

**ANÁLISIS, MANTENIMIENTO, DISEÑO, DESARROLLO E IMPLEMENTACIÓN  
DE NUEVAS FUNCIONALIDADES Y MEJORAS RELACIONADAS CON EL  
MODULO TRABAJOS DE GRADO EN LA PLATAFORMA COMA COMUNIDAD  
ACADÉMICA.**

**CARLOS ANDRES PEREZ RIVERA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2018**

**ANÁLISIS, MANTENIMIENTO, DISEÑO, DESARROLLO E IMPLEMENTACIÓN  
DE NUEVAS FUNCIONALIDADES Y MEJORAS RELACIONADAS CON EL  
MODULO TRABAJOS DE GRADO EN LA PLATAFORMA COMA COMUNIDAD  
ACADÉMICA.**

**CARLOS ANDRES PEREZ RIVERA**

**Trabajo de grado para optar al título de  
Ingeniero de Sistemas**

**Director**

**Msc. LUIS IGNACIO GONZÁLEZ RAMÍREZ**

**Magíster en Informática**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2018**

## CONTENIDO

	pág.
INTRODUCCIÓN .....	12
1. PLANTEAMIENTO DEL PROYECTO.....	13
1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL .....	13
1.2 JUSTIFICACIÓN.....	14
1.3 OBJETIVOS.....	16
1.4 IMPACTO Y VIABILIDAD.....	17
2. MARCO TEÓRICO .....	18
2.1 ARQUITECTURA CLIENTE/SERVIDOR.....	18
2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS.....	22
2.3 BASES DE DATOS.....	25
2.4 NETBEANS.....	29
2.5 SISTEMA DE CONTROL DE VERSIONES.....	29
2.6 PROGRAMACIÓN UTILIZADA.....	30
2.7 SERVIDORES WEB .....	33
3. MARCO METODOLÓGICO .....	34
3.1 PROTOTIPO EVOLUTIVO .....	34
3.2 LENGUAJE DE MODELADO UNIFICADO.....	36
3.3 ESTÁNDARES DE PROGRAMACIÓN.....	40
4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO.....	43
4.1 PROTOTIPO ESPERADO.....	43
4.2 DIAGRAMAS DE CASOS DE USO .....	47

4.3 DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA .....	49
4.4 DISEÑO Y ANÁLISIS.....	52
4.5 DESCRIPCIÓN DE LAS ENTIDADES.....	53
4.6 MODELO DE PROCESOS DEL SISTEMA.....	53
4.7 IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES.....	53
4.8 MANTENIMIENTO Y ADMINISTRACIÓN.....	55
5 PRUEBAS DEL SISTEMA .....	56
5.1 PRUEBAS DE VERIFICACIÓN .....	56
6. CONCLUSIONES .....	59
7. RECOMENDACIONES.....	60
BIBLIOGRAFÍA.....	61
ANEXOS .....	63

## LISTA DE TABLAS

	<b>pág.</b>
Tabla 1. Casos de uso: Cambiar decisión.....	49
Tabla 2. Casos de uso: Intercambio de mensajes.....	50
Tabla 3. Casos de uso: Cancelación de proyecto con plan aprobado.....	50
Tabla 4. Casos de uso: Solicitud de cancelación .....	51
Tabla 5. Descripción de las Entidades.....	53
Tabla 6. Pruebas realizadas: Toma y cambio de decisiones.....	57
Tabla 7. Pruebas realizadas: Intercambio de mensajes.....	57
Tabla 8. Pruebas realizadas: Cancelación de proyecto con plan aprobado .....	58
Tabla 9. Pruebas realizadas: Solicitud de cancelación .....	58

## LISTA DE FIGURAS9

	<b>pág.</b>
Figura 1. Modelo Cliente/Servidor.....	18
Figura 2. Modelo de acceso a JSP .....	25
Figura 3. Prototipo Evolutivo .....	34
Figura 4. Diagramas de Casos de Uso .....	37
Figura 5. Diagrama de Secuencias .....	40
Figura 6. Diagrama de Casos de Uso: Cambiar decisión.....	47
Figura 7. Diagrama de Casos de Uso: Intercambio de mensajes autor y director. ....	47
Figura 8. Diagrama de Casos de Uso: Intercambio de mensajes autor y evaluador. ....	48
Figura 9. Diagrama de Casos de Uso: Cancelación de proyecto con plan aprobado .....	48
Figura 10. Diagrama de Casos de Uso: Solicitud de cancelación .....	49
Figura11. Diagrama E/R: Cambio Decisión.....	52
Figura 12. Diagrama de secuencia: Cambiar Decisión .....	63
Figura 13. Diagrama de secuencia: Intercambio de mensajes autor y director .....	64
Figura 14. Cancelación de proyecto con plan aprobado .....	65
Figura 15. Diagrama de secuencia: Solicitud de cancelación .....	66

## LISTA DE ANEXOS

<b>Anexo A. Modelo de Procesos del Sistema .....</b>	<b>63</b>
--	-----------

## RESUMEN

**TÍTULO:** Análisis, mantenimiento, diseño, desarrollo e implementación de nuevas funcionalidades y mejoras relacionadas con el módulo trabajos de grado en la plataforma COMA Comunidad Académica.\*

**AUTOR:** Carlos Andres Perez Rivera\*\*

**PALABRAS CLAVE:** Portal Web, Trabajos de Grado, Comité, COMA, Solicitudes, Conversación, Módulo.

## DESCRIPCIÓN

En la actualidad las escuelas cuentan con un sistema de información orientado a la Web que se encarga de la administración y control de las diferentes actividades tanto académicas, como administrativas que se realizan dentro de las escuelas y las Facultades, así como del control de usuarios y servicios que se les proporcionan.

La plataforma COMA presta servicios que están orientados a promover la interacción entre los miembros de la comunidad. Actualmente el grupo Calumet brinda soporte al servicio Trabajos de Grado, en el marco de este proceso se hace necesario realizar cambios, modificaciones y nuevas implementaciones tales como: Permitir aprobar de inmediato las solicitudes de cancelación de proyecto, permitir a uno de los autores de un proyecto de grado hacer solicitud de cancelación de proyecto cuando otro autor no está matriculado, implementar notificaciones mediante correo electrónico a los miembros del Comité de Proyectos de Grado al iniciar y reiniciar una sesión de comité, creación de una interfaz que reúna todos los menús del administrador de Trabajos de Grado.

Actualmente no existe un espacio de comunicación que permita la interacción entre autores, directores y evaluadores en el módulo Trabajos de Grado. Por lo cual es necesario la creación de un servicio de intercambio de mensajes en las diferentes secciones dentro de dicho módulo.

En el comité de Trabajos de Grado luego de tomar una decisión no hay forma de revertirla, para esto se hizo necesario la creación de un servicio que permite revertir la decisión tomada durante una sesión de comité, teniendo en cuenta que dicha sesión termina al finalizar el día.

---

\* Trabajo de grado

\*\* Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Msc. Luis Ignacio González Ramírez

## ABSTRACT

**TITLE:** Analysis, maintenance, design, development and implementation of new functionalities and improvements related to Trabajos de Grado module in the COMA platform Academic Community.\*

**AUTHOR:** Carlos Andres Perez Rivera\*\*

**KEY WORDS:** Web Portal, Trabajos de Grado, Comité, COMA, Requests, Conversation, Module.

## DESCRIPTION

At present the schools have a web-oriented information system that is in charge of the administration and control of the different academic and administrative activities that are carried out within the schools and faculties, as well as the control of users and services that are provided.

The COMA platform provides services that are designed to promote interaction among community members. Currently, the Calumet group provides support to the Grade Jobs service, within the framework of this process it is necessary to make changes, modifications and new implementations such as:

Allow immediate approval of project cancellation requests, allow one of the authors of a degree project to request cancellation of project when another author is not enrolled, implement notifications by email to the members of the Project Committee of Degree initiate and restart a committee session, creating an interface that gathers all the menus of the administrator of Degree Works.

Currently, there is no communication space that allows interaction between authors, directors and evaluators in the Degree Works module. Therefore, it is necessary to create a message exchange service in the different sections within said module.

In the Committee of Degree Work after making a decision there is no way to reverse it, for this it became necessary to create a service that allows you to reverse the decision made during a committee session, taking into account that the session ends at the end of the session day.

---

\* Degree Work

\*\* Faculty of Physico-Mechanical Engineering. Department of Systems Engineering and Computing Science. Supervisor Msc. Luis Ignacio González Ramírez

## INTRODUCCIÓN

Los Portales Web Comunidad Académica son los principales canales de comunicación e integración entre los miembros de la comunidad de las escuelas y miembros en general de la comunidad UIS. Actualmente los servicios se han extendido a más escuelas de las distintas facultades gracias a la aceptación de los usuarios por lo que se adopta el objetivo de mantener y mejorar los servicios que se ofrecen, además de crear nuevos servicios que satisfagan las necesidades crecientes de los usuarios de la comunidad académica.

El grupo CALUMET, grupo de desarrollo de software de la escuela de Ingeniería de Sistemas, se encarga de desarrollar los nuevos servicios y darle mantenimiento a los existentes de manera que su actualización responda al continuo cambio. Con el fin de llevar a su cumplimiento el objetivo principal del grupo y proporcionar portales web con contenido dinámico, se cuenta con herramientas software de libre distribución como Netbeans, Github, Sqlyog, JavaScript, jQuery, entre otras.

En este documento se presenta un soporte teórico, metodológico y técnico del desarrollo realizado en el servicio de Trabajos de Grado con el fin de soportar las necesidades de la comunidad se realizaron los siguientes cambios, modificaciones y nuevas implementaciones. La creación de un nuevo servicio de intercambio de mensajes que permita la comunicación entre autores, directores y evaluadores en el módulo trabajos de Grado; Reingeniería a todos los menús que permiten cambio de decisiones en la sesión del comité; Implementación de interfaz que reúne todos los menús del administrador de Trabajos de Grado y por último la creación de nuevas funcionalidades implementadas en el módulo Trabajos de Grado.

## **1. PLANTEAMIENTO DEL PROYECTO**

### **1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL**

En la actualidad las Escuelas y las Facultades de la UIS, cuentan con un sistema de información orientado a la Web que se encarga de la administración y control de las diferentes actividades tanto académicas como administrativas que se realizan dentro de la escuela y la Facultad, así como del control de usuarios y servicios que se les proporcionan.

Los servicios de los portales Web de las escuelas deben mejorar constantemente y adaptarse a los cambios que se presenten en su entorno, a su vez deben dar solución a los problemas y necesidades que surjan por parte de los usuarios del sistema para incrementar su tiempo de vida útil y no llegar a convertirse en un software obsoleto, razón por la cual las labores de mantenimiento y actualización se hacen indispensables.

Actualmente en el módulo Trabajos de Grado no existe un espacio de comunicación que permita la interacción entre autores, directores y evaluadores en las secciones Gestionar Tema y Subir Plan, para esto se debe generar el espacio de comunicación en dichas secciones. Al momento de tomar una decisión en una sesión de Comité de Trabajos de Grado no existe actualmente una manera de revertir la decisión tomada durante la sesión, luego se hace necesario crear los menús correspondientes para poder modificar las decisiones tomadas y además implementar las funcionalidades necesarias para generar el acta correspondiente al final del día. Los menús del administrador se encuentran actualmente dispersos en diferentes secciones lo cual complica las tareas del usuario administrador, por lo tanto surge la necesidad de crear una interfaz que reúna organizadamente dichos menús para facilitar el trabajo del usuario administrador.

