

**ADMINISTRACIÓN, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DEL SISTEMA DE GESTIÓN ACADÉMICA DEL PORTAL DE LA ESCUELA DE
INGENIERÍA DE SISTEMAS**

JUAN SEBASTIAN AGUDELO HERNÁNDEZ

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

2013

**ADMINISTRACIÓN, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DEL SISTEMA DE GESTIÓN ACADÉMICA DEL PORTAL DE LA ESCUELA DE
INGENIERÍA DE SISTEMAS**

JUAN SEBASTIAN AGUDELO HERNANDEZ

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**

2013

Agradecimientos

En primer lugar a Dios, quien me da fortaleza, me ilumina con su sabiduría y me ha dado el privilegio de alcanzar este logro. A mis padres por estar siempre a mi lado y apoyar mis decisiones, por ser un ejemplo a seguir, gracias por guiarme.

Gracias a mis dos hermanas por su ayuda y consejos. Gracias a mi novia por estar siempre a mi lado y animarme en todo momento. Muchas gracias al profesor por permitirme pertenecer al grupo y guiarme en mi proceso de aprendizaje. Agradezco al grupo Calumet por acogerme y enseñarme tantas cosas, en especial a Duván, Carlos Ruiz y Yadiana 'Dayana'. Agradezco también a todos mis amigos por su apoyo y momentos de alegría. Agradezco de manera especial a Alejandro Mantilla 'Gato' por compartir sus conocimientos conmigo y ayudarme en momentos importantes de mi carrera. Gracias a todos por ayudarme a conseguir este triunfo.

Juan Sebastián Agudelo Hernández

CONTENIDO

INTRODUCCIÓN	15
1 PRESENTACIÓN DEL PROYECTO	17
1.1 ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME.....	17
1.2 ANTECEDENTES.....	18
1.3 DEFINICIÓN DEL PROBLEMA.....	18
1.4 OBJETIVOS.....	19
1.1.1. Objetivo General	19
1.1.2. Objetivos Específicos	20
1.5 JUSTIFICACIÓN	23
1.6 ALCANCES Y LIMITACIONES.....	24
2 MARCO TEÓRICO	25
2.1 ARQUITECTURA CLIENTE/SERVIDOR.....	25
2.1.1. Arquitectura Cliente/Servidor aplicada	26
2.1.2. Ventajas del esquema Cliente/Servidor.....	26
2.1.3. Desventajas del esquema Cliente/Servidor.....	27
2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS.....	27
2.2.1 Tecnología aplicada	28
2.3 BASES DE DATOS	30
2.3.1 MySQL	30
2.4 NETBEANS.....	31
2.5 SISTEMA DE CONTROL DE VERSIONES.....	32
2.5.1 Subversión	33
2.6 PROGRAMACIÓN UTILIZADA	33
2.6.1 Clases	34
2.6.2 Objetos.....	34

2.6.3	Atributos	34
2.6.4	Métodos.....	34
2.6.5	Herencia.....	34
2.6.6	Beneficios de la Programación Orientada a Objetos.....	35
2.6.7	Java y JDK (Java Development Kit)	35
2.7	SERVIDORES WEB	36
2.7.1	Servidor Jakarta Tomcat	36
3	MARCO METODOLÓGICO	37
3.1	PROTOTIPO EVOLUTIVO	37
3.2	LENGUAJE DE MODELADO UNIFICADO	39
3.2.1	Diagramas de UML	39
3.3	ESTÁNDARES DE PROGRAMACIÓN.....	41
3.3.1	Modelo de datos.....	41
3.3.2	Nombres de las tablas.....	41
3.3.3	Clases	42
3.3.4	Páginas JSP	42
3.3.5	Organización de Directorios	43
4	DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO.....	44
4.1	PROTOTIPO ESPERADO	44
4.1.1	Diagramas de Casos de Uso.....	53
4.1.2	Documentación de Casos de Uso del Sistema	60
4.1.3	Diseño y Análisis	67
4.1.4	Modelo de Procesos del Sistema	72
4.1.5	Implementación, Implantación y Pruebas Generales	86
4.2	MANTENIMIENTO Y ADMINISTRACIÓN	87
4.2.1	Actividades de Mantenimiento.....	87
4.2.2	Actividades de Soporte a Usuarios	92

4.2.3	Actividades de Administración.....	93
5	PRUEBAS DEL SISTEMA.....	94
5.1	PRUEBAS DE VERIFICACIÓN.....	94
5.1.1	Pruebas por componente.....	94
5.2	PRUEBAS DE INTEGRACIÓN.....	95
5.3	PRUEBAS DE VALIDACIÓN.....	95
6	CONCLUSIONES.....	96
7	RECOMENDACIONES.....	97
	BIBLIOGRAFÍA.....	98
	ANEXOS.....	100

LISTA DE TABLAS

Tabla 1. Caso de Uso: Ingreso al portal.....	60
Tabla 2. Caso de Uso: Personalizar portal.	60
Tabla 3. Casos de uso: Inicio del servicio	61
Tabla 4. Casos de uso: Ingreso solicitud usuario EISI.....	61
Tabla 5. Casos de uso: Ingreso solicitud administrador.....	62
Tabla 6. Casos de uso: Listado solicitudes administrador.	62
Tabla 7. Casos de uso: Orden del día - Administrador	63
Tabla 8. Casos de uso: Inicio sesión de Consejo	63
Tabla 9. Casos de uso: Desarrollo sesión de consejo de Escuela.....	64
Tabla 10. Casos de uso: Lectura Acta anterior	64
Tabla 11. Casos de uso: Desarrollo puntos del consejo.	65
Tabla 12. Casos de uso: Casos estudiantes.....	65
Tabla 13. Casos de uso: Mantenimiento y Administración del portal EISIWEB.....	66
Tabla 14. Descripción de las Entidades	71
Tabla 15. Pruebas Realizadas: Gestión Proyectos.....	114
Tabla 16. Pruebas realizadas: Ingresar Portal	114
Tabla 17. Pruebas realizadas: Personalizar Portal	115
Tabla 18. Pruebas realizadas: Inicio del servicio.	115
Tabla 19. Pruebas realizadas: Ingreso solicitud usuario EISI.	116
Tabla 20. Pruebas realizadas: Ingreso solicitud administrador.....	117
Tabla 21. Pruebas realizadas: Listado solicitudes administrador.	118
Tabla 22. Pruebas realizadas: Orden del día - Administrador.	118
Tabla 23. Pruebas realizadas: Inicio sesión de Consejo.....	119
Tabla 24. Pruebas realizadas: Desarrollo sesión de consejo de Escuela.....	119
Tabla 25. Pruebas realizadas: Lectura Acta anterior.	120
Tabla 26. Pruebas realizadas: Desarrollo puntos del consejo.	120
Tabla 27. Pruebas realizadas: Casos estudiantes.....	121

LISTA DE FIGURAS

Figura 1. Modelo Cliente/Servidor.....	25
Figura 2. Modelo de acceso a JSP	29
Figura 3. Prototipo Evolutivo.....	37
Figura 4. Diagrama de Casos de Uso: Ingreso al portal.....	53
Figura 5. Diagrama de Casos de Uso: Personalizar portal.....	54
Figura 6. Diagrama de Casos de Uso: Inicio del servicio.....	54
Figura 7. Diagrama de Casos de Uso: Ingreso solicitudes usuario EISI.....	55
Figura 8. Diagrama de Casos de Uso: Ingreso solicitud administrador.....	55
Figura 9. Diagrama de Casos de Uso: Listado solicitudes administrador.....	56
Figura 10. Diagrama de Casos de Uso: Orden del día - Administrador.....	56
Figura 11. Diagrama de Casos de Uso: Inicio sesión de Consejo.....	57
Figura 12. Diagrama de Casos de Uso: Desarrollo sesión consejo.....	57
Figura 13. Diagrama de Casos de Uso: Lectura Acta anterior.....	57
Figura 14. Diagrama de Casos de Uso: Desarrollo putos consejo.....	58
Figura 15. Diagrama de Casos de Uso: Casos estudiantes.....	58
Figura 16. Diagrama de Casos de Uso: Fin de sesión.....	59
Figura 17. Diagrama de Casos de Uso: Subsistema de Mantenimiento y Administración.....	59
Figura 18. Diagrama E/R: Portal Profesor	68
Figura 19. Diagrama E/R: Actas Consejo de Escuela.....	69
Figura 20. Continuación ddiagrama E/R: Actas Consejo de Escuela	70
Figura 21. Diagrama de secuencia: Ingresar al portal	73
Figura 22. Diagrama de secuencia: Personalizar portal	74
Figura 23. Diagrama de secuencia: Iniciar Servicio.....	75
Figura 24. Diagrama de secuencia: Ingreso solicitudes usuario EISI – Guardar solicitud.....	76
Figura 25. Diagrama de secuencia: Ingreso solicitudes – Editar solicitud.....	77

Figura 26. Diagrama de secuencia: Ingreso solicitudes Administrador – Guardar solicitud.....	78
Figura 27. Diagrama de secuencia: Revisar solicitud.	79
Figura 28. Diagrama de secuencia: Orden del día - Ordenar.	80
Figura 29. Diagrama de secuencia: Orden del día – Enviar Orden.....	81
Figura 30. Diagrama de secuencia: Iniciar sesión Consejo.	82
Figura 31. Diagrama de secuencia: Desarrollo sesión – Revisar Quórum.....	83
Figura 32. Diagrama de secuencia: Desarrollo sesión – Gestionar puntos.	84
Figura 33. Diagrama de secuencia: Desarrollo sesión – Finalizar sesión.....	85
Figura 34. Arquitectura Cliente/Servidor de Dos Capas	102
Figura 35. Arquitectura Cliente/Servidor de Tres Capas.....	103
Figura 36. Función del DBMS.....	110
Figura 37. Diagramas de Casos de Uso	111
Figura 38. Diagrama de Secuencias.....	113

LISTA DE ANEXOS

Anexo A. ARQUITECTURA CLIENTE/SERVIDOR	100
Anexo B. TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS	104
Anexo C. BASES DE DATOS.....	106
Anexo D. LEGUAJE DE MODELADO UNIFICADO (UML).....	111
Anexo E. PRUEBAS DEL SISTEMA.....	114

RESUMEN

TÍTULO: ADMINISTRACIÓN, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA DEL PORTAL DE LA ESCUELA DE INGENIERÍA DE SISTEMAS.

AUTOR: Juan Sebastián Agudelo Hernández **

PALABRAS CLAVE: Portal Web, Consejo de Escuela, Portal Profesor, Gestor de Proyectos, EISIWEB (Escuela de Ingeniería de Sistemas).

DESCRIPCIÓN

En la actualidad, la escuela de Ingeniería de Sistemas cuenta 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, así como del control de usuarios y servicios que se les proporcionan. Estos servicios ofrecen soluciones a los problemas y necesidades de los usuarios, por esta razón las labores de mantenimiento y actualización son indispensables.

Motivados por brindarle a profesores, grupos de investigación, y comunidad en general un mejor servicio y un sistema robusto capaz de suplir todas las necesidades, se hace necesaria la implantación del servicio para la gestión, control y ejecución de proyectos de diferente índole.

El portal del profesor, sitio propio de cada docente, contaba con información estática y ningún servicio. Debido al uso continuo de este portal, se amplía su funcionalidad y agregan los servicios primordiales para los docentes, además de esto, se permite la navegación entre este portal y el de la escuela.

Es función de la Escuela de Ingeniería de Sistemas, realizar consejos de escuela donde se tratan temas de profesores, estudiantes y miembros del consejo, los cuales se desarrollan según el reglamento interno de la UIS, para dar respuestas a estos asuntos. Como es debido se elabora un acta para cada consejo, donde se recopilan y se deja constancia de los puntos tratados en el desarrollo de cada sesión. Para mejorar este proceso se desarrolló un servicio que permite llevar de manera controlada las solicitudes y trámites que se generan en una sesión de consejo.

* Trabajo de grado Modalidad Trabajo de Investigación.

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: Luis Ignacio González Ramírez.

ABSTRACT

TITLE: ADMINISTRATION, ANALYSIS, DESIGN, DEVELOPMENT AND IMPLEMENTATION OF THE ACADEMIC MANAGEMENT SYSTEM FOR THE SYSTEMS ENGINEERING SCHOOL PORTAL .

AUTHORS: Juan Sebastián Agudelo Hernández

KEYWORDS: Web Portal, School Council, Professor Portal, Project manager, EISIWEB (Systems Engineering School).^{**}

DESCRIPTION

Currently, the systems engineering school have a Web information system, which is responsible for the administration and control of the several academic and administrative activities that take place within schools, as well as user control and services are provided. These services offer solutions to the problems and needs of users; therefore the maintenance and updating are essential.

Motivated by provide teachers, research groups, and the community at large a better service and a robust system able to make up for needs, it is necessary to implement the service for the management, control and execution of different kinds of projects.

The own professor's website had static information and any service. Due to the continued use of this website and add functionality expands its primary services for teachers, in addition to this, it allows navigation between this website and the website of the school.

System Engineering School purpose is making school councils that discuss issues of teachers, students and board members, which are developed according to the internal regulations of the UIS, to provide answers to these issues. Therefore it's made for each board, which is collected and it is identified as the points in the development of each session. To improve this process we developed a service that allows a controlled manner brings applications and procedures that are generated in a counseling session.

* Degree Work: Research Paper.

** Physicomechanical Engineering Faculty, System Engineering School.
Director: M.Sc. Luis Ignacio González Ramírez

INTRODUCCIÓN

El portal EISIWEB perteneciente a la Escuela de Ingeniería de Sistemas, se ha convertido en uno de los principales canales de comunicación e integración entre los miembros de la comunidad de la escuela. Gracias a la aceptación y utilización por parte de los usuarios surge el objetivo fundamental de fortalecer y mejorar los servicios que se ofrecen, además de la creación de nuevos servicios para suplir las necesidades crecientes de los miembros de la escuela.

El grupo de desarrollo de software Calumet se encarga de desarrollar, administrar y mantener el portal EISIWEB al cual se le realizan continuos cambios y actualizaciones para hacer de este una herramienta cada vez más útil, eficiente, agradable y de fácil uso para la comunidad. Con el objeto de llevar a cabo esta labor y proporcionar páginas con contenido dinámico se ha contado con herramientas software de libre distribución como JSP, Java, JavaScript, jQuery y MySQL.

El continuo soporte ofrecido al portal EISIWEB por el grupo Calumet se hace de manera integral permitiendo el mejoramiento continuo de los servicios ofrecidos, parte de este proceso se llevó a cabo mediante la adecuación e implantación del servicio de gestor de proyectos para permitir el manejo ordenado de los diferentes proyectos de extensión, dentro de este mismo proceso de mejora se amplió el portal del profesor con el fin de brindar un espacio más privado y personalizable a los docentes.

Para satisfacer las nuevas necesidades de los usuarios se hace necesaria la creación de nuevos servicios, tales como el soporte a las sesiones de consejo, que permite llevar de manera controlada, automatizada y más eficiente el proceso anteriormente mencionado.

En este documento se presenta el soporte teórico, metodológico y técnico del desarrollo web de los servicios previamente mencionados, así como de las labores de administración, mantenimiento y soporte a usuarios desempeñadas durante el desarrollo del proyecto.

1 PRESENTACIÓN DEL PROYECTO

1.1 ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME

Este documento presenta en detalle las etapas llevadas en el desarrollo del proyecto: ADMINISTRACIÓN, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN ACADÉMICA DEL PORTAL DE LA ESCUELA DE INGENIERÍA DE SISTEMAS; la información está distribuida de la siguiente forma:

CAPITULO 1. Presentación del Proyecto: Se hace un análisis de los antecedentes del proyecto, definición del problema, objetivos generales, específicos, justificación, alcances y limitaciones del proyecto.

CAPITULO 2. Marco Teórico: Se presentan los conceptos utilizados en el desarrollo técnico del proyecto.

CAPITULO 3. Marco Metodológico: Se menciona el procedimiento metodológico que se siguió para la elaboración del proyecto y las razones por la que fue elegida dicha metodología.

CAPITULO 4. Desarrollo de la herramienta: Se presenta el análisis de requisitos y el diseño de la herramienta.

CAPITULO 5. Documento de pruebas del sistema: se presenta un informe de las pruebas realizadas a la herramienta desarrollada y los resultados obtenidos.

CAPITULO 6. Conclusiones del trabajo realizado.

CAPITULO 7. Recomendaciones y sugerencias a tener en cuenta en la elaboración de futuros proyectos.

1.2 ANTECEDENTES

Debido al crecimiento de la comunidad de la Universidad Industrial de Santander y en particular la comunidad de de la Escuela de Ingeniería de Sistemas e Informática se hace necesario un canal de información y control de la misma. Es así como surge la idea de crear una herramienta que en primer lugar, facilite la interacción entre los miembros de la escuela, desarrollo llevado a cabo en el año 2004 por el grupo de desarrollo de software Calumet el cual conformado y liderado por estudiantes de pregrado de Ing. de Sistemas da vida a la primera versión del portal web de la Escuela de Ingeniería de Sistemas EISIWeb.

Dado el crecimiento y la acogida del portal web EISIWeb, apoyado en la labor de administración, mantenimiento y actualización que realizan sus integrantes, el grupo de desarrollo de software Calumet ha logrado la implementación de su portal web en las escuelas pertenecientes a las facultades de Ingenierías Fisicomecánicas y Fisicoquímicas, así como en sus respectivas decanaturas.

1.3 DEFINICIÓN DEL PROBLEMA

En la actualidad la escuela de Ingeniería de Sistemas cuenta 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, 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.

Motivados por brindarle a los profesores, grupos de investigación, y a la comunidad en general un mejor servicio y un sistema robusto capaz de suplir todas las necesidades se hace necesaria la implantación del servicio para la gestión, control y ejecución de proyectos de diferente índole.

En este momento existe el portal del profesor, sitio propio de cada docente pero que hasta el momento solo cuenta con información estática y ningún servicio. Es importante que los docentes cuenten con un espacio propio y personalizado que brinde mayor accesibilidad y eficiencia en los servicios más usados por dichos docentes.

Es trabajo de la Escuela de Ingeniería de Sistemas realizar consejos de escuela donde se tratan temas de profesores, estudiantes y miembros del consejo, según el reglamento interno de UIS, y dar respuestas a estos asuntos. Como es debido se crea un acta para cada consejo donde se deja por escrito y en constancia los puntos tratados en el desarrollo del consejo. Estas actas se están cargando en la plataforma Moodle de manera NO óptima y cómoda, además no se permite adjuntar los archivos pertinentes que se requieran.

1.4 OBJETIVOS

1.1.1. Objetivo General

Realizar las funciones de administración, mantenimiento, análisis, diseño y desarrollo de nuevos servicios para el portal web existente en la escuela de Ingeniería de Sistemas e Informática, para hacer óptimos, eficientes y sostenibles los trámites dentro de la escuela y el acceso a la información de la misma.

1.1.2. Objetivos Específicos

1.4.1.1 Realizar labores de administración del portal web de la Escuela de Ingeniería de Sistemas e Informática teniendo en cuenta principalmente:

- Generar copias de seguridad diariamente de las bases de datos.
- Salvar la información de los portales web una vez por semana por medio de copias de respaldo y mantener un histórico de los mismos en caso de alguna falla.
- Hacer seguimiento del uso que hacen los usuarios a los servicios de Foros, Eventos Propuestos, Cartelera, Archivos y Mi perfil dentro del portal EISIWEB, para detectar usos indebidos o incorrectos por parte de éstos.
- Actualizar periódicamente las bases de datos con el objeto de mantener al día la información referente a matrículas, horarios, estados y categorías de los usuarios.
- Atender las consultas y sugerencias que los usuarios hagan para proponerlas como mejoramiento y ofrecimiento de nuevos servicios en la próxima versión del portal web.
- Efectuar una revisión constante de los archivos que se suben al sitio, eliminando los que no son necesarios para evitar saturación del portal.

1.4.1.2 Llevar a cabo labores de mantenimiento al portal EISIWEB, en las que podemos encontrar:

- Implementar los nuevos servicios o mejoras realizadas por otros desarrolladores del grupo Calumet dentro del portal EISIWEB, ajustando el nuevo código, modificando las bases de datos y finalmente realizando las pruebas necesarias para verificar que el funcionamiento sea correcto.
- Hacer el seguimiento del funcionamiento de los portales para corregir posibles defectos generados por errores en el código fuente que se puedan presentar.
- Revisar y depurar la estructura de directorios y archivos del portal web.

1.4.1.3 Análisis, diseño, desarrollo e implementación de nuevos servicios o reingeniería de servicios ya existentes, tales como:

- Implantar el servicio de gestor de proyectos en la Facultad de Ingenierías Fisicomecánicas ajustado este servicio a los requerimientos de la decanatura.
- Mejorar el portal del Profesor. De esta forma cada docente contará con un espacio propio donde podrá acceder de manera fácil y rápida a los servicios más importantes. También se ofrecerá la navegación entre este portal y el de la escuela.

- Desarrollar un nuevo servicio para llevar control de las sesiones de consejo de Escuela. Se permitirá ingresar las solicitudes mediante el portal y guardarlas en base de de datos para luego ser revisadas y tratadas en la sesión de consejo correspondiente. También se permite llevar a cabo todo el desarrollo de la sesión de manera virtual. Es la primera versión.

