

**MANTENIMIENTO DEL PORTAL WEB, ANÁLISIS, DESARROLLO E  
IMPLEMENTACIÓN DE NUEVOS SERVICIOS PARA EL SISTEMA  
COMUNIDAD ACADÉMICA QUE SOPORTA LOS PORTALES WEB DE LAS  
ESCUELAS Y FACULTADES**

**OSCAR ORLANDO MOLINA PINTO**

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

**2016**

**MANTENIMIENTO DEL PORTAL WEB, ANÁLISIS, DESARROLLO E  
IMPLEMENTACIÓN DE NUEVOS SERVICIOS PARA EL SISTEMA  
COMUNIDAD ACADÉMICA QUE SOPORTA LOS PORTALES WEB DE LAS  
ESCUELAS Y FACULTADES**

**OSCAR ORLANDO MOLINA PINTO**

**Trabajo de grado presentado como requisito  
para optar al título de Ingeniero de Sistemas**

**Director**

**LUIS IGNACIO GONZÁLEZ RAMÍREZ**

**Magíster en Informática**

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

**2016**

## AGRADECIMIENTOS

*Principalmente gracias a Dios por las oportunidades brindadas y por la fortaleza que me ha dado para poder superar los diferentes obstáculos que se han presentado en la vida y que con ayuda del he podido superar y aprender de los mismos.*

*a mi madre que ha sido una gran mujer y que con su ayuda incondicional y paciencia me apoyo en este gran logro que estoy culminando, a mi padre que es un hombre sabio y que junto a sus concejos y paciencia me ha ayudado toda la vida, igualmente a mi familia hermanos que todos juntos son la base de lo que soy hoy en día.*

*A mi director de proyecto el Ingeniero Luis Ignacio por darme la oportunidad de pertenecer al grupo Calumet, igualmente a Duvan Jamid Vargas Castillo por todas las recomendaciones, tiempo, paciencia y conocimientos compartidos en este largo camino.*

*A mi amigo Rubén que siempre ha estado conmigo en los buenos y malos momentos, que me ha ayudado y con sus concejos he sabido superar diferentes obstáculos que la vida me ha puesto y por siempre estar ahí conmigo en todo igualmente a Elizabeth que en estos últimos semestres ha sido un gran apoyo en momentos difíciles que he afrontado en mi familia y que con su cariño amabilidad paciencia, alegría pude superar y a todos mis amigos que juntos a mi han compartido en la universidad en cada salón de clase, los cuales han sido un honor haber compartido con ellos y haber tenido al lado en este triunfo que estoy culminando*

## CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN .....	15
1. DEFINICIÓN DEL PROBLEMA .....	17
2. JUSTIFICACIÓN.....	19
3. OBJETIVOS.....	21
3.1 OBJETIVO GENERAL .....	21
3.2 OBJETIVOS ESPECÍFICOS.....	21
4. ALCANCES Y LIMITACIONES .....	24
5. MARCO TEÓRICO .....	25
5.1 ARQUITECTURA CLIENTE/SERVIDOR.....	25
5.1.1 Arquitectura Cliente/Servidor aplicada .....	26
5.1.2 Ventajas del esquema Cliente/Servidor .....	26
5.1.3 Desventajas del esquema Cliente/Servidor .....	27
5.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS.....	27
5.2.1 Tecnología aplicada.....	27
5.3 BASES DE DATOS.....	29
5.3.1 MySQL.....	30
5.4 NETBEANS.....	31
5.5 SISTEMA DE CONTROL DE VERSIONES .....	31
5.5.1 Subversión.....	32
5.6 PROGRAMACIÓN UTILIZADA.....	33
5.6.1 Clases.....	33
5.6.2 Objetos.....	33
5.6.3 Atributos.....	33
5.6.4 Métodos .....	34
5.6.5 Herencia.....	34
5.6.6 Beneficios de la Programación Orientada a Objetos .....	34
5.6.7 Java y JDK (Java Development Kit).....	34
5.7 SERVIDORES WEB .....	35

5.7.1 Servidor Jakarta Tomcat .....	35
6. DISEÑO METODOLÓGICO.....	37
6.1 PROTOTIPO EVOLUTIVO .....	37
6.2 LENGUAJE DE MODELADO UNIFICADO .....	39
6.2.1 Diagramas de UML .....	39
6.2.2 Diagramas de casos de uso.....	39
6.2.3 Diagramas de secuencias.....	40
6.3 ESTÁNDARES DE PROGRAMACIÓN.....	40
6.3.1 Modelo de datos .....	40
6.3.2 Nombres de las tablas .....	40
6.3.3 Clases.....	41
6.3.4 Páginas JSP .....	41
6.3.5 Organización de Directorios.....	41
7. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO.....	42
7.1 PROTOTIPO ESPERADO .....	42
7.1.1 Requerimientos de objetivo.....	42
7.1.2 Requerimientos del portal Web.....	45
7.1.3 Diagramas de casos de uso.....	50
7.1.4 Documentación de casos de USO del sistema .....	56
7.1.5 Diseño y Análisis.....	56
7.1.6 Modelo de Procesos del Sistema.....	59
7.1.7 Implementación, Implantación y Pruebas Generales.....	69
7.2 MANTENIMIENTO Y ADMINISTRACIÓN.....	70
7.2.1 Actividades de Soporte a Usuarios .....	70
7.2.2 Actividades de Administración .....	70
8. PRUEBAS DEL SISTEMA .....	72
8.1 PRUEBAS DE VERIFICACIÓN .....	72
8.1.1 Pruebas por componente.....	72
8.2 PRUEBAS DE INTEGRACIÓN .....	72

8.3 PRUEBAS DE VALIDACIÓN .....	73
9. CONCLUSIONES .....	74
10. RECOMENDACIONES.....	75
BIBLIOGRAFÍA.....	76
ANEXOS.....	79

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Descripción de las Entidades.....	58

## LISTA DE FIGURAS

	<b>Pág.</b>
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: Crear PQRS en el servicio de Consultas y Sugerencias. ....	50
Figura 5. Diagrama de Casos de USO: Enviar mensaje en el Foro del servicio de consultas y sugerencias.....	50
Figura 6. Diagrama de Casos de Uso: Administración de PQRS. ....	51
Figura 7. Diagrama de Casos de Uso: Crear PQRS y asignar encargados.....	51
Figura 8. Diagrama de casos de uso: crear correo. ....	52
Figura 9. Diagrama de casos de uso: Selección criterio de correo.....	52
Figura 10. Diagrama de casos de uso: Consultar proyectos. ....	53
Figura 11. Diagrama de casos de uso: Registrar estudiantes en el Aula.....	53
Figura 12. Diagrama de casos de uso: Crear actividad. ....	54
Figura 13. Diagrama de casos de uso: Responder taller y examen.....	54
Figura 14. Diagrama de casos de uso: Crear pregunta. ....	55
Figura 15. Diagrama de casos de uso: Subsistema de Mantenimiento y Administración. ....	55
Figura 16. Diagrama E/R: Servicio consultas y sugerencias. ....	56
Figura 17. Diagrama E/R: Módulo buscador de proyectos. ....	57
Figura 18. Diagrama E/R: Módulo Aula Virtual. ....	57
Figura 19. Diagrama de Secuencia: Crear PQRS en el servicio de Consultas y Sugerencias.....	59
Figura 20. Diagrama de Secuencia: Enviar mensaje en el Foro del servicio de consultas y sugerencias.....	60
Figura 21. Diagrama de Secuencia: Administración de PQRS. ....	61
Figura 22. Diagrama de Secuencia: Crear Correo.....	62
Figura 23. Diagrama de Secuencia: Selección criterio de correo. ....	63

Figura 24. Diagrama de Secuencia: Consultar proyectos.....	64
Figura 25. Diagrama de Secuencia: Registrar estudiantes en el Aula.....	65
Figura 26. Diagrama de Secuencia: Crear actividad.....	66
Figura 27. Diagrama de Secuencia: Responder taller y examen.....	67
Figura 28. Diagrama de Secuencia: Crear pregunta.....	68

## LISTA DE ANEXOS

	<b>Pág.</b>
Anexo A. Arquitectura Cliente/Servidor.....	80
Anexo B. Tecnologías de desarrollo de páginas web dinámicas .....	83
Anexo C. Bases de datos .....	85
Anexo D. Leguaje de modelado unificado (UML).....	90
Anexo E. Documentación de casos de uso del sistema .....	93
Anexo F. Pruebas del sistema .....	98

## RESUMEN

**TÍTULO:** MANTENIMIENTO DEL PORTAL WEB, ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE NUEVOS SERVICIOS PARA EL SISTEMA COMUNIDAD ACADÉMICA QUE SOPORTA LOS PORTALES WEB DE LAS ESCUELAS Y FACULTADES\*

**AUTOR:** OSCAR ORLANDO MOLINA PINTO\*\*

**PALABRAS CLAVE:** Correo Comunidad Escuela, buscador de proyectos, portal web.

### DESCRIPCIÓN

En la actualidad las Escuelas y la Facultad de Ingenierías Físico Mecánicas, cuentan con el Portal Web Comunidad Académica que permite el encuentro e interacción de los diferentes miembros de estas comunidades. Este Portal soporta las diferentes actividades académicas que se realizan dentro de las escuelas y la Facultad, así como define a los usuarios los permisos y servicios que se les proporcionan.

