas antes mencionadas se tomó para el desarrollo de la aplicación la Arquitectura en tres Niveles o Capas.

3.3. Patrón de arquitectura (Modelo Vista Controlador)

Para el diseño de la aplicación se utilizó el patrón de diseño MVC ya que permite separar el Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos. Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas. Crea independencia de funcionamiento y facilita el mantenimiento en caso de errores.

Al aplicar este patrón se logró que la aplicación facilitara la reusabilidad y la comprensión durante y después del desarrollo del sistema ya que es posible modificar cualquiera de estos tres componentes sin que esto afecte el funcionamiento del resto

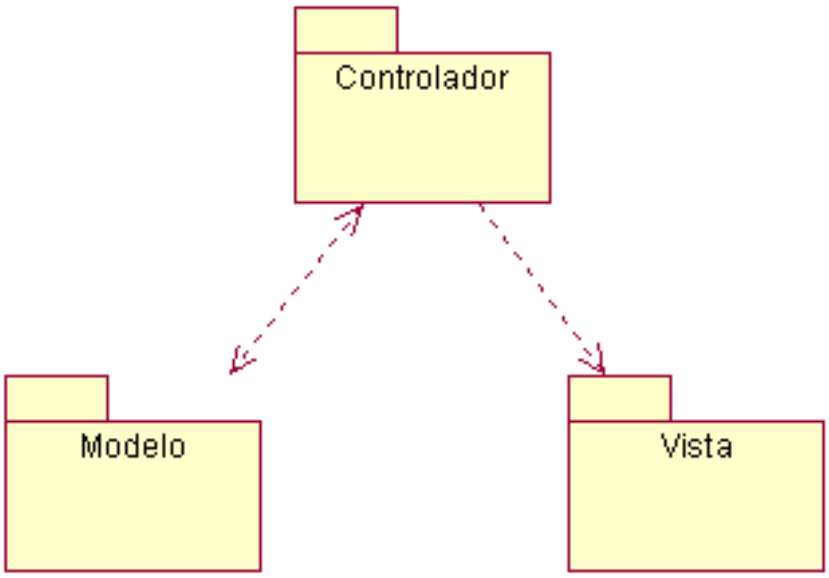

FIGURA 6: MODELO VISTA CONTROLADOR.

El modelo representa la información con la que trabaja la aplicación, su lógica de negocio. La vista transforma el modelo en una interfaz que permite al usuario interactuar con ella. El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La entrada de los datos se realiza a través de la Vista y en esta se ubican todas las interfaces que representan al modelo en un formato adecuado para interactuar con el usuario. En este proceso de entrada de datos participa el Controlador el cual toma decisiones, responde a eventos en este caso acciones del usuario y a su vez invoca cambios en el modelo que pueden también modificar la vista. El procesamiento de datos se realiza en el Modelo que es la representación específica del dominio de la información sobre

la cual funciona la aplicación, ocurriendo posteriormente la salida de los datos a través de la Vista.

3.4. Diseño.

El diseño es el refinamiento de los modelos de análisis donde se tienen en cuenta principalmente los requisitos no funcionales para lograr más tarde una solución sólida y factible. Esto es lo que ayuda a mantener una arquitectura estable y sólida para poder crear un plano del modelo de implementación.

3.5. Diagrama de clases del diseño.

El Modelo de Diseño es un proceso para la ilustración detallada de un sistema con el fin de la realización física de los casos de uso para cubrir las funciones que realizará el sistema y otras restricciones del entorno de implementación que tienen impacto en el mismo, por tanto en él se definen las clases del diseño que conformarán el sistema que se va a implementar.

- **Páginas clientes:** Son las páginas encargadas de permitir a los usuarios interactuar con el sistema tanto para hacer solicitudes como para que sean mostradas las respuestas a las mismas.
- **Páginas servidoras:** Son las encargadas de la construcción de forma dinámica de las páginas clientes y sirven de enlace entre estas y el resto de las clases.
- **Páginas controladoras:** Son las responsables de realizar las operaciones que responden a los procesos de negocio y dar respuestas a las solicitudes hechas por el usuario.
- **Clases entidad:** Son las responsables de la persistencia de los datos físicamente.

Diagrama de clase del diseño.

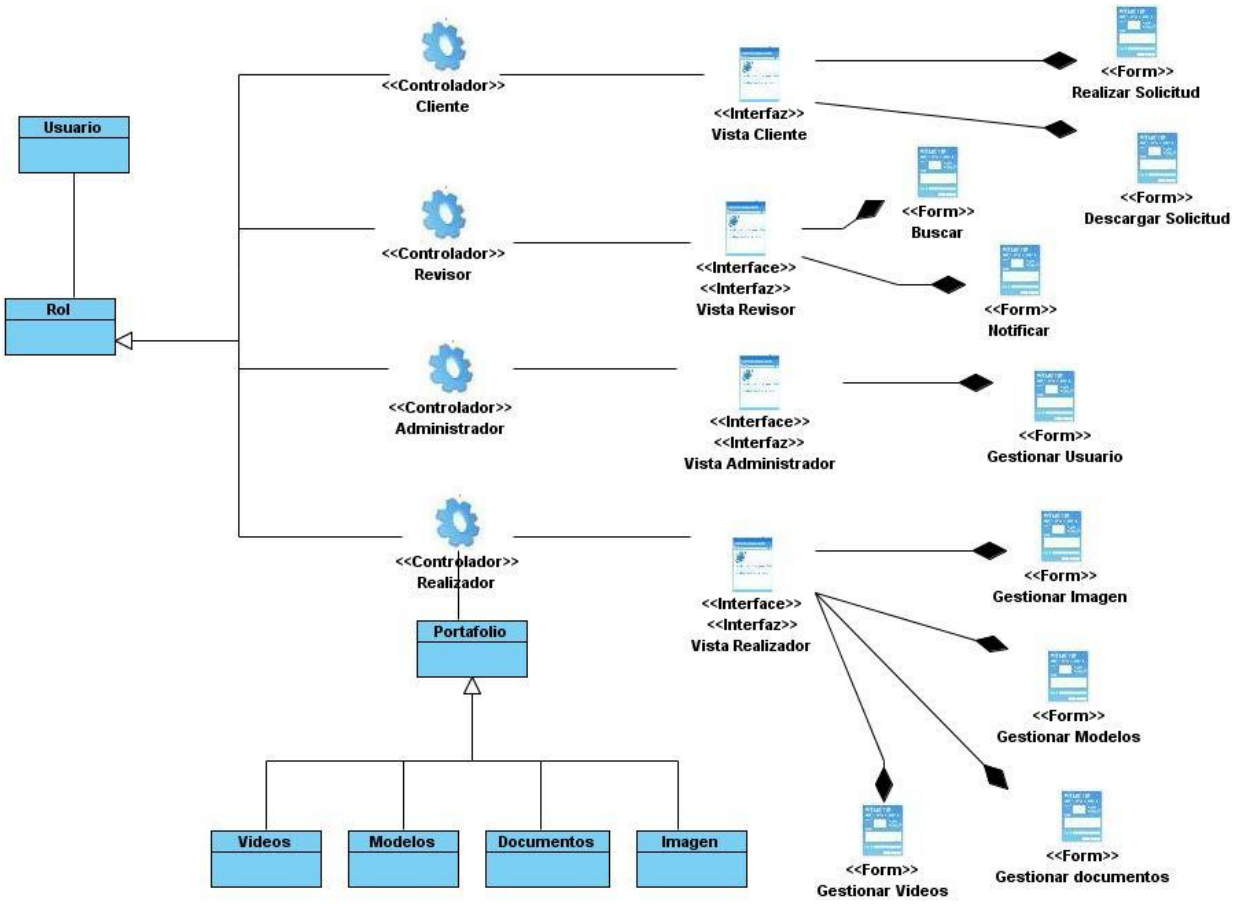

FIGURA 6: DIAGRAMA DE CLASES DE DISEÑO.

3.6. Diagramas de secuencia.

Los diagramas de secuencia representan gráficamente las interacciones del actor y las operaciones generadas que da origen. En estos se muestran como los objetos se comunican entre ellos con el objetivo de cumplir con los requerimientos planteados.

Diagrama de secuencia CU Gestionar Modelo:

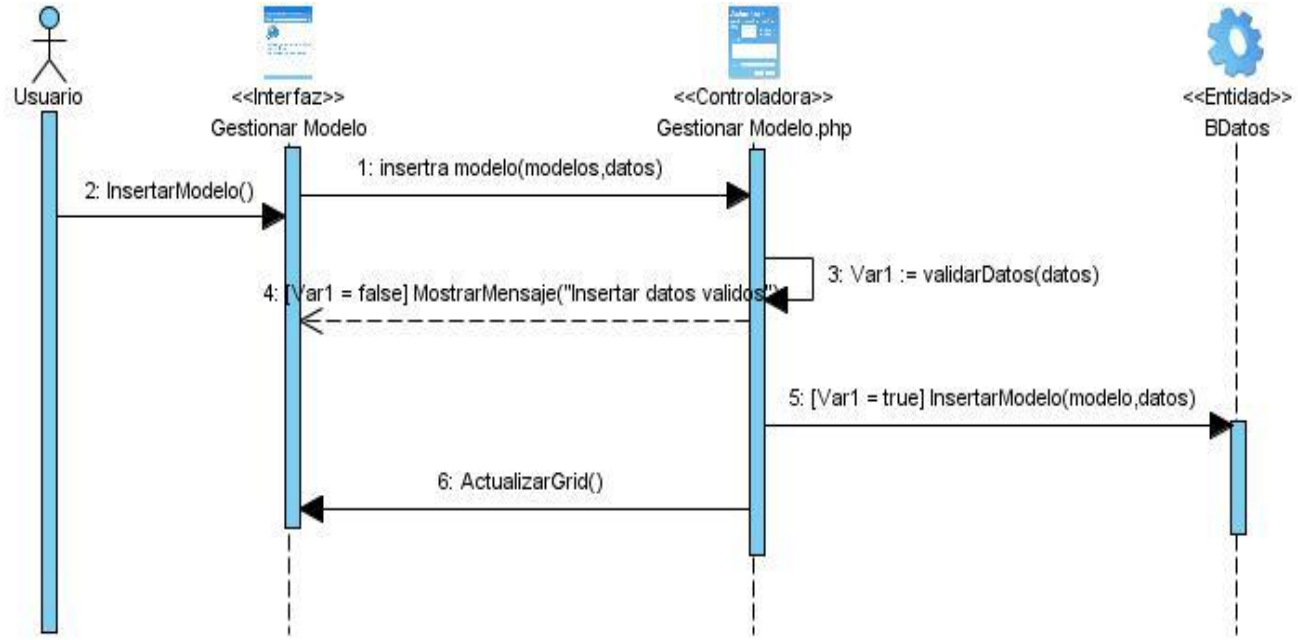

FIGURA 7: DIAGRAMA DE SECUENCIA CU GESTIONAR MODELO (INSERTAR MODELO)