1.4.1.4 Desempeñar labores de soporte a los usuarios del portal EISIWEB, brindando así solución a sus diferentes necesidades o situaciones que se puedan presentar. Entre estas labores se destacan:

- Capacitar usuarios y estudiantes del primer nivel en el uso de servicios dentro del Portal EISIWEB, promoviendo así su utilización.
- Atender usuarios por olvido de la contraseña, creación de grupos, solicitudes de propuestas de eventos o de cartelera y creación de agendas con eventos y foros con sus respectivas conversaciones.
- Crear usuarios de forma manual, para personas no pertenecientes a la escuela y que por algún motivo necesitan registrarse en el sitio.
- Modificar los estados de los usuarios de acuerdo a la relación con la Escuela (Activo, Inactivo, Suspendido).

1.4.1.5 Capacitar a los estudiantes que recibirán las funciones de administración, mantenimiento, creación, y mejora de nuevos servicios dentro del portal EISIWEB en cuanto a:

- Implantación del sitio local para la creación de nuevos servicios y realización de pruebas.
- Realizar inducción en cuanto al manejo y utilización de los JSP, Beans, base de datos, JavaScript y jQuery.
- Llevar a cabo la familiarización con el entorno del portal web.

1.5 JUSTIFICACIÓN

Los servicios del portal de las escuelas deben mejorar constantemente y proporcionalmente a los cambios que se presenten en su entorno, a su vez deben dar correcta solución a los problemas y necesidades que surjan por parte de los usuarios, razón por la cual las labores de mantenimiento y actualización son indispensables.

Motivados por brindarle a los profesores, grupos de investigación, y a la comunidad en general un mejor servicio y un sistema robusto capaz de suplir todas las necesidades se hace necesaria la implantación del servicio para la gestión, control y ejecución de proyectos de diferente índole.

Los docentes deben contar con un espacio personalizado en el que puedan llevar a cabo las tareas más comunes e importantes de su labor docente, tareas como enviar correos, crear foros, solicitudes de trabajos de grado, aula virtual y otros más.

Los procesos académicos y administrativos de la Escuela deben realizarse de manera eficiente y confiable, para esto es útil contar con servicios web que permitan flexibilidad y control a dichos procesos. En una sesión del Consejo de

Escuela se necesita llevar control de las peticiones hechas por los estudiantes así como los demás miembros del consejo, la automatización de estas labores permite realizar procesos más eficientes y reducir tiempos de operación.

1.6 ALCANCES Y LIMITACIONES

El gestor de proyectos permite la creación de diferentes clases de proyectos asociándolos a un autor (docente - grupo) así como la asignación de actividades por semanas. Se permite ingresar proyectos cuyas fechas de inicio sean anteriores a la fecha de ingreso al sistema.

La nueva versión del Portal del profesor permite a los docentes acceder a los servicios más utilizados y recurrentes dentro de sus labores académicas y administrativas. Por otra parte se crea la primera iniciativa para una navegación interactiva entre este portal, el de la escuela, los grupos y así mismo la producción intelectual (portal que está en proceso de creación).

El servicio de Actas del consejo de Escuela surge como herramienta para optimizar labores administrativas. Se permite el estudio de casos y solicitudes pertinentes a la Escuela y a la Facultad, para esta última se deja estructurado de tal manera que más adelante se permita la conexión entre las escuelas y la facultad.

2 MARCO TEÓRICO

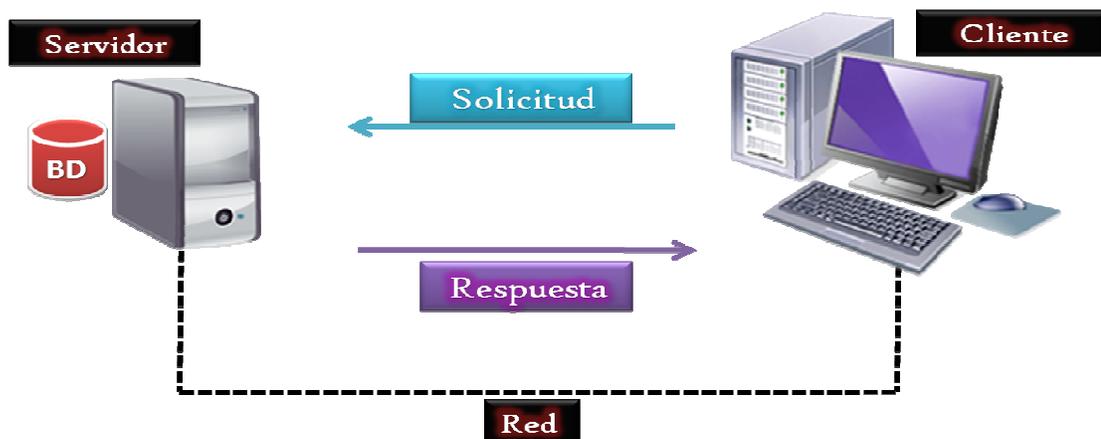
2.1 ARQUITECTURA CLIENTE/SERVIDOR

Se define como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En el modelo C/S, el cliente envía un mensaje solicitando un servicio a un servidor (hacer una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio) (ver figura 1). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

La arquitectura C/S es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento.

Figura 1. Modelo Cliente/Servidor



Fuente: Internet. <http://ulyla003.wordpress.com/page/4/>

La arquitectura C/S presenta unas características y clasificaciones que permiten la aplicación correcta de este modelo. (Ver Anexo A).

2.1.1. 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.

2.1.2. 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 maquinas 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.

2.1.3. 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, las cuales se pueden agrupar en Código del Lado del Cliente (Client Side Scripts) y Código del Lado del Servidor (Server Side Scripts). (Ver Anexo B).

2.2.1 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.

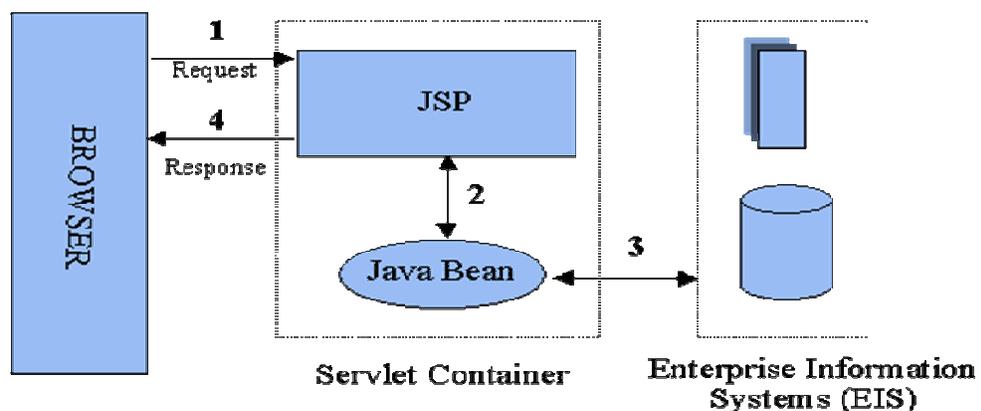
Una página JSP se compila a una aplicación Java la primera vez que se invoca, y de esta aplicación Java se crea una clase que empieza a ejecutarse en el servidor como un servlet. Un JSP es una página web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe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

2.2.1.1 Modelo de acceso a JSP

1. 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.

2. Después de recibir la petición del cliente, el archivo JSP pide información de un JavaBean si es necesario.
3. El JavaBean en turnos puede pedir información de otro JavaBean o de una base de datos.
4. 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: Internet: <http://geneura.ugr.es/~jmerelo/JSP/modelo1.gif>

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.

Dentro de las bases de datos existen diferentes modelos, entre los cuales encontramos bases de datos jerárquicas, de red y relacionales. Así mismo para tener conexión a estos modelos es necesario utilizar conectores los más comunes son ODBC, MDB y JDBC. (Ver Anexo C).

Existen diferentes manejadores de bases de datos como MySQL, ORACLE, FoxPro, Microsoft Access o PowerBuilder, en el desarrollo de los portales FIFMEWeb y EISIWeb se utiliza MySQL.

2.3.1 MySQL

Es un sistema de base de datos operacional considerado uno de los más importantes y utilizados 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.

2.3.1.1 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 7.1.2. 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. El NetBeans C/C++ Pack soporta proyectos de C/C++.

- Modularidad. 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, permitiéndole 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 de que esta hecho 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 Jakarta 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 el Portal de la de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander se propone como metodología de desarrollo el prototipo evolutivo.

Figura 3. Prototipo Evolutivo.



Fuente: Autor.

La elección de esta metodología se debe a las siguientes razones:

- El portal ESIWEB presenta constantemente necesidades por parte de los usuarios, por lo tanto se requiere el desarrollo de nuevos servicios y así

mismo la mejora de los ya existentes, debido a que el sistema no es un producto final, el mismo se considera en constante reconstrucción.

- El desarrollo de los aspectos visuales del sistema puede estar sujeto a cambios por parte de los usuarios del portal EISIWEB durante el desarrollo del proyecto.
- Durante la primera etapa de los nuevos servicios es necesario tener una clara interpretación de las especificaciones dadas por los usuarios y escuelas, pero muchas veces los usuarios no tienen una idea clara de lo que necesitan. Por esto la construcción de prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado.
- Otra ventaja es tener la posibilidad de realizar cambios en etapas tempranas y crear varios prototipos evaluables durante el desarrollo, obteniéndose de este modo una metodología integral para el proceso de evaluación del programa.
- Esta metodología favorece la autocrítica al sistema, lo que conlleva a que se produzca muchas pruebas antes de dar un nuevo prototipo, así como mejoras rápidas a problemas que puedan surgir durante su uso.

Procedimiento a seguir para la metodología planteada:

- Se toma nota de los requisitos para la construcción de los prototipos.
- Se especifican en detalle los objetivos globales del software a realizar, mediante una reunión entre el desarrollador y el usuario, en la cual se

identifican los requisitos básicos y se concluyen las áreas donde se necesita mayor definición.

- Posteriormente se presenta al usuario el diseño de un prototipo que se enfoca en los aspectos visuales del software, métodos de entrada y formatos de salida, para proceder a la construcción.
- El prototipo es evaluado por el usuario y se utiliza 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.

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.

3.2.1.1 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. (Ver anexo D).

3.2.1.2 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. (Ver anexo D).

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 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.

Se han definido tres categorías para las diferentes tablas que conforman la base de datos. Dada la categoría de la tabla, se antepondrá 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 a anteponer a los nombres de estas tablas es “TB_”, es decir la tabla que almacena los puntos que se tratan en un consejo de escuela es llamada “TB_PuntosConsejo”.

- *Tabla de Relación:* Surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deberán ser descriptivos para cada relación. El prefijo a anteponer a los nombres de estas tablas es “TR_”, es decir la tabla de “MiembrosConsejo” es conocida como “TR_MiembrosConsejo”.
- *Tabla Principal:* Aquella cuya población 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 a 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 las clases deben ser sustantivos en plural, la primera letra de cada palabra debe ser mayúscula. Estos deben ser simples, descriptivos como por ejemplo: ConsejoEscuela.java, PortalProfesor.java.

3.3.4 Páginas JSP

Los nombres de las paginas JSP que componen el portal EISIWeb serán escritos inicializando en mayúscula precedido de letras minúsculas, en caso de que el nombre del JSP compuesto por dos o más palabras, entonces la primera de cada palabra interna debe ir en mayúscula , por ejemplo:

IngresarSolicitud.jsp, DesarrolloActa.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. Por ejemplo:

- El sitio cuenta con un directorio llamado “images”; en éste se encuentran almacenados todos los archivos .jpg, .gif, .png.
- Para el desarrollo de este proyecto se creó el directorio “ConsejoEscuela”, el cual almacena los archivos jsp correspondientes para el funcionamiento de dichos servicios.
- El sitio cuenta con un directorio llamado Script, donde están todos los archivos de JavaScript “.js” necesarios para el buen funcionamiento del portal. Para este proyecto se creó un directorio llamado “ConsejoEscuela”, el cual almacena los archivos correspondientes para el funcionamiento de dichos servicios.

4 DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

Como se mencionó anteriormente 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 inicio el proyecto no se tenía una concepción clara de cómo sería el producto final, sin embargo durante el 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 exitosamente el desarrollo a la solución de las necesidades de los usuarios.

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 listan los requerimientos detallados de este, los cuales se cumplieron para el prototipo final.

- 1. Implantar el servicio de gestor de proyectos en la Facultad de Ingenierías Fisicomecánicas ajustándolo a los requerimientos de la decanatura, permitir el ingreso de proyectos cuya semana de inicio sea inferior a la**

fecha de ingreso al sistema y mejorar las interfaces de usuarios permitiendo mayor usabilidad en el servicio.

Objetivo Inicial:

- Implantar el servicio de gestor de proyectos en la Facultad de Ingenierías Fisicomecánicas ajustado este servicio a los requerimientos de la decanatura.

Requisitos finales del objetivo:

- Permitir asignar fechas anteriores a la fecha de iniciación de proyecto.
- Brindar la información del proyecto de forma clara y agradable mediante el uso de interfaces gráficas que permitan mayor usabilidad del servicio.
- Ajustarse a los requerimientos de la Facultad de Ingenierías Fisicomecánicas.
- Revisar y participar en las actividades que tiene el proyecto en cada semana.

2. Mejorar el portal del profesor y crear un espacio interactivo para que los docentes y usuarios EISIWeb, puedan interactuar con los servicios más importantes dentro de la labor docente. Además permitir el ingreso a este portal mediante el nombre de usuario y contraseña del portal de la escuela, para así clasificar a los usuarios y otorgar permisos acorde a sus privilegios. También se ofrece la navegación entre este portal y el de la Escuela manteniendo vigente la sesión.

Objetivo inicial:

- Integrar en el portal del profesor los servicios necesarios para que cada docente cuente con un espacio propio donde tenga acceso fácil y rápido a los servicios del portal.

Requisitos finales del objetivo:

- Este portal maneja los mismos usuarios y sus respectivos permisos que el portal de la escuela.
- Se permite el inicio de sesión y así desplegar un menú de servicios acorde a los permisos del usuario sesionado.
- Permite el uso normal de los servicios mostrados según los permisos del usuario sesionado.
- Se puede navegar entre este portal y el portal de la escuela manteniendo la sesión vigente.
- Se detecta cuando hay cambio de sesión en los demás portales y se actualiza la misma.
- El sistema detecta cuando un profesor ingresa a su portal y le da la opción de personalizarlo.

3. Desarrollar un servicio que de soporte y permita llevar de manera automatizada y organizada las sesiones de consejo de escuela, de manera que todos los trámites se puedan realizar mediante el portal de la escuela. También se ofrece como medio de información rápido y pertinente para los usuarios interesados en estos trámites.

Objetivo inicial:

- Implementar un servicio que soporte las sesiones del consejo de Escuela, permitiendo llevar registro de las solicitudes realizadas por los estudiantes y demás miembros autorizados, así como de las actas con sus respectivos puntos y respuestas generadas a ellos.

Requisitos finales del objetivo:

Los siguientes son los requisitos del módulo administrador de Actas de Consejo:

- Permite iniciar el servicio indicando el número de acta a partir de la cual se continuará la numeración.
- Informar al usuario la realización de una solicitud mediante un correo electrónico.
- Permite ingresar todo tipo de solicitud, indicando a qué punto del consejo pertenece. Además puede ingresar esta solicitud en nombre de otro usuario en caso que sea conveniente. Se puede adjuntar archivos “pdf”.
- Permite revisar las solicitudes que se han registrado e indicarle al usuario que la hizo si es necesario realizar cambios.
- Permite modificar el orden del día y enviar un correo a los miembros del consejo indicando este orden y el lugar de la próxima sesión del consejo.
- Permite iniciar la sesión del consejo, se confirma el lugar.
- Permite revisar el quórum, se señalan los miembros que han llegado y se confirma su llegada.
- Permite indicar al sistema las decisiones, comentarios y respuestas tomadas por los miembros del consejo en cada uno de los asuntos tratados.

Los siguientes son los requisitos para los miembros del consejo de Actas Consejo:

- Permite ingresar solicitudes con archivos “pdf” adjuntos a los puntos del consejo según sus permisos.
- Informar al usuario la realización de una solicitud mediante un correo electrónico.

- Permite editar las solicitudes que realizadas que el administrador ha devuelto para cambios.
- Permite ver y comentar el orden del día para la próxima sesión de consejo.
- Permite ver y comentar las solicitudes vigentes para la próxima sesión de consejo.

Los siguientes son los requisitos para Actas Consejo público:

- Permite ingresar solicitudes con archivos “pdf” adjuntos a los puntos del consejo según sus permisos.
- Informar al usuario la realización de una solicitud mediante un correo electrónico.
- Permite editar las solicitudes que realizadas que el administrador ha devuelto para cambios.

Para aquellos servicios que lo requieran, se debe incluir la rutina de login de usuario y verificación de sesión, requerimiento de seguridad intrínseco de los sitios web a los que presta sus servicios el grupo CALUMET.

Administración.

Objetivos iniciales:

- Efectuar labores de administración del portal web de la Escuela de Ingeniería de Sistemas; teniendo en cuenta entre otras:
 - Generar Backups (copias de respaldo) diariamente de la Base de Datos.
 - Salvar la información de los sitios web semanalmente, por medio de copias de respaldo y mantener un histórico de los mismos en caso de alguna falla.

- Hacer seguimiento del uso que hacen los usuarios en cuanto a los servicios de Foros, Eventos Propuestos, Cartelera, Archivos y Mi perfil dentro del portal EISIWeb para detectar usos indebidos o incorrectos por parte de éstos.
- Actualizar periódicamente las Bases de Datos con el objeto de mantener al día la información referente a matrículas, horarios, estados y categorías de los usuarios.
- Atender consultas y sugerencias que los usuarios hagan para proponerlas como mejoramiento y ofrecimiento de nuevos servicios en la próxima versión del portal web.
- Realizar una revisión constante de los archivos que se suben al sitio, eliminando los que no son necesarios para evitar saturación del portal.

Requisitos finales del objetivo:

Para usuarios con perfil de administrador:

- Corregir cada uno de los fallos que a diario se presentan en los portales web.
- Mantenimiento a la bases de datos para eliminar tablas que ya no se usan o crear nuevas que se necesitan para que los nuevos servicios funcionen.
- Actualizar el diagrama Entidad/Relación de la base de datos Diamante y subirlo al portal EISIWeb, en el espacio del grupo Calumet; para realizar futuras actualizaciones sobre estos últimos.
- Revisar la estructura de directorios del portal, borrar los archivos que ya no se usan y agregar nuevos servicios implantados en el portal.
- Realizar limpiezas en cuanto a conversaciones y datos que ya no sean necesarios dentro de las Bases de datos.
- Mantener los JavaBeans actualizados y subirlos al portal EISIWeb, en el espacio del grupo Calumet; para realizar futuras modificaciones sobre estos últimos.

Cada una de las actividades mencionadas en el objetivo inicial se llevó a cabo durante la práctica y desarrollo del proyecto.

Mantenimiento.

Objetivos iniciales:

- Llevar a cabo labores de mantenimiento al portal EISIWeb en las que podemos encontrar:
 - Implementar los nuevos servicios o mejoras realizadas por otros desarrolladores del grupo CALUMET dentro del portal EISIWeb, ajustando el nuevo código, modificando la Base de Datos y realizando las pruebas necesarias.
 - Hacer el seguimiento del funcionamiento de los portales para corregir posibles defectos generados por errores en el código fuente que se puedan presentar y dañar el funcionamiento del sistema.
 - Revisar y depurar la estructura de directorios y archivos del portal web.
 - Realizar el mantenimiento de un archivo clasificado de las solicitudes de cambios y correcciones que hagan los usuarios.

Requisitos finales del objetivo:

Para usuarios con perfil de administrador:

- Revisar regularmente el historial de cambios para supervisar el uso correcto que los usuarios le dan a los servicios.
- Realizar copias periódicas de las bases de datos Division y Diamante y de los directorios del sitio EISIWeb.

- Eliminar de la base de datos diamante información que no se usa, ya sean archivos o conversaciones.
- Hacer la actualización periódica de la base de datos Diamante.

Cada una de las actividades mencionadas en el objetivo inicial se llevó a cabo durante la práctica.

Soporte a Usuarios.

Objetivo inicial:

- Desempeñar labores de soporte a los usuarios del portal EISIWeb, brindando así solución a sus diferentes necesidades, conflictos o situaciones que se puedan presentar. Entre estas labores se destacan:
 - Capacitar usuarios y estudiantes del primer nivel en el uso de servicios dentro del portal EISIWeb, promoviendo así su utilización.
 - Atender usuarios por olvido de la contraseña, creación de grupos, solicitudes de propuestas de eventos, solicitudes de cartelera y creación de agendas con eventos y foros.
 - Crear usuarios de forma manual para personas no pertenecientes a EISIWeb y que por algún motivo necesitan registrarse en el sitio web correspondiente.
 - Modificar los estados de los usuarios de acuerdo a la relación con la escuela (activo, inactivo, suspendido).

Requisitos finales del objetivo:

Para usuarios con perfil de administrador:

- Conocer el total funcionamiento del portal EISIWeb, a fin de dar solución a las posibles dificultades que se presenten.
- Investigar las soluciones a nuevas dificultades que se presenten.

Cada una de las actividades mencionadas en el objetivo anterior se llevó a cabo durante la práctica en repetidas ocasiones. Además, se dio orientación a los usuarios sobre el uso de determinados servicios.

Capacitación a nuevos integrantes del grupo Calumet.

Objetivo inicial:

- Capacitar a los estudiantes que relevaran las funciones de administración, mantenimiento, creación, y mejora de nuevos servicios dentro del portal EISIWeb en cuanto a:
 - Implantación del sitio local para la creación de nuevos servicios y realización de pruebas.
 - Realizar inducción en cuanto al manejo y utilización de los JSP, JavaBeans y Base de Datos.
 - Llevar a cabo la familiarización con el entorno de los portales web.