Con el fin de atender las necesidades de los usuarios en los portales, se implementó el servicio de consultas y sugerencias. A este servicio pueden acceder tanto miembros de la comunidad, como personas que visitan los portales y no son miembros de la misma

El servicio Correos Comunidad Escuela permite el envío de mensajes a los miembros de otras escuelas. Se implementaron las nuevas opciones en el servicio de Correos comunidad Escuela ayudara a los diferentes usuario poder enviar correos a toda la comunidad teniendo en cuenta filtros como estudiantes por carrera y nivel entre otros.

El servicio e buscador de proyectos permite a los docentes buscar los diferentes proyectos. Se realice la implementación del buscador de proyectos en donde se incluyeron nuevos módulos de búsqueda como son los proyectos tanto dirigidos, evaluados y calificados, ayudara a los profesores y directores de proyecto consultar los proyectos de los cuales están asignados.

De los servicios que brindan actualmente los Portales web de las diferentes Escuelas y Facultades, se encuentra el aula virtual. La implementación de crear actividades evaluables dentro del aula virtual y la posibilidad de poder asignar a varios profesores a la misma aula facilitara el manejo de la misma y la unión de la comunidad.

---

\* Trabajo de grado: Modalidad Trabajo de Investigación.

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

## ABSTRACT

**TÍTULO:** WEB PORTFOLIO MAINTENANCE, ANALYSIS, DEVELOPMENT AND IMPLEMENTATION OF NEW SERVICES FOR THE ACADEMIC COMMUNITY SYSTEM SUPPORTING WEB SITES OF SCHOOLS AND FACULTIES \*

**AUTOR:** OSCAR ORLANDO MOLINA PINTO\*\*

**KEYWORDS:** Mail Community School, Project Search, web portal.

### DESCRIPCIÓN

At present the Schools and the Faculty of Mechanical Physical Engineering, have the Web Portal Academic Community that allows the encounter and interaction of the different members of these communities. This Portal supports the different academic activities that are carried out within the schools and the Faculty, as well as it defines to the users the permissions and services that are provided to them.

In order to meet the needs of the users in the portals, the service of consultations and suggestions was implemented. This service can be accessed by members of the community, as well as people who visit the portals and are not members of the same

The Mail Community School service allows sending messages to members of other schools. They implemented the new options in the service of Mail Community School will help the different users to be able to send mails to the whole community taking into account filters like students per race and level among others.

The service and project finder allows teachers to search for different projects. The implementation of the project searcher where new search modules are included as the projects are both directed, evaluated and qualified, will help teachers and project managers consult the projects of which they are assigned.

Of the services currently offered by the Web Portals of the different Schools and Faculties, there is the virtual classroom. The implementation of evaluable activities within the virtual classroom and the possibility of being able to assign several teachers to the same classroom facilitate the management of the same and the union of the community.

---

\* Degree Work: Research Paper.

\*\* Faculty of Physical Mechanical Engineering, School of System Engineering and Computing Science. Director: Luis Ignacio González Ramírez.

## INTRODUCCIÓN

Debido al crecimiento de la comunidad de la Universidad Industrial de Santander e inicialmente la comunidad de la escuela de Ingeniería de Sistemas, se hace necesario un canal de información y control dentro 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 comunidad, 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 Ingeniería 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.

Los portales WEB pertenecientes a la escuela y a la facultad, se han convertido en uno de los principales canales de comunicación e integración entre los miembros de la comunidad de la escuela y la facultad. 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 y la facultad.

El grupo de desarrollo de software Calumet se encarga de desarrollar, administrar y mantener los portales EIMTWEB y FIFMEWEB a los cuales se les realizan continuos cambios y actualizaciones para hacer de estos una herramienta eficiente.

A 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 portal web de las Escuelas, en línea con su objetivo de aportar lugares para la comunidad, permite a sus usuarios crear conversaciones o proponer eventos que se pueden dirigir a la comunidad de la escuela o a las comunidades de varias escuelas (comunidad UIS). La creación de conversaciones y eventos permiten el envío de correos a las personas de la comunidad con el fin de invitarlas a la participación de diferentes formas. Consistente con nuestro enfoque comunidad UIS, se hace necesario generar mensajes de notificación que permitan al creador, como a los demás participantes enterarse de lo acontecido a partir de su creación.

Dentro de las funciones de los administradores de los Portales web de las diferentes Escuelas y Facultades, se encuentra la de generar y actualizar los menús que se mostrarán en el Portal de Profesor. Esta labor se realiza por un servicio creado para este fin. Para actualizar los menús del portal de profesor en este momento, es necesario eliminar manualmente de la Base de Datos los registros en la tabla correspondiente, para de esta forma volver a crearlos todos. Por consiguiente, surge la necesidad de crear un servicio que consulte la tabla, muestre los menús vigentes, y actualice rutas del menú actual, de tal forma que sea sencillo actualizarlos, reubicarlos, eliminarlos o agregar uno nuevo.

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. DEFINICIÓN DEL PROBLEMA

En la actualidad las Escuelas y la Facultad de Ingenierías Físico Mecánicas, cuentan con el Portal Web Comunidad Académica que permite el encuentro e interacción de los diferentes miembros de estas comunidades. Este Portal soporta las diferentes actividades académicas que se realizan dentro de las escuelas y la Facultad, así como define a los usuarios los permisos y servicios que se les proporcionan.

Los servicios de Portal Web Comunidad Académica 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..

Con el fin de atender las necesidades de los usuarios en los portales, se desea mejorar el servicio de consultas y sugerencias. A este servicio pueden acceder tanto miembros de la comunidad, como personas que visitan los portales y no son miembros de la misma; los miembros de la comunidad podrán llevar un control de sus consultas y podrán participar en foros con el administrador para resolver sus inquietudes. Existirán diferentes tipos de solicitudes, y de acuerdo a estas se dirigirá a un usuario específico, asignado para atender este tipo.

Para facilitar la consulta de los proyectos de grado por parte de los profesores de la comunidad académica se busca implementar un buscador de proyectos de grado en los cuales se contemplarán los proyectos dirigidos, los evaluados y los calificados. Además el profesor verá el plan, el historial y la información de los autores (tanto académica como del perfil).

Uno de los servicios con los que cuenta actualmente la escuela es el de correos comunidad escuela, el cual facilita la comunicación entre las diferentes personas que pertenecen a la comunidad de la escuela. A este servicio se le desean agregar nuevas opciones para el envío de correos como son: programas académicos, estudiantes por programa, estudiantes por nivel, egresados por año de egreso, entre otros,

Con el fin de hacer más sencilla la gestión del aula por parte del Profesor, se hace necesario implementar nuevas herramientas para las actividades evaluables dentro del aula virtual. Se requiere el desarrollo de un módulo donde se permita la generación de planes de actividades evaluables desde la interfaz en la cual se dirige la asignatura. Esto hará más sencillo la creación de planes y actividades por parte del profesor.

## 2. 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, estudiantes y a la comunidad en general un mejor servicio se implementaran nuevas herramientas en el Aula virtual el cual no es el único lugar de encuentro de la Comunidad de la Escuela, pero si el más importante, por ser esta una Comunidad de Aprendizaje. Es allí, donde se desarrolla, por excelencia, la misión en su eje de docencia.

Con el fin de facilitar a los profesores de la comunidad académica se implementó un buscador de proyectos en los cuales el docente podrá visualizar los proyectos dirigidos los evaluados y los calificados en los cuales está siendo evaluador o calificador de los planes, Además el profesor podrá ver el plan, el historial y la información de los autores (tanto académica como del perfil) de los planes calificados y en curso.