## 1.2 JUSTIFICACIÓN

El Portal Web Comunidad Académica (COMA), que desarrolla y mantiene el grupo Calumet, es una plataforma que provee lugares, adicionales a los ya existentes, de encuentro y comunicación a la comunidad de los programas de las diferentes escuelas. Esta plataforma contribuye a la realización de la misión en sus tres ejes: extensión, investigación y docencia. En ella, la Comunidad podrá mantener actualizada su hoja de vida, publicar noticias, proponer eventos, consultar y crear agendas, compartir documentos, enviar correos, y otros servicios que podrán hacer más fácil la realización de la misión institucional. Los servicios de los portales web deben actualizarse constantemente, mejorar y adaptarse a los cambios que se presenten en su entorno, satisfaciendo las necesidades de los usuarios del sistema. Esto permite aumentar el tiempo de vida útil del software evitando que pase a ser obsoleto con la creación de nuevos servicios, y con el mantenimiento realizado a los existentes.

En el módulo de Trabajos de Grado se deben realizar los siguientes cambios, modificaciones y nuevas implementaciones para soportar las necesidades de la comunidad:

- Permitir aprobar de inmediato las solicitudes de cancelación de proyecto con plan aprobado por parte del Comité de Proyectos de Grado.
- Permitir a uno de los autores de un proyecto de grado hacer solicitud de cancelación de proyecto de grado cuando otro autor no está matriculado, además que el director del proyecto pueda dar aval a la solicitud y así se pueda tomar decisiones en el Comité de Proyectos de Grado.
- Servicio de intercambio de mensajes que permita la interacción entre autores, directores, evaluadores y comité en el módulo Trabajos de Grado, en las secciones Gestionar Tema, Subir Plan y Avalar Solicitud. Además la notificación de las participaciones de los usuarios mediante correo electrónico.
- Implementación de notificaciones mediante correo electrónico a los miembros del Comité de Proyectos de Grado al iniciar y reiniciar una sesión de comité,

informando la fecha de inicio y reinicio de la sesión, el usuario desde el cual se inicia y reinicia la sesión, el lugar de la sesión y los asistentes a la sesión.

- Realizar reingeniería a todos los menús que permiten cambio de decisiones en la sesión del Comité de Proyecto de Grado.
- Implementar una interfaz que reúna todos los menús del administrador de Trabajos de Grado.

## 1.3 OBJETIVOS

**1.3.1 Objetivo general** Realizar las funciones de mantenimiento, análisis, diseño, desarrollo e implementación del nuevo servicio de interacción entre autores, directores y evaluadores en el módulo trabajos de grado y nuevas funcionalidades en la sesión del comité, entre otros, con el fin de hacerlos óptimos, eficientes y sostenibles, permitiendo acceder a la información de manera más sencilla, ágil, óptima y eficiente.

### 1.3.2 Objetivos específicos

- Realizar la aprobación inmediata de solicitudes de cancelación de proyecto con plan aprobado por parte del comité de proyectos de grado.
- Permitir hacer solicitudes de cancelación de proyecto de grado, dar aval por parte del director y tomar decisiones en el comité de proyecto de grado cuando un autor no está matriculado.
- Crear nuevo servicio de intercambio de mensajes que permita la comunicación entre autores, directores, evaluadores y comité en el módulo trabajos de grado considerando los siguientes aspectos:
  - La implementación de este servicio en tema, plan y solicitudes dentro del módulo trabajos de grado.
  - Notificar las participaciones de los usuarios a través de correo electrónico.
- Enviar notificaciones vía correo electrónico a los miembros del comité de proyectos al iniciar y reiniciar una sesión del comité, informando la fecha y hora de inicio o reinicio de sesión, el usuario desde el cual se inicia y reinicia la sesión, el lugar de reunión y los asistentes.
- Realizar reingeniería a todos los menús que permiten cambio de decisiones en la sesión del comité.
- Implementar una interfaz que reúna todos los menús del administrador de Trabajos de Grado.

## **1.4 IMPACTO Y VIABILIDAD**

**1.4.1 Impacto.** Los portales web de las escuelas han sido una herramienta útil para el manejo de la información, por lo tanto, es necesario realizar labores de administración y mantenimiento, para ofrecer al usuario un sitio más confiable.

Se pretende que los procesos que se realizan en las escuelas cada día sean más ágiles, dinámicos, seguros y eficientes, permitiendo una mejor organización de la información, razón por la cual se crean nuevos servicios y se hace reingeniería a servicios existentes para que se ajusten a las necesidades que puedan surgir.

**1.4.2 Viabilidad.** La administración del sitio soporte a usuarios, mantenimiento y desarrollo de nuevos servicios es completamente viable pues se usará software de libre distribución, recurso humano preparado para tal fin, servidores que marchan de manera legal en la escuela y la facultad, equipos disponibles y todo el soporte tecnológico necesario para el desarrollo del mismo.

Además, se cuenta con la supervisión por parte del director del proyecto y la colaboración del equipo de trabajo CALUMET, agentes de gran apoyo en la realización de este proyecto.

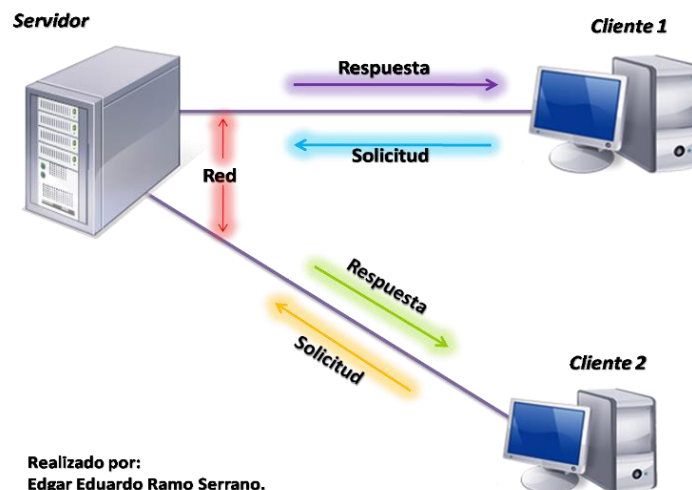
## 2. MARCO TEÓRICO

### 2.1 ARQUITECTURA CLIENTE/SERVIDOR

En la arquitectura cliente/servidor cada uno de los clientes produce un mensaje solicitando un determinado servicio a un servidor (hace una petición) estos envían uno o varios mensajes como respuesta (responden peticiones o provee un servicio). La mayoría del trabajo pesado (procesos de base de datos, procesar la lógica de la aplicación entre otros) está a cargo de los servidores, el cliente comúnmente se encarga de las funciones de administración de la interfaz de usuario, interacción con el usuario, recibir resultados del servidor, generar requerimientos de base de datos, entre otros.

Esta idea se puede aplicar tanto a programas que se están ejecutando en una sola máquina, pero es más ventajosa en un sistema operativo multiusuarios distribuidos a través de una red de computadores.

Figura 1. Modelo Cliente/Servidor



Fuente: <http://cramercontracramer.blogspot.com.co/2013/12/modelo-cliente-servidor.html>

**2.1.1 Características de la arquitectura Cliente/Servidor.** Las características básicas de una arquitectura Cliente / Servidor son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
- El proceso del cliente da la interface entre usuarios y el resto del sistema, maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades independientes.
- Las tareas del cliente y el servidor tienen diferentes requerimientos como: velocidad del procesador, memoria o capacidad del disco, por tanto, la plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma y eso se conoce como ambiente heterogéneo.
- La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar el rendimiento y la escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores. Se puede realizar independientemente cambios en las plataformas de los clientes o de los servidores, ya sea actualización o reemplazo tecnológico, de manera transparente para el usuario final.

**2.1.2 Clasificación de las arquitecturas Cliente/Servidor.** Los sistemas Cliente/Servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece. Se distinguen tres componentes básicos de software para la clasificación:

- **Presentación:** Presentación de resultados al usuario de forma comprensible.
- **Lógica de aplicación:** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Está compuesta por los archivos que contienen los datos persistentes de la aplicación.

La siguiente es la clasificación de la arquitectura Cliente/Servidor

**2.1.2.1 Arquitectura Cliente/Servidor de dos capas.** Consiste en una capa de presentación y lógica de la aplicación; y otra de la base de datos, cuando el cliente solicita recursos entonces el servidor responde directamente a la solicitud con sus propios recursos. Normalmente esta arquitectura es utilizada en las siguientes situaciones:

- Cuando se requiere poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

**2.1.2.2 Arquitectura Cliente/Servidor de tres capas.** Define como organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, es decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no conscientes de ningún detalle o interfaz de las superiores, está compuesta de:

- Un equipo cliente con una interfaz de usuario (habitualmente se utiliza un navegador web), que solicita los recursos.
- El servidor de aplicaciones (o software intermedio), cuya tarea es prestar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos que almacena y proporciona al servidor de aplicaciones los datos que requiere.

**2.1.3 Arquitectura Cliente/Servidor aplicada.** En el desarrollo de este proyecto se recurre a arquitectura de tres capas, debido a las ventajas ofrecidas como: Escalabilidad, fácil mantenimiento y el manejo de un mayor número de usuarios que la ofrecida por la arquitectura C/S de dos capas. La arquitectura es aplicada de la siguiente forma:

- Capa de Cliente: Interfaz con el usuario, se usa un navegador web.
- Capa Intermedia: Para los servicios del negocio se utiliza un computador configurado como servidor web, el cual almacena el portal web conformado por páginas JSP y JavaBeans. Allí se realizan los procesos complejos, y se solicitan los servicios del servidor de datos cuando es necesario acceder a la información almacenada en la base de datos.
- Capa de Servidor: Se utiliza el motor de bases de datos MySQL, el cual se encuentra en el mismo servidor web.

#### **2.1.4 Ventajas del esquema Cliente/Servidor**

- La existencia de plataformas de software y hardware de varios fabricantes y cada vez más a económicas contribuye a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Este esquema facilita la integración entre sistemas heterogéneos y comparte información permitiendo que las máquinas existentes puedan ser utilizadas con interfaces amigables al usuario, de esta forma integrar los computadores con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Facilita a los diferentes departamentos de una organización soluciones locales, permitiendo la integración de la información principal totalmente.

### **2.1.5 Desventajas del esquema Cliente/Servidor**

- El mantenimiento de los sistemas es complejo pues implica la interacción de diferentes partes hardware y software de diferentes proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con escasas herramientas para la administración y ajuste del desempeño de los sistemas, además se deben tener estrategias para el manejo de errores y para salvaguardar la consistencia de los datos.
- La seguridad del esquema C/S es preocupante, un ejemplo: las validaciones y verificaciones que se deben hacer tanto en el cliente como en el servidor.
- El desempeño es un aspecto a tener en cuenta en el esquema C/S, problemas de este estilo pueden presentarse por congestión en la red.

## **2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS**

Las páginas dinámicas aportan grandes beneficios porque permiten entrar a bases de datos para extraer información que pueda presentarse al usuario, dependiendo de algunos permisos y de la misma forma para almacenar información.

Existen diferentes tecnologías para el desarrollo de páginas dinámicas entre ellas están:

**2.2.1 Código del Lado del Cliente (Client Side Scripts).** Código ejecutado por los navegadores, el cual los computadores clientes tienen instalados. Las tecnologías más comunes de este tipo son:

- JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite, crear ventanas, mostrar texto en movimiento y verificar las entradas a un formulario.

- **Controles Activos:** tecnología Microsoft que permite la creación de aplicaciones Windows, como pueden ser Visual Basic Script o Visual C. Es la respuesta de Microsoft a los Applets de Java.
- **Java Applets:** Programas escritos en lenguaje de programación Java, se incrustan en HTML y se ejecutan en el navegador gracias a la Máquina Virtual de Java (JVM) que lleva éste incorporado.