Requisitos finales del objetivo:

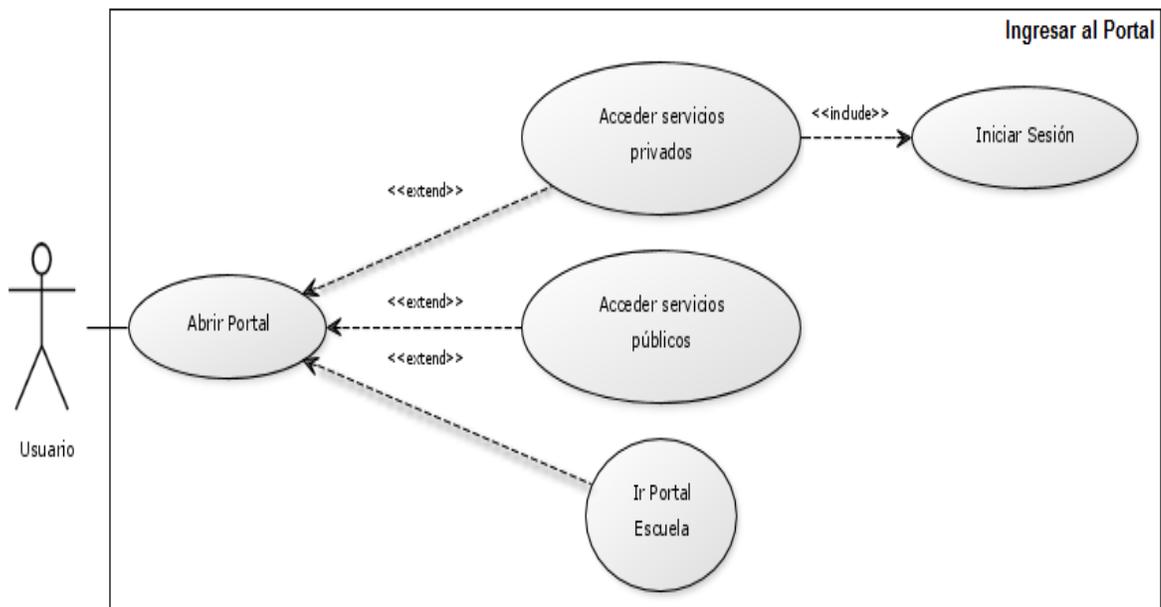
- Para usuarios con perfil de administrador.
- Dar orientación a los nuevos integrantes del grupo sobre posibles errores en la instalación del sitio local.

- Dar orientación básica a los nuevos integrantes del grupo sobre la programación en JSP, uso de los JavaBeans y funcionalidad de cada tabla dentro de la base de datos.
- Dar capacitación a los nuevos integrantes del grupo sobre los estándares que se siguen para la programación en cuanto a nombres y estilos.
- Orientar a los nuevos integrantes sobre el procedimiento para crear y habilitar nuevos servicios en el sitio.

4.1.1 Diagramas de Casos de Uso

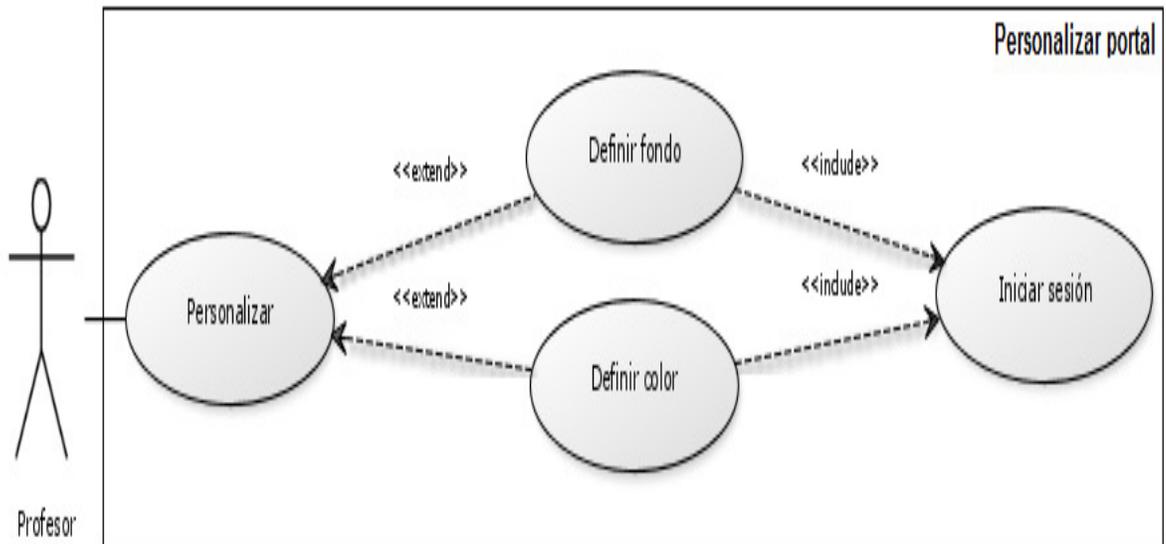
4.1.1.1 Portal Profesor

Figura 4. Diagrama de Casos de Uso: Ingreso al portal.



Fuente: Autor.

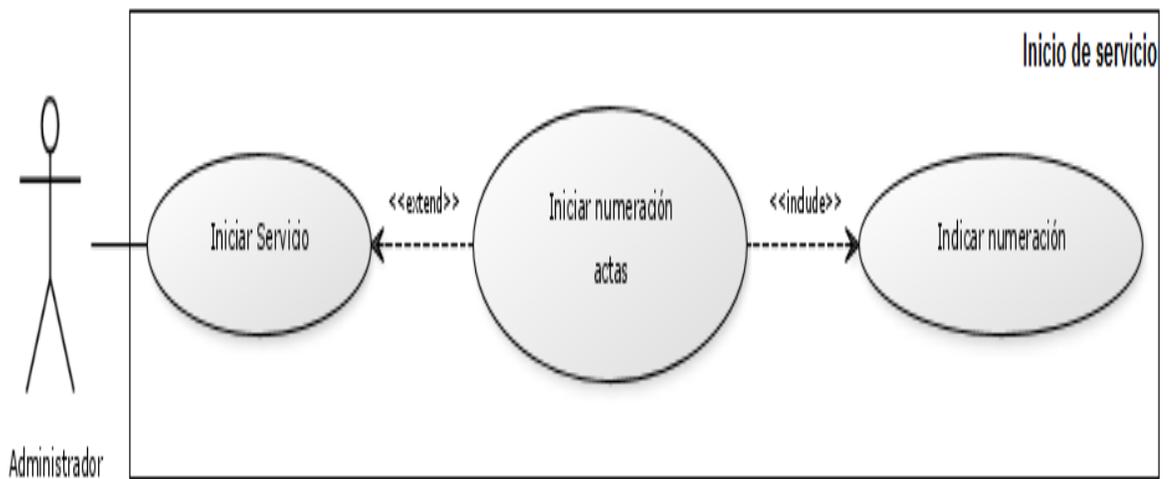
Figura 5. Diagrama de Casos de Uso: Personalizar portal.



Fuente: Autor.

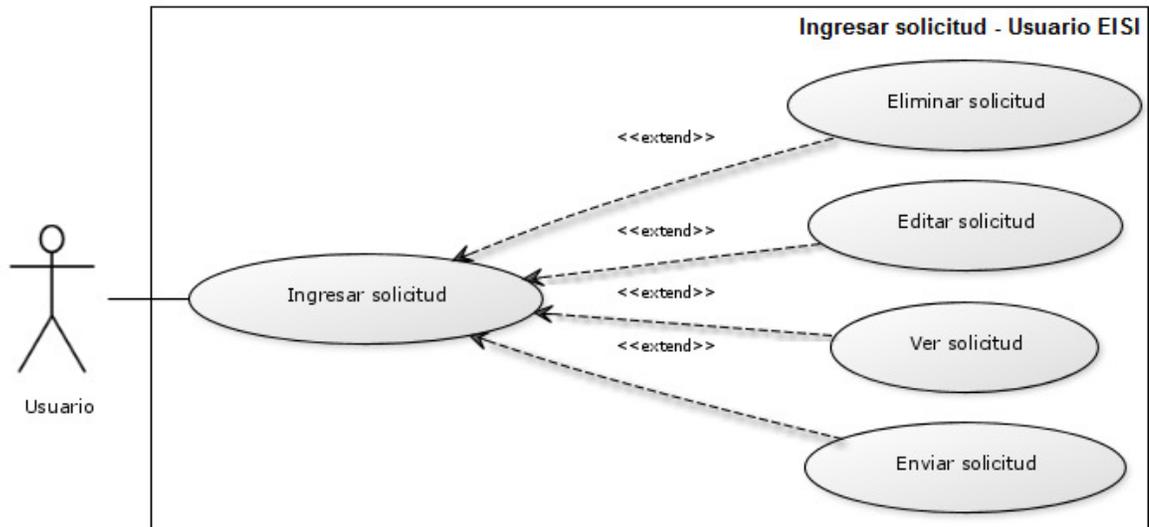
4.1.1.2 Servicio de Actas de Consejo

Figura 6. Diagrama de Casos de Uso: Inicio del servicio.



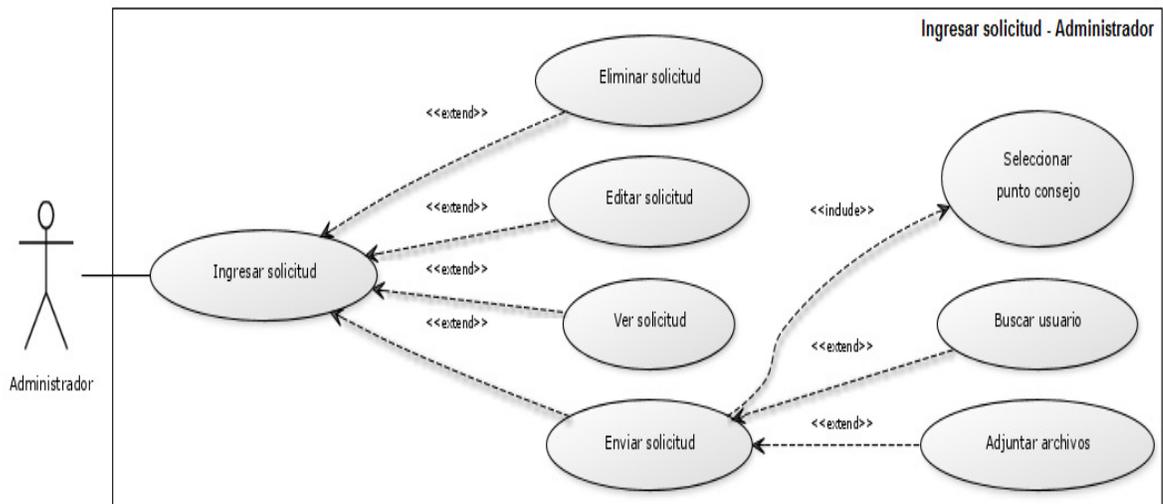
Fuente: Autor.

Figura 7. Diagrama de Casos de Uso: Ingreso solicitudes usuario EISI



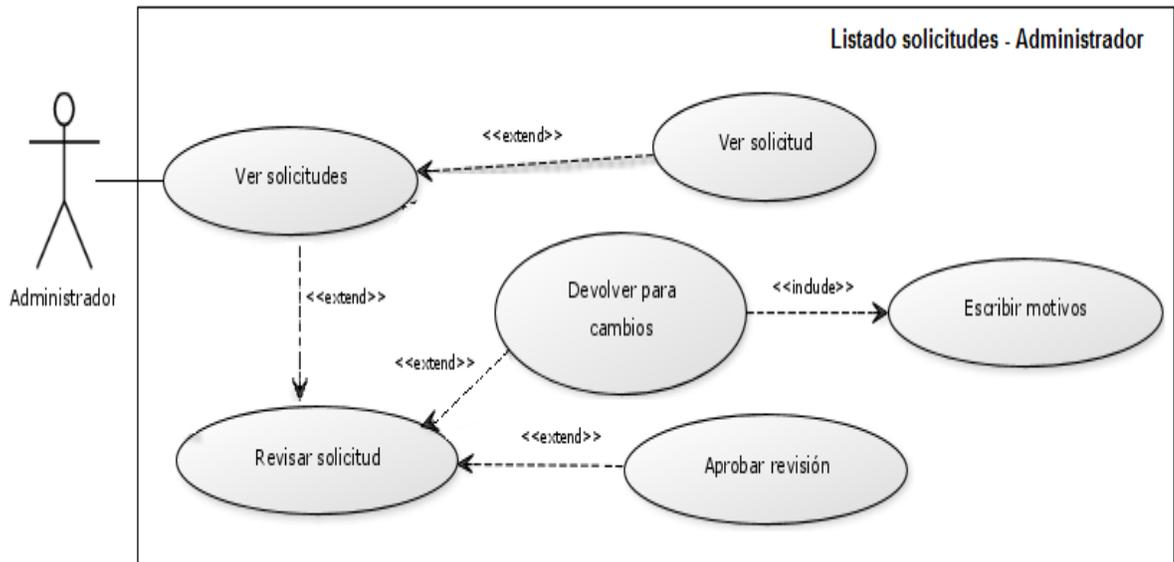
Fuente: Autor.

Figura 8. Diagrama de Casos de Uso: Ingreso solicitud administrador.



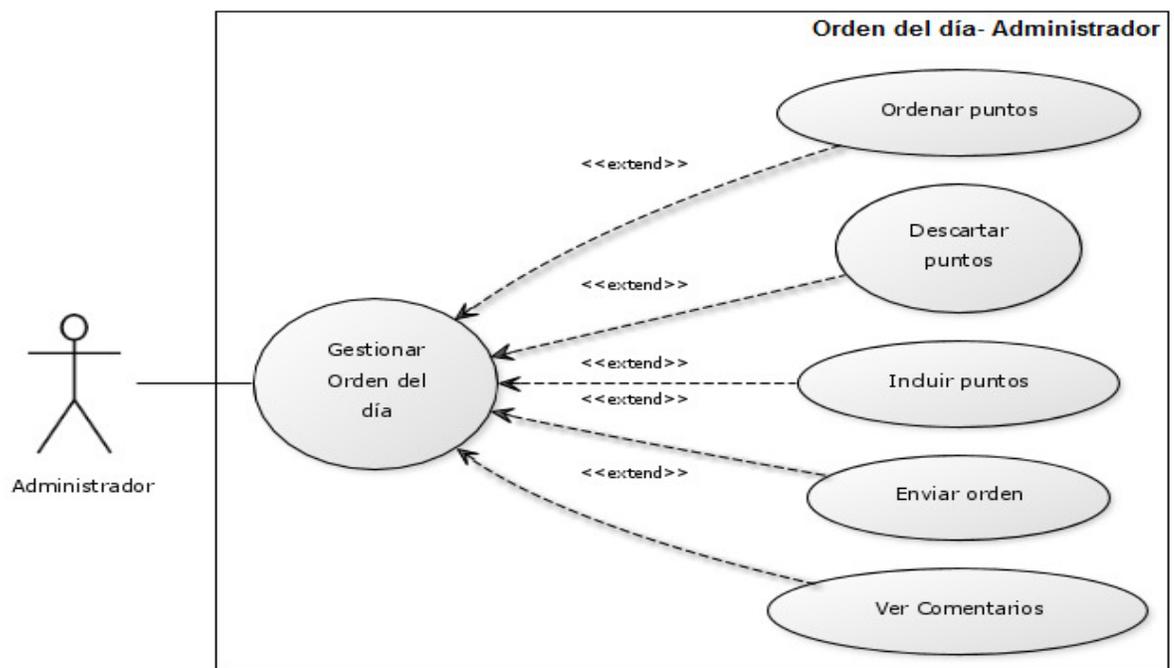
Fuente: Autor.

Figura 9. Diagrama de Casos de Uso: Listado solicitudes administrador.



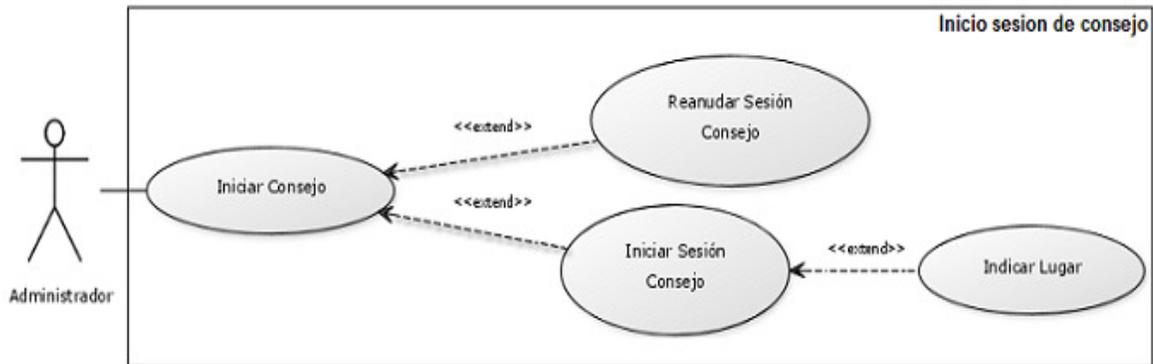
Fuente: Autor.

Figura 10. Diagrama de Casos de Uso: Orden del día - Administrador.



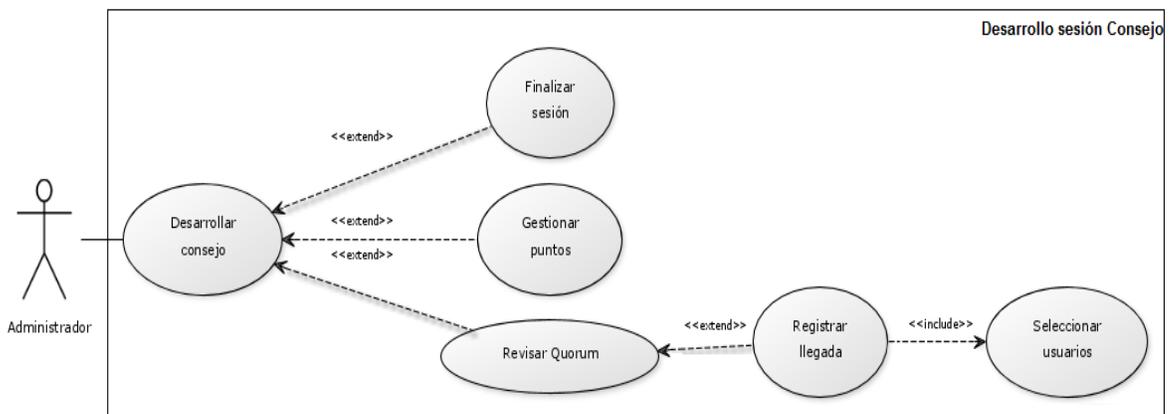
Fuente: Autor.

Figura 11. Diagrama de Casos de Uso: Inicio sesión de Consejo.



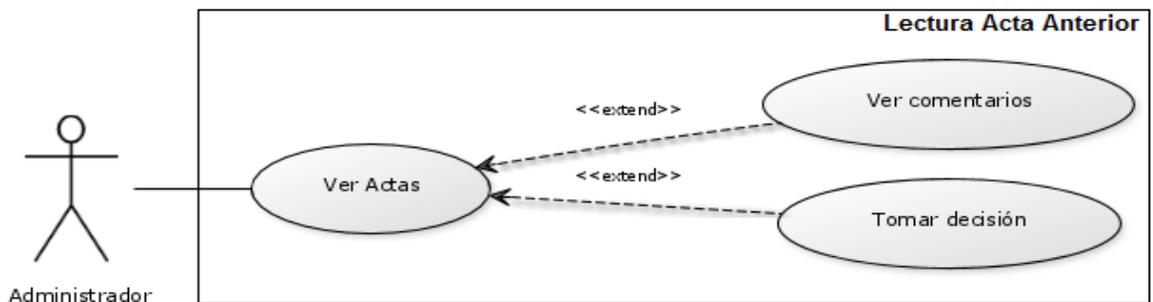
Fuente: Autor.

Figura 12. Diagrama de Casos de Uso: Desarrollo sesión consejo.



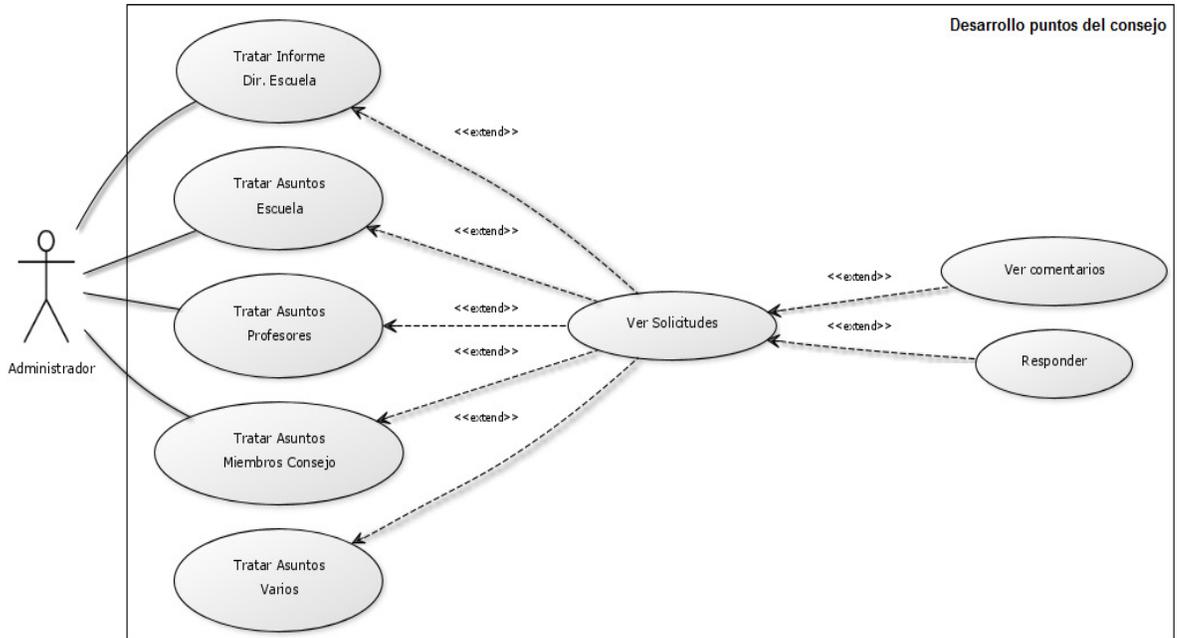
Fuente: Autor.