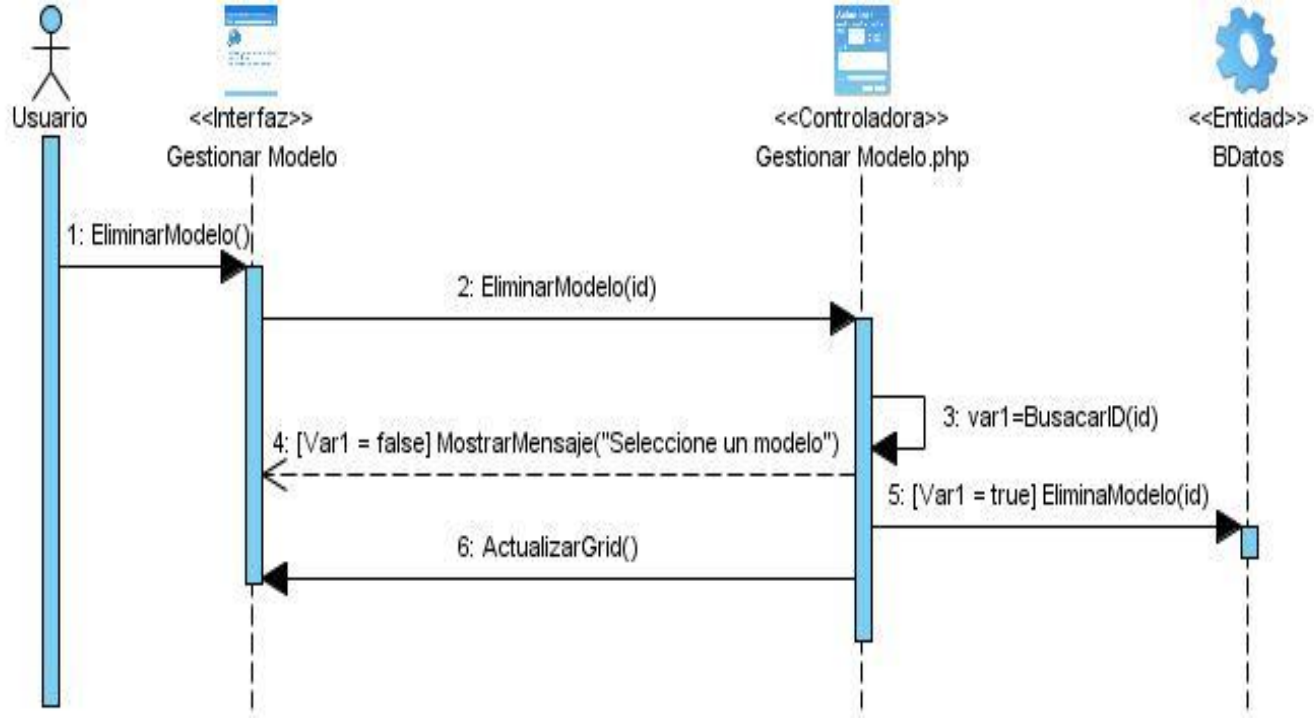

FIGURA 8: DIAGRAMA DE SECUENCIA CU GESTIONAR MODELO(ELIMINAR MODELO)

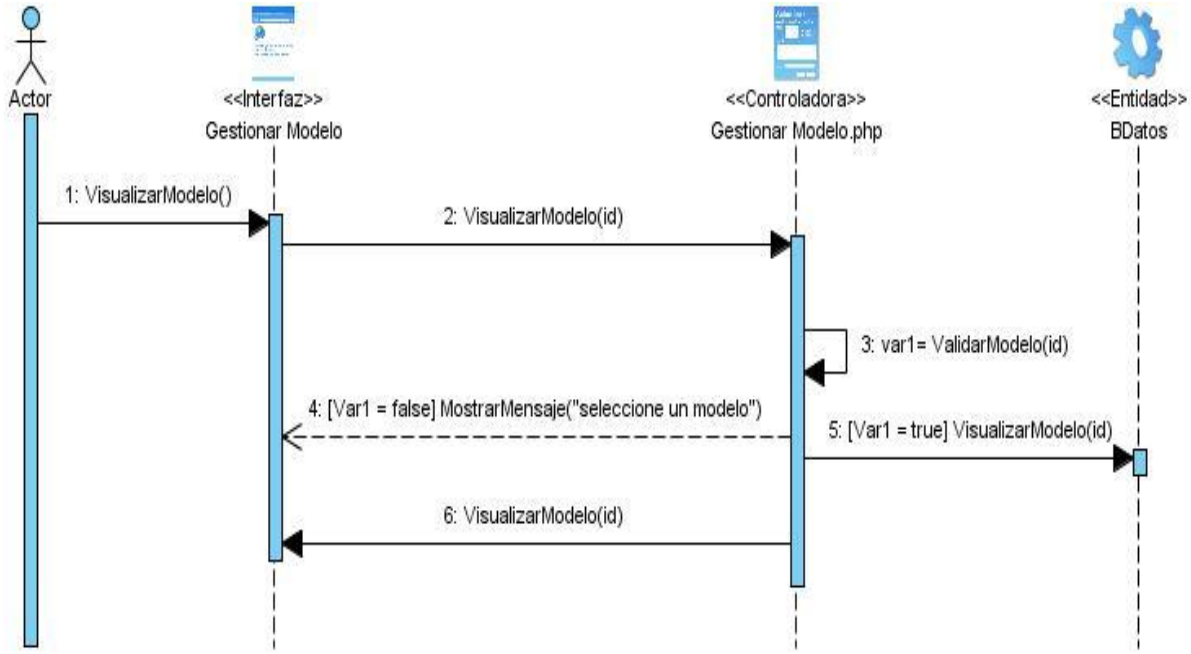

FIGURA 9: DIAGRAMA DE SECUENCIA CU GESTIONAR MODELO(VISUALIZAR MODELO).

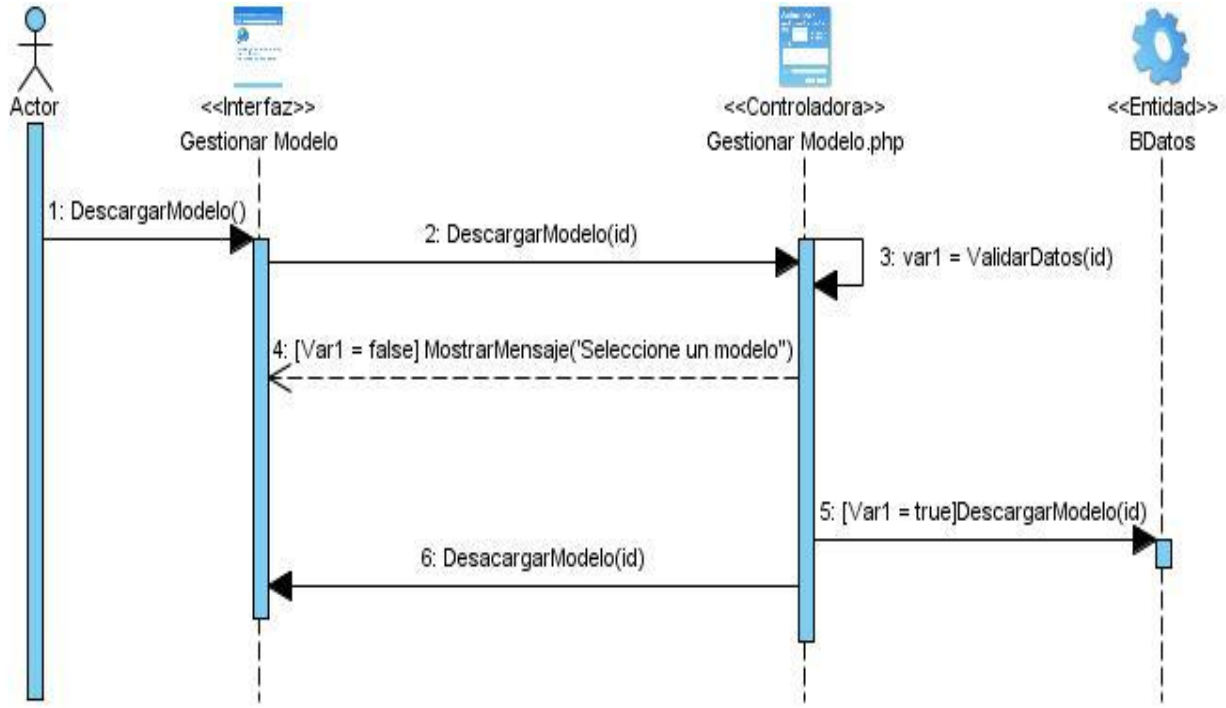

FIGURA 10: DIAGRAMA DE SECUENCIA CU GESTIONAR MODELO(DESCARGAR MODELO).

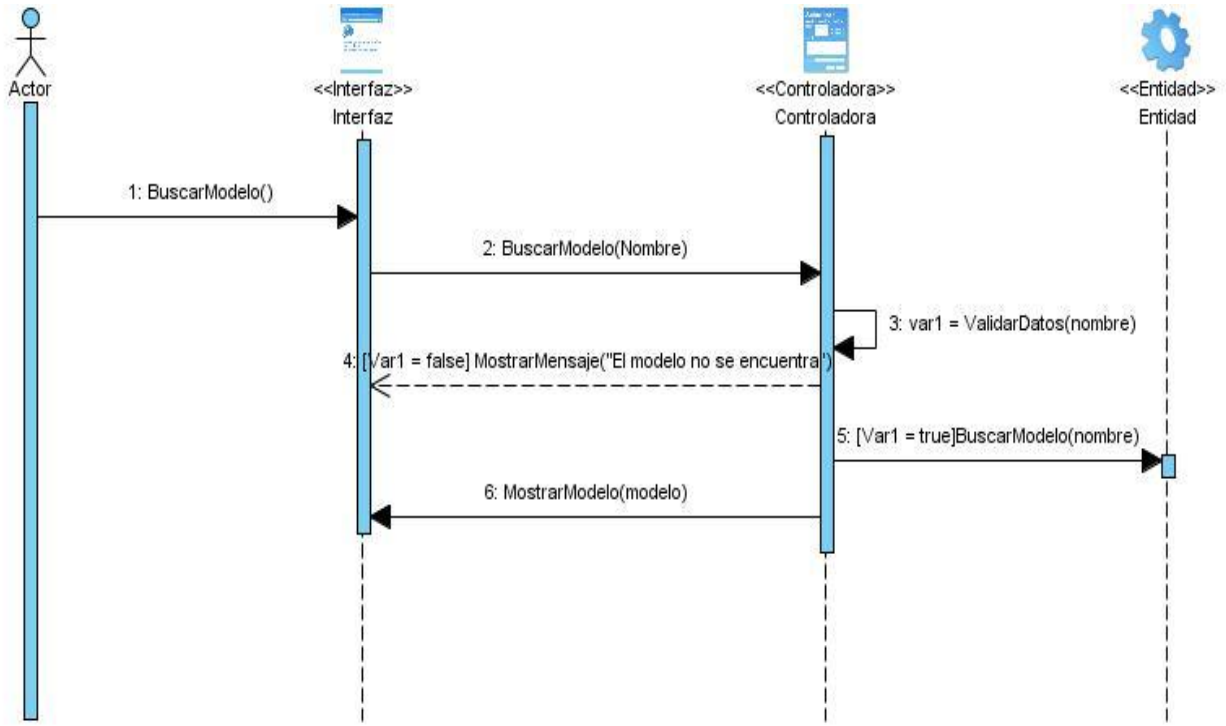

FIGURA 11: DIAGRAMA DE SECUENCIA CU GESTIONAR MODELO(BUSCAR MODELO).