**2.2.2 Código del Lado del Servidor (Server Side Scripts).** Código que se ejecuta en el servidor. Para su actividad el programa ejecuta y procesa los datos o peticiones que el usuario envía desde su navegador, para luego enviar los resultados del programa en una página HTML que el usuario verá normalmente en su navegador. Los más usados son:

- **ASP (Active Server Pages):** Permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables, requiere de un computador configurado como Servidor Web de Microsoft (Microsoft Web Server), el navegador del cliente es indiferente pues el trabajo se realiza del lado del servidor. Da gran uso en la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras.
- **PHP (PHP Hypertext Pre-processor):** Lenguaje de programación interpretado, diseñado para la creación de páginas web dinámicas. Es un lenguaje de código abierto (Open Source) y gratuito. Su gran potencia se encuentra en la interacción con los motores de bases de datos como Oracle y MySQL.
- **JSP (Java Server Pages):** tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Permiten la utilización de código Java mediante scripts.

**2.2.3 Tecnología aplicada.** La tecnología aplicada para la creación del portal web fue JSP, por lo tanto, los nuevos servicios son desarrollados con esta misma tecnología, ya que permite producir aplicaciones independientes de la plataforma y portables a otros sistemas operativos y servidores web.

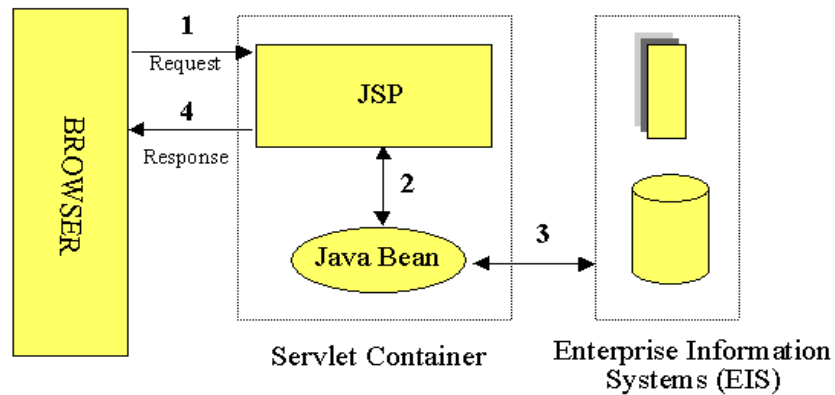
Las paginas JSP y servlets se ejecutan en la Máquina Virtual de Java, lo cual permite que se puedan usar en cualquier tipo de computador, siempre y cuando este instalada la Máquina Virtual de Java. Cada JSP se ejecuta en su propio contexto (llamado también hilo o hebra); pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo. Su persistencia permite hacer cosas de forma más eficiente como la conexión a bases de datos y manejo de sesiones.

Una página JSP se compila a una aplicación Java la primera vez que se invoca, y de esta aplicación Java se crea una clase que empieza a ejecutarse en el servidor como un servlet. Un JSP es una página web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

#### **2.2.3.1 Modelo de acceso a JSP**

- Un usuario en su navegador web cliente hace una petición que es enviada a un archivo JSP. Este archivo accede a componentes del servidor que generan contenido dinámico y lo presentan en el navegador.
- Después de recibir la petición del cliente, el archivo JSP pide información de un JavaBean si es necesario.
- El JavaBean en turnos puede pedir información de otro JavaBean o de una base de datos.
- Una vez el JavaBean genera el contenido, el archivo JSP puede consultar y presentar el contenido del JavaBean al navegador.

Figura 2. Modelo de acceso a JSP



Fuente: <http://geneura.ugr.es/~jmerelo/JSP/>

La primera vez que un archivo JSP es invocado, este es compilado en un objeto, la respuesta del objeto es HTML estándar, el cual es interpretado por el navegador para ser presentado al usuario. Después de la compilación, el objeto de la página es almacenado en la memoria del servidor. En las peticiones posteriores a esta página, el servidor revisa si el archivo JSP ha cambiado. Si no ha cambiado, el servidor utiliza el objeto de la página compilada guardado en memoria para generar la respuesta al cliente, en caso contrario el servidor automáticamente compila el archivo de la página y reemplaza el objeto en la memoria.

### 2.3 BASES DE DATOS

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, con una redundancia controlada y una estructura que refleja las interrelaciones y restricciones existentes en el mundo real. En la base de datos se almacena información considerada necesaria para una determinada organización o negocio.

Existen modelos que describen la estructura de una base de datos (entidades, atributos y relaciones), la mayoría de los modelos de datos poseen un conjunto de operaciones básicas como consultar y actualizar y eliminar.

**2.3.1 Modelos de Bases de Datos.** Las bases de datos se clasifican de acuerdo con su modelo de administración de datos. Algunos modelos utilizados con frecuencia son:

**2.3.1.1 Base de Datos Jerárquica.** Estas bases de datos almacenan su información en una estructura escalonada, organizando los datos en forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos, el nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se les conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Esta limitado por su incapacidad de representar eficientemente la redundancia de datos.

**2.3.1.2 Base de Datos de Red.** En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

**2.3.1.3 Base de Datos Relacional.** Es el más utilizado para modelar problemas reales y administrar datos dinámicamente. Su fundamento es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas

de una tabla). Los datos pueden ser recuperados o almacenados mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es el Lenguaje Estructurado de Consultas (Structured Query Language, SQL), un estándar implementado por los principales manejadores de bases de datos relacionales.

**2.3.2 Manejadores o Gestores de Bases de Datos.** El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de Datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser capturados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos a través de las sentencias en Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos.
- Respaldo y recuperación: Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia: consiste en controlar la interacción entre los usuarios concurrentes para preservar la consistencia de los datos.
- Seguridad e Integridad: consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

**2.3.3 MySQL.** Es un sistema de base de datos operacional considerado uno de los más importantes, utilizado por usuarios del medio para el diseño y programación de base de datos de tipo relacional. MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo. La característica más interesante de MySQL es que permite recurrir a las bases de datos multiusuario a través de la web y en diferentes lenguajes de programación y diferentes plataformas que se adaptan a diferentes necesidades y requerimientos, además MySQL es conocida por desarrollar alta velocidad de búsqueda de datos e información, a diferencia de sistemas anteriores.

#### **2.3.4 Ventajas de MySQL**

- El MySQL es un Open Source, o sea código abierto que puede ser usado y modificado.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consume puede ser ejecutado en una maquina con escasos recursos sin ningún problema.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL altamente apropiado para acceder a bases de datos en internet.

## **2.4 NETBEANS**

Es un entorno de desarrollo integrado (IDE), siendo una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas escritos en JAVA, pero puede servir para cualquier otro tipo lenguaje de programación. Netbeans es un producto libre y gratuito sin restricciones de uso.

- El Netbeans es un entorno de desarrollo integrado de código abierto escrito completamente en Java usando la plataforma Netbeans, soporta desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).
- La versión actual es NetBeans IDE 8.1 Desde NetBeans IDE 6.5 se extienden las características existentes del Java EE (incluyendo Soporte a Persistencia, EEJB 3 y JAX-WS). Adicionalmente, el Netbeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML.
- Todas las funciones del IDE son provistas por módulos. Cada Módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. Netbeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permite al usuario comenzar a trabajar inmediatamente.

## **2.5 SISTEMA DE CONTROL DE VERSIONES**

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, o el código fuente de un programa.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Además, el

repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

Existen multitud de sistemas de control de versiones, pero sin duda, el más popular es CVS (Concurrent Versions System). CVS tuvo el mérito de ser el primer sistema usado por el movimiento de código abierto para que los programadores colaboran remotamente mediante el envío de parches. Es de uso gratuito, código abierto, y emplea fusión de cambios. Subversión se creó para igualar y mejorar la funcionalidad de CVS, preservando su filosofía de desarrollo.

### **2.5.1 Subversión.**

Sistema de control de versiones iniciado por CollabNet Inc. Emplea licencia Apache/BSD. Se usa para mantener versiones actuales e históricas y los cambios de archivos tales como los de código fuente, páginas web y/o documentación. Esto permite recuperar versiones antiguas de los datos o examinar cómo han ido evolucionando esto. Su objetivo es ser un sucesor prácticamente compatible del ampliamente usado Concurrent Version system (CVS).

Subversión puede trabajar a través de redes, lo que permite que las personas que estén en diferentes computadores puedan usarlo, con la posibilidad de que varias personas modifiquen y gestionen el mismo conjunto de datos desde sus sitios promueve la colaboración, y como el trabajo está versionado, ya que, si se produce algún cambio incorrecto de los datos, sólo hace falta deshacerlo.

## **2.6 PROGRAMACIÓN UTILIZADA**

Para el desarrollo de este proyecto se usó la Programación Orientada a Objetos (P.O.O.). La P.O.O. es una de las formas más populares de programas que usa objetos y sus interacciones para diseñar aplicaciones y programas de computador, intenta simular el mundo real a través del significado de objetos que contienen

características y funciones; abstrae algunas características de sistemas naturales complejos como son:

- Atributos: estado del objeto.
- Métodos: comportamiento del objeto.
- Herencia: comportamientos comunes entre objetos relacionados para hallar relaciones de especialización y generalización de comportamientos.

**2.6.1 Clases.** Definición de todos los elementos que componen un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase. Por lo tanto, para realizar la abstracción de sistemas naturales, observamos y analizamos un grupo de cosas con características comunes, el resultado de esta abstracción será válido para todas estas cosas.

**2.6.2 Objetos.** Cualquier cosa real o abstracta, que posee atributos y un conjunto de operaciones que manipulan esos atributos que da un comportamiento particular. Un objeto es una instancia de una clase, el estado del objeto se determina por el estado (valor) de sus propiedades o características (atributos).

**2.6.3 Atributos.** Características de un objeto siendo un conjunto de datos (valores) y calificadores para aquellos datos. Estos atributos pueden ser desde tipos de datos simples (enteros, caracteres, cadenas de texto) hasta otros objetos.

**2.6.4 Métodos.** Son funciones o procedimientos propios de la clase que pueden tener acceso a los atributos de la misma para realizar las operaciones para los que son programados.

**2.6.5 Herencia.** Se fundamenta en usar una clase ya creada para tomar sus características en clases más especializadas o derivadas de ésta para reutilizar el código que sea común con la clase base, y solamente definir nuevos métodos o redefinir algunos de los existentes para ajustarse al comportamiento particular de esta subclase.

### **2.6.6 Beneficios de la Programación Orientada a Objetos**

- Permite obtener aplicaciones modificables y fácilmente extensibles a partir de componentes reutilizables.
- Disminución en el tiempo de desarrollo gracias a la reutilización del código.
- El desarrollo del software es más intuitivo porque las personas piensan naturalmente en términos de objetos más que en términos de algoritmos de software.

A continuación, se presenta una breve descripción de Java, el lenguaje de programación orientado a objetos que se usó en el desarrollo de este proyecto:

**2.6.7 Java y JDK (Java Development Kit).** Java es un lenguaje desarrollado por Sun Microsystems, en el año 2009 fue adquirida por la compañía Oracle.

Permite escribir aplicaciones que puedan ejecutarse en casi cualquier plataforma. El lenguaje toma parte de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Además, cuenta con una característica denominada “recolección de basura”, que examina la memoria y libera cualquier variable u objeto que no esté siendo usado. El JDK es un software que provee herramientas de desarrollo para la creación de programas en java.

Para trabajar con Java se necesita un kit de desarrollo que proporciona:

- Un compilador: javac
- Un intérprete: java.

- Un generador de documentación: *javadoc*
- Un visor de applet para generar sus vistas previas, ya que un applet carece de método main y no se puede ejecutar con el programa java: Appletviewer.