Figura 13. Diagrama de Casos de Uso: Lectura Acta anterior.



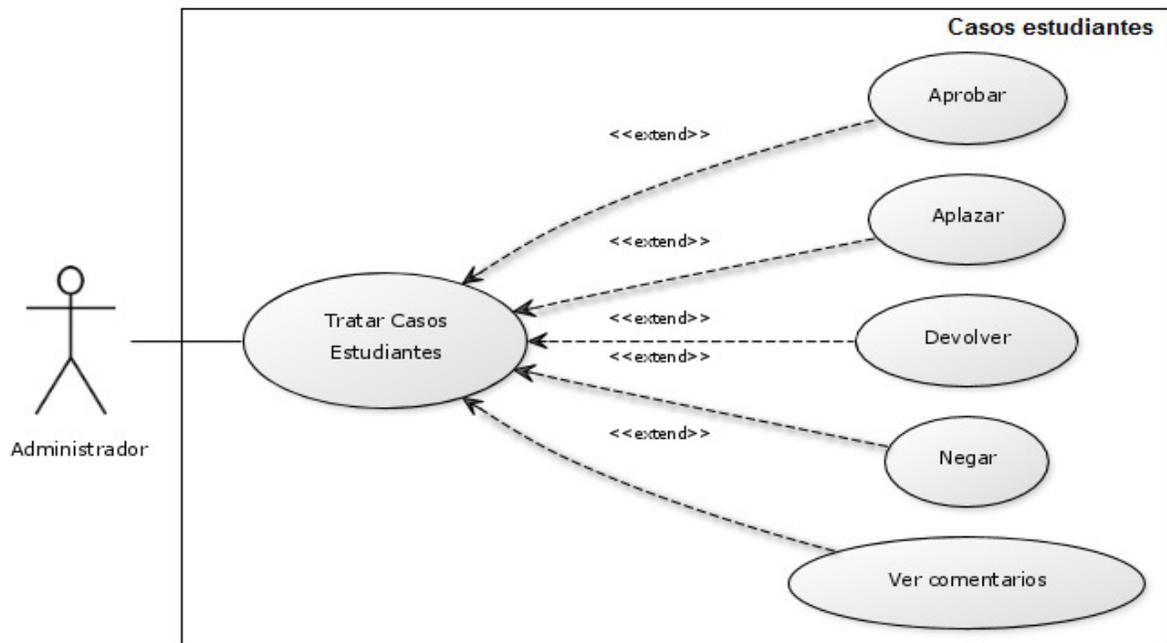
Fuente: Autor.

Figura 14. Diagrama de Casos de Uso: Desarrollo puntos consejo.



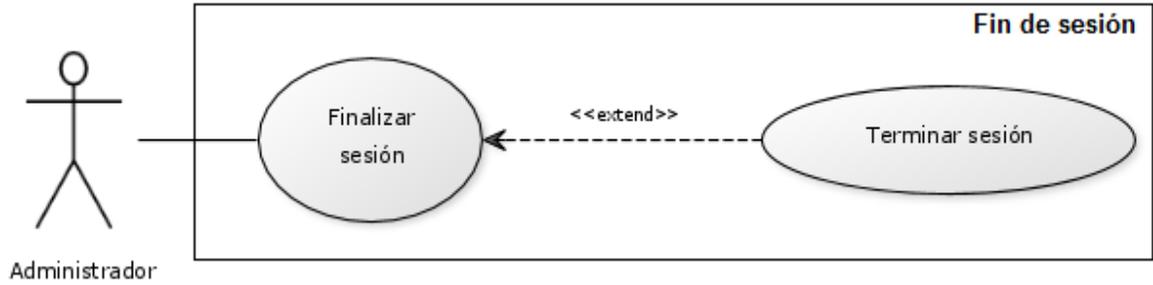
Fuente: Autor.

Figura 15. Diagrama de Casos de Uso: Casos estudiantes.



Fuente: Autor.

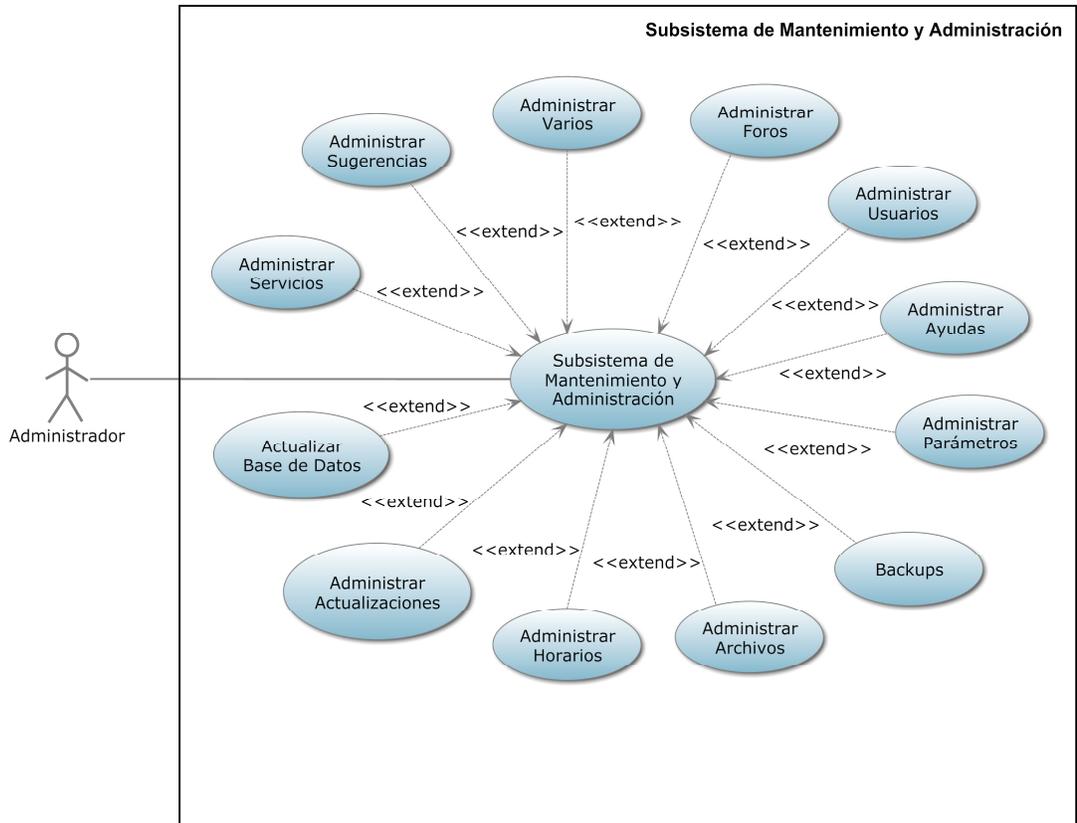
Figura 16. Diagrama de Casos de Uso: Fin de sesión.



Fuente: Autor.

4.1.1.3 Mantenimiento y Administración del portal EISIWEB.

Figura 17. Diagrama de Casos de Uso: Subsistema de Mantenimiento y Administración.



Fuente: Autor.

4.1.2 Documentación de Casos de Uso del Sistema

4.1.2.1 Portal Profesor

Tabla 1. Caso de Uso: Ingreso al portal.

TÍTULO	FUNCIONES PRIMARIAS
Abrir portal	Permite ir al portal del profesor seleccionado.
Acceder servicios privados	Navegar entre los servicios que tiene permitido el usuario sesionado.
Iniciar sesión	Ingresar con usuario y contraseña para poder acceder a los servicios privados
Acceder servicios públicos	Navegar en los servicios que no tienen restricción de usuarios.
Ir al portal de la Escuela	Ir al portal de la Escuela sin perder la sesión iniciada.

Fuente: Autor.

Tabla 2. Caso de Uso: Personalizar portal.

TÍTULO	FUNCIONES PRIMARIAS
Personalizar	Acceder al módulo que permite personalizar el portal.
Definir fondo	Indicar un estilo para el fondo del header y footer.
Definir color	Seleccionar una imagen para asignarla como background.
Iniciar sesión	Ingresar con usuario y contraseña para verificar que es el profesor del portal y lo puede personalizar.

Fuente: Autor.

4.1.2.2 Servicio de Actas de Consejo

Tabla 3. Casos de uso: Inicio del servicio

TÍTULO	FUNCIONES PRIMARIAS
Iniciar Servicio	Acceder al servicio que permite iniciar la numeración de las actas y el inicio del servicio.
Iniciar numeración	Iniciar el servicio y generar la primera acta.
Indicar Numeración	Indicar que numero de acta tendrá la primera acta generada.

Fuente: Autor.

Tabla 4. Casos de uso: Ingreso solicitud usuario EISI

TÍTULO	FUNCIONES PRIMARIAS
Ingresar solicitud	Acceder al servicio que permite digitar una nueva solicitud ante el consejo de escuela.
Guardar solicitud	Guardar la solicitud digitada por el usuario.
Ver solicitud	Permite ver los detalles de las solicitudes realizadas por el usuario.
Editar solicitud	Permite realizar cambios a una solicitud que ha sido devuelta por el administrador.
Eliminar solicitud	Permite a un usuario desistir de una solicitud siempre y cuando el Consejo de Escuela no la haya tratado.

Fuente: Autor.

Tabla 5. Casos de uso: Ingreso solicitud administrador

TÍTULO	FUNCIONES PRIMARIAS
Ingresar solicitud	Acceder al servicio que permite digitar una nueva solicitud ante el consejo de escuela.
Enviar solicitud	Envía la solicitud digitada por el usuario.
Ver solicitud	Permite ver los detalles de las solicitudes realizadas por el usuario.
Editar solicitud	Permite realizar cambios a una solicitud que ha sido devuelta por el administrador.
Eliminar solicitud	Permite al administrador desistir de sus solicitudes siempre y cuando el Consejo de Escuela no la haya tratado.
Seleccionar punto consejo	Se debe indicar a qué punto del consejo desea ingresar la solicitud.
Buscar usuario	Permite buscar un usuario alternativo para registrar la solicitud.
Adjuntar archivos	Permite adjuntar los archivos necesarios.

Fuente: Autor.

Tabla 6. Casos de uso: Listado solicitudes administrador.

TÍTULO	FUNCIONES PRIMARIAS
Ver solicitudes	Listar todas las solicitudes vigentes y clasificarlas según su estado.
Revisar solicitud	Permite revisar una solicitud y dar un visto bueno o devolverla para cambios.
Aprobar revisión	Permite aprobar una solicitud.

Devolver para cambios	Permite indicar que una solicitud necesita cambios para ser estudiada.
Indicar motivos	Permite indicar al usuario porque se ha devuelto su solicitud.

Fuente: Autor.

Tabla 7. Casos de uso: Orden del día - Administrador

TÍTULO	FUNCIONES PRIMARIAS
Gestionar orden del día	Permite acceder al servicio para gestionar el orden del día.
Ordenar puntos	Permite asignar un orden a los puntos del consejo a tratar en la próxima sesión.
Descartar puntos	Permite descartar un punto del orden del día de la próxima sesión del consejo.
Incluir puntos	Permite incluir en el orden del día un punto anteriormente descartado.
Enviar orden	Permite enviar a los miembros del consejo el orden del día y el lugar de la próxima sesión.
Ver comentarios	Permite revisar los comentarios acerca del orden del día realizados por los miembros del consejo.

Fuente: Autor.

Tabla 8. Casos de uso: Inicio sesión de Consejo

TÍTULO	FUNCIONES PRIMARIAS
Iniciar consejo	Permite acceder al servicio para iniciar una nueva sesión de consejo.
Reanudar sesión	Permite entrar a una sesión de consejo en curso.

Iniciar sesión consejo	Permite iniciar una nueva sesión de consejo.
Indicar lugar	Permite indicar el lugar en el que se llevará a cabo la sesión de consejo.

Fuente: Autor.

Tabla 9. Casos de uso: Desarrollo sesión de consejo de Escuela

TÍTULO	FUNCIONES PRIMARIAS
Desarrollar consejo	Acceder al módulo donde se tratarán los puntos del consejo con sus respectivos ítems.
Revisar Quórum	Permite acceder al servicio para revisar el quórum.
Registrar Llegada	Permite registrar la asistencia de los miembros del consejo presentes.
Seleccionar usuarios	Permite indicar los usuarios que están presentes.
Gestionar puntos	Acceder a los servicios donde se tratarán cada punto del consejo.
Finalizar sesión	Permite dar por terminada una sesión de consejo.

Fuente: Autor.

Tabla 10. Casos de uso: Lectura Acta anterior

TÍTULO	FUNCIONES PRIMARIAS
Ver actas	Acceder al servicio que permite ver las actas por aprobar.
Ver comentarios	Permite revisar los comentarios que han hecho los miembros del consejo acerca de un acta.

Tomar decisión	Permite tomar una decisión acerca del acta por aprobar.
----------------	---

Fuente: Autor.

Tabla 11. Casos de uso: Desarrollo puntos del consejo.

TÍTULO	FUNCIONES PRIMARIAS
Tratar informe del Director de Escuela	Acceder al servicio para estudiar las solicitudes de este punto.
Tratar Asuntos de la Escuela	Acceder al servicio para estudiar las solicitudes de este punto.
Tratar Asuntos de Profesores	Acceder al servicio para estudiar las solicitudes de este punto.
Tratar Asuntos Miembros Consejo	Acceder al servicio para estudiar las solicitudes de este punto.
Tratar Asuntos Varios	Acceder al servicio para estudiar las solicitudes de este punto.
Ver solicitudes	Permite ver las solicitudes a estudiar para un punto del consejo específico.
Ver comentarios	Permite revisar los comentarios de una solicitud.
Responder	Permite emitir una respuesta y un concepto a una solicitud tratada.

Fuente: Autor.

Tabla 12. Casos de uso: Casos estudiantes.

TÍTULO	FUNCIONES PRIMARIAS
Tratar casos estudiantes	Permite ver las solicitudes a estudiar correspondientes al punto casos estudiantes.

Aprobar	Acceder al servicio para estudiar las solicitudes de este punto.
Aplazar	Acceder al servicio para estudiar las solicitudes de este punto.
Devolver	Acceder al servicio para estudiar las solicitudes de este punto.
Negar	Acceder al servicio para estudiar las solicitudes de este punto.
Ver comentarios	Revisar los comentarios que los miembros del consejo han hecho a una solicitud de estudiante.

Fuente: Autor.

4.1.2.3 Mantenimiento y Administración del portal EISIWEB.

Tabla 13. Casos de uso: Mantenimiento y Administración del portal EISIWEB.

TÍTULO	FUNCIONES PRIMARIAS
Administrar Servicios	Crear, modificar y eliminar servicios del portal. Establecer parámetros.
Administrar Sugerencias	Revisar todas las sugerencias para atender o eliminar.
Administrar Varios	Ejecutar labores de mantenimiento que no clasifican en otras categorías. Llevar el contador de visitas a cero. Insertar, cambiar y eliminar banner.
Administrar Foros	Ver información y eliminar conversaciones de los usuarios en foros.
Administrar Usuarios	Crear usuarios manualmente. Ver información de usuarios. Cambiar parámetros.

	<p>Cambiar clave.</p> <p>Autorizar servicios a categorías y perfiles.</p> <p>Crear, modificar, eliminar categorías y perfiles.</p> <p>Enviar Correos.</p>
Administrar Ayudas	Crear, modificar y eliminar las ayudas en el portal de ayudas de las escuelas.
Administrar Parámetros	Crear, modificar y eliminar los diferentes parámetros del portal.
Backups	Generar copias de la base de datos División y Diamante.
Administrar Archivos	Ver información, descargar y eliminar archivos cargados por los usuarios.
Administrar Horarios	Cambiar, restaurar, ver incoherencias y cambios de horarios. Crear grupo de materia.
Administrar Actualizaciones	Ver historial y estadísticas de cambios del portal web.
Actualizar Base de Datos	Actualizar tablas de datos de División y Diamante. Ver contadores de registros.

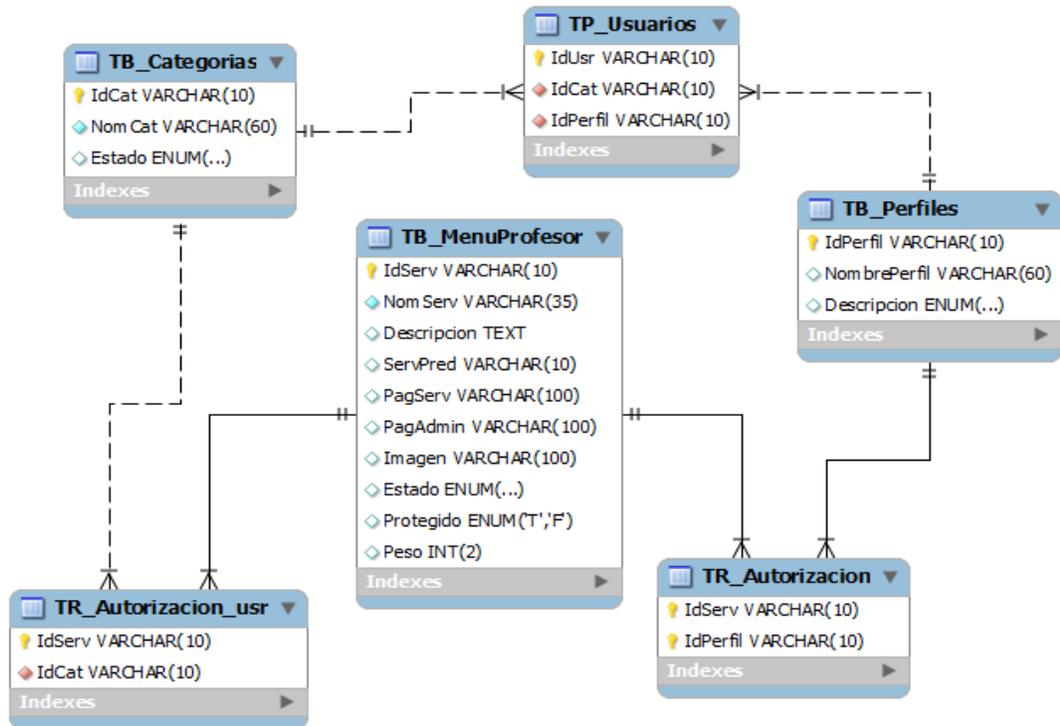
Fuente: Autor.

4.1.3 Diseño y Análisis

4.1.3.1 Diagrama Entidad/Relación de los servicios desarrollados

- **Portal Profesor:**

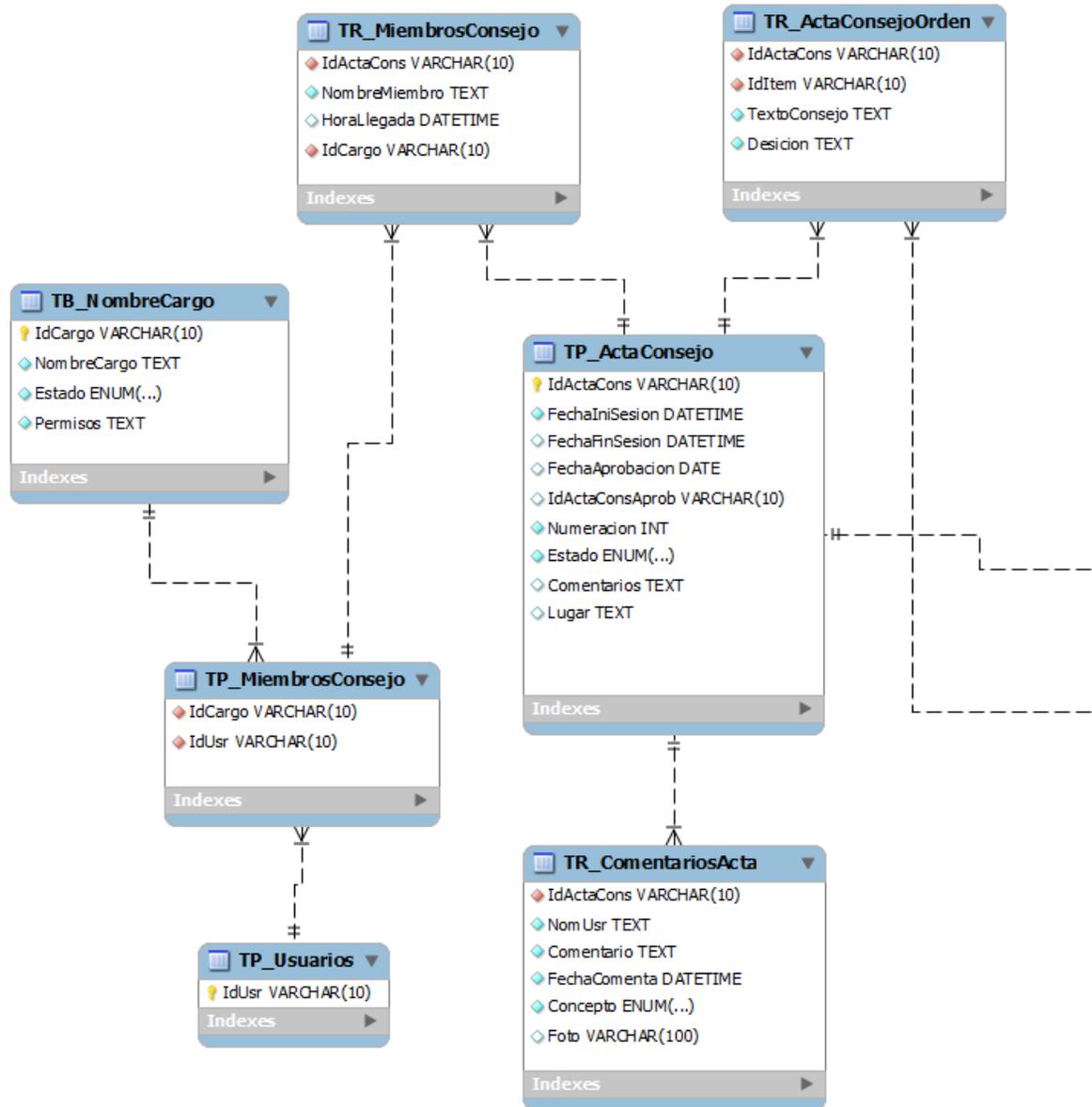
Figura 18. Diagrama E/R: Portal Profesor



Fuente: Autor.