Diagrama de secuencia CU Gestionar Usuario:

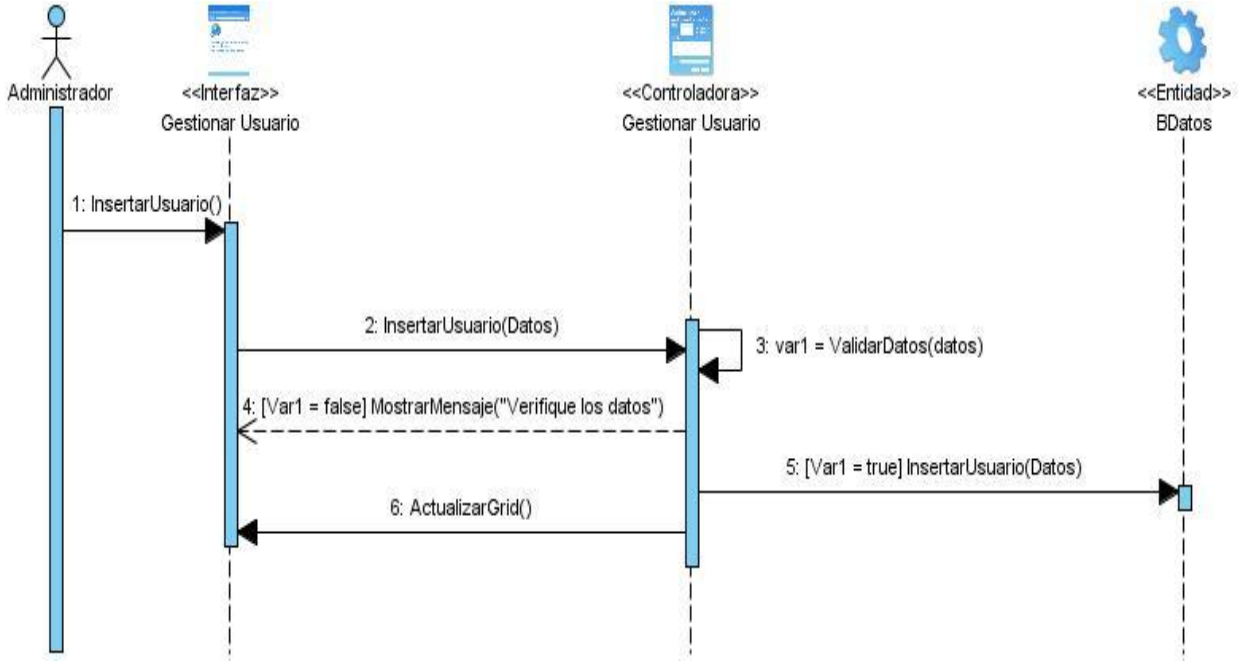

FIGURA 12: DIAGRAMA DE SECUENCIA CU GESTIONAR USUARIO(INSERTAR USUARIO).

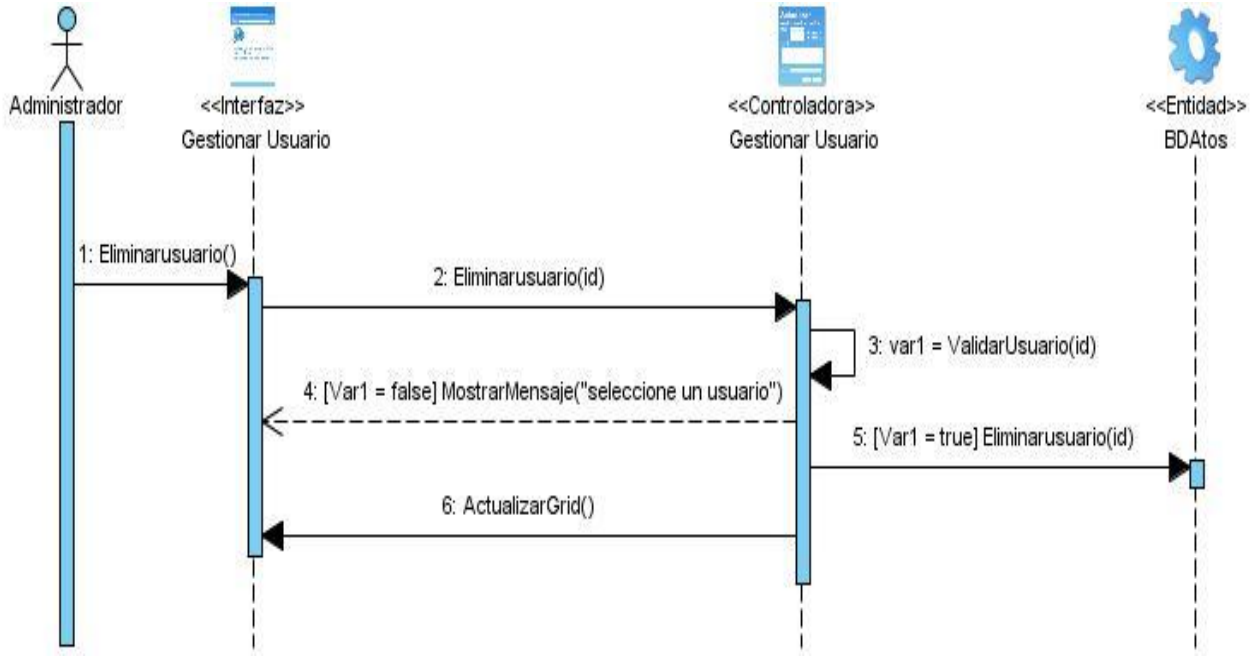

FIGURA 13: DIAGRAMA DE SECUENCIA CU GESTIONAR USUARIO(ELIMINAR USUARIO).

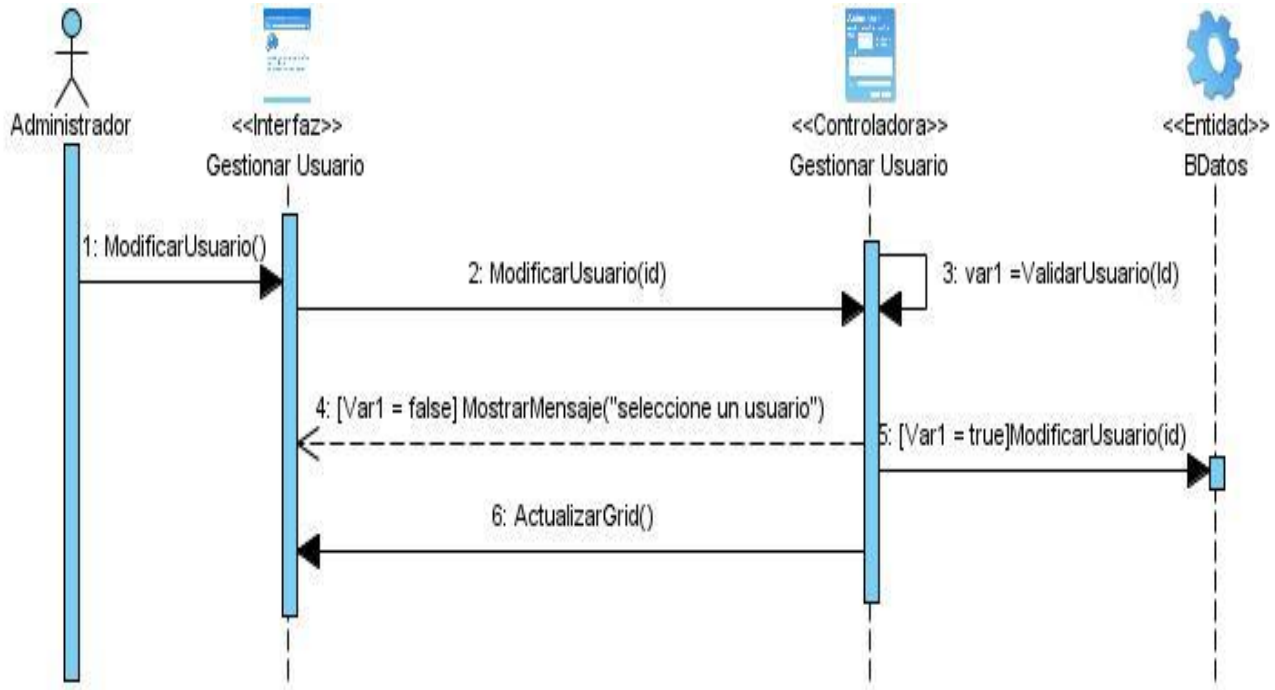

FIGURA 14: DIAGRAMA DE SECUENCIA CU GESTIONAR USUARIO(MODIFICAR USUARIO).

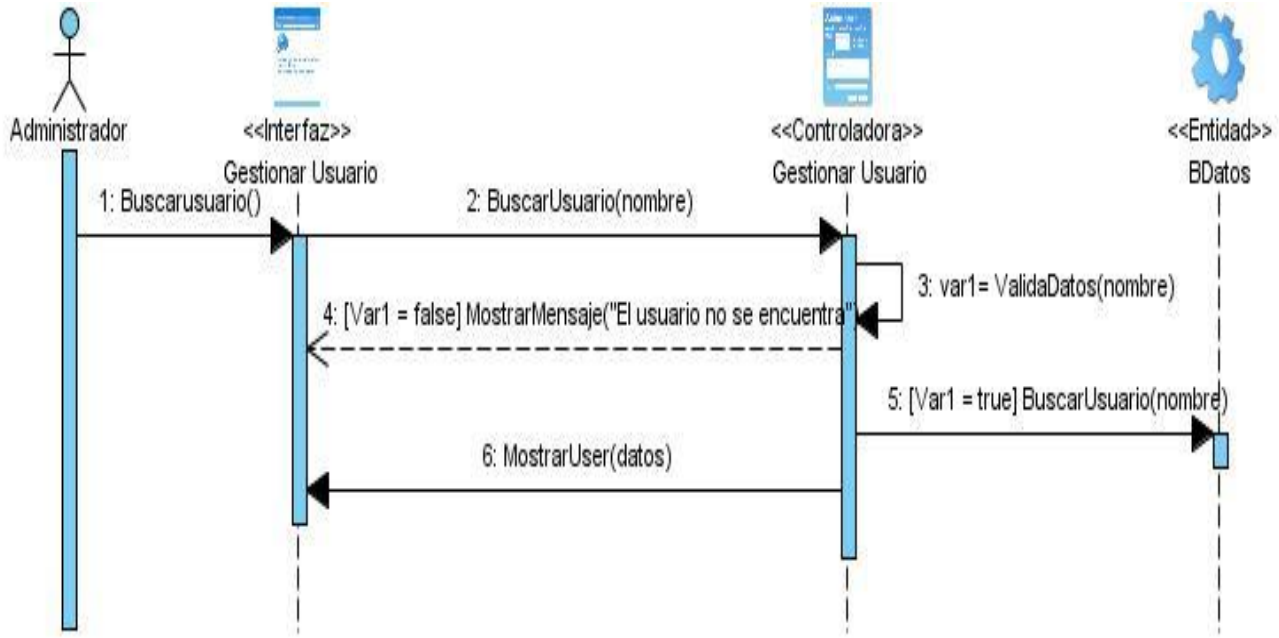

FIGURA 15: DIAGRAMA DE SECUENCIA CU GESTIONAR USUARIO(BUSCAR USUARIO).