## **2.7 SERVIDORES WEB**

Es un tipo de software que se encuentra a la espera de una petición hecha por una aplicación cliente y da respuesta a dicha petición a través de una página web. Para cada transacción el servidor debe realizar dos acciones básicas: integrar todos los componentes de la página (texto, imágenes, vídeo, scripts, etc.) y enviarla rápidamente al usuario. A continuación, se describe el servidor Web que se ajusta a la tecnología escogida para el proyecto.

**2.7.1 Servidor Apache Tomcat.** Servidor de aplicaciones Java basado en los estándares definidos por Sun Microsystems. Tomcat es desarrollado como parte del proyecto de código abierto Jakarta de la fundación de software Apache y es uno de los servidores de aplicaciones Java más utilizados, en especial porque es liviano, cumple con todos los estándares, sencillo de instalar, tiene muy buena documentación y es gratuito, además por ser escrito en Java funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java (JVM).

Es posible ejecutarlo desde la línea de comandos (consola o terminal), después de configurar algunas variables de entorno, sin embargo, configurar cada variable de entorno y seguir los parámetros de la líneas de comando usados por Tomcat es tedioso y expuesto a errores, en su lugar se proporciona código existente para arrancar y detener el servicio.

### 3. MARCO METODOLÓGICO

#### 3.1 PROTOTIPO EVOLUTIVO

Para realizar los nuevos servicios para los portales web comunidad académica de las diferentes escuelas y facultades de la Universidad Industrial de Santander se propone como metodología de desarrollo el prototipo evolutivo.

Figura 3. Prototipo Evolutivo



La elección de la metodología de prototipo evolutivo se debe a las siguientes razones:

- Es deseable tener un bosquejo de lo que se desee mejorar o crear para poder incorporar sugerencias de cambio por parte de los usuarios del portal de las escuelas en las etapas tempranas del desarrollo.
- Por otra parte, es necesario saber lo antes posible si hemos interpretado correctamente las especificaciones y las necesidades de las escuelas y de los profesores.
- En muchos casos los usuarios no tienen una idea definida de lo que desean, por lo tanto, debemos tomar decisiones y suponer qué es lo que el usuario quiere.
- Por este motivo, la emisión de los prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado. Con el prototipo evolutivo se comienza diseñando y construyendo las partes más importantes de la aplicación en un prototipo que posteriormente se refinara y ampliará hasta que el prototipo se termine. Este prototipo será el software que se entregará al final.
- La decisión se fundamenta en la ventaja de la realización de los cambios en etapas tempranas y la posibilidad de emisión de varios prototipos evaluables durante el desarrollo, obteniéndose de este modo, y de forma paralela, una metodología integral también para el proceso de evaluación del programa.
- Esta metodología propicia un intercambio de conocimientos y de autocrítica al sistema, lo que conlleva a que se produzcan muchas pruebas antes de liberar una nueva versión, así como mejoras rápidas a problemas que puedan surgir durante su uso.

Procedimiento a seguir para la metodología planteada:

- Hacer un análisis de los requerimientos para la construcción de los prototipos.

- Desglosar los objetivos globales con el fin de tener una idea más detallada del software a realizar, mediante reuniones entre los desarrolladores y los usuarios, en las cuales se identifican los requerimientos de los usuarios y se concluyen los aspectos que requieren una mayor definición.
- Presentar al usuario el diseño de un prototipo enfocado en los aspectos visuales del software, métodos de entrada y formatos de salida, para proceder a la construcción del mismo.
- Evaluación del prototipo por parte del usuario para filtrar los requisitos del software a desarrollar.
- Se produce un proceso interactivo en el que el prototipo es depurado para satisfacer necesidades del usuario, de igual forma el desarrollador obtiene una mejor comprensión de lo que hay que hacer para la entrega del producto final de ingeniería requerido por el usuario.

### **3.2 LENGUAJE DE MODELADO UNIFICADO**

El Lenguaje de Modelado Unificado o Unified Modeling Language (UML), es el más utilizado en la actualidad. Es un lenguaje gráfico estándar para visualizar, especificar, construir y documentar un sistema para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

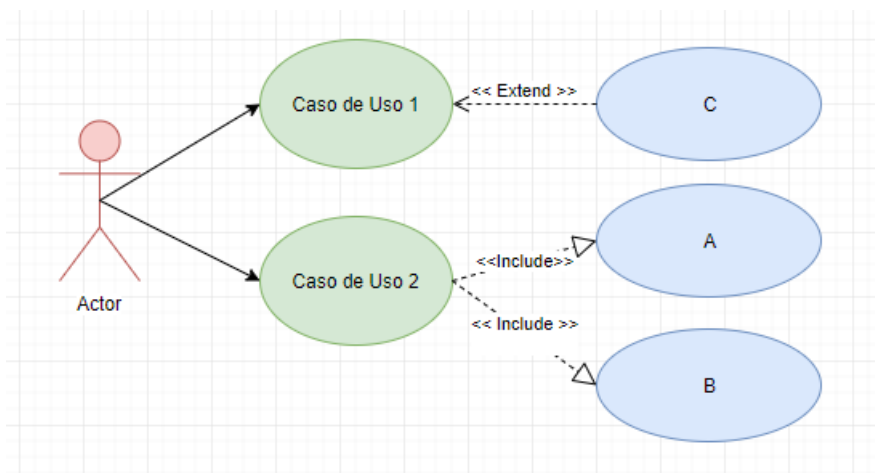
UML no es un método de desarrollo porque no indica los pasos a seguir para llegar al código, es decir, no especifica como pasar del análisis al diseño y de este al código. Al no ser un método de desarrollo resulta ser independiente del ciclo de desarrollo que se siga, puede encajar en un ciclo en cascada, evolutivo, espiral o en métodos ágiles de desarrollo.

**3.2.1 Diagramas de UML.** Los diagramas UML utilizados en el desarrollo de este proyecto fueron diagramas de casos de uso y diagramas de secuencias. Las principales razones por las cuales se prefirió UML como el lenguaje de modelado son:

- UML tiene una notación gráfica muy expresiva que permite representar todas las fases de un proyecto informático, desde el análisis con casos de uso, el diseño con diagramas de clases, objetos, etc.
- UML facilita el entendimiento de la información, la función y el comportamiento de un sistema, haciendo fácil el análisis de los requerimientos, ya que sirve de apoyo en los procesos de análisis de un problema.
- UML permite a los creadores de sistemas realizar diseños que faciliten la comunicación a otras personas de manera convencional.
- UML permite generar un punto de comparación entre lo logrado y lo planificado.

**3.2.2 Diagramas de casos de uso.** Representación gráfica del entorno del sistema (actores) y su funcionalidad principal. Describe lo que hace el sistema desde el punto de vista de un observador externo, concentrándose en expresar lo que hace el sistema y no en dar respuesta de cómo lograr su comportamiento.

Figura 4. Diagramas de Casos de Uso



**Actores:** Un actor en un caso de uso representa un rol, que alguien o algo puede desempeñar dentro un sistema y no un alguien o algo específico.

En este proyecto se destacan tres clases de actores:

- **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios tienen un perfil de administrador, con el cual tiene permisos extras a los que tiene un usuario comúnmente dentro del sitio; alguno de estos son los auxiliares de administración del portal, los profesores, las secretarías de las escuelas, entre otras. Dentro de esta categoría se incluye también el súper administrador.
- **Súper Administrador:** Es el usuario que puede administrar, controlar y modificar los portales web de las escuelas, sus parámetros y sus usuarios.
- **Usuario Portal Web Comunidad Académica:** Es el tipo de usuario común de los portales web y a quien van dirigidas las páginas de servicio. Este usuario solo tiene control sobre sus servicios permitidos.

**Inclusión (include–uses):** Es una forma de interacción, un caso de uso dado puede "incluir" otro. Una inclusión es utilizada para indicar que un caso de uso depende de otro, es decir, la funcionalidad de determinado caso de uso se requiere para realizar las tareas de otro. En la figura 4 el caso de uso "Caso de uso 1" depende de los casos de uso "A" y "B".

**Extensión (extend):** Es otra forma de interacción, una extensión representa una variación de un caso de uso a otro, es decir, una dependencia específica entre los casos de uso, a través de la cual un caso de uso puede extender a otro.

**3.2.3 Diagramas de secuencias.** Es aquel que muestra la forma en que los objetos interactúan entre sí al transcurrir el tiempo. Consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

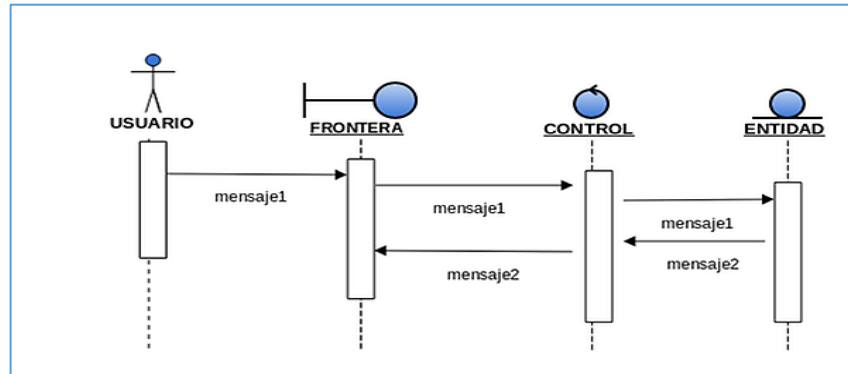
**Objetos:** Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la *línea de vida* de un objeto. Con la línea de vida se encuentra un pequeño rectángulo conocido como *activación*, el cual representa la ejecución de una operación que realiza el objeto.

**Mensaje:** Un mensaje que va de un objeto a otro pasa la línea de vida de un objeto a otro. Un objeto puede enviarse un mensaje a sí mismo. Un mensaje puede ser simple, sincrónico o asincrónico.

**Tiempo:** El diagrama representa al tiempo en dirección vertical. Inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.

**GUI:** (Siglas en Ingles) La interfaz gráfica de usuario; es la interfaz de interacción del usuario y en la que más interactividades se presentan con otros objetos.

Figura 2. Diagrama de Secuencias



### 3.3 ESTÁNDARES DE PROGRAMACIÓN

**3.3.1 Modelo de datos.** Es un lenguaje utilizado para la descripción de una base de datos, por lo general permite describir estructuras de datos de la base de datos (el tipo de datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregar, borrar, modificar).

**3.3.2 Nombres de las tablas.** Los nombres de los campos, así como de las tablas de la base de datos, se escriben en minúsculas, exceptuando los prefijos TP, TR y TB que indican si la tabla es principal, relacional o básica, respectivamente, y exceptuando también la primera letra de cada palabra que conforme su nombre; si es un nombre compuesto por dos o más palabras, los nombres tendrán en mayúscula la primera letra de cada palabra que la forma.

Como se mencionó, se han definido tres categorías para las diferentes tablas que conforman la base de datos. Dada la categoría de la tabla, se agrega un prefijo a su nombre que permita conocer la categoría a la que pertenece. Las categorías son:

- **Tabla básica:** Aquella cuyos registros son necesarios para el correcto funcionamiento de la base de datos. Estas tablas no experimentan muchos cambios en los datos. El prefijo para anteponer a los nombres de estas tablas es “TB\_”, es decir la tabla que almacena los estados en los que se puede encontrar un trabajo de grado es llamada “TB\_Estados”, por ejemplo.
- **Tabla de Relación:** Surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deben ser siempre descriptivos para cada relación. El prefijo para anteponer a los nombres de estas tablas es “TR\_”, por ejemplo, la tabla “ComiteTemporal” que almacena las decisiones tomadas por el comité de trabajos de grado tiene por nombre “TR\_ComiteTemporal”.
- **Tabla Principal:** Aquella cuyo número de registros tiende a crecer en gran cantidad y que además no es posible clasificar como tabla básica o de relación. Un ejemplo de tabla principal es la tabla que almacena los usuarios del portal EISIWeb. El prefijo para anteponer a los nombres de estas tablas es “TP\_”, es decir la tabla “Usuarios”, es conocida como “TP\_Usuarios”.