Respondiendo a las necesidades de los usuarios se mejoró el servicio de consultas y sugerencias, el cual permite tanto a miembros de la comunidad como usuarios que visiten los portales. Con el fin de que el usuario pueda manejar su consulta y resolver sus inquietudes mediante un foro que se realizaría con el administrador y pueda disponer de ellos cuando lo requiera.

Se realizó reingeniería al portal de Producción intelectual, el cual consistió en realizar una interface amigable para el usuario, y ampliar la navegabilidad entre los Diferentes portales que ofrece el grupo Calumet a la comunidad, los cuales son:

portal de grupos, portal del profesor, y portal de la escuela.

Con el fin de ofrecer un mejor servicio se hizo reingeniería al módulo que cuenta actualmente la escuela es el de correos comunidad escuela, el cual facilitaría a la comunicación entre las diferentes personas que pertenecen a la comunidad de la escuela. Este servicio facilitaría el envío de correos como son: programas académicos, estudiantes por programa, estudiantes por nivel, egresados por año de egreso, estudiantes por materia, entre otros.

### **3. OBJETIVOS**

#### **3.1 OBJETIVO GENERAL**

Realizar las funciones de administración, soporte a usuarios, mantenimiento del portal web, análisis, desarrollo e implementación de nuevos servicios para el portal web de la escuela de Metalúrgica y la Facultad de Ingenierías Fisicomecánicas para hacer óptimos, eficientes y sostenibles los trámites dentro de las escuelas y el acceso a la información de las mismas.

#### **3.2 OBJETIVOS ESPECÍFICOS**

**Efectuar labores de Administración del Portal web de la Escuela de Metalúrgica y Facultad de Ingenierías Fisicomecánicas teniendo en cuenta principalmente:**

- Generar BackUps (copias de seguridad) diariamente de las bases de datos.
- Hacer seguimiento del uso que hacen los usuarios a los servicios de Foros, Eventos Propuestos, Cartelera, Archivos y Mi perfil dentro de los portales, 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 de los portales web.
- Efectuar una revisión constante de los archivos que se suben al sitio, eliminando los que no son necesarios para evitar saturación de los portales.

**Llevar a cabo labores de mantenimiento a los portales, en las que podemos encontrar:**

- 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 de los portales web.

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

- Implementar reingeniería al servicio de consultas y sugerencias a los usuarios en general.
- Implementar buscador de proyectos para los profesores contemplando los diferentes tipos de proyectos (dirigidos, evaluados y calificados).
- Implementar reingeniería al servicio correo comunidad escuela, de manera que agregue algunas nuevas opciones como estudiantes por nivel, entre otros.
- Implementar dentro del aula virtual la creación de actividades evaluables durante el curso.

**Desempeñar labores de soporte a los usuarios de los portales, 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 los portales, promoviendo así su utilización.
- Atender usuarios por olvido de la contraseña, creación de grupos, solicitudes de propuestas de eventos o de cartelera en el índice y creación de agendas con eventos y foros con sus respectivas conversaciones.
- Crear usuarios de forma manual, para personas no pertenecientes a la escuela o la facultad y que por algún motivo necesitan registrarse en el sitio.

- Modificar los estados de los usuarios de acuerdo a la relación con la Escuela o la Facultad (Activo, Inactivo, Suspendido).

**Capacitar a los estudiantes que recibirán las funciones de administración, mantenimiento, creación, y mejora de nuevos servicios dentro de los portales 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.

#### **4. ALCANCES Y LIMITACIONES**

El servicio de consultas y sugerencias a nivel de Comunidad UIS como de personas que visitan los portales y no son miembros de la misma, les brinda un control y manejo de sus consultas mediante un foro el cual podrá ser realizado con el administrador encargado de resolver el tipo de consulta que la persona requiera.

El Servicio de Correo Comunidad Escuela, mediante una interface amigable permite el envío de correos a toda la comunidad universitaria (estudiantes, egresados, trabajadores, profesores, entre otros), con la información del asunto y los destinatarios del mismo, a su vez, da la opción seleccionar varios miembros específicos, y poder agilizar el envío de sus correos.

Con el fin de un mejor manejo de los planes se implementó un buscador de proyectos donde el profesor podrá realizar la búsqueda de un plan en el cual pertenezca como calificador y a su vez hacer seguimiento del mismo, también podrá ver la información del plan como de los estudiantes, igualmente el profesor podrá consultar los planes a los cuales se le asigno como calificador o los cuales que ya fueron calificados por el mismo.

Se implementaron nuevas herramientas en el servicio de aula virtual para así hacer más sencilla el manejo de la misma, entre los cuales se encuentran para realizar actividades evaluables como talleres, foros, trabajos, exposiciones y exámenes. Igualmente se implementó un Nuevo módulo el cual facilitara al profesor la creación de planes y actividades.

## 5. MARCO TEÓRICO

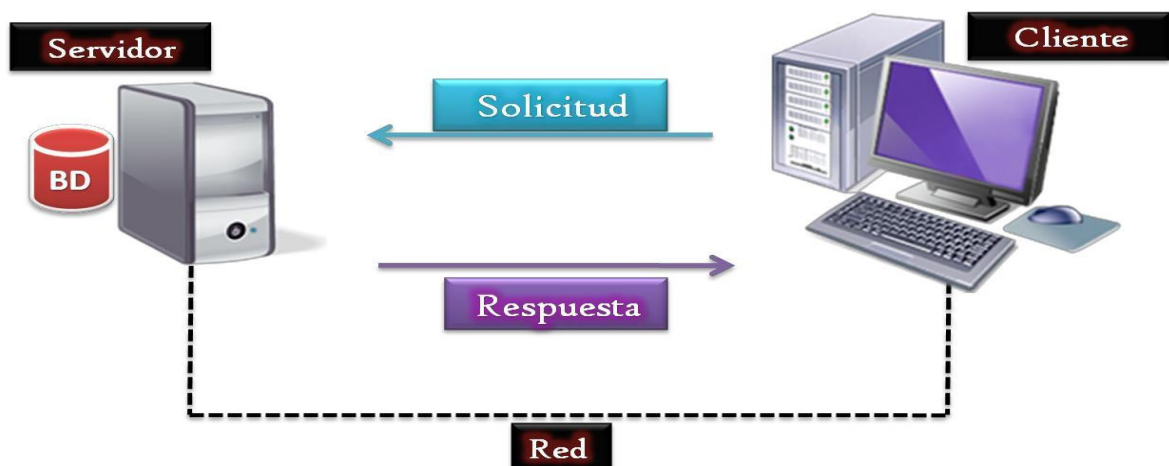
### 5.1 ARQUITECTURA CLIENTE/SERVIDOR

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: Arquitectura cliente servidor. [en línea]. <http://ulyla003.wordpress.com/page/4/>. [citado el 01 Mayo de 2014]

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

**5.1.1 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.

### **5.1.2 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.

### **5.1.3 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.

## **5.2 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).

**5.2.1 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.

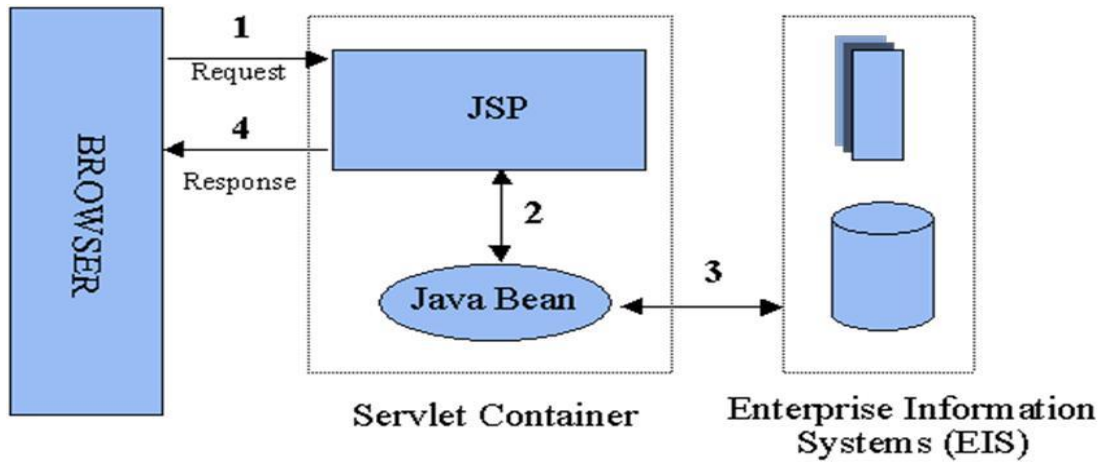
Las paginas JSP y servlets se ejecutan en la Máquina Virtual de Java, lo cual permite que se puedan usar en cualquier tipo de computador, siempre y cuando este instalada la Máquina Virtual de Java. Cada JSP se ejecuta en su propio contexto (llamado también hilo o hebra); pero no se comienza a ejecutar 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.