Diagrama de secuencia CU Autenticar:

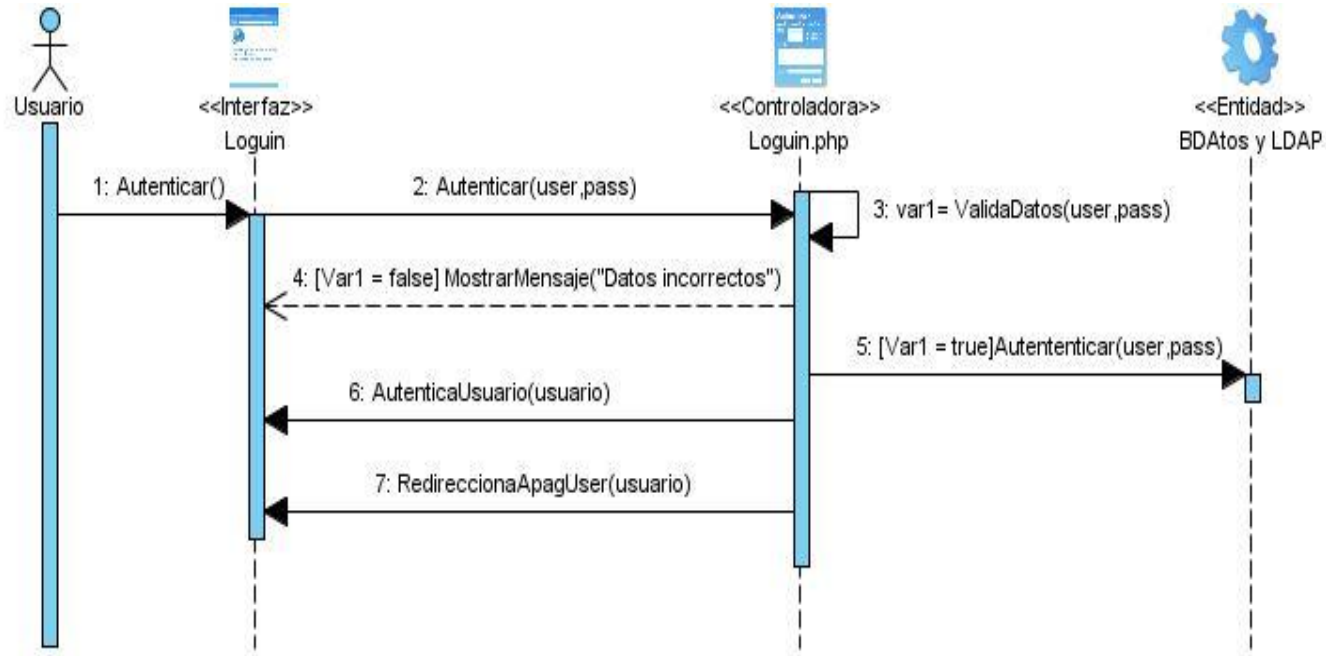

FIGURA 16: DIAGRAMA DE SECUENCIA CU AUTENTICAR USUARIO(AUTENTICAR USUARIO).

3.7. Diagrama de clases persistentes.

Como parte del diseño de la aplicación se ha desarrollado el diagrama de clases persistentes, que muestra las clases persistentes de nuestro sistema y las relaciones que existen dentro de ellas.

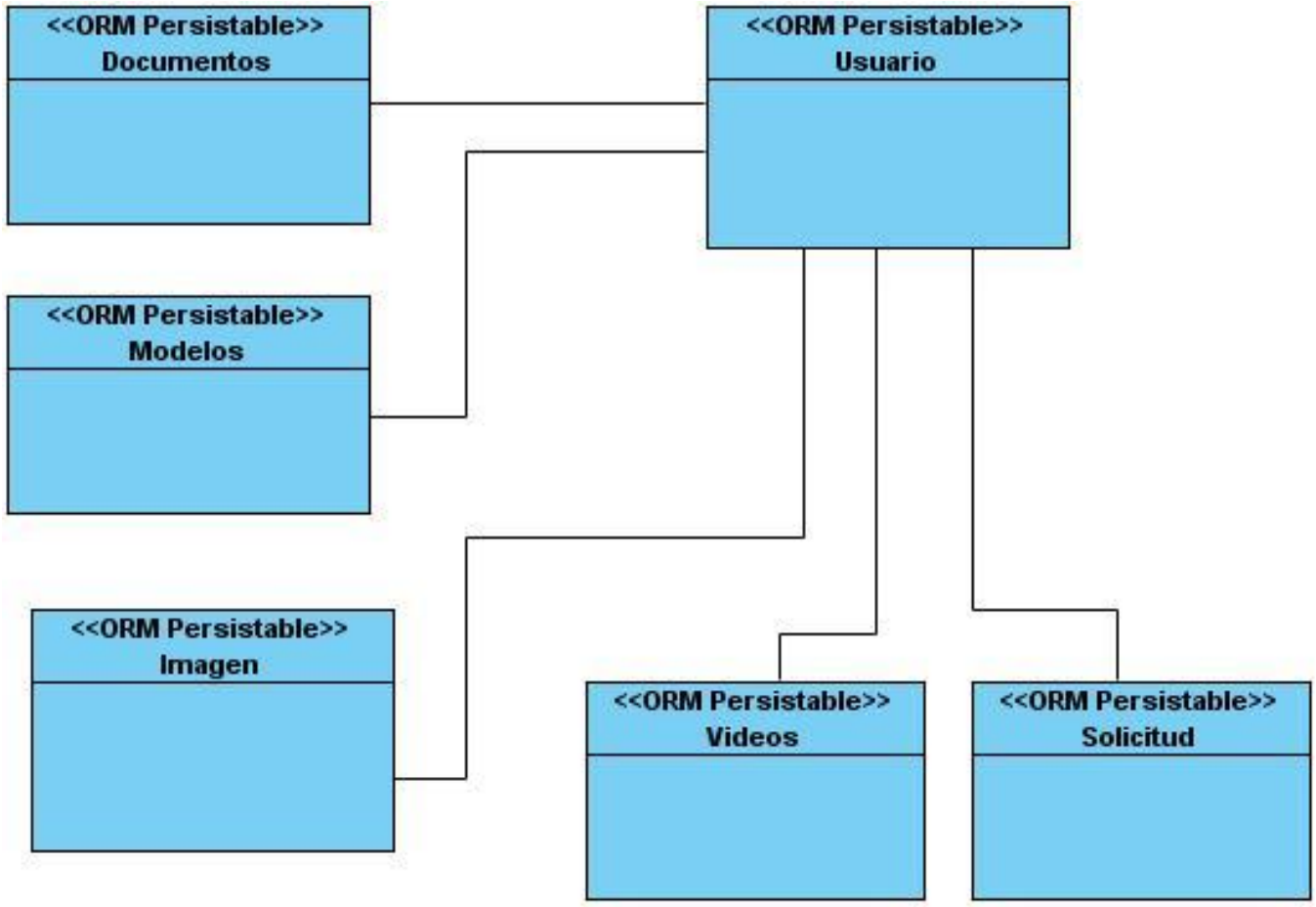

FIGURA 17: DIAGRAMA DE CLASES PERSISTENTES.

3.8. Diseño de la Base de Datos.

Al desarrollar un correcto diseño de Base de Datos se garantiza obtener un acceso fiel a la información sin ningún tipo de inconveniente. Una definición fuerte de su estructura nos facilita mucho el trabajo a la hora de almacenar los datos, reconocer el contenido, recuperar información y siempre tiene que responder a las necesidades que se desean desarrollar. El mantener un base de Datos fuerte mente estructurada nos permite no tener que depender de muchas clases al llevar a cabo nuestra aplicación.

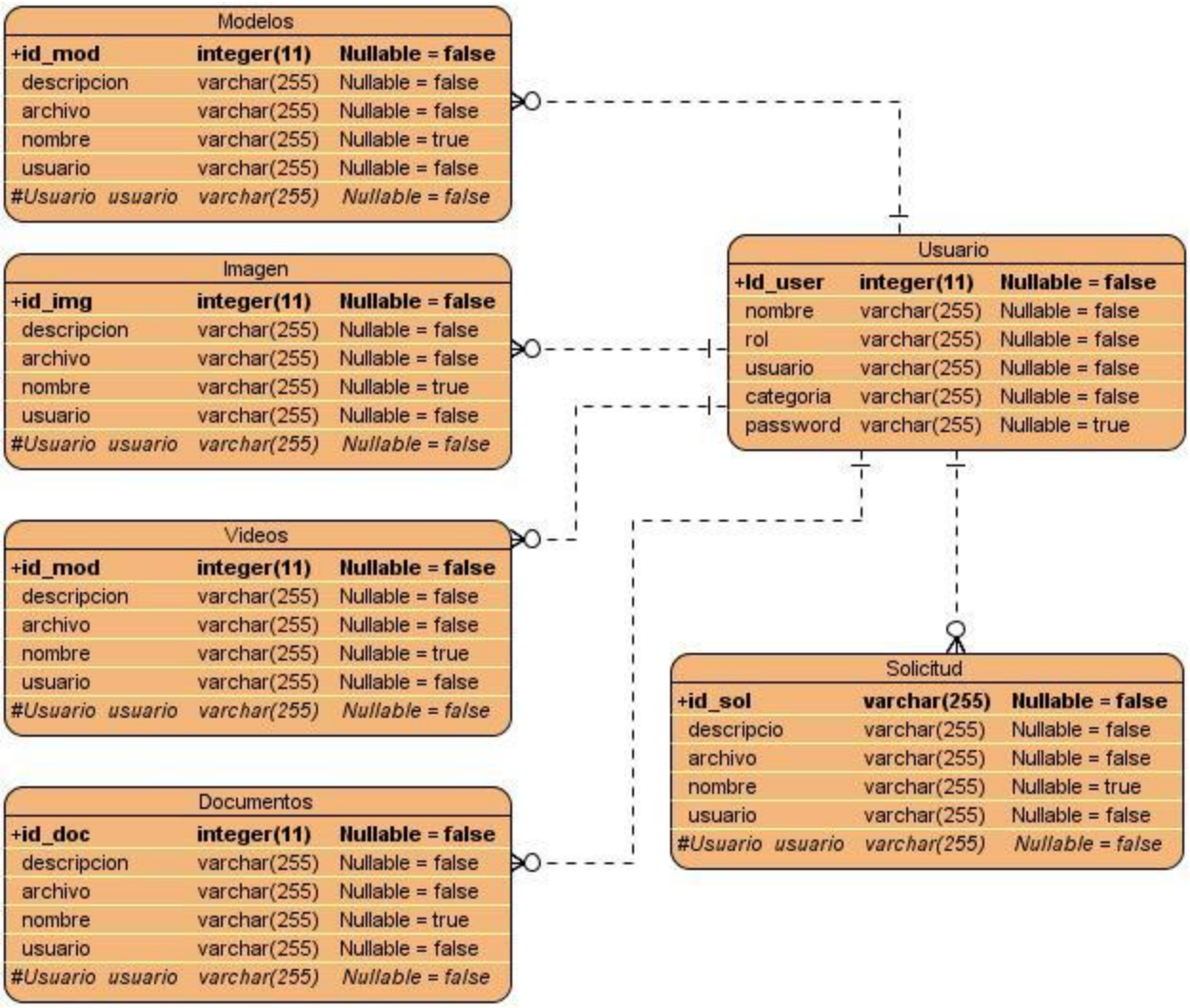

FIGURA 18: DISEÑO DE BASE DATOS.