**3.3.3 Clases.** Los nombres de los campos, así como de las tablas de la base de datos, se escriben en minúsculas, exceptuando los prefijos TP, TR y TB que indican si la tabla es principal, relacional o básica, respectivamente, y exceptuando también la primera letra de cada palabra que conforme su nombre; si es un nombre compuesto por dos o más palabras, los nombres tendrán en mayúscula la primera letra de cada palabra que la forma.

**3.3.4 Páginas JSP.** Los nombres de las páginas JSP que componen los portales web comunidad académica son escritos de manera que la primera letra es una mayúscula seguido de letras minúsculas, en caso de que el nombre del JSP sea compuesto por dos o más palabras, entonces la primera de cada palabra debe ir en mayúscula, por ejemplo, ModificarDecision.jsp, GestionarAdm.jsp.

**3.3.5 Organización de Directorios.** Los directorios del sitio están organizados de tal manera que los archivos que se almacenen en ellos correspondan a lo que describe el nombre del directorio.

## **4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO**

Para el desarrollo de este proyecto se siguió la metodología de prototipo evolutivo. Al iniciar el proyecto se elaboró un primer prototipo durante la fase de requerimientos, el cual fue mejorado con la inclusión de nuevos requerimientos surgidos en la fase de desarrollo, a medida que se generaba un prototipo, el mismo era sometido a pruebas de funcionamiento y se le realizaban los refinamientos pertinentes a partir del resultado de dichas pruebas.

### **4.1 PROTOTIPO ESPERADO**

Al iniciar el proyecto aún no se contaba con una concepción clara de lo sería el producto final, sin embargo, en el transcurso del desarrollo y evolución de los prototipos, las pruebas y análisis del sistema se pudo comprobar que se estaba acercando a los requerimientos iniciales, esto con el fin de enfocar el desarrollo a la solución de las necesidades de los usuarios.

**4.1.1 Requerimientos de objetivo** El objetivo específico inicial y los requisitos que surgieron se dieron gracias a la realización de prototipos y la realimentación con el cliente. Para cada objetivo se listaron los requerimientos detallados de este, los cuales se cumplieron para el prototipo final.

**4.1.1.1 Realizar reingeniería a todos los menús que permiten cambio de decisiones en la sesión del comité.**

#### **Objetivo inicial:**

- Crear un servicio que permita al usuario comité poder modificar las decisiones tomadas durante una sesión e implementar las funcionalidades necesarias para generar el acta correspondiente al final del día.

**Requisitos finales del objetivo:**

- Guardar las decisiones tomadas por el comité de trabajos de grado en una tabla temporal en base de datos.
- Crear las interfaces correspondientes donde se podrá revertir las decisiones tomadas durante una sesión de comité.
- Una decisión solo podrá ser modificada el mismo día en la que sesiona el comité.
- Una decisión podrá ser modificada en más de una ocasión durante el día.
- Al final del día se debe generar el acta correspondiente con las decisiones tomadas y hacer los respectivos cambios en base de datos.

**4.1.1.2 Nuevo servicio de intercambio de mensajes que permita la comunicación entre autores, directores y evaluadores.****Objetivo inicial:**

- Crear un nuevo servicio de intercambio de mensajes que permita la comunicación entre autores, directores y evaluadores en el módulo trabajos de Grado.

**Requisitos finales del objetivo:**

- El autor o los autores de un proyecto podrán comunicarse con su respectivo director de proyecto al inscribir, modificar y solicitar la aprobación de tema.
- El autor o los autores de un proyecto podrán comunicarse con su respectivo director de proyecto al inscribir, modificar y solicitar la aprobación de un plan.
- El autor o los autores de un proyecto podrán comunicarse con el evaluador de su plan de proyecto siempre y cuando el comité tome la decisión de aplazar el plan por concepto del evaluador.
- Al momento de participar en la conversación se debe enviar un correo electrónico informando de la nueva participación a los usuarios interesados.

#### **4.1.1.3 Implementar una interfaz que reúna todos los menús del administrador de Trabajos de Grado.**

##### **Objetivo inicial:**

- Creación e Implementación de una interfaz que reúna todos los menús del administrador de Trabajos de Grado.

##### **Requisitos finales del objetivo:**

- Se debe crear la interfaz para el administrador de trabajos de grado en pregrado.
- Se debe crear la interfaz para el administrador de trabajos de grado en posgrado.
- Se debe conservar la estructura visual usada en las interfaces de trabajo de grado.
- En la interfaz se deben reunir los diferentes menús que permitan al usuario administrador realizar acciones correspondientes a trabajos de grado.

#### **4.1.1.4 Enviar notificaciones vía correo electrónico a los miembros del comité de proyectos al iniciar y reiniciar una sesión del comité.**

##### **Objetivo inicial:**

- Enviar notificaciones vía correo electrónico a los miembros del comité de proyectos al iniciar y reiniciar una sesión del comité.

##### **Requisitos finales del objetivo:**

- Informar a los usuarios pertenecientes al comité la fecha y hora de inicio y reinicio de una sesión de comité.
- Informar a los usuarios pertenecientes al comité el usuario desde el cual se inicia y se reinicia una sesión de comité.
- Informar a los usuarios pertenecientes al comité el lugar de reunión de la sesión de comité.

- Informar a los usuarios pertenecientes al comité los asistentes a la sesión de comité.

#### **4.1.1.5 Realizar la aprobación inmediata de solicitudes de cancelación de proyecto con plan aprobado por parte del comité de proyectos de grado.**

##### **Objetivo inicial:**

Realizar la aprobación inmediata de solicitudes de cancelación de proyecto con plan aprobado por parte del comité de proyectos de grado.

##### **Requisitos finales del objetivo:**

- Se debe permitir la aprobación inmediata de una solicitud de cancelación de proyecto con plan aprobado por parte del comité de trabajos de grado.

#### **4.1.1.6 Permitir a uno de los autores de un proyecto de grado hacer solicitud de cancelación de proyecto de grado cuando otro autor no está matriculado, además que el director del proyecto pueda dar aval a la solicitud y así se pueda tomar decisiones en el Comité de Proyectos de Grado.**

##### **Objetivo inicial:**

Permitir a uno de los autores de un proyecto de grado hacer solicitud de cancelación de proyecto de grado cuando otro autor no está matriculado.

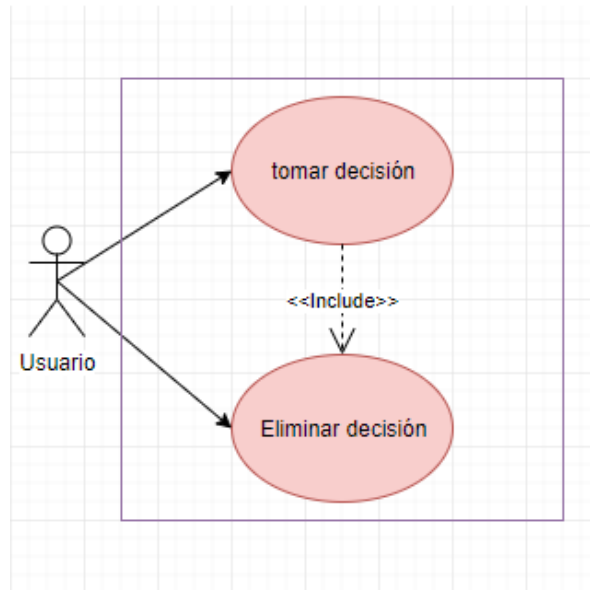
##### **Requisitos finales del objetivo:**

- Se debe permitir que el director del proyecto de grado pueda dar aval a la solicitud.
- Se debe permitir la aprobación inmediata por parte del comité de trabajos de grado.

## 4.2 DIAGRAMAS DE CASOS DE USO

### 4.2.1 Cambio de decisión por parte del comité de trabajos de grado

Figura 3. Diagrama de Casos de Uso: Cambiar decisión



### 4.2.2 Servicio intercambio de mensajes

Figura 7. Diagrama de Casos de Uso: Intercambio de mensajes autor y director.

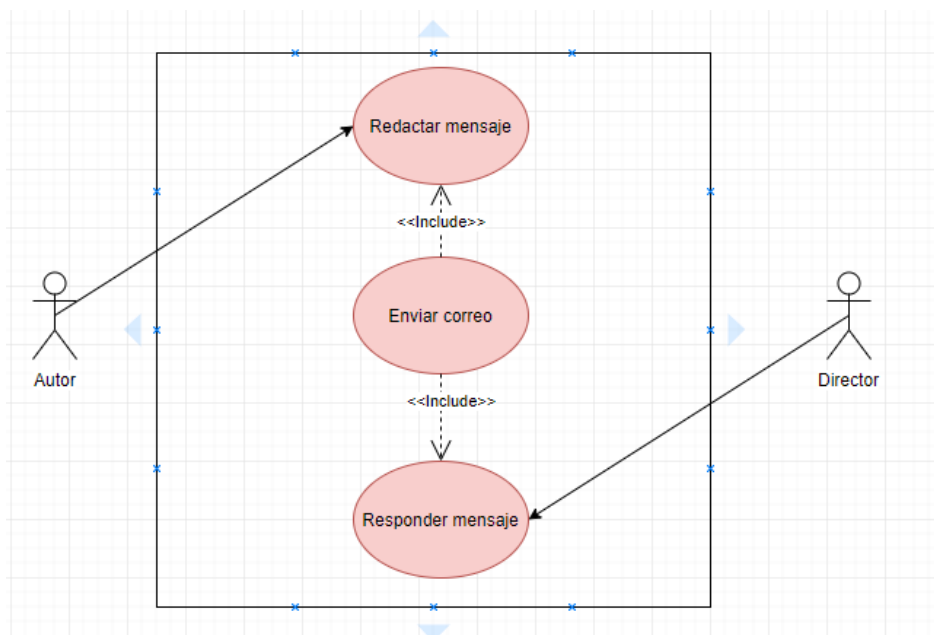
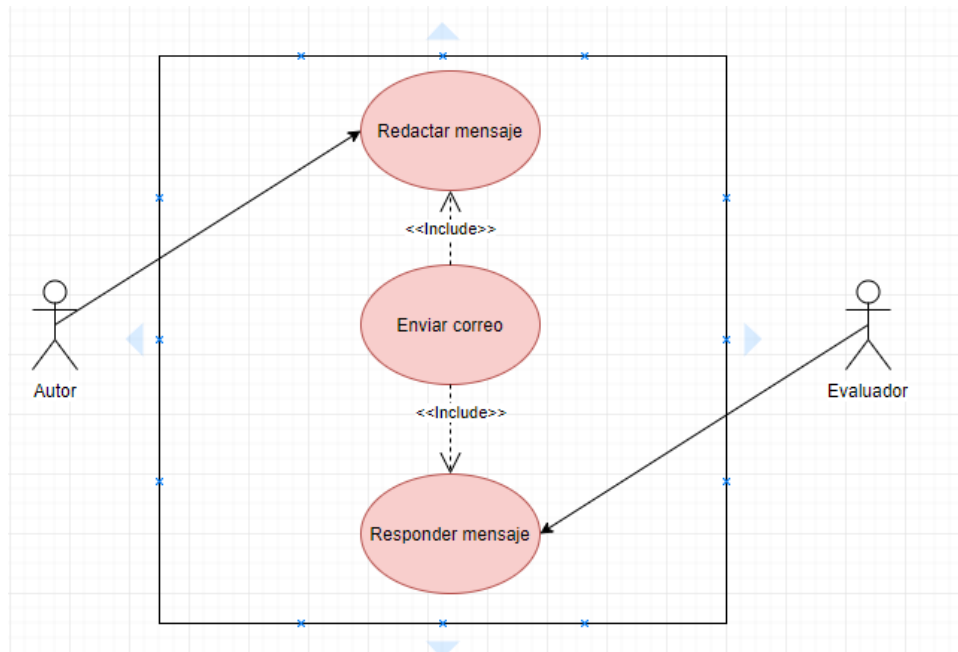
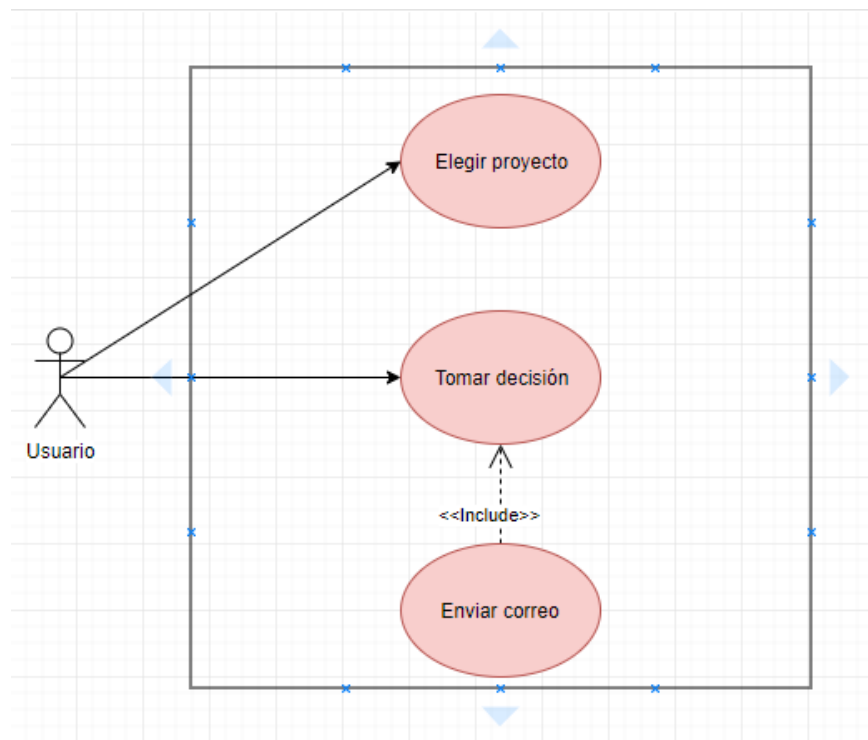