### **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: Modelo JSP. [en línea]. <http://geneura.ugr.es/~jmerelo/JSP/modelo1.gif> . [citado el 05 Mayo de 2014]

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.

### 5.3 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 GEOWEB y FIFMEWEB se utiliza MySQL.

**5.3.1 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.

### **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.

## **5.4 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.

## **5.5 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.

**5.5.1 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.

## 5.6 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.

**5.6.1 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.

**5.6.2 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).

**5.6.3 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.

**5.6.4 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.

**5.6.5 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.

#### **5.6.6 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.

**5.6.7 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.

## 5.7 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.

**5.7.1 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.

## 6. DISEÑO METODOLÓGICO

### 6.1 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.



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

- Los portales EIMTWEB y FIFMEWEB presentan 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 de los portales EIMTWEB y FIFMEWEB

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.

## 6.2 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.

**6.2.1 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.

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

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

## **6.3 ESTÁNDARES DE PROGRAMACIÓN**

**6.3.1 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).

**6.3.2 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 antepone 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 EIMTWEB. El prefijo a anteponer a los nombres de estas tablas es “TP\_”, es decir la tabla “Usuarios”, es conocida como “TP\_Usuarios”.

**6.3.3 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: ProductoIntelectual.java, PortalProfesor.java.

**6.3.4 Páginas JSP.** Los nombres de las paginas JSP que componen los portales EIMTWEB y FIFMEWEB 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: IngresarProductoIntelectual.jsp, CorreosPorPoseidon.jsp.

**6.3.5 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.
- El sitio cuenta con un directorio llamado Script, donde están todos los archivos de JavaScript “.js” necesarios para el buen funcionamiento del portal.

## **7. 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.

### **7.1 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.

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

**7.1.1.1 Crear Ofrecer la posibilidad de crear un Aula virtual a una comunidad más amplia, soportando no solo las materias y grupos reportados en la matrícula, sino también soportar cursos, capacitaciones y otros.**

Objetivo Inicial:

- Implementar dentro del aula virtual la creación de actividades evaluables

durante el curso.

Requisitos finales del objetivo:

- Se implementa la interfaz para poder agregar temas a los planes y así poder asignarlos al momento de realizar alguna actividad en el curso.
- El Aula virtual que se crea puede ser ligada a una material de la matricula o independiente de ella.
- Se crea nuevos módulos para que el profesor pueda generar los planes de sus actividades.
- Se incluye una interfaz nueva donde el docente podrá agregar las preguntas correspondientes a una actividad ya sea taller o examen.

#### **7.1.1.2 Implementar reingeniería al servicio de consultas y sugerencias a los usuarios en general.**

Objetivo inicial:

- Mejorar el servicio de consultas y sugerencias de tal forma que se pueda llevar un control de las consultas de los usuarios.

Requisitos finales del objetivo:

- Permitir enviar una PQRS al administrador encargado de atenderla según cada escuela
- Permitirle al miembro de la comunidad llevar un control de sus consultas en una interfaz amigable con el usuario.
- El usuario podrá participar en un foro con el administrador encargado de atender su consulta para de esta forma atender de mejor forma su consulta.
- Permitir al administrador por medio de una interfaz nueva atender las PQRS que se le asignen, donde podrá observar las consultas que están por atender, las respondidas y las finalizadas por el administrador.

- Enviar correos a los miembros de la comunidad y personas que visitan la página informándoles sobre la consulta que realizaron.

**7.1.1.3 Implementar un buscador de proyectos para los profesores donde puedan visualizar los planes a los proyectos a los cuales fueron asignados como evaluadores o calificadores, en la nueva interfaz podrán visualizar también la información del plan sus objetivos y los datos de los estudiantes.**

Objetivo inicial:

- Implementar buscador de proyectos para los profesores contemplando los diferentes tipos de proyectos (dirigidos, evaluados y calificados).

Requisitos finales del objetivo:

- Permitir al profesor visualizar los proyectos a los cuales fue asignado como evaluador o calificador.
- Brindarle al profesor una interfaz donde podrá realizar la búsqueda con diferentes requerimientos ya sea de proyectos de pregrado, posgrado o maestría.
- Brindarle al profesor la información del proyecto al cual fue asignado entre los que están el nombre del plan, los autores, los objetivos, estado del plan entre otros.
- Permitir al profesor visualizar el historial de los planes a los cuales fue asignado tanto como evaluador o calificador.

**7.1.1.4 Implementar reingeniería al servicio correo comunidad escuela, de manera que agregue algunas nuevas opciones Como estudiantes por nivel, entre otros.**

Objetivo inicial:

- Uno de los servicios con los que cuenta actualmente la escuela es el de correos comunidad escuela, el cual facilita la comunicación entre las diferentes personas que pertenecen a la comunidad de la escuela. A este servicio se le desean agregar nuevas opciones para el envío de correos como son: programas académicos, estudiantes por programa, estudiantes por nivel, egresados por año de egreso, entre otros.

Requisitos finales del objetivo:

- Diseñar una interfaz amigable para el usuario, en la cual pueda redactor información la cual enviara a diferentes usuarios de la comunidad universitaria.
- Permitir al usuario buscar personas dela comunidad universitaria entre los que están estudiantes pregrado, posgrado, egresados, administrativos, profesores entre otros.
- Permitir al usuario enviar correos con a miembros de la comunidad con diferentes tipos de categoría ya sea programas académicos, estudiantes por programa, estudiantes por nivel, egresados por año de egreso, entre otros.

**7.1.2 Requerimientos del portal Web.** Para aquellos servicios que lo requieran, se debe incluir la rutina de logueo de usuario y verificación de sesión, requerimiento de seguridad intrínseco de los sitios web a los que presta sus servicios el grupo CALUMET.

#### **7.1.2.1 Administración.**

Objetivos iniciales:

- Efectuar labores de administración de los portales web; teniendo en cuenta entre otras:
- Generar Backups (copias de respaldo) diariamente de la Base de Datos.

- Hacer seguimiento del uso que hacen los usuarios en cuanto a los servicios de Foros, Eventos Propuestos, Cartelera, Archivos y Mi perfil dentro de los portales, 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.

#### **7.1.2.2 Mantenimiento.**

Objetivos iniciales:

- Llevar a cabo labores de mantenimiento a los portales en las que podemos encontrar:
- 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.

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 División, Diamante y de los directorios de los sitios web.
- 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.

### **7.1.2.3 Soporte a Usuarios.**

Objetivo inicial:

- Desempeñar labores de soporte a los usuarios de los portales web, 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 de los portales web, 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 los portales web 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 de los portales web, 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.

#### **7.1.2.4 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 de los portales web 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.

### 7.1.3 Diagramas de casos de uso.

- Servicio de Consultas y Sugerencias.

Figura 4. Diagrama de Casos de Uso: Crear PQRS en el servicio de Consultas y Sugerencias.

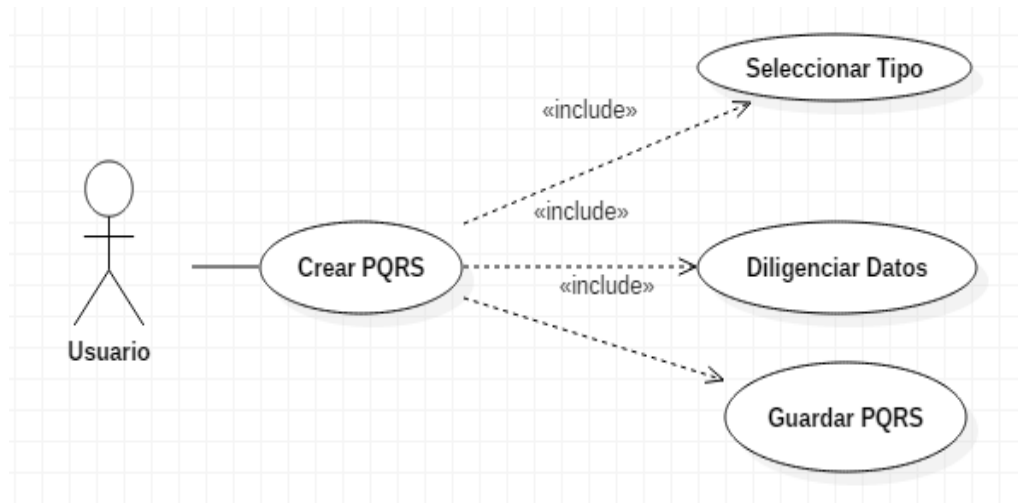


Figura 5. Diagrama de Casos de USO: Enviar mensaje en el Foro del servicio de consultas y sugerencias