3.9. Conclusiones.

Con la elaboración de este capítulo se mostraron los resultados de la etapa del diseño del sistema para abrir paso a la implementación de la aplicación. Realizando satisfactoriamente todos los diagramas de la parte del diseño de nuestra aplicación.

4

Capítulo 4. Implementación y validación del Sistema.

4.1. Introducción.

En este capítulo se representarán un conjunto de diagramas que reflejan los componentes por los que está constituido el sistema, así como la distribución física de los mismos facilitando la comprensión de la estructura y composición de la aplicación.

4.2. Modelo de implementación.

En el modelo de Implementación se describe como los elementos del modelo del diseño se implementan en términos de componentes y como estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Estos componentes representan la composición física de la implementación del sistema.

4.3. Diagrama de Componentes.

El Diagrama de componentes se usa para estructurar el modelo de implementación y mostrar las relaciones entre los elementos de implementación. Este diagrama muestra la estructura de alto nivel del modelo de implementación.

Diagrama de componente CU Autenticar.

Este diagrama de componente del CU Autenticar es uno de los pasos críticos a la hora de implementar y poner en marcha nuestra aplicación, ya que si el usuario no se autentica el sistema no tiene conocimiento de quien es el usuario que esta interactuando con él y la información que guarde o modifique se pierde.

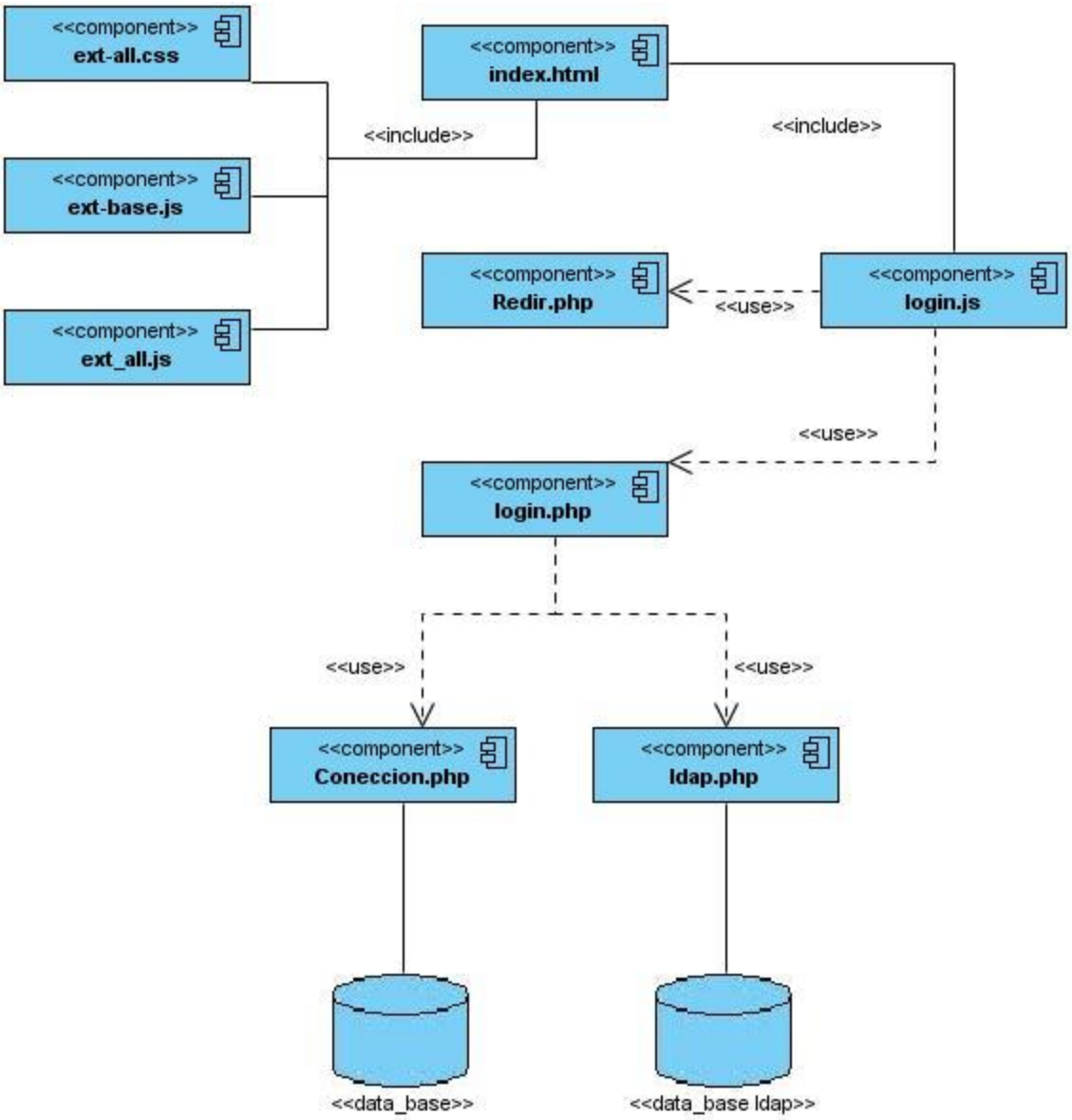

FIGURA 19:DIAGRAMA DE COMPONENTE CU AUTENTICAR.

Diagrama de componente CU Gestionar modelos.

Este es otro de los CU críticos del sistema, aquí es donde se gestionan los modelos tridimensionales, Esta parte del sistema brinda las funcionalidades de, Adicionar, Eliminar,

Descargar y Visualizar modelos 3D la principal funcionalidad de nuestro sistema

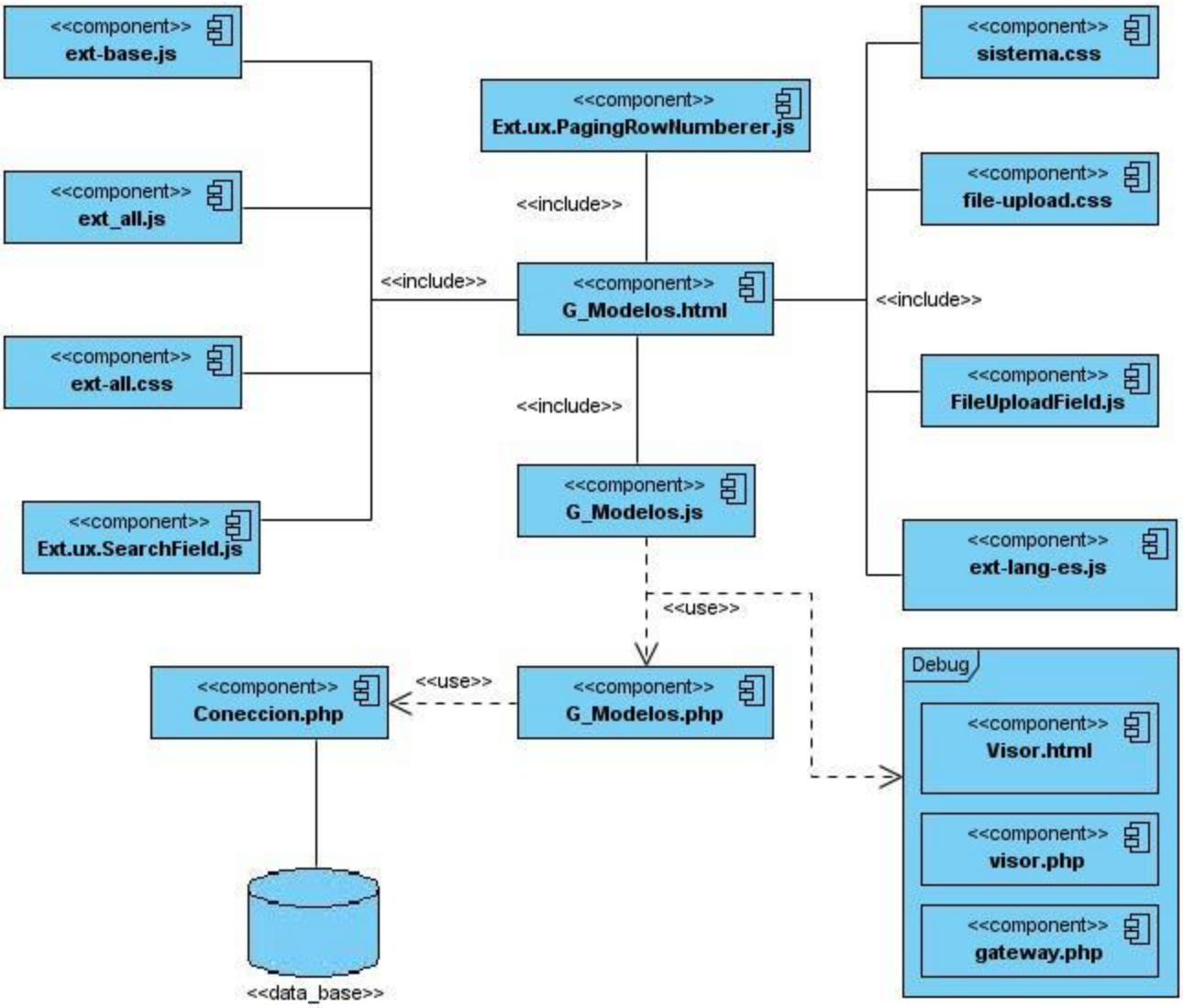

FIGURA 20: DIAGRAMA DE COMPONENTE CU GESTIONAR MODELOS.

Diagrama de Componente CU Gestionar Solicitud

Este diagrama representa los componentes de las gestión de solicitudes por parte del cliente. El cual brinda las funcionalidades de Adicionar una nueva solicitud o Descargar un solicitud, independiente de esto también brinda las opción de realizar búsquedas de

solicitud.