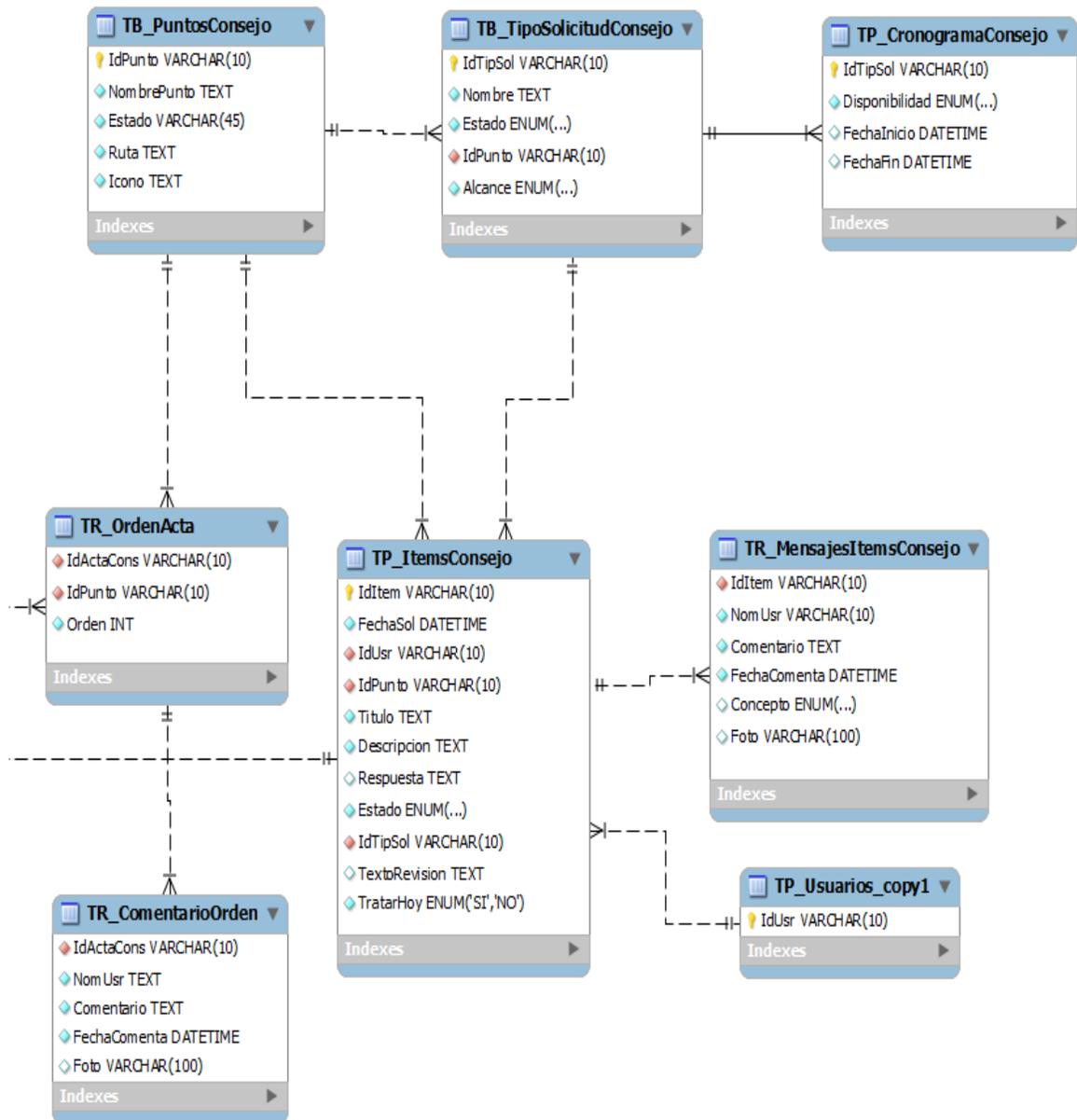
- Servicio de Actas de Consejo de Escuela.

Figura 19. Diagrama E/R: Actas Consejo de Escuela



Fuente: Autor.

Figura 20. Continuación ddiagrama E/R: Actas Consejo de Escuela



Fuente: Autor.

4.1.3.2 Descripción de las Entidades

Definición de las entidades utilizadas y creadas en la base de datos Poseidón ubicada en el servidor de la escuela de Ingeniería Civil y de la base de datos Diamante ubicada en el servidor de los portales de las escuelas GEOWEB y E3TWeb.

Tabla 14. Descripción de las Entidades

ENTIDAD	DESCRIPCIÓN
TB_Categorias	Contiene las categorías que puede tener un usuario.
TB_MenuProfesor	Contiene los servicios para el portal del profesor
TB_Perfiles	Contiene los perfiles para los usuarios administradores.
TP_Usuarios	Contiene la información de todos los usuarios registrados en el portal.
TR_Autorizacion	Contiene los servicios permitidos para los usuarios con perfil de administrador.
TR_Autorizacion_usr	Contiene los servicios permitidos para los usuarios normales.
TB_NombreCargo	Contiene los nombres de los cargos de los miembros del consejo.
TB_PuntosConsejo	Contiene los puntos que se pueden desarrollar en una sesión de consejo de Escuela
TB_TipoSolicitudConsejo	Contiene las solicitudes que se pueden presentar ante el consejo de Escuela.

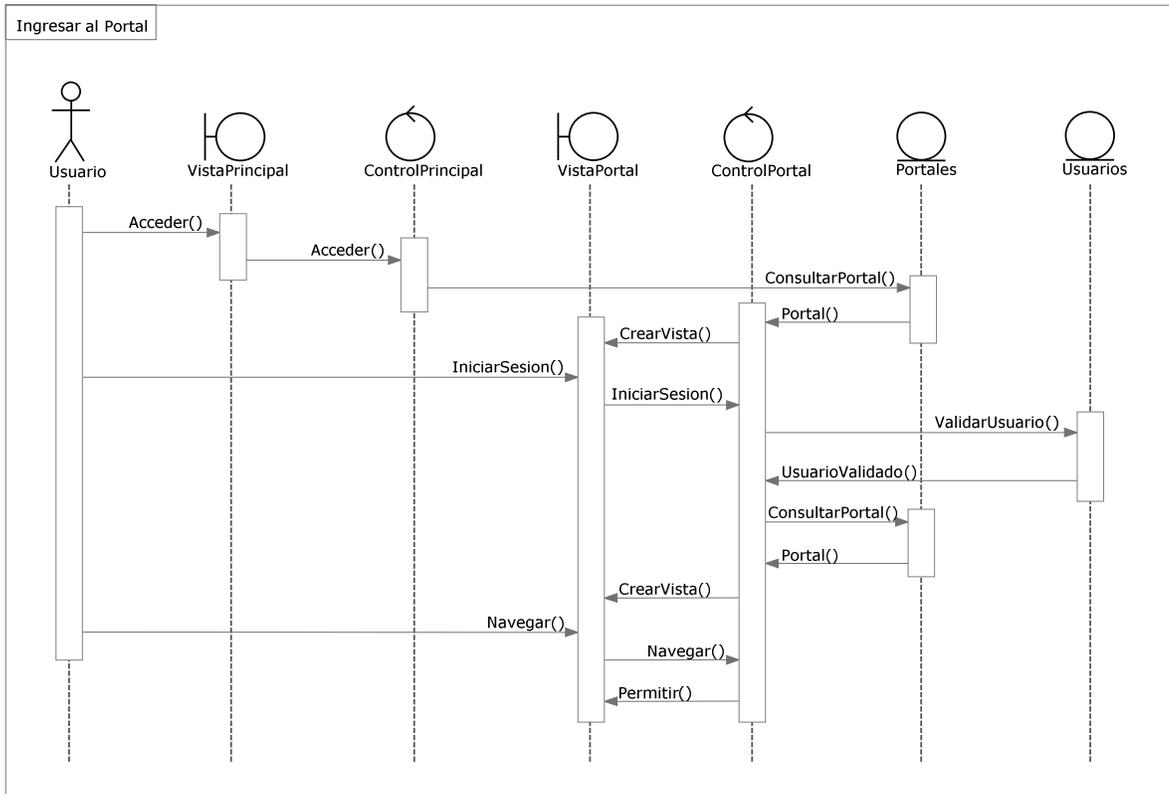
TP_ActaConsejo	Contiene las actas de consejo de cada una de las sesiones.
TP_CronogramaConsejo	Contiene las fechas en las cuales estará activa una solicitud.
TP_ItemsConsejo	Contiene las solicitudes presentadas por los usuarios ante el consejo de Escuela.
TP_MiembrosConsejo	Contiene los miembros del consejo de Escuela.
TR_ActaConsejoOrden	Relaciona un Acta de consejo con una o más solicitudes.
TR_ComentariosActa	Contiene los comentarios que se han hecho a cada Acta.
TR_ComentarioOrden	Contiene los comentarios que se han hecho a cada orden del día.
TR_MensajesItemsConsejo	Contiene los comentarios que se han hecho a cada solicitud hecha.
TR_MiembrosConsejo	Relaciona los miembros del consejo con su participación en un acta.
TR_OrdenActa	Contiene el orden del día de cada acta.

Fuente: Autor.

4.1.4 Modelo de Procesos del Sistema

4.1.4.1 Portal de Profesor

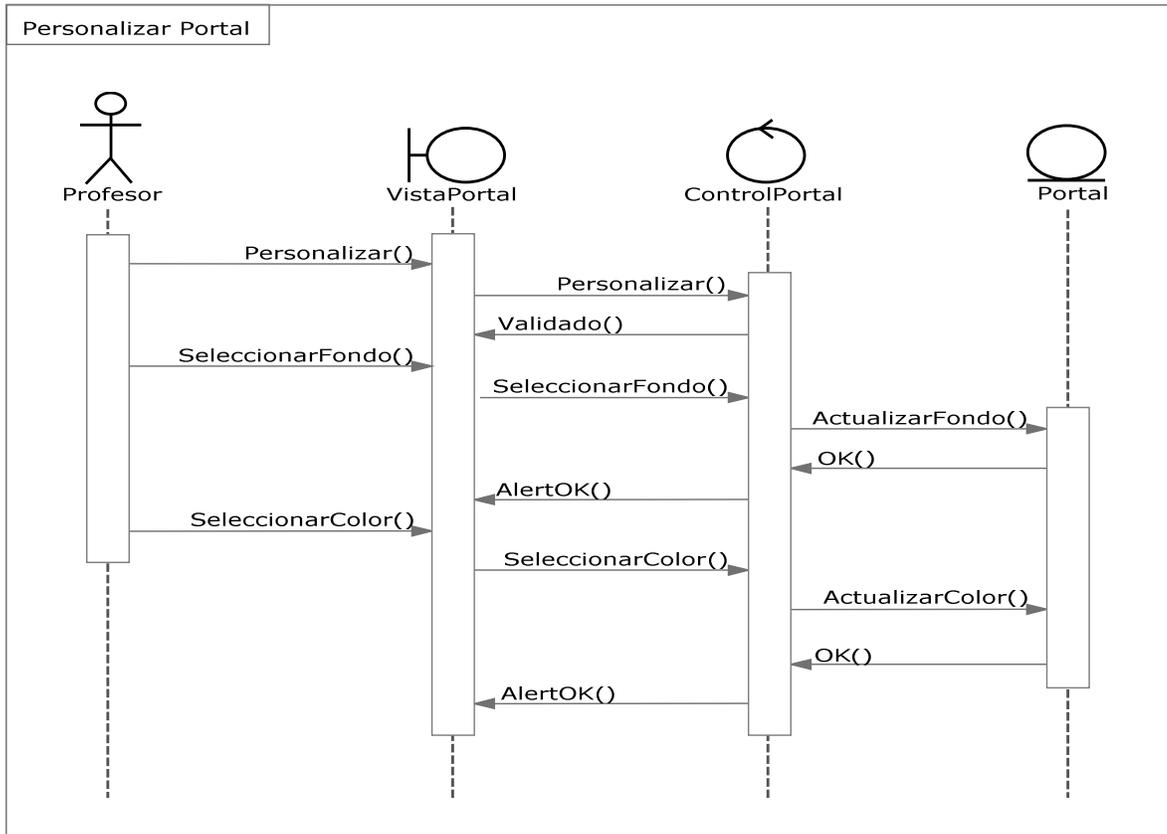
Figura 21. Diagrama de secuencia: Ingresar al portal



Fuente: Autor.

1. El Usuario solicita acceder al portal en la vista VistaPrincipal.
2. El control ControlPrincipal recibe la petición, consulta en base de datos la información del portal y crea la vista VistaPortal.
3. El usuario solicita iniciar sesión en la vista VistaPortal. El control ControlPortal recibe la petición y valida la información ingresada con la base de datos.
4. El control ControlPortal valida si el usuario sesionado tiene portal y crea la vista VistaPortal.
5. El usuario solicita navegar en la vista VistaPortal. El control ControlPortal recibe la petición, valida los permisos y permite la navegación.

Figura 22. Diagrama de secuencia: Personalizar portal

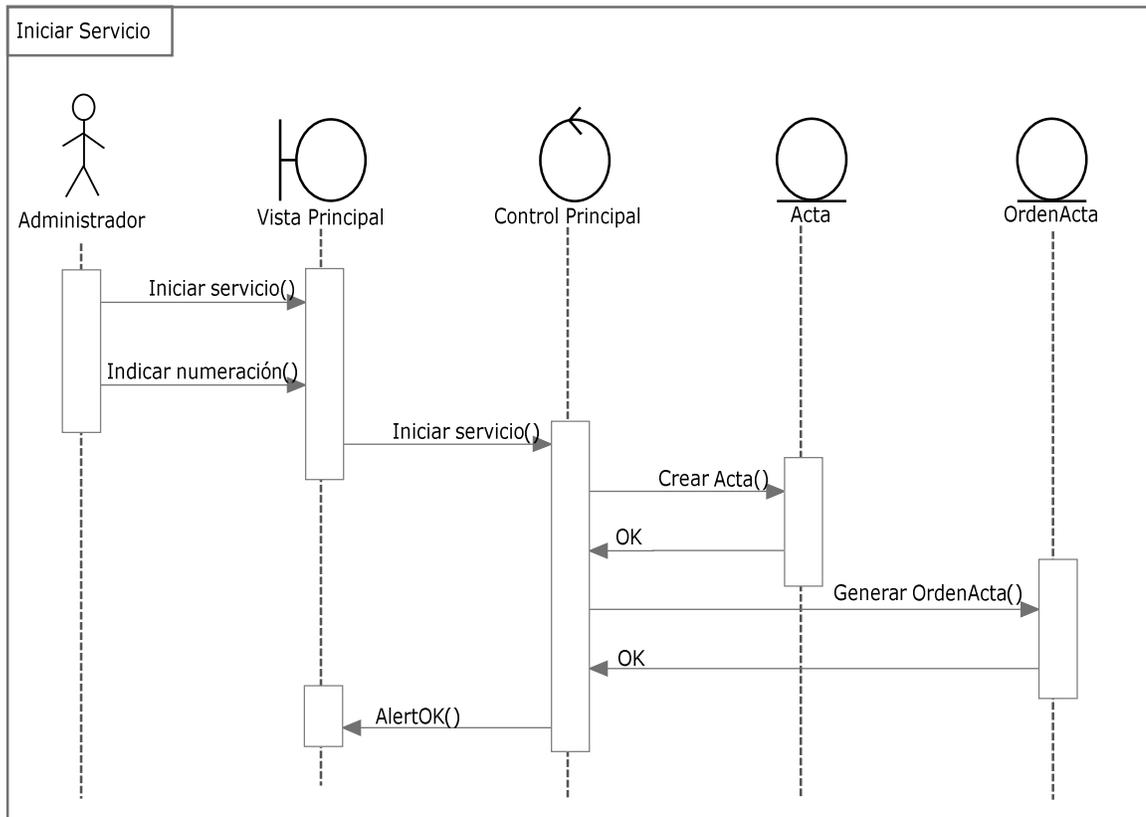


Fuente: Autor.

1. El Usuario solicita personalizar en la vista VistaPortal.
2. El control ControlPortal recibe la petición, valida permisos y permite ver el módulo de personalización en la VistaPortal.
3. El usuario selecciona el fondo y solicita guardar. El ControlPortal recibe la petición y guarda el fondo en la base de datos.
4. El ControlPortal indica al usuario que se ha guardado el fondo mediante la VistaPortal.
5. El usuario selecciona el color y solicita guardar. El ControlPortal recibe la petición y guarda el color en la base de datos.
6. El ControlPortal indica al usuario que se ha guardado el color mediante la VistaPortal

4.1.4.2 Servicio de Actas de Consejo de Escuela

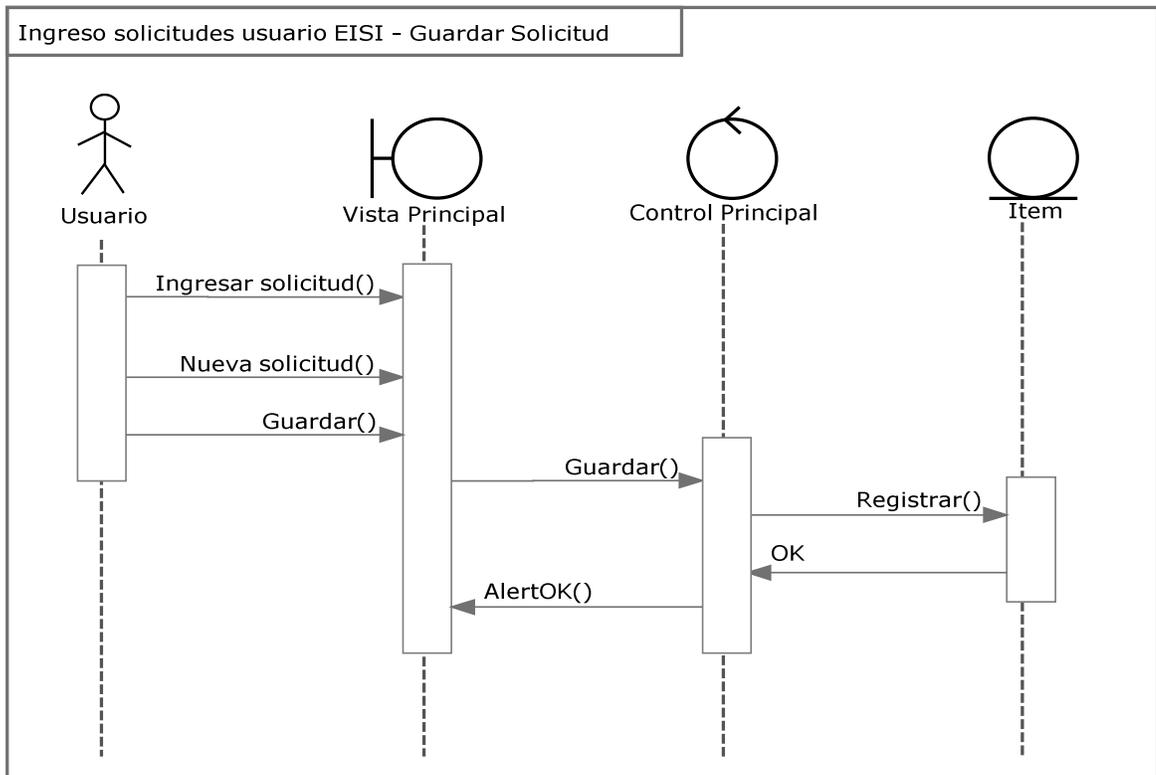
Figura 23. Diagrama de secuencia: Iniciar Servicio.



Fuente: Autor.

1. El Administrador solicita iniciar el servicio en la vista VistaPrincipal.
2. El Administrador indica la numeración de Acta y presiona Iniciar.
3. El ControlPrincipal recibe la petición de iniciar el servicio y crea la primer Acta con la numeración indicada por el usuario. Se verifica que se creó el Acta.
4. El ControlPrincipal genera un Orden del día para el Acta recién creada y envía una alerta de OK al usuario mediante la VistaPrincipal.