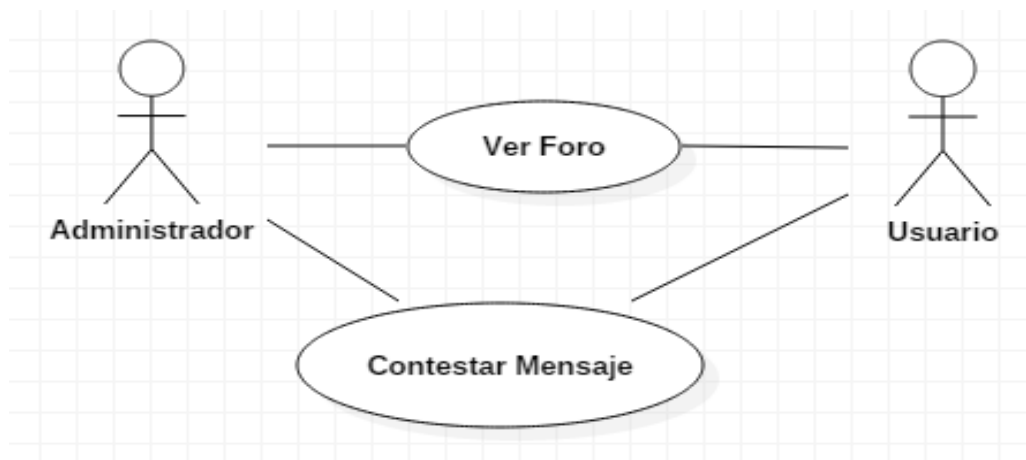


Figura 6. Diagrama de Casos de Uso: Administración de PQRS.

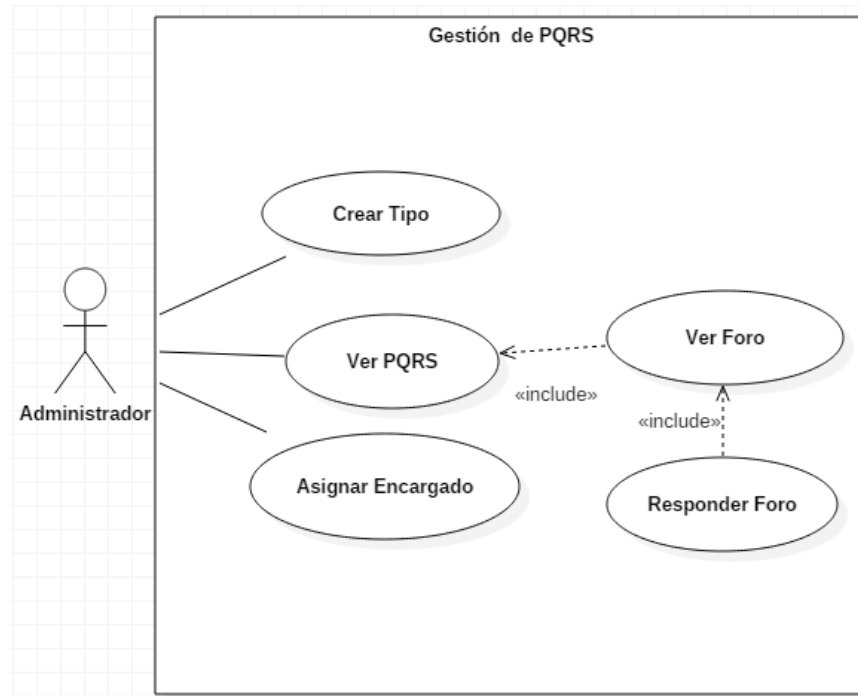
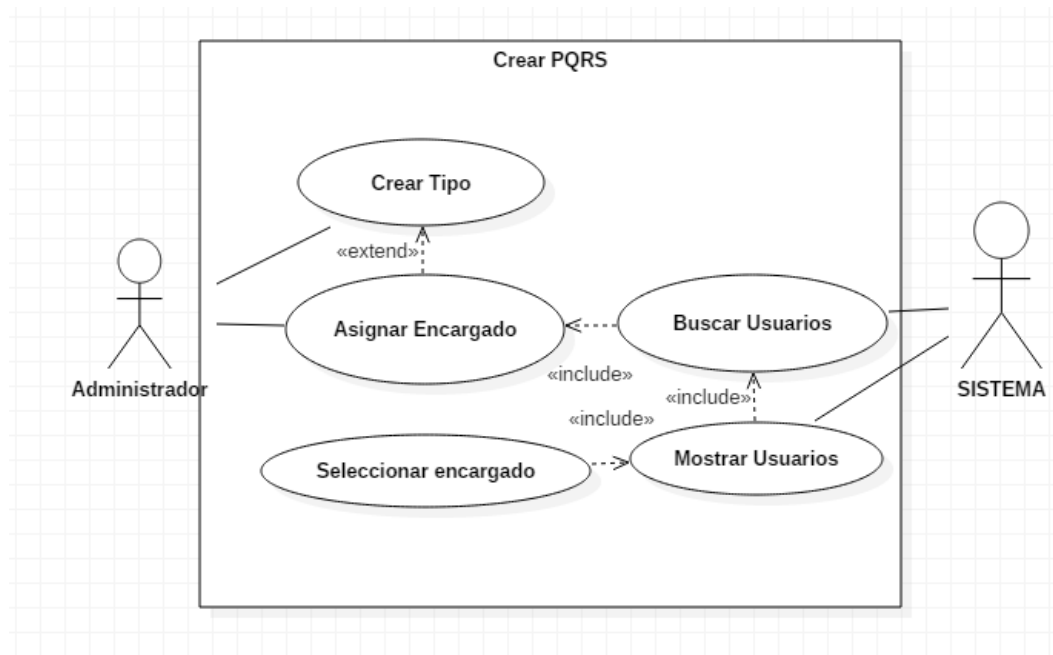


Figura 7. Diagrama de Casos de Uso: Crear PQRS y asignar encargados.



- **Correo Comunidad Escuela.**

Figura 8. Diagrama de casos de uso: crear correo.