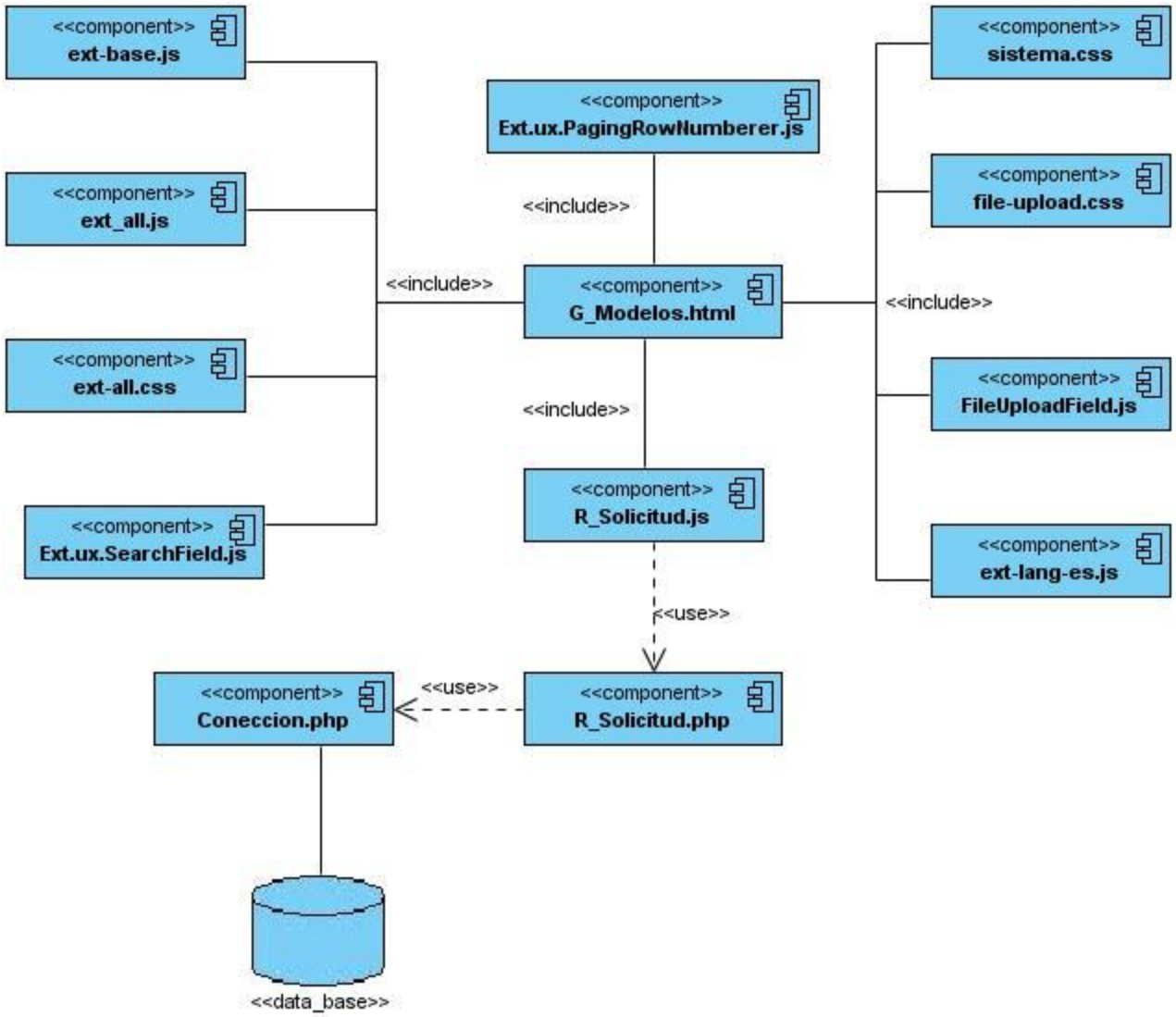

FIGURA 21: DIAGRAMA DE COMPONENTES CU GESTIONAR SOLICITUD

4.4. Modelo de Despliegue.

Con el Modelo de Despliegue se muestra la configuración del hardware del Sistema, así como los nodos físicos por los que está compuesto y sus respectivas conexiones.

El sistema se desarrolló siguiendo la arquitectura cliente servidor, donde se interactúa con el usuario a través de una estación de trabajo cliente desde donde se solicita una determinada petición al servidor. Del lado del servidor estará funcionando el servidor de base de datos PostgreSQL así como el servidor web Apache. La comunicación con el nodo del cliente que es donde se van a mostrar las interfaces para interactuar con el sistema se va a realizar por medio del protocolo HTTP.

Modelo de despliegue:

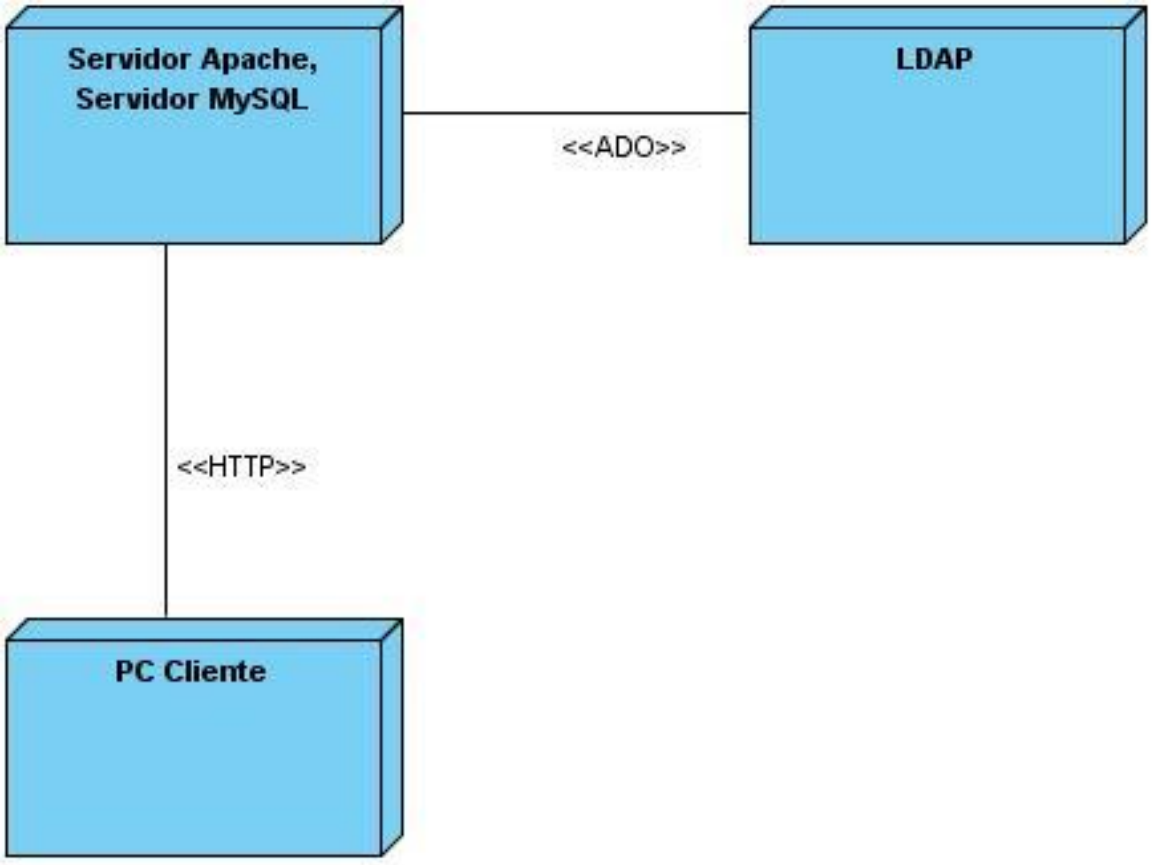

Figura 22: modelo de despliegue.

4.5. Interfaz de usuario.

La creación de una página web es posible con la unión de diferentes factores necesarios para su funcionamiento: aquello que se ve y es visible para el usuario - el diseño web y el contenido - y aquello que no se ve pero que es igual o más importante - la programación web, accesibilidad y la interfaz web. Todo junto forma una estructura virtual que tiene como función hacer llegar al usuario o cliente toda la información de manera adecuada y accesible. [22]

El desarrollo de la interfaz de nuestra aplicación se realizó utilizando las ventajas que nos ofrece el framework de desarrollo ExtJS. Este framework nos brinda excelentes efectos y ventanas que fueron fundamentales en el desarrollo de la aplicación. La visión de los implicados en el desarrollo de la aplicación fue el de acercar la interfaz de usuario lo más posible a una interfaz de aplicación de escritorio. A continuación los mostramos las principales funcionalidades de estas vistas.