Figura 8. Diagrama de Casos de Uso: Intercambio de mensajes autor y evaluador.



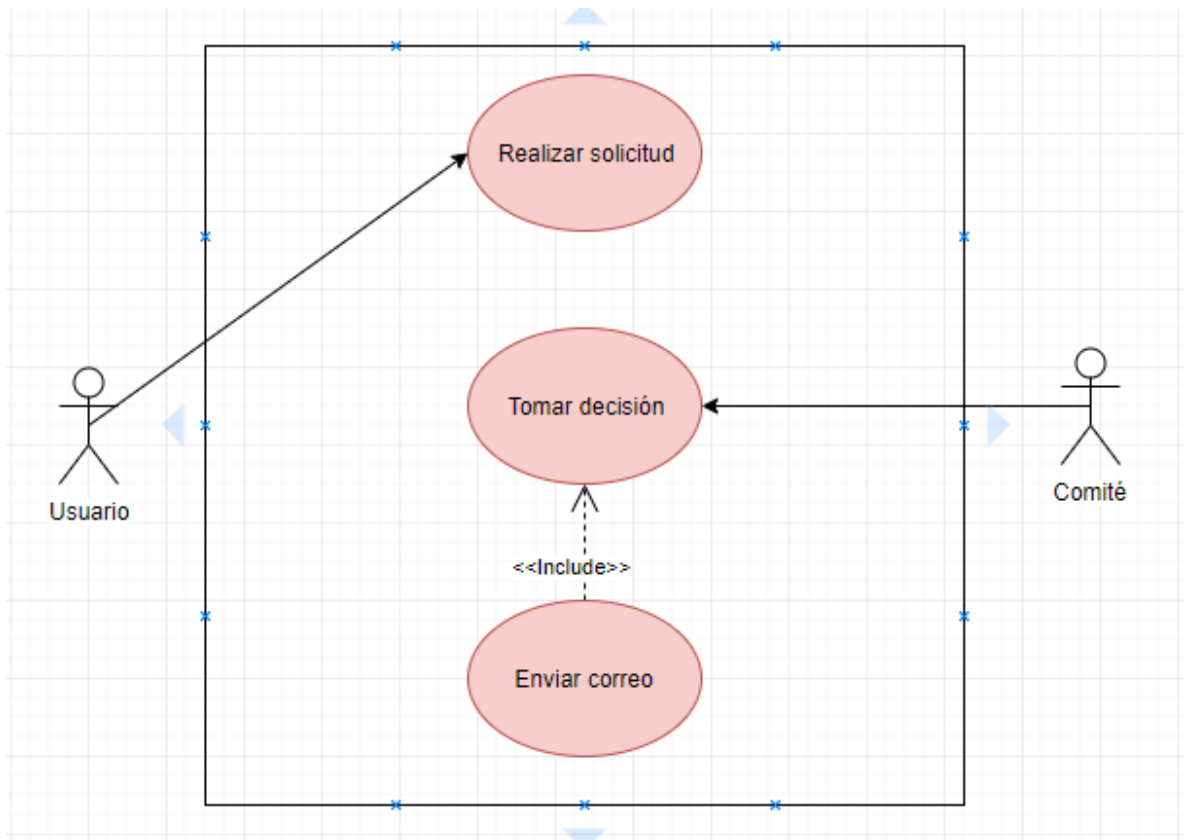
### 4.2.3 Cancelación de proyecto con plan aprobado

Figura 9. Diagrama de Casos de Uso: Cancelación de proyecto con plan aprobado



#### 4.2.4 Hacer solicitud de cancelación

Figura 10. Diagrama de Casos de Uso: Solicitud de cancelación



### 4.3 DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA

#### 4.3.1 Cambio de decisión por parte del comité de trabajos de grado

Tabla 1. Casos de uso: Cambiar decisión

TÍTULO	FUNCIONES PRIMARIAS
Tomar decisión	El comité toma una decisión sobre un tema, plan, solicitud, asignación de evaluadores o calificadores.
Eliminar decisión	Elimina una decisión tomada anteriormente por el comité.

### 4.3.2 Servicio intercambio de mensajes

Tabla 2. Casos de uso: Intercambio de mensajes.

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Redactar mensaje	El usuario crea una participación en la conversación.
Enviar correo	Enviar un correo a los participantes de la conversación.
Responder mensaje	El usuario crea una participación en respuesta a mensaje anterior.

### 4.3.3 Cancelación de proyecto con plan aprobado

Tabla 3. Casos de uso: Cancelación de proyecto con plan aprobado

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Elegir proyecto	Selecciona el proyecto que se va a cancelar.
Tomar decisión	El comité toma una decisión sobre la cancelación del proyecto.
Enviar correo	Enviar un correo a los usuarios sobre la decisión tomada.

#### 4.3.4 Hacer solicitud de cancelación

Tabla 4. Casos de uso: Solicitud de cancelación

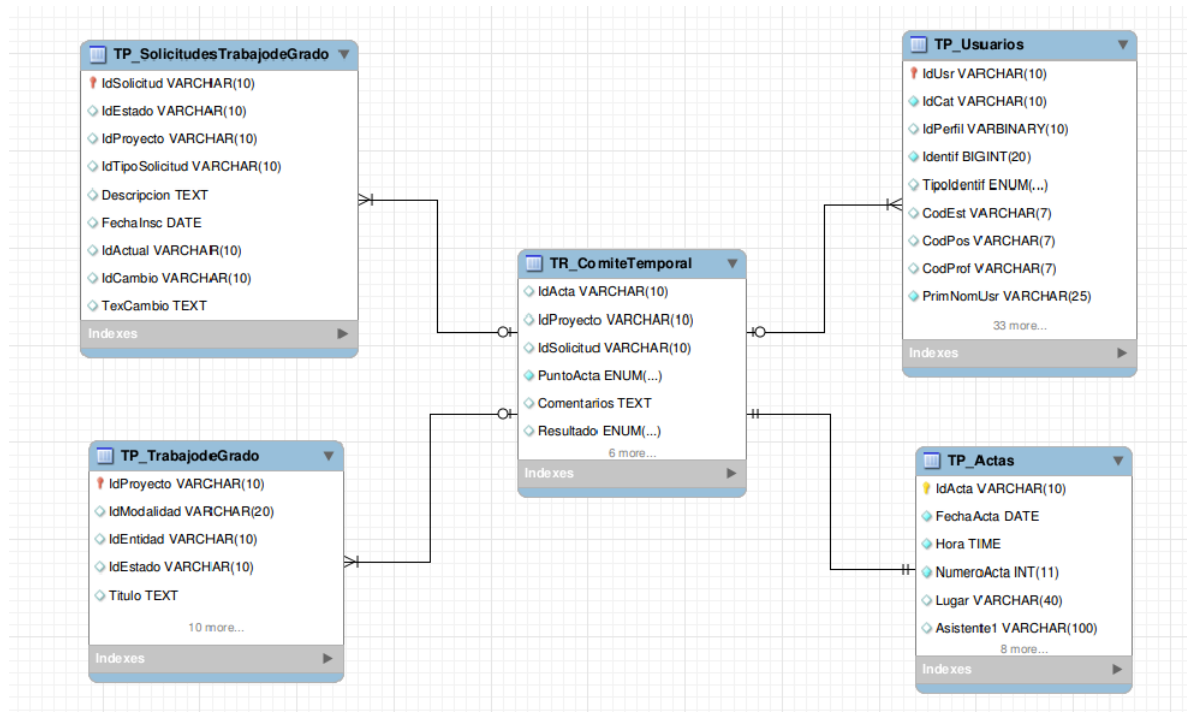
<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Realizar solicitud	El usuario realiza una solicitud de cancelación del proyecto
Tomar decisión	El comité toma una decisión sobre la solicitud de cancelación del proyecto.
Enviar correo	Enviar un correo a los usuarios sobre la decisión tomada.

## 4.4 DISEÑO Y ANÁLISIS

### 4.4.1 Diagrama Entidad/Relación de los servicios desarrollados

#### 4.4.1.1 Cambio de decisión por parte del comité de trabajos de grado

Figura11. Diagrama E/R: Cambio Decisión



#### 4.5 DESCRIPCIÓN DE LAS ENTIDADES.

Descripción de las entidades creadas y utilizadas en las bases de datos Diamante ubicada en el servidor de cada portal y Poseidón ubicada en el servidor del portal web de la Vicerrectoría Académica.

*Tabla 5. Descripción de las Entidades*

<b>ENTIDAD</b>	<b>DESCRIPCION</b>
<b>TP_Usuarios</b>	Contiene la información de todos los usuarios registrados en el portal.
<b>TP_TrabajodeGrado</b>	Contiene toda la información de los trabajos de grado.
<b>TP_SolicitudesTrabajodeGrado</b>	Contiene la información de las solicitudes de trabajos de grado hechas hacia el comité de trabajos de grado.
<b>TP_Actas</b>	Contiene la información de las actas que se generan después de finalizar una sesión de comité.
<b>TR_ComiteTemporal</b>	Contiene la información sobre las decisiones tomadas en una sesión de comité de trabajos de grado.