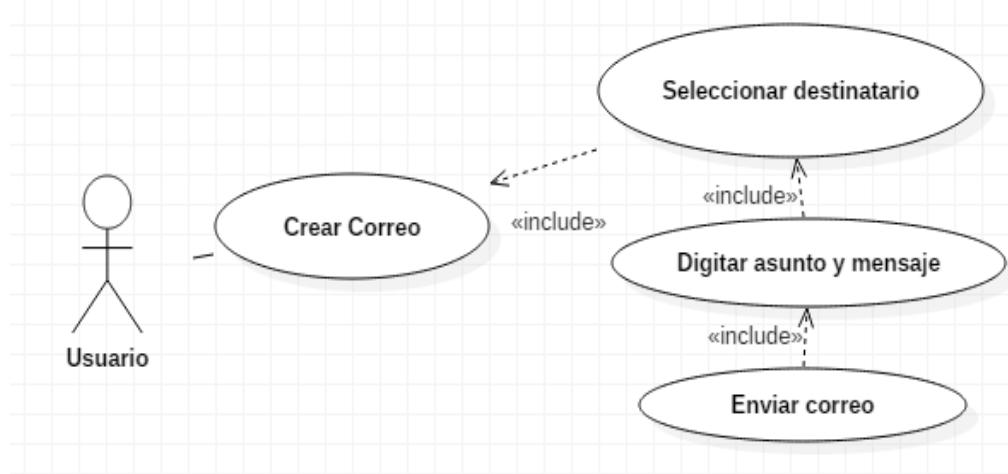
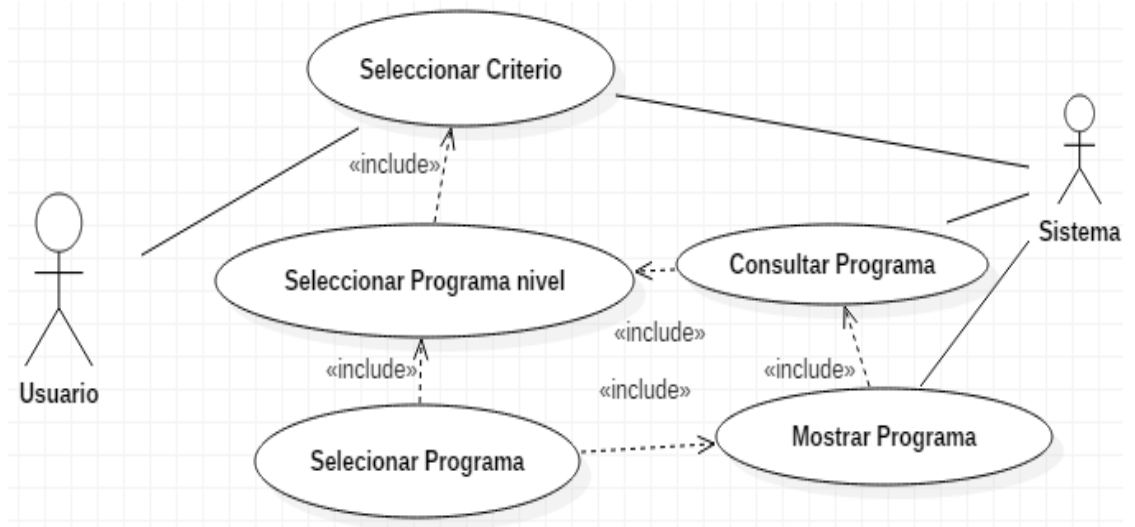
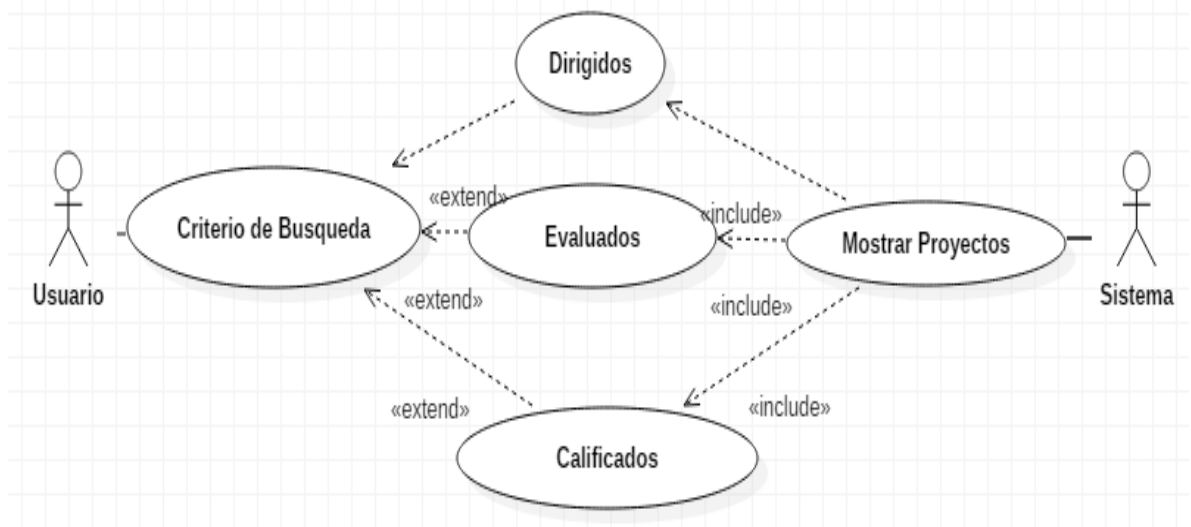


Figura 9. Diagrama de casos de uso: Selección criterio de correo.



- **Buscador de proyectos**

Figura 10. Diagrama de casos de uso: Consultar proyectos.



- **Aula virtual.**

Figura 11. Diagrama de casos de uso: Registrar estudiantes en el Aula

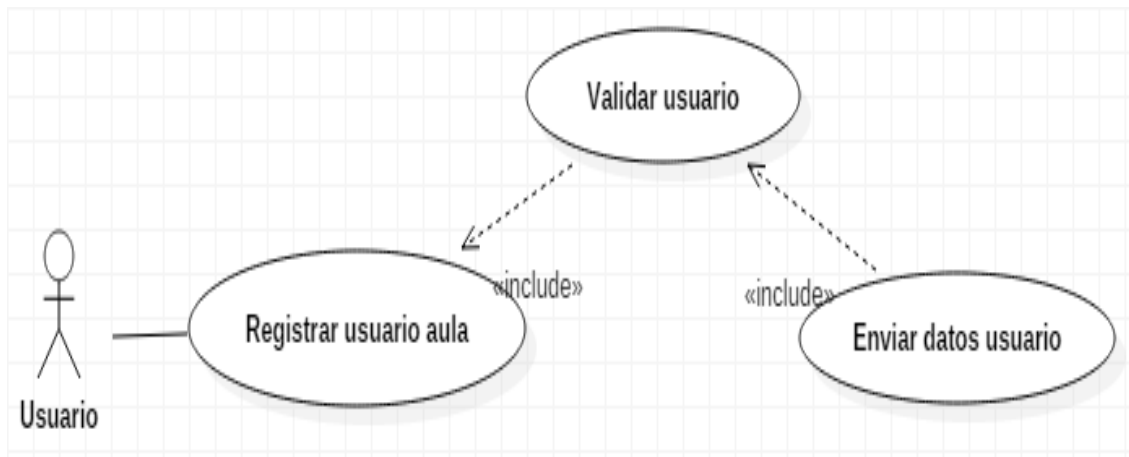


Figura 12. Diagrama de casos de uso: Crear actividad.

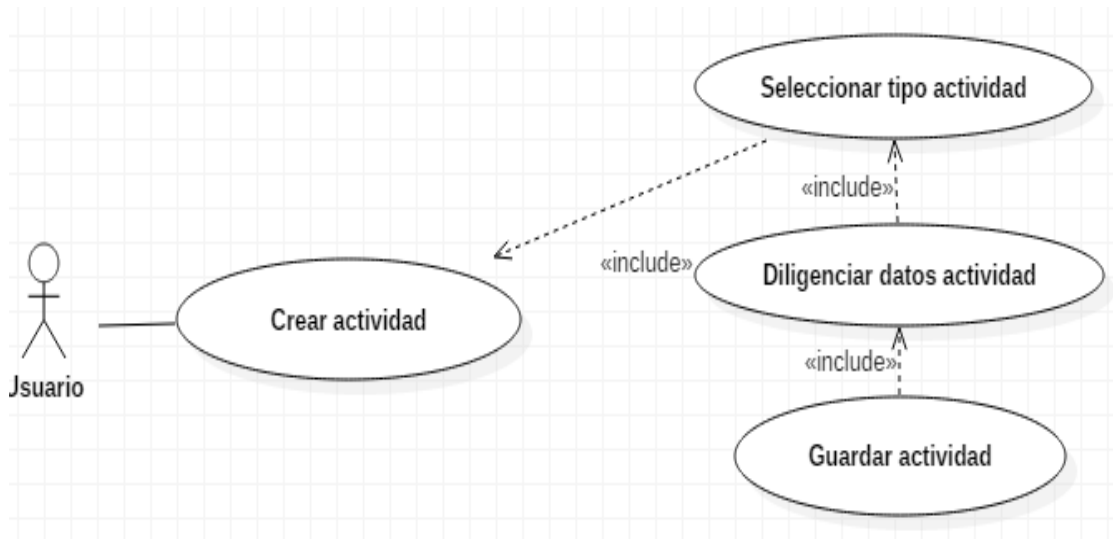


Figura 13. Diagrama de casos de uso: Responder taller y examen.