Gestionar Solicitud

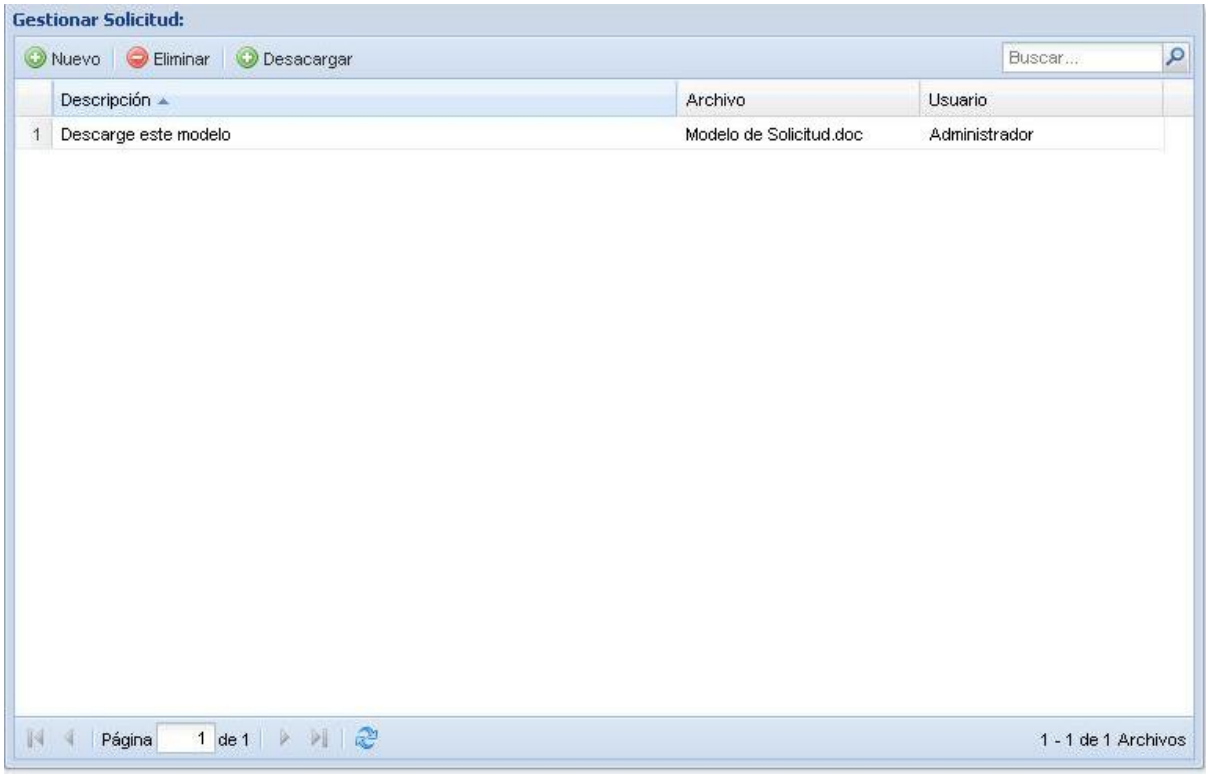

FIGURA 23: GESTION DE USUARIO

Gestión de Usuario

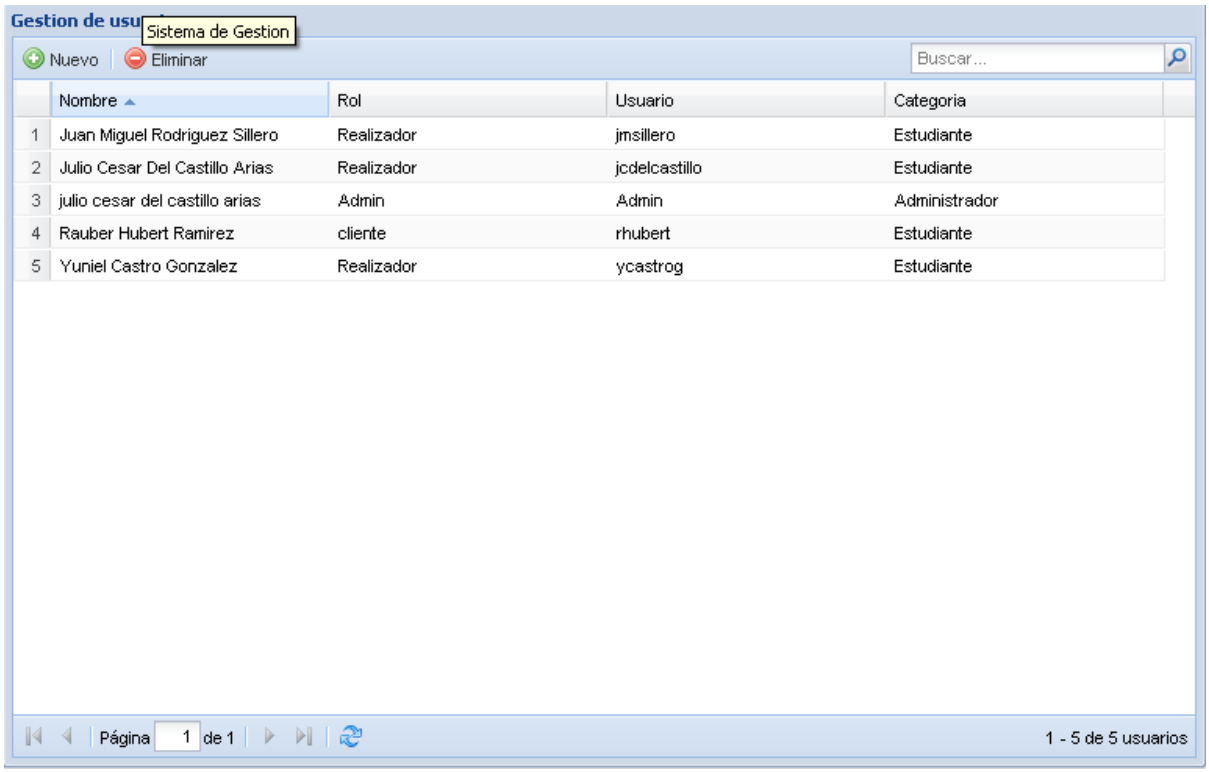

FIGURA 24: GESTIÓN DE USUARIO.

Gestión de Modelos

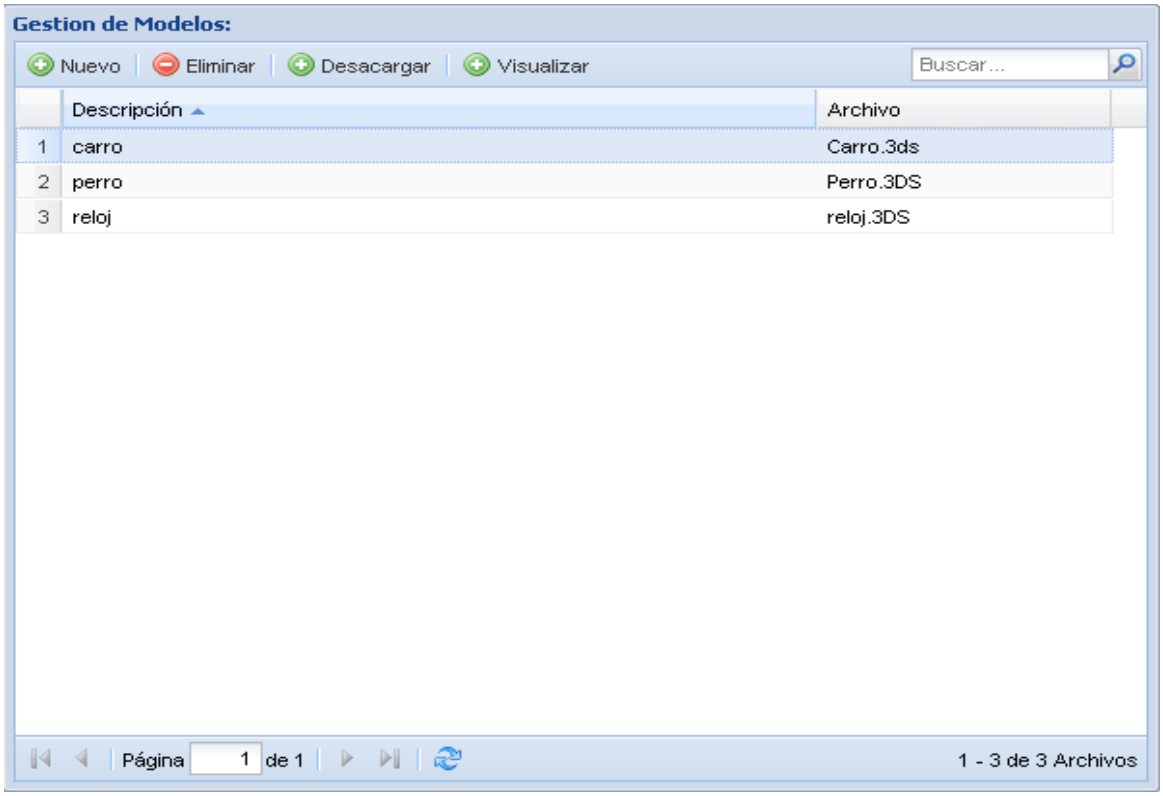

Figura 25: Gestión de modelo

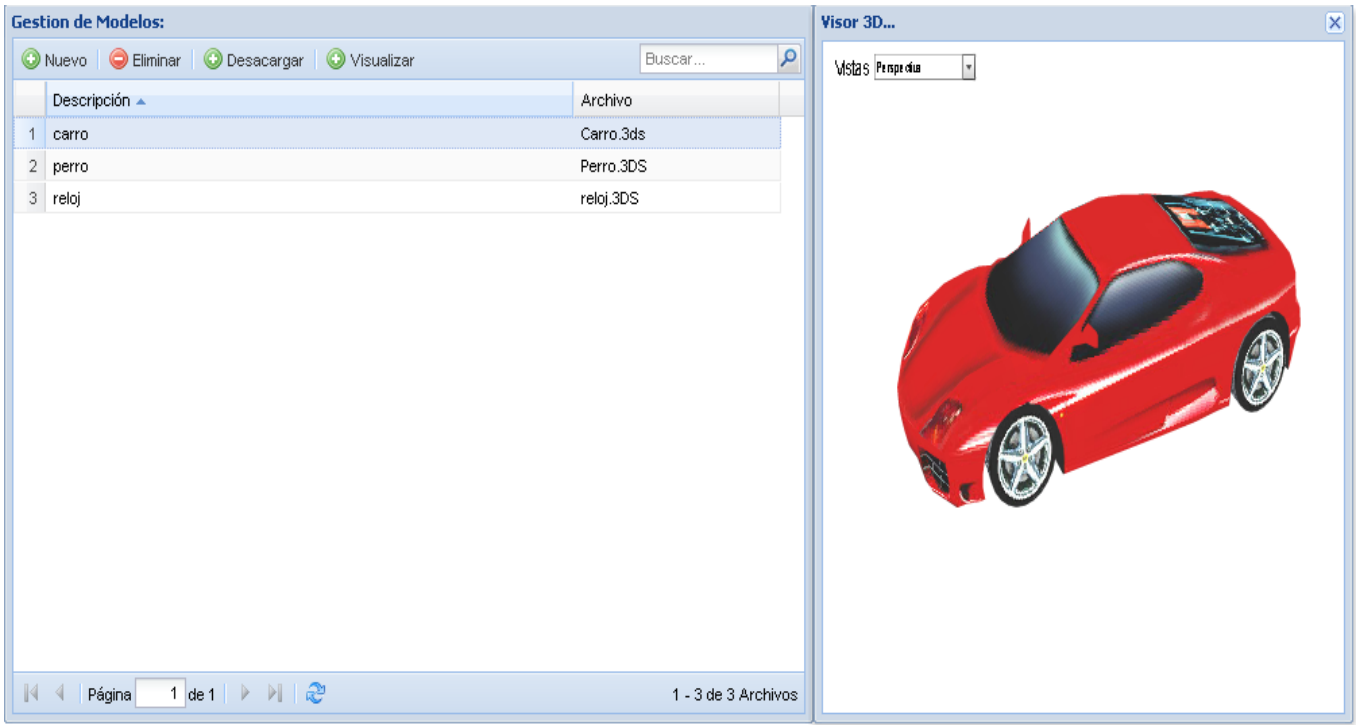

FIGURA 26: VISOR 3D.

El portafolio de trabajo de los RH de proyecto está compuesto por el Gestionar Modelos, Gestionar Videos, Gestionar Documentos y Gestionar Imagen, estos tres últimos tiene las mismas funcionalidades y el mismo aspecto que la figura que se les muestra a continuación Su única diferencia es el nombre ya que realizan las mismas funcionalidades.

Gestión de Videos

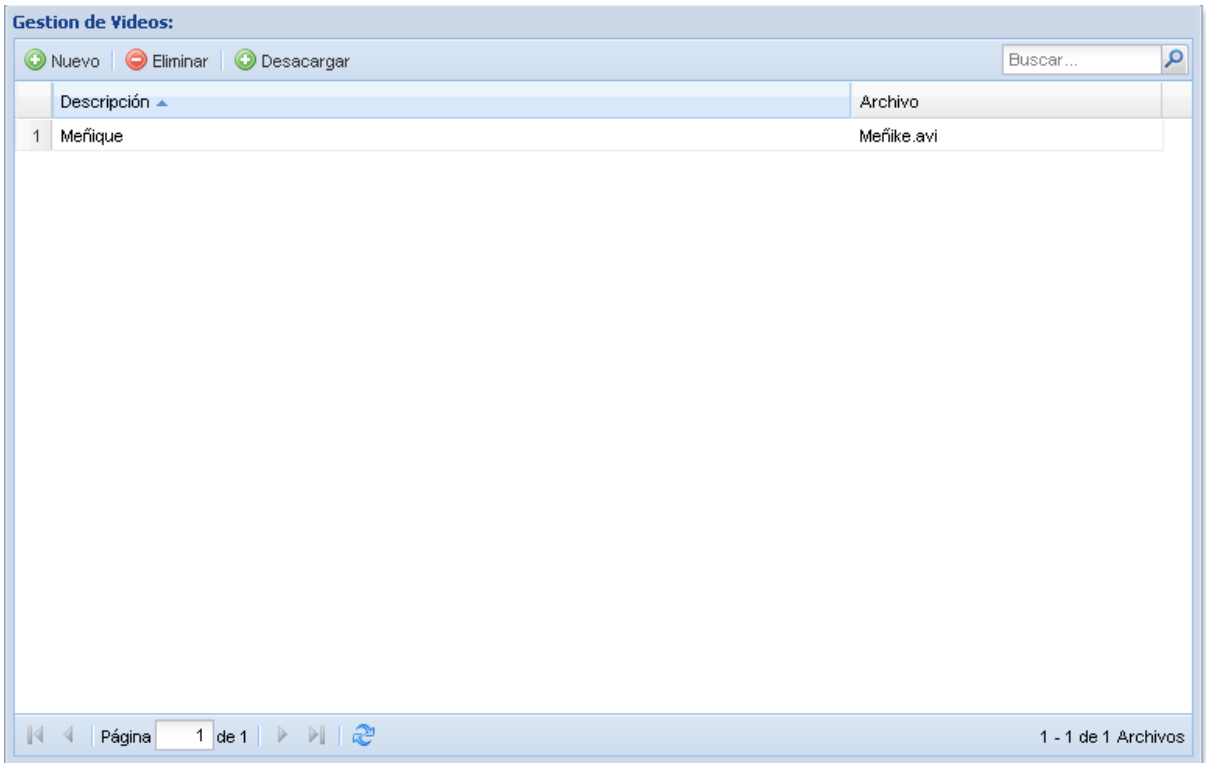

FIGURA 27: GESTION DE VIDEOS.