#### 4.6 MODELO DE PROCESOS DEL SISTEMA.

Para una mejor interpretación de los modelos de los procesos del sistema se realizaron los diagramas de secuencia necesarios para cada caso de uso en los que se explica con detalle los pasos para el funcionamiento de cada uno de los servicios (Ver Anexo A).

#### 4.7 IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES.

Para la implementación de los servicios se utilizaron las siguientes herramientas:

- Lenguaje de marcado para la elaboración de páginas web, HTML.
- Lenguaje de programación orientado a la web, JSP.
- Lenguaje de programación interpretado orientado a objetos, JavaScript.
- Hojas de estilo en cascada, CSS.
- Lenguaje Java.
- NetBeans, IDE para desarrollar las clases de Java y los archivos JSP.
- Servidor Jakarta Tomcat.
- Manejador de Base de datos, MySQL 5.0.

Haciendo uso de las anteriores herramientas, y con la asesoría y seguimiento del director de proyecto se diseñaron los nuevos servicios y se estructuraron los datos para el primer prototipo. También se tomaron en cuenta sugerencias recibidas por miembros del Grupo Calumet, para realizar un posterior refinamiento a las interfaces.

Para el desarrollo se trabajó con la base de datos "Diamante" ubicada en todos los servidores de los portales de las escuelas, y con la base de datos centralizada "Poseidón" ubicada en el servidor del portal web de la Vicerrectoría Académica. En las cuales se crearon las tablas necesarias para el funcionamiento de los servicios. Se llevaron a cabo pruebas para cada subsistema propuesto, verificando que el resultado correspondiera con lo esperado, de esta forma, se evidenció el correcto funcionamiento en la captura de datos, selección de ítems y almacenamiento de información.

La implantación de los servicios se realizó en primer lugar en el portal web de pruebas del Grupo Calumet, donde se realizan las pruebas a todos los componentes. Finalmente se implementan en todos los portales web a los cuales el Grupo Calumet presta soporte.

## **4.8 MANTENIMIENTO Y ADMINISTRACIÓN**

**4.8.1 Actividades de Mantenimiento.** Dentro de las funciones que se realizan en la administración y mantenimiento de los servicios de los portales web se encuentra la tarea de corregir errores, las cuales se denominan incidencias, que se corrigen durante la primera fase como integrantes del Grupo Calumet.

**4.8.2 Actividades de Soporte a Usuarios.** Las escuelas cuentan en sus portales con un servicio de consultas y sugerencias, a través del cual se responden preguntas y se resuelven problemas de los usuarios. Diariamente se reciben consultas de estudiantes y profesores, que necesitan asesoría en el funcionamiento de servicios o soporte en el manejo de su usuario y contraseña. También se resuelven problemas de manera presencial, donde el usuario acude directamente a las oficinas del Grupo Calumet y es atendido directamente por alguno de sus miembros, que le prestan la asesoría necesaria. Algunas de las labores realizadas comúnmente son:

- Restablecimiento de contraseña a usuarios.
- Dar respuesta y solución a las inquietudes y problemas que tengan los usuarios de los portales respecto al uso de los servicios.
- Facilitar orientación a los estudiantes nuevos de las escuelas a las cuales presta sus servicios el grupo Calumet, en cuanto al registro en el portal y el uso de sus principales servicios.

**4.8.3 Actividades de Administración.** Dentro de las actividades realizadas por los miembros del Grupo Calumet, se encuentra la tarea de administrar los servidores de los portales de las escuelas, cada semestre se asignan nuevos administradores, los cuales cuentan con un usuario dentro de uno de los portales a los cuales presta soporte el grupo. Como administrador las tareas que se realizan comúnmente son:

- Realizar periódicamente copias de las bases de datos Diamante y División.
- Dar aval a las solicitudes de publicación de los usuarios en la cartelera para que puedan ser accedidas por la comunidad.
- Atender las sugerencias hechas por los usuarios del sistema a través del servicio de consultas y sugerencias.
- Actualizar periódicamente las bases de datos con respecto a la información que ofrece la División de Servicios de Información de la Universidad Industrial de Santander, para que el portal cuente con información actualizada.

## **5 PRUEBAS DEL SISTEMA**

Para garantizar el correcto desarrollo de los nuevos servicios creados y a los que se le hizo reingeniería, se realizaron las siguientes pruebas:

### **5.1 PRUEBAS DE VERIFICACIÓN**

Esta prueba es una de las más utilizadas en desarrollo de software mediante esta se aplican diferentes técnicas para detectar errores en el sistema antes de ser utilizado.

Se efectúa ejecutando paso a paso el proceso del servicio de manera que se explora cada funcionalidad que tiene el módulo desarrollado, realizando verificaciones de validación, los campos que son obligatorios no pueden quedar vacíos, por ejemplo.

Las siguientes tablas describen las pruebas de cada caso de uso de los servicios que fueron desarrollados.

### 5.1.1 Pruebas por componente.

Esta prueba se realizó para cada caso de uso de cada servicio desarrollado y descrito anteriormente:

#### 5.1.1.1 Cambio de decisión por parte del comité de trabajos de grado

*Tabla 6. Pruebas realizadas: Toma y cambio de decisiones*

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Tomar Decisión	Se tomó una decisión de cada caso donde el comité puede(Aprobar, Rechazar, Aplazar, Poner en Estudio, Aplazar por Evaluador, Cancelar)	✓
Eliminar Decisión	Se eliminó una decisión tomada por parte del comité de proyectos de grado, se verifico que la decisión fuera revertida y se pudieran tomar acciones nuevamente.	✓

#### 5.1.1.2 Servicio intercambio de mensajes

*Tabla 7. Pruebas realizadas: Intercambio de mensajes*

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Redactar Mensaje	Se participó en una conversación y se verifico que llegara el mensaje al destinatario correctamente.	✓
Enviar Correo	Se verifico que al participar en la conversación se enviaran los respectivos correos electrónicos a los participantes.	✓
Responder Mensaje	Se verifico que al momento de responder el mensaje se desplegara correctamente la conversación.	✓

### 5.1.1.3 Cancelación de proyecto con plan aprobado

Tabla 8. Pruebas realizadas: Cancelación de proyecto con plan aprobado

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Elegir Proyecto	Se verifico que se desplegaran los proyectos correctamente y al momento de elegirlos la información debía ser la correspondiente.	✓
Tomar Decisión	Se verifico que al momento de tomar una decisión (Aprobado, Rechazado, Aplazado, Cancelado, Aplazado por Evaluador, Espera) esta se hiciera correctamente y el estado del proyecto sea el adecuado.	✓
Enviar Correo	Se verifico el envío de correo electrónico después de tomada la decisión por parte del comité.	✓

### 5.1.1.4 Hacer solicitud de cancelación

Tabla 9. Pruebas realizadas: Solicitud de cancelación

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Realizar Solicitud	Se realizó una solicitud de cancelación de proyecto confirmando que uno de los autores no estuviera matriculado.	✓
Tomar Decisión	Se verifico que al momento de tomar una decisión sobre la solicitud (Aprobado, Rechazado, Aplazado, Cancelado, Estudio, Espera) esta se hiciera correctamente.	✓
Enviar Correo	Se verifico el envío de correo electrónico después de tomada la decisión por parte del comité.	✓

## 6. CONCLUSIONES

- El servicio de Trabajos de Grado es beneficioso para la comunidad académica ya que facilita el trabajo para los miembros del comité a través del nuevo servicio que les permite revertir una decisión tomada durante una sesión de comité.
- Permitir que los autores de un proyecto puedan interactuar con su director de proyecto y evaluador de plan a través del servicio de intercambio de mensajes es de gran ayuda ya que además de brindar un espacio de comunicación más ágil permite llevar un registro de los avances, modificaciones y observaciones del proyecto.
- Reunir todos los menús del administrador de trabajos de grado en una sola interfaz reduce el trabajo y promueve el buen uso de estos servicios.
- La metodología de trabajo del grupo Calumet permite que los estudiantes brinden apoyo de manera continua al proyecto COMA a través de las capacitaciones a estudiantes nuevos y el manejo de estándares se hace continuo uso del trabajo grupal ocasionando así una continua mejora y soporte sobre el proyecto sin perder la esencia de este.

## **7. RECOMENDACIONES**

Para mantener una buena comunicación con los usuarios de los portales, es fundamental dar pronta y precisa respuesta a las sugerencias de los mismos dando un soporte oportuno a sus inquietudes.

Socializar con los administrativos acerca de la nueva interfaz de administración de trabajos de grado y brindar capacitación sobre el manejo de los diferentes menús para que así puedan hacer uso de manera óptima de este servicio.

Capacitar a los miembros del comité de trabajos de grado sobre el nuevo servicio de cambio de decisión implementado para que puedan hacer buen uso de este servicio.

Socializar con los autores, directores y evaluadores de un proyecto de grado sobre el nuevo servicio de intercambio de mensajes para que puedan hacer buen uso de este y puedan comunicarse entre ellos de una forma más rápida.

## BIBLIOGRAFÍA

JAVASCRIPT-YA. Tutoriales del lenguaje de programación JavaScript. [En línea] Disponible en: <http://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=0>>.

JOHNSON, James. Bases de datos: modelos lenguajes, diseño. 1ª ed. México: Oxford University Press, 2000. 1015. ISBN: 970-613-461-1.

MANUALES. Manuales básicos de JSP. [En línea] Disponible en: <http://www.desarrolloweb.com/manuales/73/> >.

PIATTINI VELTHUIS, Mario, *et al.* Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Madrid: RA-MA S.A. Editorial y Publicaciones, 2007, 480. ISBN-10: 8478977767

PROGRAMACIÓN. Curso para aprender java. [En línea] Disponible en: [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=68&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188) >.

PROGRAMACIÓN FÁCIL. Manual de programación para Java Jsp (Java Server Pages). [En línea] Disponible en: [http://www.programacionfacil.com/programacion:manual\\_java\\_jsp](http://www.programacionfacil.com/programacion:manual_java_jsp)>.

SCHMULLER, Joseph. Aprendiendo UML en 24 horas. México: Alhambra Mexicana S.A, 2000, 387.

SQL. Documentación con ejemplos de SQL. [En línea] Disponible en: <http://www.1keydata.com/es/sql/> >

STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005, 208. ISBN-13: 978-0-13-230998-1

STUMPF, Robert, TEAGUE, Lavette. Object-Oriented Systems Analysis and Design with UML. New Jersey: Pearson Prentice Hall, 2004, 428. . ISBN 0131434063

WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. México: Thomson International, 2003, 678.

WIKISPACES. Información acerca de la Arquitectura Cliente Servidor. [En línea] Disponible en: <http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor>>.

W3SCHOOLS. Tutoriales de cómo programar en HTML. [En línea] Disponible en: <http://www.w3schools.com/html/>>.