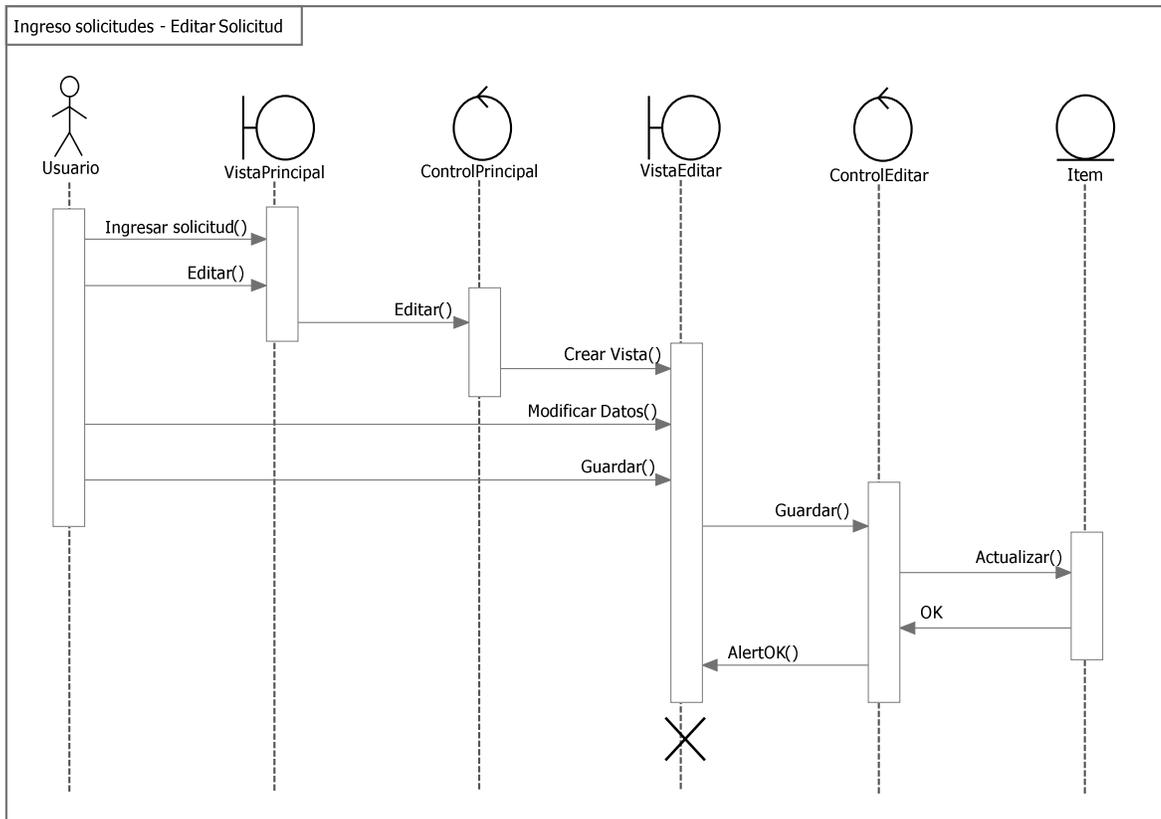
Figura 24. Diagrama de secuencia: Ingreso solicitudes usuario EISI – Guardar solicitud



Fuente: Autor.

1. El Usuario solicita Nueva solicitud en la vista VistaPrincipal.
2. El usuario ingresa los datos y archivos adjuntos necesarios y los guarda.
3. El controlador ControlPrincipal recibe la petición y hace la inserción en Base de datos.
4. El ControlPrincipal retorna al usuario una alerta de éxito en la operación mediante la VistaPrincipal.

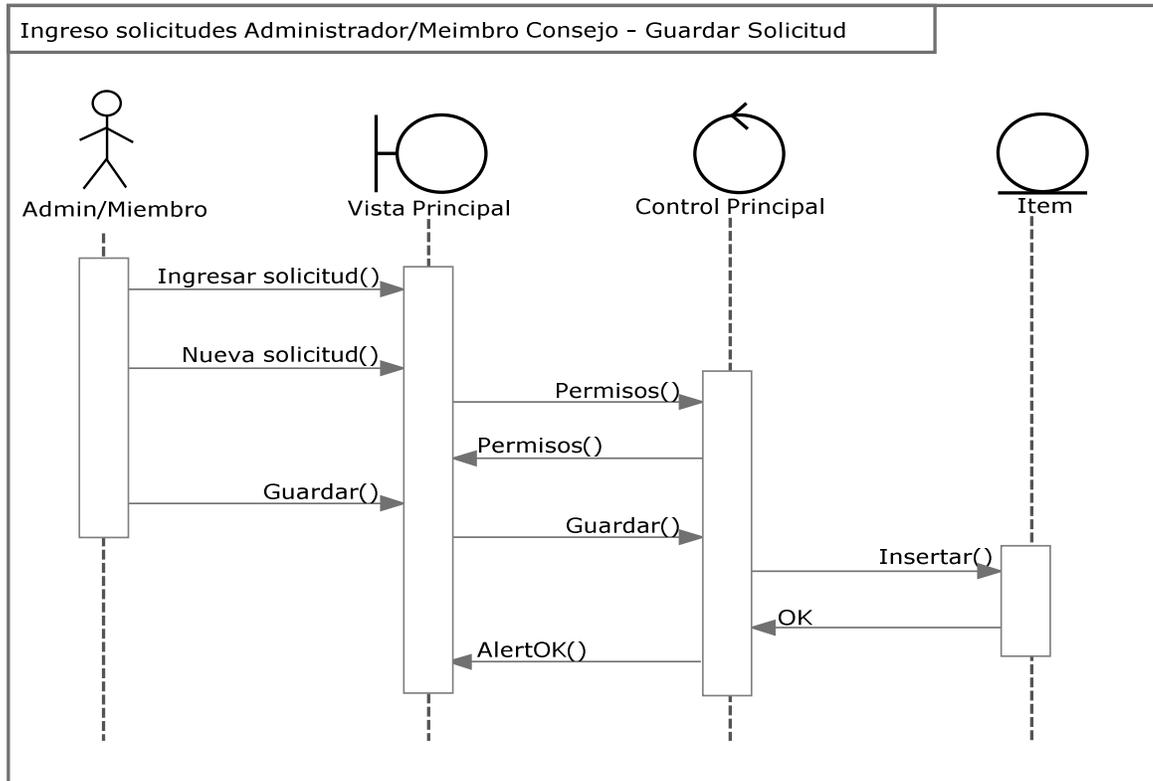
Figura 25. Diagrama de secuencia: Ingreso solicitudes – Editar solicitud.



Fuente: Autor.

1. El Usuario solicita editar una solicitud en la vista VistaPrincipal.
2. El controlador ControlPrincipal recibe la petición, verifica permisos y despliega la vista VistaEditar.
3. El Usuario modifica la información y guarda los cambios.
4. El controlador ControlEditar recibe la petición de guardar los cambios, actualiza el registro en base de datos y retorna un mensaje de Ok al usuario en la vista VistaEditar.
5. Se cierra la vista VistaEditar.

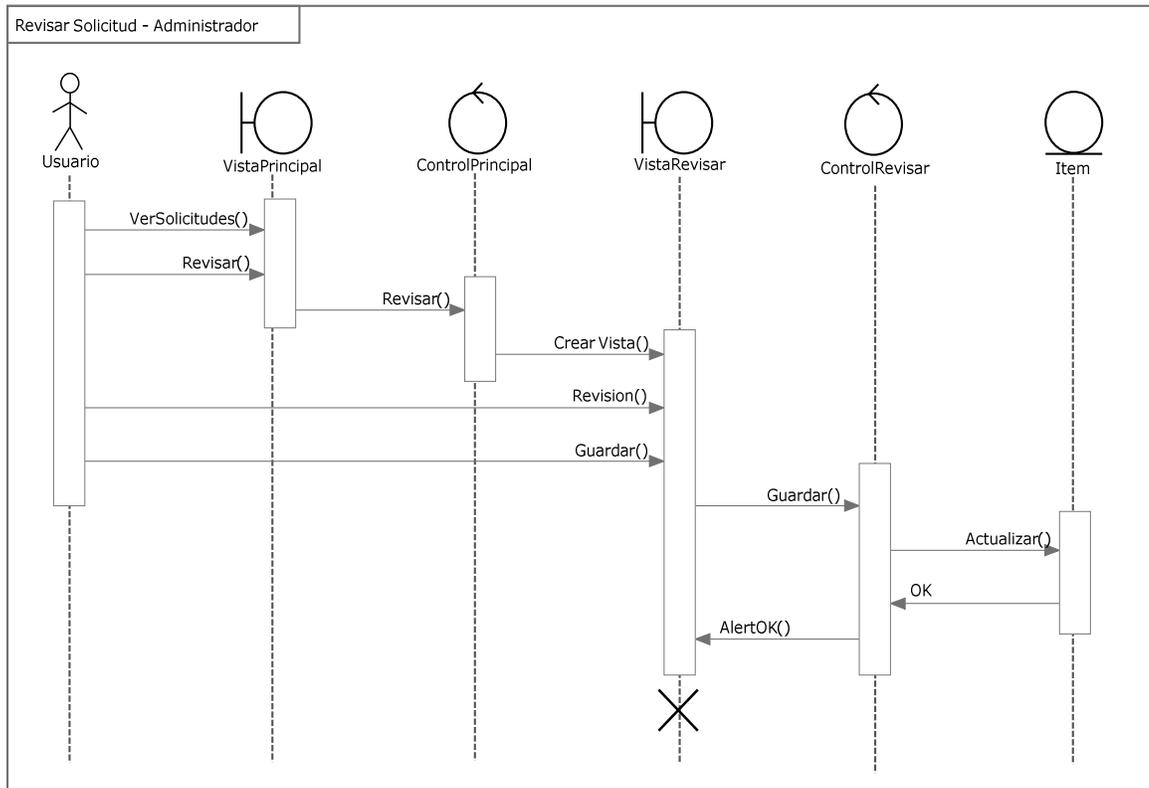
Figura 26. Diagrama de secuencia: Ingreso solicitudes Administrador – Guardar solicitud.



Fuente: Autor.

1. El Usuario solicita ingresar solicitud en la vista VistaPrincipal.
2. El usuario solicita nueva solicitud en la VistaPrincipal.
3. El controlador ControlPrincipal verifica los permisos del usuario y le indica a la VistaPrincipal que solicitudes puede ingresar.
4. El Usuario ingresa los datos y archivos adjuntos de su solicitud y los guarda.
5. El controlador ControlPrincipal recibe la petición, valida la información y la guarda en la base de datos.
6. La base de datos indica el resultado de la transacción y el ControlPrincipal la muestra al Usuario mediante la VistaPrincipal.

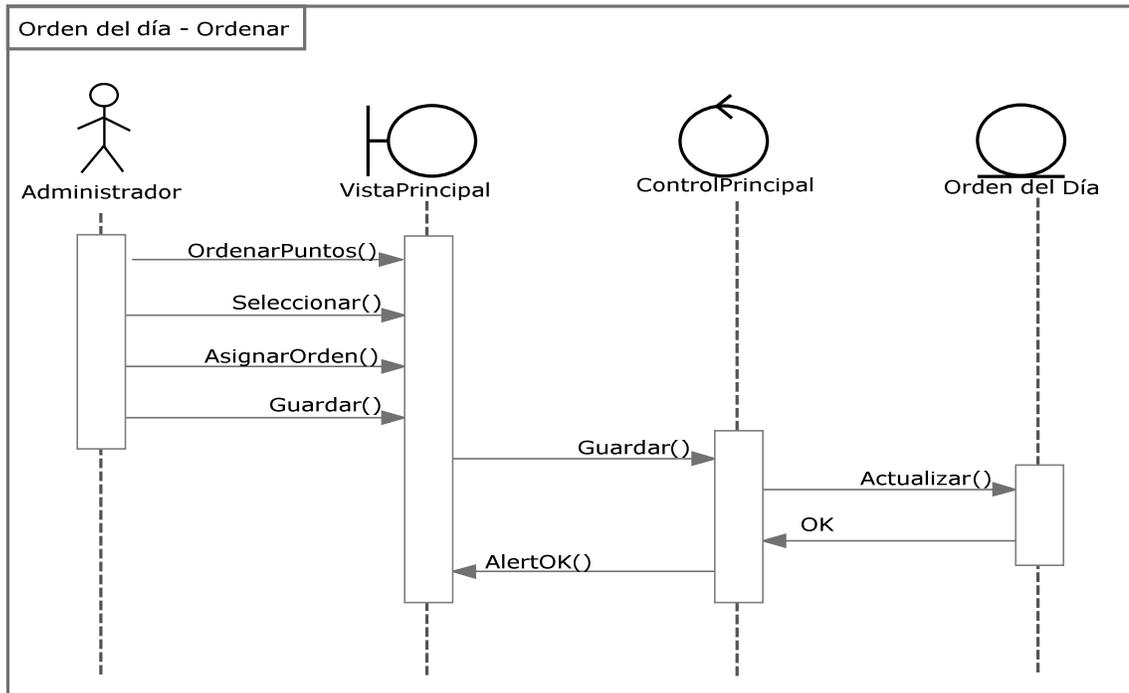
Figura 27. Diagrama de secuencia: Revisar solicitud.



Fuente: Autor.

1. El Usuario solicita ver solicitudes en la vista VistaPrincipal.
2. El Usuario indica cual solicitud desea revisar en la vista VistaPrincipal. El control ControlPrincipal recibe la petición y crea la vista VistaRevisar.
3. El Usuario revisa y comenta la solicitud. Luego solicita guardar en la vista VistaRevisar.
4. El ControlRevisar recibe la petición y actualiza la solicitud en base de datos.
5. La base de datos retorna la respuesta al control ControlRevisar y este indica al usuario el resultado.
6. Se cierra la vista VistaRevisar y se actualiza la vista VistaPrincipal.

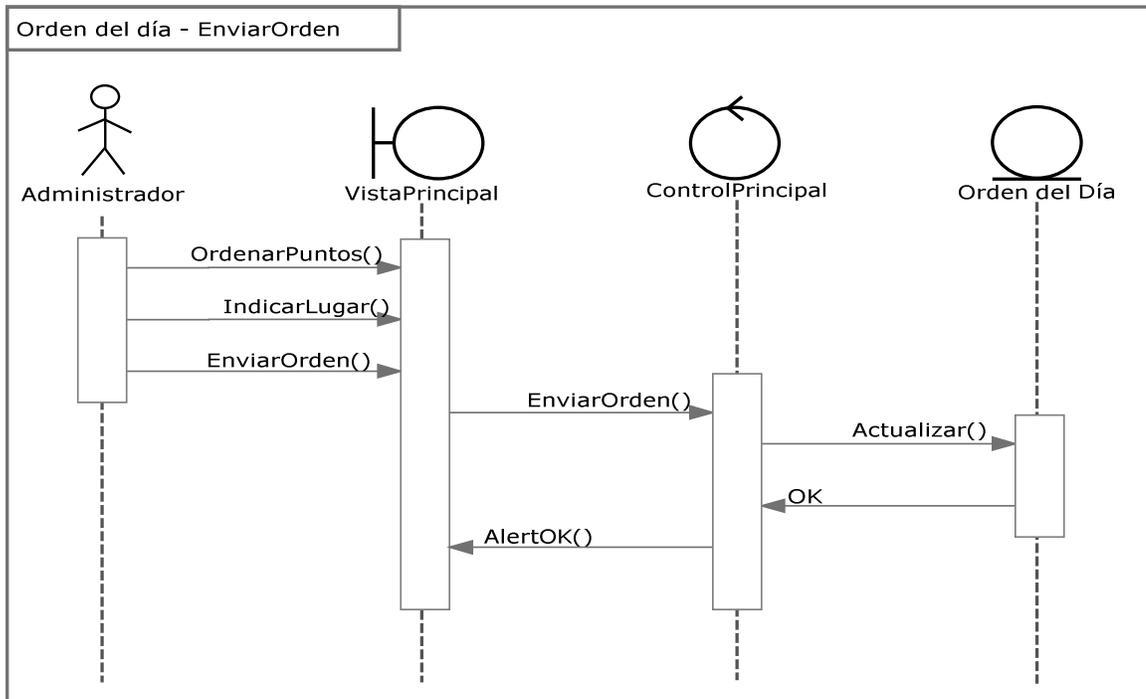
Figura 28. Diagrama de secuencia: Orden del día - Ordenar.



Fuente: Autor.

1. El Usuario solicita ordenar puntos en la vista VistaPrincipal.
2. El Usuario indica los puntos y el orden de los mismos en la VistaPrincipal.
3. El Usuario selecciona guardar orden en la vista VistaPrincipal. EL control ControlPrincipal recibe la petición.
4. El ControlPrincipal guarda el orden en la base de datos, recibe la respuesta y se le indica el resultado al Usuario en la VistaPrincipal.

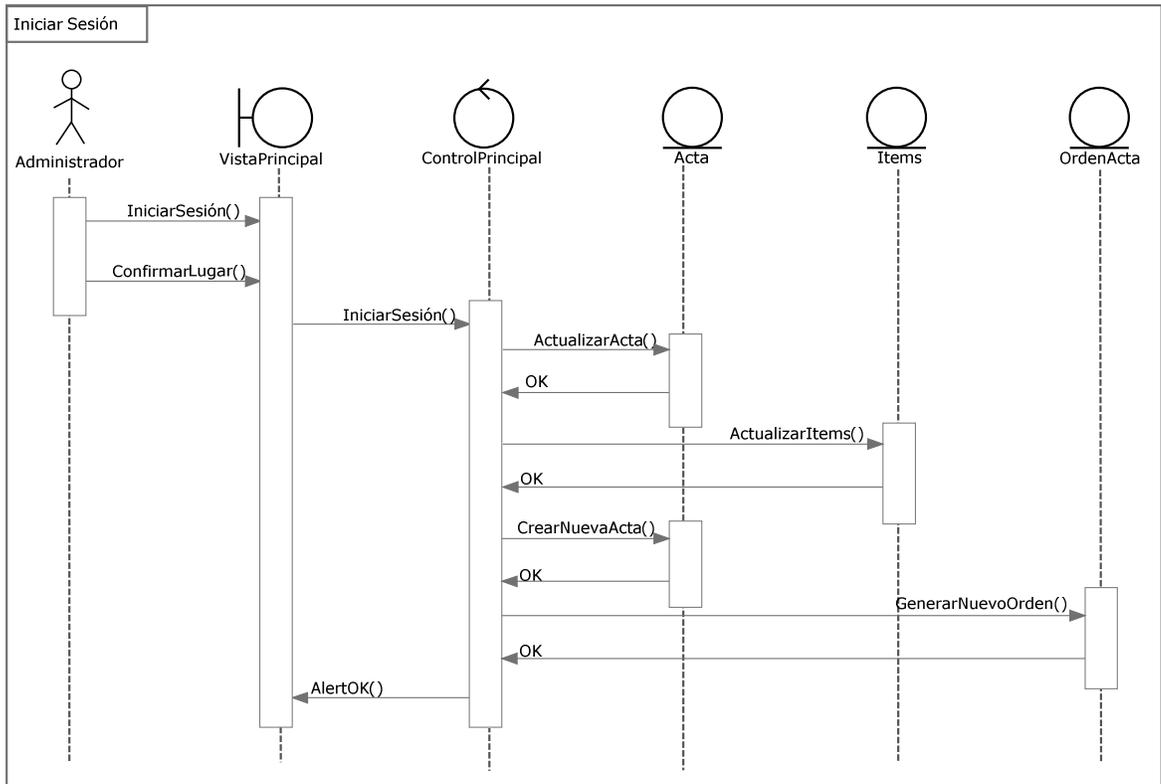
Figura 29. Diagrama de secuencia: Orden del día – Enviar Orden.



Fuente: Autor.

1. El Usuario solicita ordenar puntos en la vista VistaPrincipal.
2. El Usuario indica el lugar y selecciona enviar orden en la vista VistaPrincipal.
3. El control ControlPrincipal recibe la petición, guarda el orden en la base de datos y envía un correo a los miembros del Consejo indicando el orden del día y el lugar de la próxima sesión.
4. El control ControlPrincipal indica al usuario el resultado de la operación mediante la vista VistaPrincipal.