Estas son las funcionalidades principales de nuestro sistema.

4.6. Pruebas.

El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades. Es por esto que es necesario ir realizando pruebas al sistema.

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.

Diseño de caso de prueba CU Gestionar Modelo.

Descripción general: El caso de uso comienza cuando el realizador o el revisor decide realizar una acción en la sección de Modelos de su portafolio de trabajo, Adicionar, Insertar, Eliminar, Descargar o visualizar modelos 3D.

Condiciones de ejecución:

Debe estar autenticado en la aplicación.

Nombre de la sección	Escenario de sección	Funcionalidad	Flujo Central
SC1:Modelos	EC1.1: Gestionar Modelos.	Mostrar Gestor de Modelos.	El actor ya autenticado selecciona la opción de su portafolio “Modelos ”. El sistema muestra la interfaz donde se gestionan estos modelos.
	EC 1.2 Adicionar Modelo.	Adicionar Modelo	El actor decide adicionar un nuevo modelo y da clic en el botón Nuevo. El sistema muestra un ventana para cargar el modelo. Al dar clic en Grabar el modelo se carga y muestra en la interfaz.
	EC 1.3 Eliminar Modelo.	Eliminar Modelo.	El actor decide eliminar un modelo y da clic en el botón Eliminar.

			Si no tiene seleccionado el modelo el sistema le pide que seleccioné el modelo a eliminar. Elimina el modelo y lo quita de la interfaz.
	EC 1.4: Descargar Modelo.	Descargar Modelo.	El actor decide descargar un modelo y da clic en el botón Descargar. Si no tiene seleccionado el modelo el sistema le pide que seleccioné el modelo a descargar. El sistema muestra una ventana de descarga y salva el modelo.
	EC 1.5: Visualizar Modelo.	Visualizar Modelo.	El actor decide visualizar un modelo y da clic en el botón Visualizar. Si no tiene seleccionado el modelo el sistema le pide que seleccioné el modelo a visualizar. Cuando el usuario da clic en Visualizar se invoca y se muestra una ventana con el Visor 3D y el modelo cargado

TABLA 7: CASO DE PRUEBA CU GESTIONAR MODELO.

Diseño de caso de prueba CU Realizar Solicitud.

Descripción general: El caso de uso se inicia cuando el cliente o el administrador decide realizar una operación sobre la sección solicitud de servicios.

Condiciones de ejecución:

Debe estar autenticado en la aplicación.

Nombre de la sección	Escenario de sección	Funcionalidad	Flujo Central
SC 2: Solicitud	EC 2.1: Gestionar Solicitud.	Mostrar Gestor de Solicitud.	El actor ya autenticado selecciona la opción Solicitud. El sistema muestra la interfaz donde se gestionan las Solicitudes.
	EC 2.2 Adicionar Solicitud.	Adicionar Solicitud	El actor decide adicionar una nueva solicitud y da clic en el botón Nuevo. El sistema muestra un ventana para cargar la solicitud. Al dar clic en Grabar la solicitud se carga y muestra en la interfaz.
	EC 1.3 Eliminar Solicitud.	Eliminar Solicitud.	El actor decide eliminar una solicitud y da clic en el botón Eliminar. Si no tiene seleccionado la solicitud el sistema le pide que seleccioné la solicitud a eliminar. Elimina la solicitud y la quita de la interfaz.
	EC 1.4: Descargar Solicitud.	Descargar	El actor decide descargar un modelo y

		Modelo.	da clic en el botón Descargar. Si no tiene seleccionado el modelo el sistema le pide que seleccioné el modelo a descargar. El sistema muestra una ventana de descarga y salva el modelo.

TABLA 8: CASO DE PRUEBA CU REALIZAR SOLICITUD.

4.7. Conclusiones.

En este capítulo se representó un conjunto de diagramas que reflejan los componentes por los que está constituido el sistema, así como la distribución física de los mismos facilitando la comprensión de la estructura y composición de la aplicación.

Conclusiones.

Con la realización de este trabajo de diploma se alcanzaron los objetivos del mismo, se logró desarrollar una aplicación web funcional que permite gestionar los artefactos de escenarios 3D, el portafolio de trabajo de la comunidad y las solicitudes de los clientes

La aplicación web Sistema de gestión del portafolio de los recursos humanos y tareas de Escenarios 3D, le proporciona a los integrantes de la línea de desarrollo Imagen y video basados en Rendering a la cual pertenece nuestro proyecto Escenarios 3D una amplia gama de facilidades, estos pueden publicar sus trabajos en la misma, lo profesores o encargados de revisar las tareas que son asignadas a los miembros del proyecto no necesitan ir hasta el proyecto para ver cómo va la evolución de la tarea, con solo conectarse a nuestra aplicación puede revisar a distancia la evolución de dichas tareas.

Otra de las grandes ventajas que ofrece nuestra aplicación es la interacción con posibles clientes, estos a través de nuestra aplicación pueden ver los trabajos publicados por los miembros del proyecto, y realizar solicitudes de alguno de los servicios que brinde el proyecto.

La interfaz de nuestra aplicación se desarrolló utilizando el framework de desarrollo ExtJS, este framework posibilita la creación de interfaz de aplicación muy creativas e interesantes, ahora el uso de este framework puede traer algunos problemas con el navegador que se esté usando, por lo que se decidió optimizar la aplicación para que funcione a través del Firefox por lo que le rogamos nos disculpen por cualquier molestia que esto pueda provocarles.

Recomendaciones.

Con el desarrollo del Sistema de gestión del portafolio de los recursos humanos y tareas de Escenarios 3D. Se logró alcanzar el objetivo por el cual fue creada, es necesario seguir perfeccionando la aplicación para acercarla cada vez más a las necesidades del proyecto.

Se les recomienda prestar gran atención a las tecnologías y en especial a las de desarrollo web ya que estas no están exceptas de la evolución y todos los años publican en internet nuevas funcionalidades de estas o sale una versión mejor que la anterior. Y estas nuevas tecnologías pueden acercarnos cada vez más a la excelencia.

También se les recomienda realizar una investigación más profunda acerca de los motores 3d para la web, con el objetivo de ir perfeccionando cada vez más la visualización de los modelos 3d y para una posible migración hacia el X3D.

Además se les recomienda incluirle nuevas funcionalidades que mejoren la comunicación del usuario con la aplicación o con el resto de los usuarios. Otra de las recomendaciones es que se cree un portal web para el para la línea de desarrollo Escenarios 3D que reciba la aplicación.

Se recomienda integrarle un chat a la aplicación, para establecer una comunicación y un proceso de retroalimentación entre los usuarios del sistema.

Glosario de términos.

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.

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.

GNU/GPL: Licencia Pública General de GNU o más conocida por su nombre en inglés GNU *General Public License* o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la *Free Software Foundation* a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

GNU/Linux: Es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente.

Interfaz de usuario: Engloba la forma en la que el operador interactúa con el ordenador, los mensajes que éste recibe en pantalla, las respuestas del ordenador a la utilización de periféricos de entrada de datos, etc.

Citas Bibliográficas.

1. La importancia de internet hoy en dia http://www.universopyme.com.mx/index.php?option=com_content&task=view&id=2520&Itemid=373.

2. ¿Que son los sistemas de gestion? http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion

3. *¿Por que son importantes?* http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion.

4. *Metodologías* de desarrollo de software http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html

5. Rational Rose Enterprise. [Online] 2007. http://www.rational.com.ar/herramientas/roseenterprise.html.

6. Proceso de desarrollo OpenUp http://cbasqa.wordpress.com/2008/09/02/proceso-de-desarrollo-openup/

7. Herramientas Case http://www.mitecnologico.com/Main/HerramientasCase

8. Metodologia Rup http://www.buenastareas.com/ensayos/Metodologia-De-Rup/2051528.html

9. El lenguaje Unificado De Modelado (UML) http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html.

10. Yamisleydis Borrero Lao (2009), Sistema de Gestión Información Gineco-Obstétrico, Ciudad de la Habana.

11. Rodolfo Smesch, Aplicaciones web http://www.rodolfosemsch.com/articulos/080902-webapps.php.

12. Aplicaciones de Escritorio http://www.desarrolloweb.com/wiki/aplicacion-de-escritorio.html

13. Conceptos básicos del servidor web http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php

14. Capítulo 10. Servidor Apache HTTP http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-httpd.html

15. Ext JS lo bueno, lo malo y lo feo http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/

16. Los diferentes lenguajes de programación para la web http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/

17. Glosario de términos http://www.slideshare.net/danielajimenez/glosario-de-trminos-4219280

18. Sistemas de gestión de BD http://limared.blogspot.com/2010/12/sistemas-de-gestion-de-bases-de-datos.html

19. MySQL http://www.desarrolloweb.com/directorio/bases_de_datos/mysql/

20. PostgreSQL http://danielpecos.com/docs/mysql_postgres/x15.html

21. Interfaz de usuario http://www.drauta.com/interfaces-web/

22. Todoroms. *Todoroms*. [Online] Marzo 25, 2011. [Cited: junio 14, 2011.] http://www.todoroms.com/away3d-3-6-essentials-3

Bibliografía.

1. *Cursos de Extjs por Crysfel* http://www.quizzpot.com/2009/02/ext-js-framework-en/
2. Programacióon en Internet 2006-2007. Universidad de Alicante.
3. Introducción a AJAX www.librosweb.es.
4. Introducción a JavaEscript www.librosweb.es.
5. Manual de PHP 2006.
6. Curso de PHP-MySql por Jesus Conde http://www.illasaron.com/.
7. Curso de PHP http://www.webtaller.com/info/curso-php.php .
8. Metodologias de Desarrollo de Software por María A. Mendoza Sánchez http://www.informatizate.net
9. ¿Que son las Aplicaciones Web? http://www.esenciahumana.com.mx
10. Historia de las Aplicaciones Web http://www.cibernetia.com.
11. PHP Programming solutions by Vikram Vaswani.
12. PostGreSQL vs MySQL por Daniel pecos http://danielpecos.com/docs/mysql_postgres/index.html
13. Metodologia de Desarrollo de Software www.bloger.com.
14. Metodología de desarrollo de Software http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
15. Rup http://www.rational.com.ar/herramientas/rup.html
16. Sistemas Gestores de Base Datos http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html
17. [MySQL_Manual] *Manual de MySQL*, http://www.mysql.com/documentation/index.html.

18. [PostGreSQL_Manual] *Manual de PostgreSQL*, http://www.mysql.com/documentation/index.html.

19. [Article_MySQL-PostGreSQL] *Artículo comparativo*, http://www.phpbuilder.com/columns/tim20000705.php3?page=1 .

20. Visual Parading http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(MÍ)_14720_p/ .

21. Rational Rose Enterprise http://www.rational.com.ar/herramientas/roseenterprise.html

22. Away3d http://www.esedeerre.com/portada/17/away3d-flash-ejemplos-tutoriales-actionscript3.0

23. Away3D, motor 3D Para Flash http://away3d.com.

24. Tutoriales away3D http://www.whoarethispeople.com/blog/index.php?post=Tutorial-Away3D-(I).