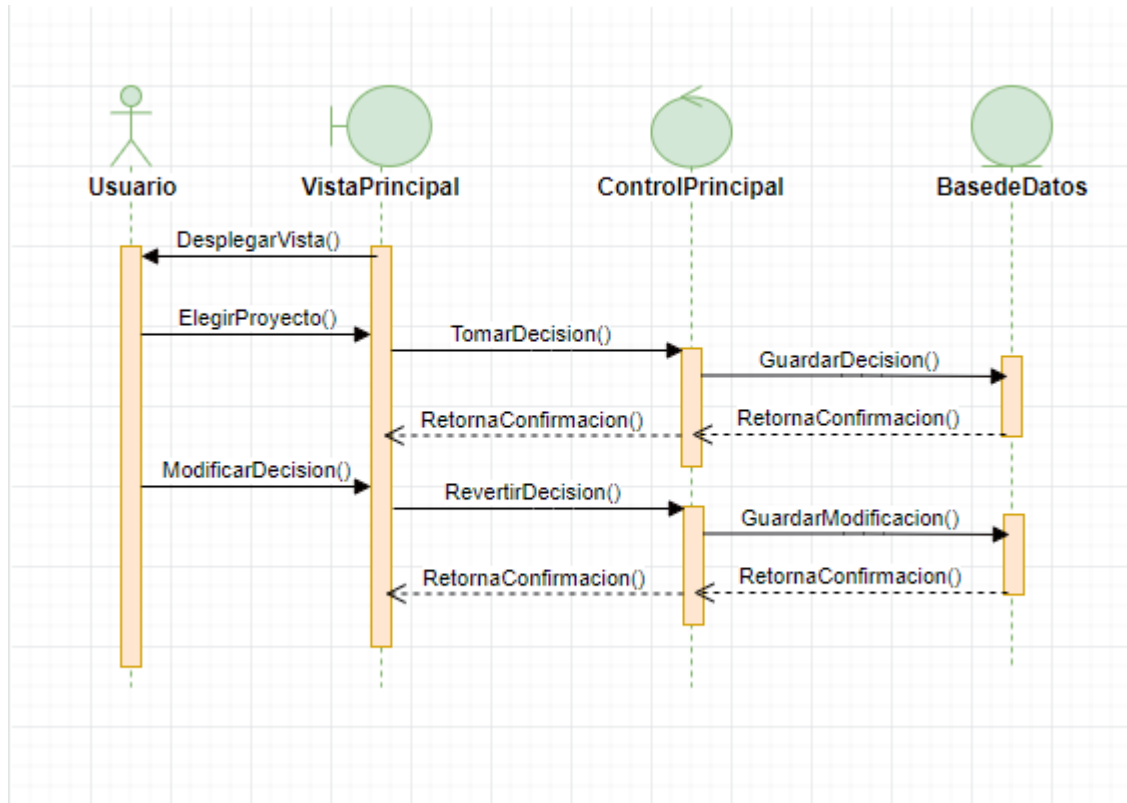
YOUBLISHER. Tutoriales básicos sobre UML. [En línea] Disponible en: <http://www.tutorialspoint.com/uml/> >

## ANEXOS

### Anexo A. Modelo de Procesos del Sistema

#### Cambio de decisión por parte del comité de trabajos de grado

Figura 12. Diagrama de secuencia: Cambiar Decisión

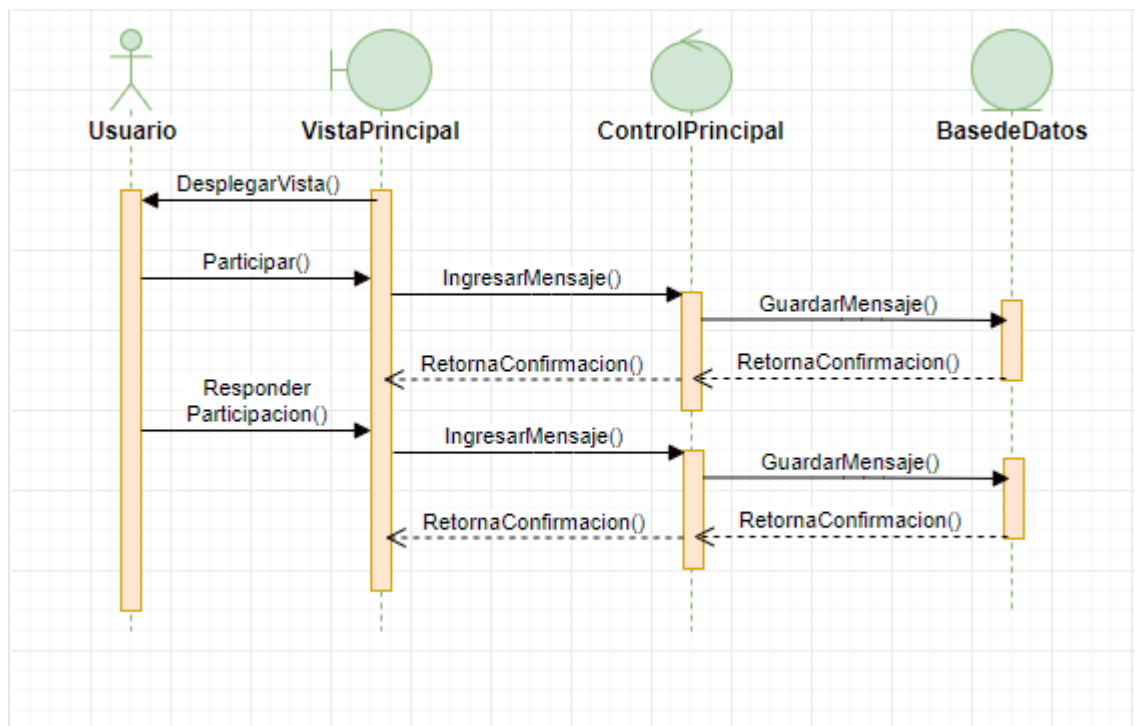


1. La vista Principal despliega al usuario los proyectos de grado que tienen acciones por realizar.
2. El usuario toma una decisión de un proyecto de grado (aprobado, rechazado, estudio, aplazado, espera, cancelado).
3. El controlador principal envía los datos de la decisión tomada a la base de datos.
4. El controlador principal hace una petición a base de datos solicitando las decisiones tomadas.

5. La vista modificar muestra al usuario la decisión tomada.
6. El usuario procede a revertir la decisión.
7. El controlador recibe los datos de la decisión y los envía a la base de datos.
8. El usuario puede repetir nuevamente el paso 1.

### Servicio intercambio de mensajes

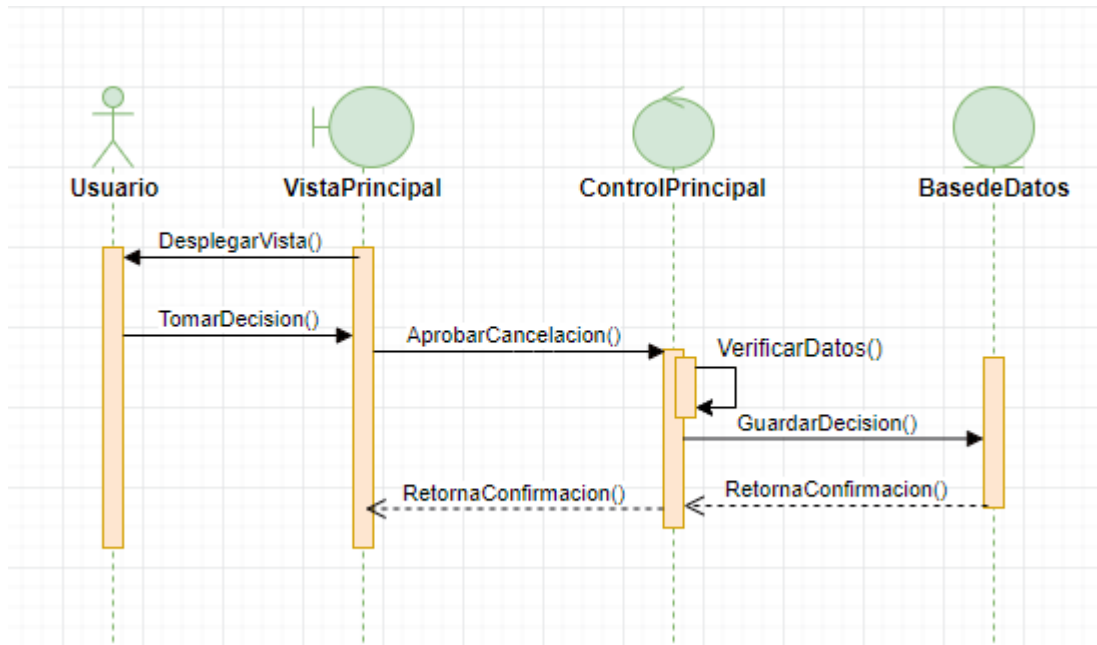
Figura 13. Diagrama de secuencia: Intercambio de mensajes autor y director



1. La vista principal despliega la información del proyecto y la sección de intercambio de mensajes.
2. El usuario ingresa el mensaje que desea enviar y da click al botón enviar comentario.
3. El controlador valida los datos seleccionados y los envía a la base de datos.
4. La base de datos guarda la información y da una respuesta al Controlador y la Vista Principal muestra un mensaje.
5. Se repiten los pasos anteriores al responder el mensaje.

## Cancelación de proyecto con plan aprobado

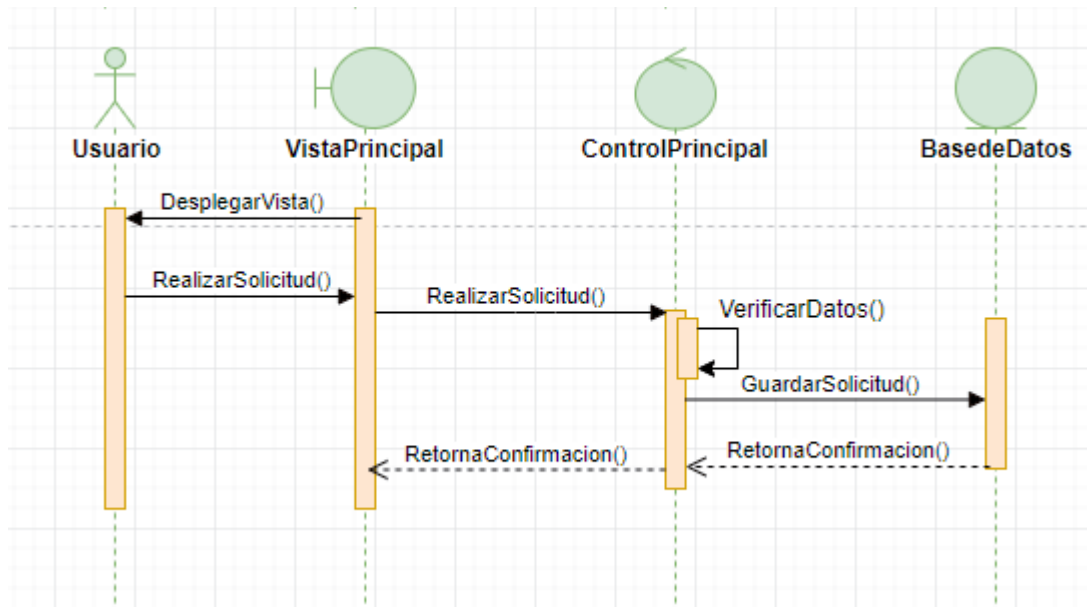
Figura 14. Cancelación de proyecto con plan aprobado



1. La vista principal despliega la información del proyecto y las acciones a tomar por parte del usuario (aprobar, rechazar, aplazar, estudio, cancelar).
2. El Controlador valida que los datos sean correctos y envía los datos a la base de datos.
3. La base de datos guarda la información y da una respuesta de confirmación al controlador y la vista principal muestra un mensaje.

## Hacer solicitud de cancelación

Figura 15. Diagrama de secuencia: Solicitud de cancelación



1. La vista principal despliega los diferentes tipos solicitudes.
2. El usuario selecciona la solicitud cancelación de proyecto, llena los datos respectivos y da click al botón enviar.
3. El Controlador valida que los datos sean correctos y envía los datos a la base de datos.
4. La base de datos guarda la información y da una respuesta de confirmación al controlador y la vista principal muestra un mensaje.