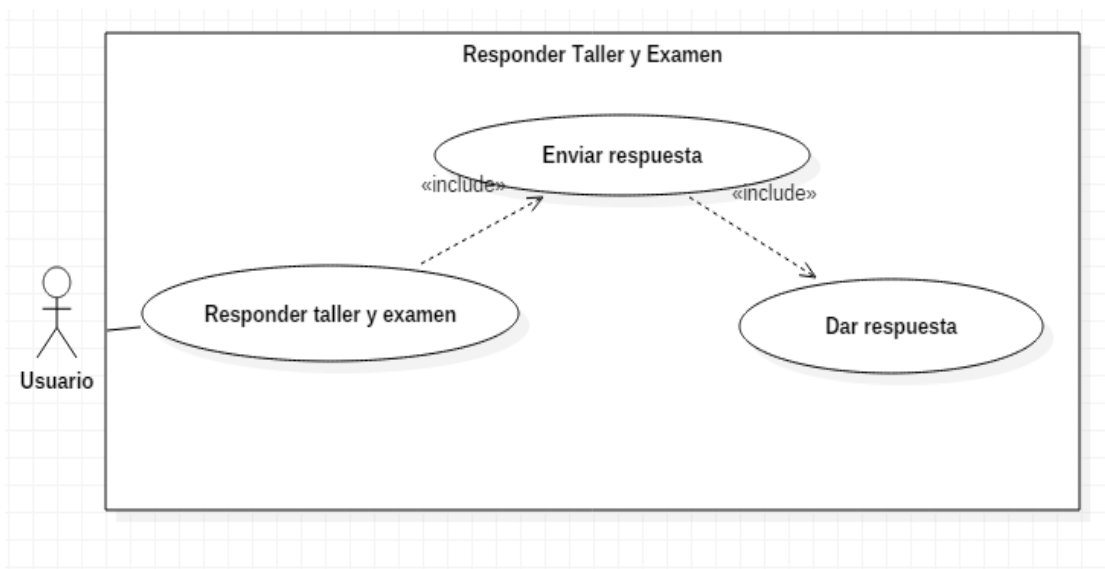
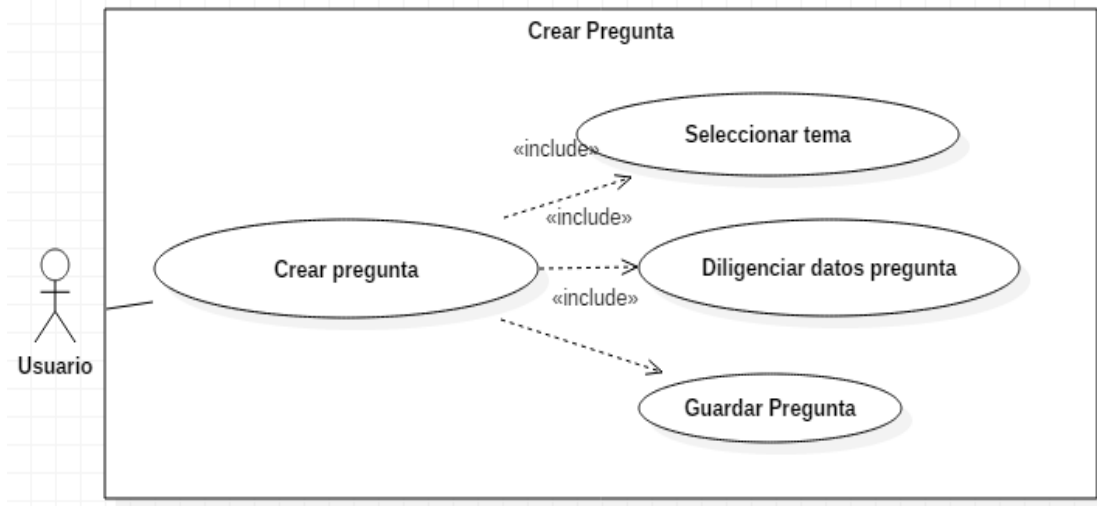
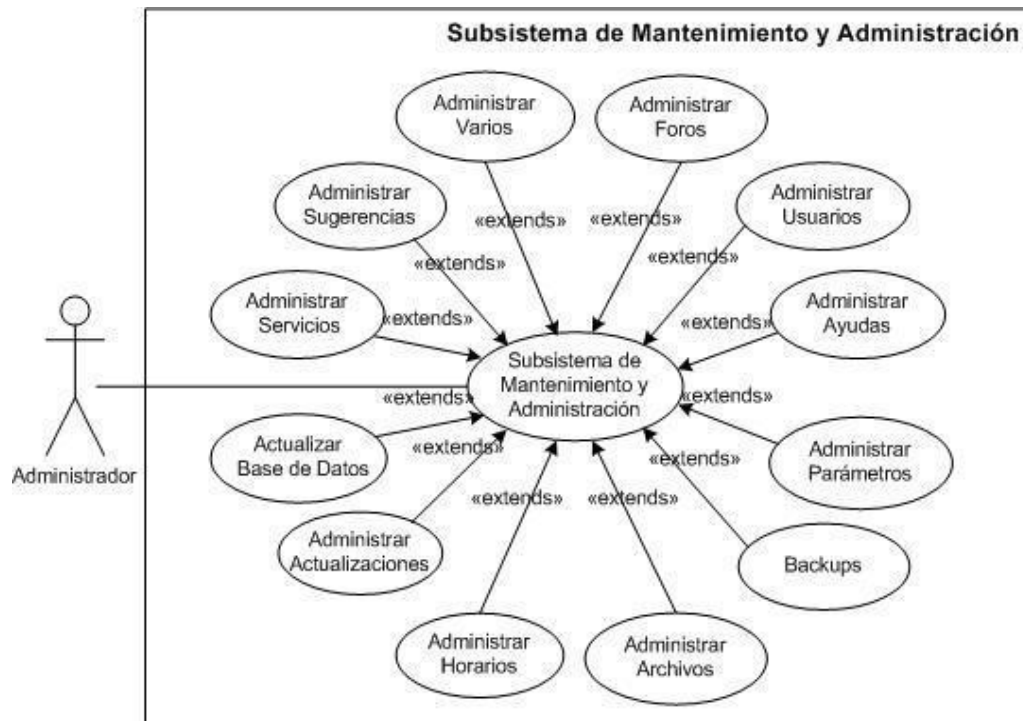


Figura 14. Diagrama de casos de uso: Crear pregunta.



- **Mantenimiento y Administración de los portales EIMTWEB y FIFMEWEB**

Figura 15. Diagrama de casos de uso: Subsistema de Mantenimiento y Administración.



**7.1.4 Documentación de casos de USO del sistema.** Una vez identificados los casos de uso, procedemos a documentar sus pasos, describiendo cada una de las acciones que se llevan a cabo en los casos de uso anteriormente mencionados. Los detalles completos se documentan en los anexos de este libro. (Ver Anexo E)

### 7.1.5 Diseño y Análisis.

#### 7.1.5.1 Diagrama Entidad/Relación de los servicios desarrollados.

- Servicio consultas y sugerencias

Figura 16. Diagrama E/R: Servicio consultas y sugerencias.

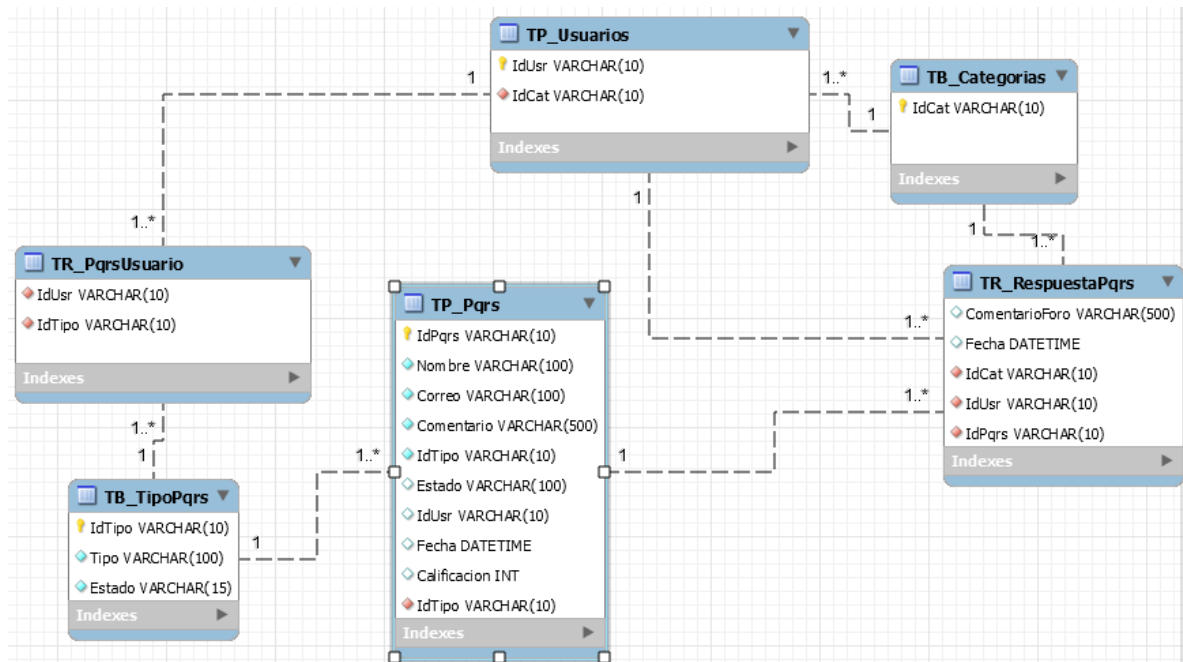


Figura 17. Diagrama E/R: Módulo buscador de proyectos.