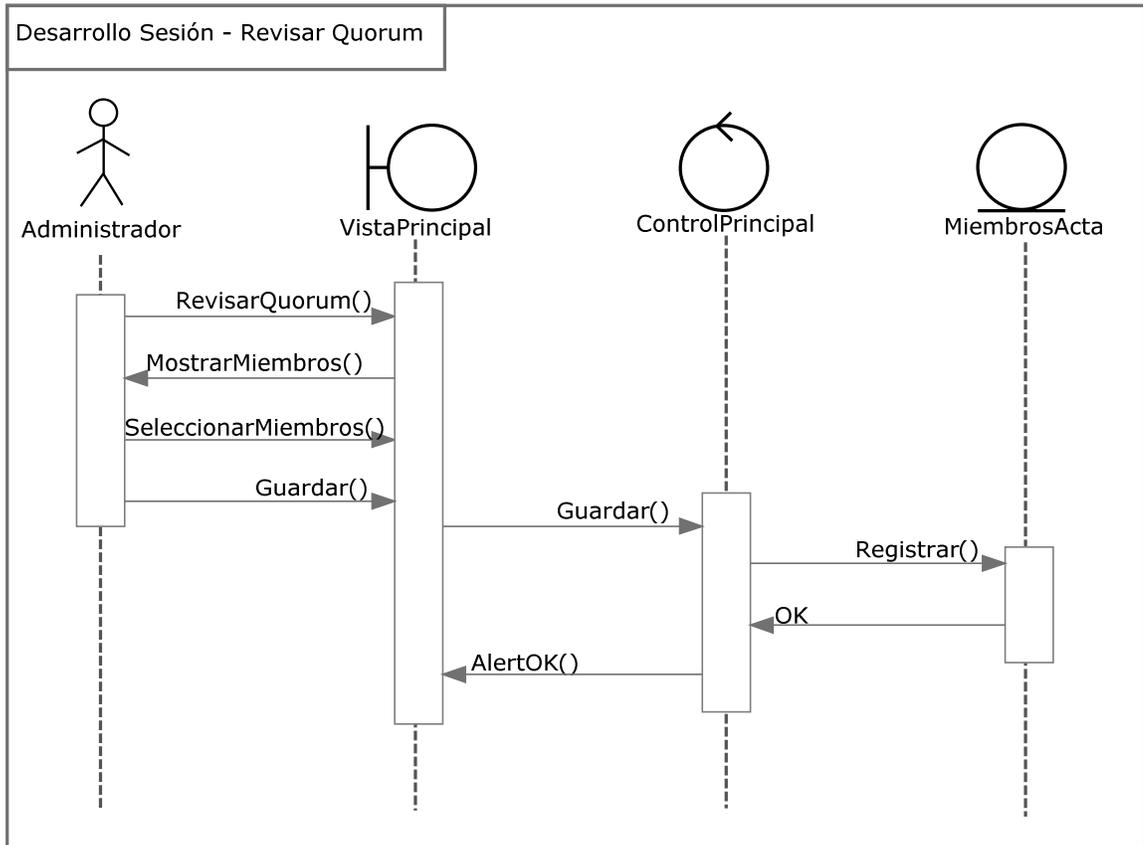
Figura 30. Diagrama de secuencia: Iniciar sesión Consejo.



Fuente: Autor.

1. El Usuario solicita iniciar sesión en la vista VistaPrincipal.
2. El Usuario confirma el lugar de reunión y selecciona iniciar.
3. El control ControlPrincipal recibe la petición y actualiza el Acta en base de datos.
4. El control ControlPrincipal actualiza los Ítems a tratar en la sesión de Consejo iniciada.
5. El control ControlPrincipal crea una nueva Acta para la próxima sesión y genera su respectivo orden del día.
6. El control ControlPrincipal indica al usuario que se iniciado la sesión mediante la VistaPrincipal.

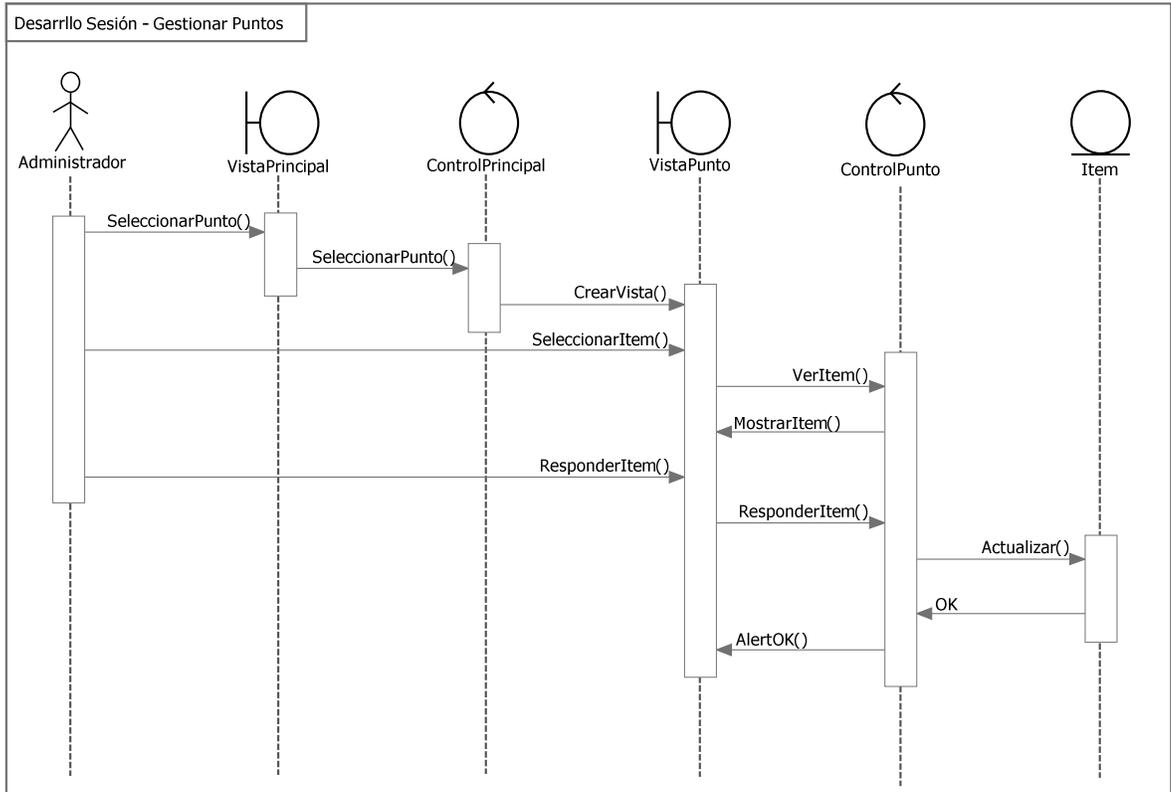
Figura 31. Diagrama de secuencia: Desarrollo sesión – Revisar Quórum.



Fuente: Autor.

1. El Usuario solicita revisar quórum en la vista VistaPrincipal, donde se muestran los miembros actuales.
2. El Usuario selecciona los miembros y solicita guardar en la VistaPrincipal. El control ControlPrincipal recibe la petición y registra en base de datos los miembros seleccionados.
3. El control ControlPrincipal recibe la respuesta y la comunica al Usuario mediante la vista VistaPrincipal.

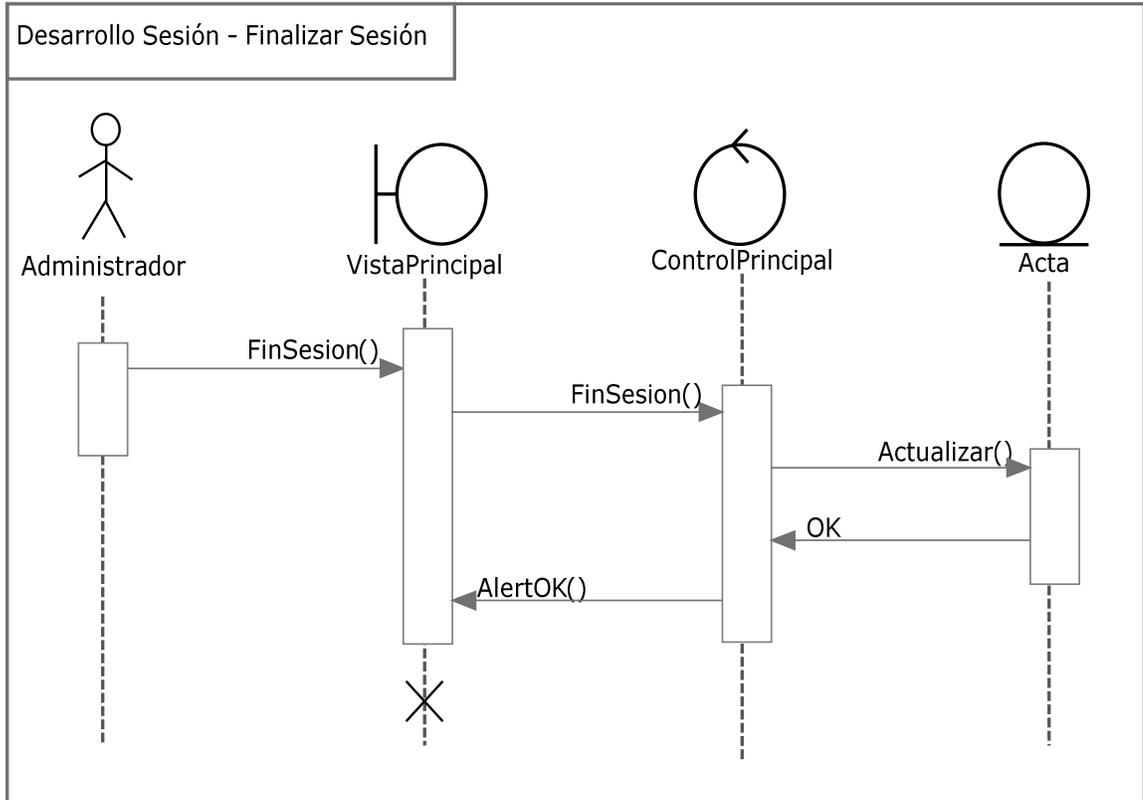
Figura 32. Diagrama de secuencia: Desarrollo sesión – Gestionar puntos.



Fuente: Autor.

1. El Usuario solicita el punto que desea revisar en la vista VistaPrincipal.
2. El control ControlPrincipal recibe la petición y crea la vista VistaPunto.
3. El Usuario selecciona el ítem que desea revisar en la vista VistaPunto, el control ControlPunto recibe la petición y muestra el ítem al Usuario.
4. El Usuario selecciona una decisión, escribe un texto (opcional) y selecciona guardar. El control ControlPunto recibe la petición y guarda la respuesta en la base de datos.
5. El control ControlPunto recibe la respuesta y la comunica al Usuario mediante la vista VistaPunto.

Figura 33. Diagrama de secuencia: Desarrollo sesión – Finalizar sesión.



Fuente: Autor.

1. El Usuario solicita finalizar sesión en la vista VistaPrincipal.
2. El control ControlPrincipal recibe la petición y guarda la respuesta en la base de datos.
3. El control ControlPrincipal recibe la respuesta de la base de datos y la comunica al Usuario mediante la vista VistaPunto, luego se cierra la vista VistaPrincipal.

4.1.5 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 Escuela de Ingeniería Civil. 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 los resultados correspondieran 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.

4.2 MANTENIMIENTO Y ADMINISTRACIÓN

4.2.1 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 y que se corrigen durante la primera fase como integrantes del Grupo Calumet. Las siguientes son las incidencias que se realizaron paralelamente al desarrollo de este proyecto.

- **Foros comunidad y aula, correos, orden y permisos.**

Funcionalidad:

El Portal de las Escuelas soporta los debates y conversaciones a través de los foros. Estos foros están abiertos para la comunidad, en general, como también para el Aula Virtual y los grupos de la Escuela..

Desarrollo del problema:

Para los Foros del Aula se crearon dos botones para invertir el orden en que se muestran los mensajes. Cada botón envía un parámetro diferente para ver el orden de los mensajes.

Para los Foros de la Comunidad se crearon dos botones para invertir el orden en que se muestran los mensajes. Cada botón envía un parámetro diferente para llamar un método diferente y ordenar los mensajes.

Al crear una nueva conversación se llama una nueva función que envía una petición Ajax para enviar correos a los Usuarios y al director del aula en el que se crea la conversación.

Problemática:

En los foros abiertos (no evaluables) dentro del Aula Virtual al crear una conversación no se está enviando correos a los directores de la asignatura, además no se está comprobando sesión ni permiso de servicio al abrir y/o opinar (hay que revisar por el lado del estudiante, como por el profesor).

Además es necesario invertir el orden cronológico con que se muestran las participaciones, es decir, el último en opinar es el primero en salir. Este arreglo hay que hacerlo para los foros de la comunidad, como los del Aula tanto evaluables como abiertos.

Descripción:

Para los foros del Aula en los archivos `/eisi/web/Aula/ForosAula/ForosMensajesGrupoAula.jsp` y `/eisi/web/Aula/ForosAula/ForosMensajesGuionAula.jsp` según la bandera que envíe cada botón se ordena la consulta de los mensajes de manera diferente.

Para los Foros de la Comunidad en el archivo `/eisi/web/Foros/forosMensajesS.jsp` se crearon dos botones y cada uno consulta en el bean

/eisi/src/java/beans/Mensajes.java la función respectiva. En este bean se creó una nueva función llamada consultarMensajesDesc.

Para enviar correos al crear una nueva conversación dentro del Aula, en el archivo /eisi/web/Aula/ForosAula/ForosCrearAula.jsp se creó la función Correos() que envía una petición Ajax al archivo /eisi/web/Aula/ForosAula/EnviarCorreoCrearAula.jsp donde se envían correos a los respectivos usuarios del Aula.

- **Revisar visibilidad en la cartelera del Aula.**

Funcionalidad:

Dentro del Aula Virtual se ha creado una cartelera que permite a profesores y estudiantes dejar mensajes que podrán leer al momento de ingresar. Estos mensajes están dirigidos a todos los grupos relacionados con el aula, o solo a un grupo, o solo al profesor.

Desarrollo del Problema:

Se corrigió la visibilidad de una publicación de manera que solo la vea el grupo al cual está dirigida, además se le dio visibilidad al modificar una publicación. Esto se hizo por el lado del estudiante y del profesor.

Problemática:

Al publicar en cartelera del aula virtual a un grupo determinado, este lo está habilitando a otros. Es necesario revisar que cada grupo vea lo que le corresponde. También revisar publicación para el profesor, que solo él lo vea.

Descripción:

En el archivo /eisi/web/Aula/Clase/FormModificar.jsp se recoge el parámetro grupMater enviado desde /eisi/web/Aula/Clase/CarteleraAulaVirtual.jsp y se inserta el campo "Grupo" en la actualización de la tabla TP_CarteleraAulaVirtual cuando se cumple VisualizarA.equals("Grupo"). Esto es para el lado del estudiante.

En el archivo /eisi/web/Aula/GuiarClase/FormModificarProf.jsp se recoge el parámetro grupMater enviado desde /eisi/web/Aula/GuiarClase/CarteleraAulaVirtualProf.jsp y se inserta el campo "Grupo" en la actualización de la tabla TP_CarteleraAulaVirtual cuando se cumple VisualizarA.equals("Grupo"). Esto es para el lado del estudiante.

- **Detención automática y correos del Aula virtual.**

Funcionalidad:

El Aula Virtual ha sido diseñada para que corra automáticamente semana a semana sin la intervención del profesor. Esto permite ir cerrando y abriendo actividades sin la intervención del profesor. Al llegar a la semana final el Aula debe de detenerse automáticamente.

Desarrollo del Problema:

Ahora se envía correo al director cuando un Aula se detiene por falta de uso o por terminar las semanas, además se informa cuando inicia una nueva semana.

Problemática:

Un Aula de Clase no se detuvo al llegar a la semana final, sino que continuó en la siguiente semana. Además es necesario avisar a los profesores con un correo de inicio de una nueva semana, o de la detención por llegar a la última semana. Ya existe un correo para cuando han transcurrido más de 13 o 14 días sin acciones.

Descripción:

En el archivo /eisi/web/Aula/AccederAula.jsp se actualiza la tabla TP_GuionClase y se pone el Estado=Detenido cuando han pasado más de 14 días o ha terminado la semana final. Si pasa una nueva semana, se actualiza la tabla TP_GuionClase y se aumenta en 1 la Semana Actual. Para cada una de estas situaciones se envía correo a los directores informando lo acontecido.

- **Participaciones en Foros del Aula virtual.**

Funcionalidad:

Dentro del Aula existen Foros evaluables y no evaluables. Estas actividades permiten que el profesor interactúe con sus estudiantes acerca de un tema específico. Los profesores pueden entrar y revisar las participaciones de los estudiantes.

Desarrollo del Problema:

Ahora se envía correo al director cuando un Aula tiene participaciones nuevas en algún foro.

Problemática:

Los profesores y directores no se enteraban si los estudiantes habían participado en los foros.

Descripción:

En el archivo /eisi/web/Aula/AccederAula.jsp se ejecuta automáticamente una función que envía una petición Ajax a un archivo creado llamado EnviarCorreoCrearAula.jsp, donde se revisan todas las Aulas actualmente creadas y se revisa si en sus foros existen participaciones que no han enviado correo. En un correo se informa a los directores de un Aula las participaciones hechas en días anteriores.

4.2.2 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.

4.2.3 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.
- Realizar copias de seguridad diarias 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 servicios realizados, se realizaron las siguientes pruebas:

5.1 PRUEBAS DE VERIFICACIÓN

Esta prueba es una de las más prácticas y más utilizadas mediante la cual se aplican diferentes técnicas para detectar errores en el sistema antes de ser utilizado.

Se efectúa elaborando actividades para cada servicio desarrollado, realizando pruebas como: verificar que los campos que contienen datos obligatorios no queden vacíos, o examinar que los campos contengan el tipo de dato que le corresponde.

5.1.1 Pruebas por componente

Esta prueba se realizó para cada caso de uso de cada servicio desarrollado y descrito anteriormente. De igual manera en las mejoras a los servicios ya existentes.

Los resultados obtenidos con la realización de las pruebas, evidencian el cumplimiento de los objetivos propuestos, en la medida en que los servicios y mejoras funcionan correctamente. (Ver anexo E).

5.2 PRUEBAS DE INTEGRACIÓN

Una vez creados todos los servicios se verificó que cada uno funcionara correctamente y no interfirieran con el funcionamiento del portal. Se verificó que las consultas a la base de datos se hicieran de manera correcta, que en el sistema se visualizaran correctamente los servicios y que cada tipo de usuario tuviera acceso a ellos dependiendo del perfil y categoría al que está asociado.

5.3 PRUEBAS DE VALIDACIÓN

Con el fin de probar el funcionamiento de cada servicio implementado se realizaron pruebas de validación elaboradas por integrantes del grupo Calumet, administrativos de las escuelas y el director de proyecto. Para algunos de los servicios observamos irregularidades las cuales fueron corregidas durante la marcha hasta obtener los resultados requeridos.

6 CONCLUSIONES

- La implementación del Portal del Profesor se consolida como un espacio propio para el docente que permite agilizar sus labores académicas y administrativas gracias a que brinda de manera ordenada y entendible un menú de servicios dirigido a la labor docente.
- El servicio Gestor de Proyectos surge como herramienta de gran utilidad para el desarrollo y control de un proyecto, ya que permite llevar registro de las fechas y actividades a desarrollar dentro del mismo, lo cual facilita la visión y optimización del trabajo.
- El servicio de Actas de Consejo se propone como solución óptima, eficiente y ecológica para el manejo de procesos académicos y administrativos, dentro de una Escuela, debido a que permite sistematizar y tramitar la inscripción de solicitudes, pertinentes al Consejo de Escuela, hasta llegar a una instancia final dentro de su proceso.
- Los servicios web ofrecidos surgen como apoyo a la política de Gobierno denominada “Cero Papel”.
- El uso de nuevas tecnologías en el desarrollo de los servicios realizados, permite brindar una funcionalidad eficiente y competitiva en el control académico y administrativo dentro del portal.

7 RECOMENDACIONES

- Continuar y ampliar el servicio Actas de Consejo para llevarlo a las Facultades y unificar este proceso entre ellos y sus respectivas Escuelas.
- Crear los portales necesarios para navegar entre la Escuela, grupos y profesores.
- Integrar en el portal del profesor los grupos a que pertenece, así como su producción intelectual desarrollada.
- Ampliar las funcionalidades del gestor de proyectos para permitir reasignar fechas a las actividades.
- Para mantener una buena comunicación con los usuarios de los portales, es fundamental dar pronta y precisa respuesta a las sugerencias de los mismos. De esta manera se logra dar soporte oportuno a sus inquietudes.

BIBLIOGRAFÍA

1. JOHNSON, James. Bases de datos: Modelos lenguajes y diseño. 1ª ed. Oxford, 2000. Presenta temas de teoría de bases de datos: modelos y métodos de acceso, administración, diseño de aplicaciones.
2. PIATTINI, Mario, CALVO-MANZANO, José A., CERVERA, Joaquín, FERNANDEZ, Luis. Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Alfaomega, 2000. Este libro contiene información sobre técnicas para el buen modelado de aplicaciones informáticas.
3. SCHMULLER, JOSEPH. Aprendiendo UML en 24 horas. 1ª ed. México: Alhambra Mexicana S.A, 2000. En este libro se encuentra una guía muy práctica que permite conocer y entender sobre UML.
4. STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005. Este libro se ocupa de los conceptos completos de las características de los sistemas operativos.
5. STUMPF, Robert, TEAGUE, Lavette. Object-Oriented Systems Analysis and Design with UML. Prentice Hall. 2004. Este libro introduce los conceptos y métodos del análisis y diseño de sistemas orientados a objetos.
6. WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. Thomson International, 2003. En este libro se encuentra información sobre desarrollo de software orientado a objetos.
7. <http://api.jquery.com/>. Api de la librería jQuery.

8. <http://cormoran.uis.edu.co/eisi/Estandar>. Documentación de una librería del grupo para el diseño de interfaces.
9. <http://docs.oracle.com/javase/docs/api/>. Api de Java.
10. <http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor>. Información acerca de la Arquitectura Cliente Servidor.
11. <http://manuales.dgsca.unam.mx/jsp>. Manuales básicos de JSP.
12. <http://www.htmlquick.com/es/tutorials.html>, tutoriales de cómo programar en html de gran ayuda en detalles básicos y sencillos.
13. <http://www.mysql.com>. Sitio oficial de MySQL, que permite participar en capacitaciones en línea y tener las ultimas noticias acerca del producto para su mayor aprovechamiento.
14. <http://www.programacion.com/java/tutorial>. En este sitio se encuentran una gran variedad de tutoriales relacionados con: Los APIS, las herramientas y los servicios que proporciona los Servlets y las Java Server Pages (JSP), los tópicos necesarios para la programación de acceso a bases de datos en Java.

ANEXOS

Anexo A. ARQUITECTURA CLIENTE/SERVIDOR

1. Características de la arquitectura Cliente/Servidor

Las características básicas de una arquitectura Cliente / Servidor son:

- El proceso del cliente da la interface entre usuarios y el resto del sistema (Interacción con usuarios y el proceso del servidor), 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. 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:

- 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.

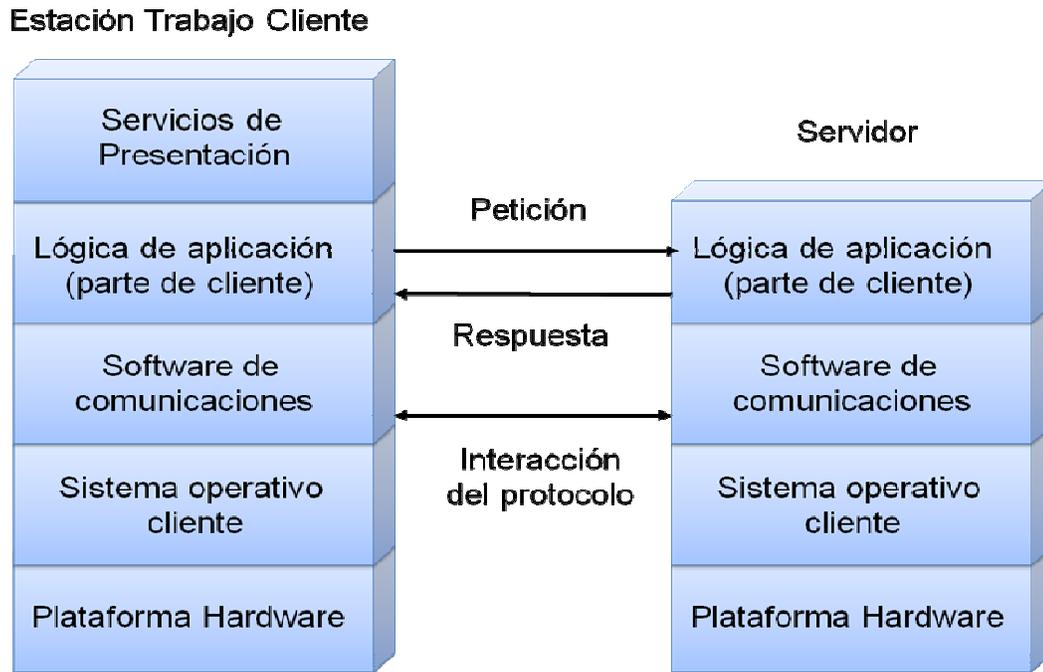
A continuación se muestra la clasificación de los sistemas Cliente / Servidor:

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 requiera 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.

Figura 34. Arquitectura Cliente/Servidor de Dos Capas



Fuente: Autor.

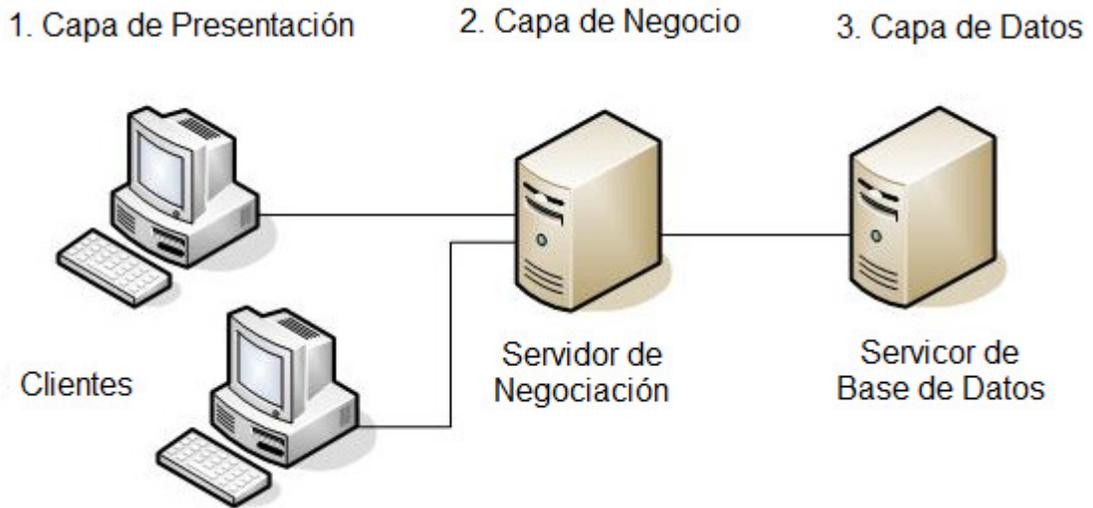
2.2. ARQUITECTURA CLIENTE/SERVIDOR DE TRES CAPAS

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.

Figura 35. Arquitectura Cliente/Servidor de Tres Capas



Fuente: Internet. http://www.ecured.cu/index.php/Archivo:Tres_capas.JPG

Anexo B. TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

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. 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. De 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.

Anexo C. BASES DE DATOS

1. Modelos de Bases De Datos

Este modelo describe la estructura de una base de datos, donde se incluye conceptos como: entidades, atributos y relaciones, la mayoría de los modelos de datos poseen un conjunto de operaciones básicas para especificar consultar y actualizaciones. Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. Algunos modelos utilizados con frecuencia son:

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.

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.

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. Acceso a Base de Datos

Para desarrollar aplicaciones que conecten bases de datos, se utilizan interfaces y programas estándar que envían demandas escritas en SQL, y procesan los resultados. Para conectarse a un motor de bases de datos determinado, se necesita un driver o una interfaz estándar que medie entre la aplicación y la base de datos.

3. Conectores más utilizados

- ODBC (Open DataBase Connectivity): Estándar de acceso a Bases de datos, su objetivo es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de

datos de la aplicación en comandos que el DBMS entienda. Para que esto funcione tanto la aplicación como el DBMS deben ser compatibles con ODBC.

- MDB: Esta aplicación permite trabajar con tablas de base de datos creadas en Access 97/2000. Es posible abrir tablas en SQL, visualizarlas, navegar, crear y borrar índices, fijar relaciones, copiar, etc.
- JDBC (Java Database Connectivity): Interfaz de programación de aplicaciones que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Para el desarrollo de los portales y cada uno de sus módulos se empleó el conector JDBC. Uno de sus mayores beneficios es su capacidad para crear aplicaciones cuya programación sea independiente de la base de datos, es decir, pueden ser migradas a otro servidor de bases de datos sin complicaciones. Sin embargo, dos elementos siguen estando ligados a una base de datos en particular, el nombre de la clase que se usa para cargar el controlador JDBC y la dirección URL (Universal Resource Locator) para acceder a la base de datos.

Los servlets y las páginas JSP usan JDBC habitualmente de la misma manera que cualquier otra aplicación en Java, típicamente los datos del controlador JDBC, la cadena de conexión, el nombre de usuario y contraseña para conectarse a la base de datos son codificados dentro del programa.

Las operaciones primordiales realizadas durante la ejecución del JDBC son:

- Cargar un controlador JDBC.
- Utilizar el controlador para abrir una conexión con la base de datos.

- Formular instrucciones SQL a través de la conexión.
- Procesar los conjuntos de resultados devueltos por las operaciones SQL.

4. Manejadores o Gestores de Bases de Datos

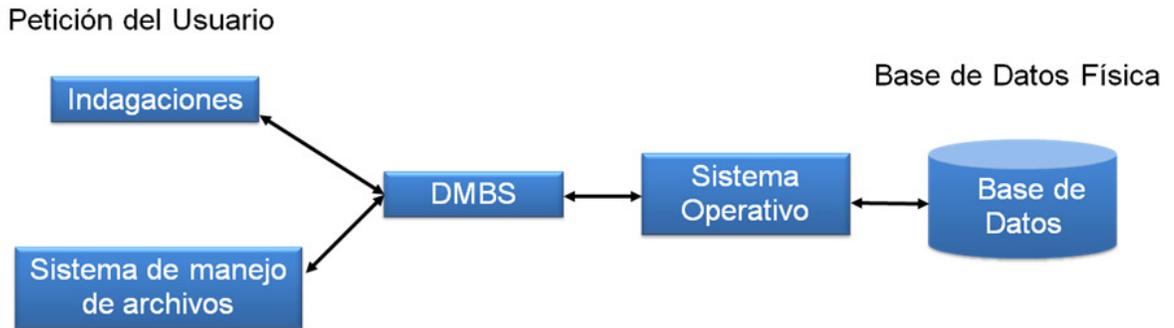
El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

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.

Figura 36. Función del DBMS.



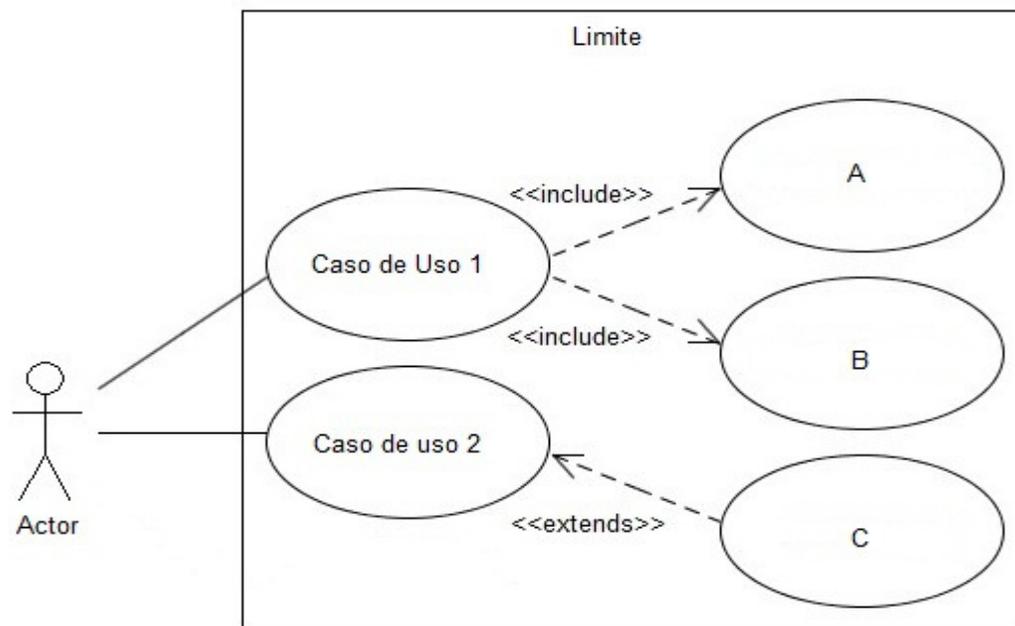
Fuente: Autor.

La figura muestra el DBMS como interface entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal. Es decir el DBMS es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

Anexo D. LEGUAJE DE MODELADO UNIFICADO (UML)

1. Diagramas de casos de uso

Figura 37. Diagramas de Casos de Uso



Fuente: Autor.

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 dos clases de actores:

- **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios tienen un perfil de administrador, con el cual pueden desempeñar ciertas labores diferentes a un usuario normal dentro del sitio. Estos son: los Auxiliares de administración del portal, profesores, secretaria con ciertos

privilegios. Dentro de esta categoría se incluye también el súper administrador.

- Usuario EISIWEB: Es el tipo de usuario común del portal EISIWEB y a quien van dirigidos los servicios. Este usuario solo tiene control sobre sus privilegios.

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 7 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.

2. Diagramas de secuencias

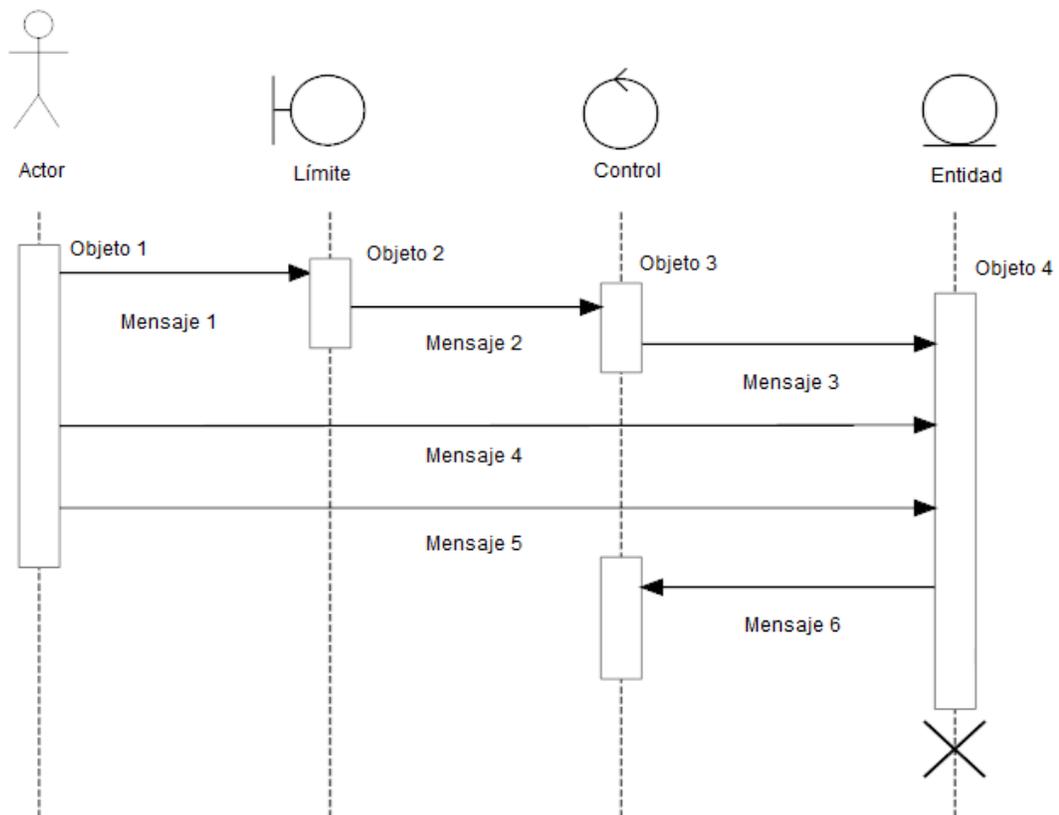
Objetos: Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la *línea de vida* de un objeto. Con la línea de vida se encuentra un pequeño rectángulo conocido 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 38. Diagrama de Secuencias



Fuente: Autor.

Anexo E. PRUEBAS DEL SISTEMA

1. Gestor de Proyectos

Tabla 15. Pruebas Realizadas: Gestión Proyectos.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Iniciar Proyecto	Seleccionar una fecha de inicio anterior a la fecha actual.	✓
Ver Actividades Semanales	Navegar entre semanas viendo las actividades respectivas.	✓
Participar	Participar en una actividad vigente.	✓
Ver proyectos	Mostrar los proyectos de manera ordenada y con sus estados respectivos.	✓

Fuente: Autor.

2. Portal Profesor

Tabla 16. Pruebas realizadas: Ingresar Portal

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Abrir portal	Acceder al portal del profesor indicado.	✓
Acceder servicios privados	Permitir la navegación con los servicios permitidos según el usuario sesionado.	✓
Iniciar sesión	Validar sesión y enviar al portal propio en caso de tener uno.	✓

Acceder servicios públicos	Permitir la navegación de los servicios que no tienen restricción de usuario.	✓
Ir al portal de la Escuela	Acceder al portal de la escuela manteniendo la sesión vigente.	✓

Fuente: Autor.

Tabla 17. Pruebas realizadas: Personalizar Portal

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Personalizar	Acceder al modulo de personalización verificando que tenga permisos.	✓
Definir fondo	Guardar el fondo seleccionado.	✓
Definir color	Guardar el color seleccionado.	✓
Iniciar sesión	Verificar permios del usuario sesionado.	✓

Fuente: Autor.

3. Servicio de Actas de Consejo

Tabla 18. Pruebas realizadas: Inicio del servicio.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Iniciar Servicio	Al acceder al servicio se verifica que sea el administrador el que está accediendo.	✓

Iniciar numeración actas	Iniciar el servicio y generar la primera acta con el número, fecha de inicio y estado correcto.	✓
Indicar Numeración	Validar el número y asignarlo a la primera acta.	✓

Fuente: Autor.

Tabla 19. Pruebas realizadas: Ingreso solicitud usuario EISI.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ingresar solicitud	Muestra las solicitudes permitidas según el usuario sesionado.	✓
Guardar solicitud	Valida la información ingresada por el usuario.	✓
Ver solicitud	Permite ver sólo las solicitudes realizadas por el usuario.	✓
Editar solicitud	Verifica que el usuario que está accediendo sea el propietario y además valida la información editada.	✓
Eliminar solicitud	Validar el estado de la solicitud y que el usuario que está solicitando sea el que la inscribió.	✓

Fuente: Autor.

Tabla 20. Pruebas realizadas: Ingreso solicitud administrador.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ingresar solicitud	Muestra las solicitudes permitidas según el usuario sesionado.	✓
Enviar solicitud	Valida la información ingresada por el usuario.	✓
Ver solicitud	Permite ver sólo las solicitudes realizadas por el usuario.	✓
Editar solicitud	Verifica que el usuario que está accediendo sea el propietario y además valida la información editada.	✓
Eliminar solicitud	Validar el estado de la solicitud y que el usuario que está solicitando sea el que la inscribió.	✓
Seleccionar punto consejo	Muestra los puntos a los que tiene permiso el usuario.	✓
Buscar usuario	Buscar el usuario según el criterio seleccionado.	✓
Adjuntar archivos	Validar la extensión y tamaño del archivo.	✓

Fuente: Autor.

Tabla 21. Pruebas realizadas: Listado solicitudes administrador.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ver solicitudes	Listar las solicitudes que estén vigentes.	✓
Revisar solicitud	Verificar que sea el administrador el que revisa.	✓
Aprobar revisión	La solicitud cambia al estado indicado.	✓
Devolver para cambios	La solicitud cambia al estado indicado. Validar las sugerencias escritas.	✓
Indicar motivos	Validar el texto que ha ingresado el usuario.	✓

Fuente: Autor.

Tabla 22. Pruebas realizadas: Orden del día - Administrador.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Gestionar orden del día	Muestra las opciones de acuerdo al tipo de usuario.	✓
Ordenar puntos	Sólo el administrador lo pueda realizar. Además se verifica que el orden guardado sea el indicado por el usuario.	✓
Descartar puntos	Se descarta el punto seleccionado y no se guarda en el orden.	✓
Incluir puntos	El punto incluido se guarda en el nuevo orden.	✓

Enviar orden	El correo contiene el orden y lugar indicado.	✓
Ver comentarios	Ver los comentarios acerca del orden del día realizados por los miembros del consejo.	✓

Fuente: Autor.

Tabla 23. Pruebas realizadas: Inicio sesión de Consejo.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Iniciar consejo	Verificar que el usuario tenga permisos para este servicio.	✓
Reanudar sesión	Ingresar a una sesión en vigencia.	✓
Iniciar sesión consejo	Se inicia sesión cuando no hay una sesión vigente	✓
Indicar lugar	Validar el texto indicado por el usuario.	✓

Fuente: Autor.

Tabla 24. Pruebas realizadas: Desarrollo sesión de consejo de Escuela.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Desarrollar consejo	Verificar el tipo de usuario y mostrar los puntos a tratar.	✓
Revisar Quórum	Acceder al servicio según permisos.	✓
Registrar Llegada	Guardar la llegada de los miembros del consejo seleccionados.	✓

Seleccionar usuarios	Mostrar los miembros actuales del consejo.	✓
Gestionar puntos	Mostrar las solicitudes para el punto seleccionado.	✓
Finalizar sesión	El estado del Acta se cambie al indicado.	✓

Fuente: Autor.

Tabla 25. Pruebas realizadas: Lectura Acta anterior.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ver actas	Mostrar todas las Actas que están por aprobar.	✓
Ver comentarios	Ver los comentarios que han hecho los miembros del consejo a un acta.	✓
Tomar decisión	Validar la información ingresada por el administrador. Actualizar el Acta.	✓

Fuente: Autor.

Tabla 26. Pruebas realizadas: Desarrollo puntos del consejo.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Tratar informe del Director de Escuela	Se muestran los informes pendientes.	✓

Tratar Asuntos de la Escuela	Se muestran los ítems que pertenecen a este punto.	✓
Tratar Asuntos de Profesores	Se muestran los ítems que pertenecen a este punto.	✓
Tratar Asuntos Miembros Consejo	Se muestran los ítems que pertenecen a este punto.	✓
Tratar Asuntos Varios	Se muestran los ítems que pertenecen a este punto.	✓
Ver solicitudes	Mostrar la solicitud en el punto correspondiente. Mostrar sólo las solicitudes vigentes	✓
Ver comentarios	Revisar los comentarios de una solicitud.	✓
Responder	Guardar la respuesta de manera correcta.	✓

Fuente: Autor.

Tabla 27. Pruebas realizadas: Casos estudiantes.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Tratar casos estudiantes	Se muestran las solicitudes realizadas por los estudiantes.	✓
Aprobar	Cambiar el estado de la solicitud a ' <i>Aprobada</i> '.	✓

Aplazar	Cambiar el estado de la solicitud a <i>'En estudio.'</i>	✓
Devolver	Cambiar el estado de la solicitud a <i>'Mejorar para consejo.'</i>	✓
Negar	Cambiar el estado de la solicitud a <i>'Negada.'</i>	✓
Ver comentarios	Ver los comentarios que han hecho los miembros del consejo a un ítem.	✓

Fuente: Autor.