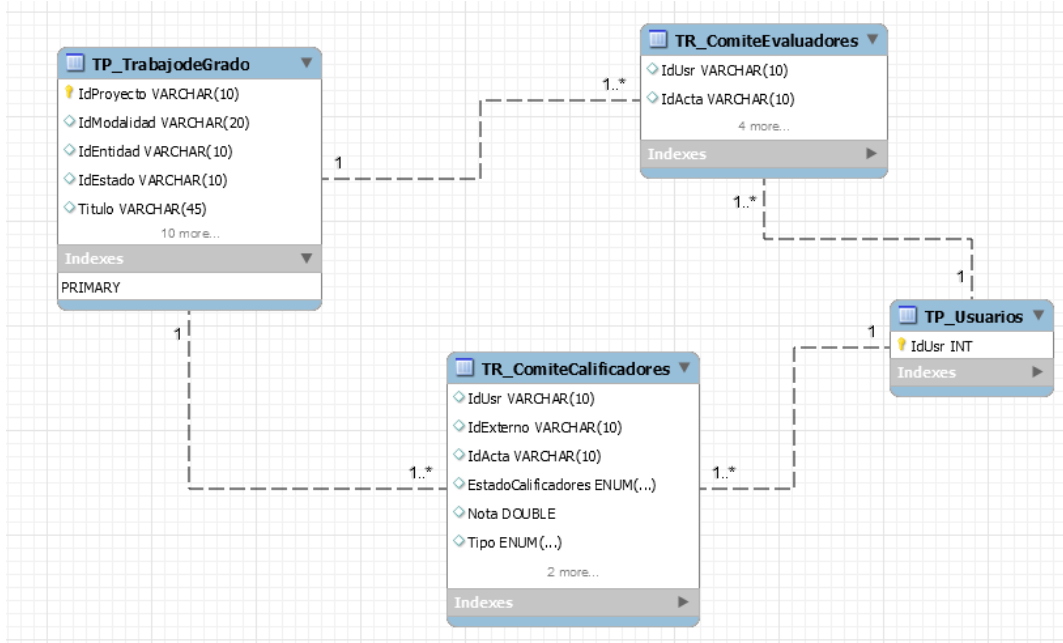
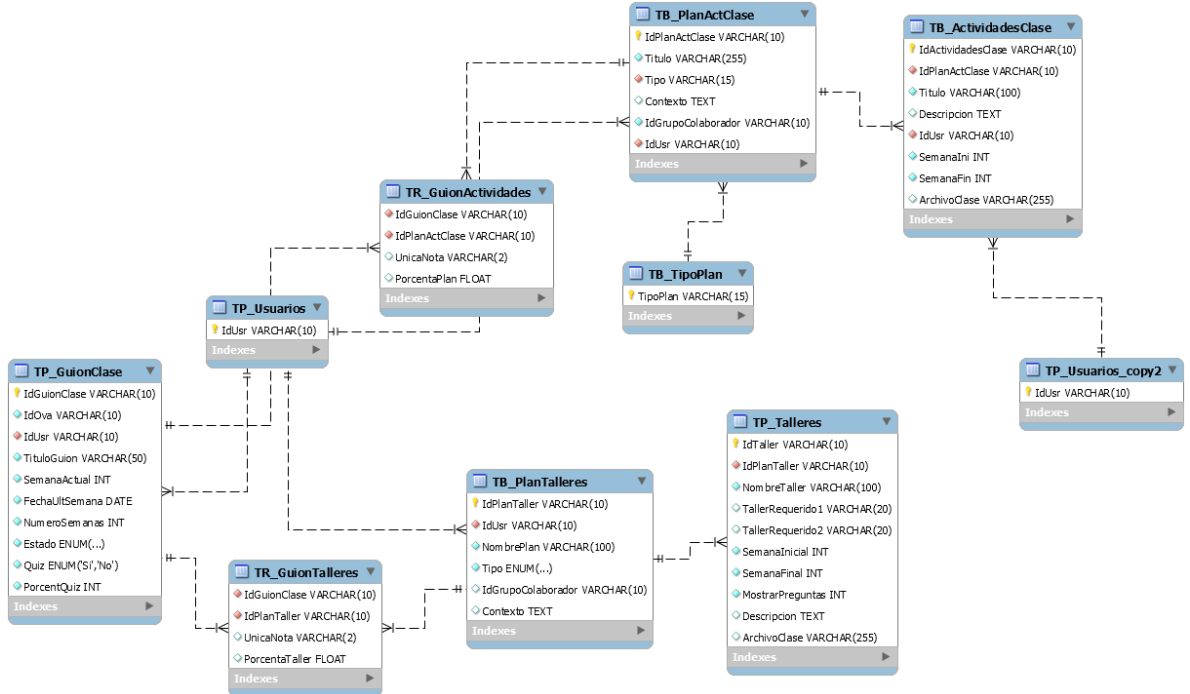


Figura 18. Diagrama E/R: Módulo Aula Virtual.



**7.1.5.2 Descripción de las Entidades.** Definición de las entidades utilizadas 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 EIMTWEB y E3TWeb.

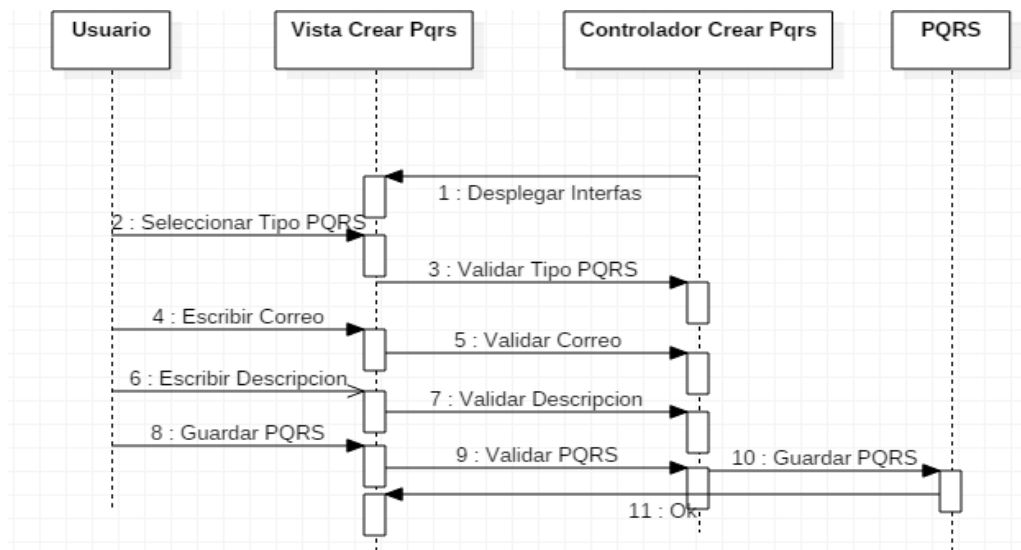
Tabla 1. Descripción de las Entidades

<b>ENTIDAD</b>	<b>DESCRIPCIÓN</b>
TP_Usuarios	Contiene la información de todos los usuarios registrados en el portal.
TR_RespuestasPqrs	Contiene las respuestas de los foros que se realizan en el módulo de foro.
TTR_PqrsUsuario	Contiene las personas encargadas de dar respuesta a las PQRs, según a la que sea asignada.
TB_TipoPqrs	Contiene todos los tipos de PQRs.
TP_Pqrs	Contiene las PQRs que se han realizado.
TP_TrabajodeGrado	Contiene toda la información de los proyectos de grado.
TR_ComiteEvaluadores.	Contiene la información del comité de evaluadores.
TR_ComiteCalificadores	Contiene la información del comité de calificadores.
TP_GuionClase	Contiene la información de los guiones o aulas virtuales creadas en el portal.
TR_GuionTalleres	Relaciona los planes de talleres que pertenecen a un guion.
TB_PlanTalleres	Contiene la información de cada plan de talleres
TR_GuionActividades	Relaciona los planes de actividades de clase que pertenecen a un guion.
TB_TipoPlan	Contiene los posibles planes que se puedan crear para un guion o aula determinada.
TP_Talleres	Contiene la información de cada taller que se crea para usar en el aula
TB_PlanActClase	Contiene la información de cada plan de actividades de clase que ha sido creado.
TB_ActividadesClase	Contiene las actividades de clase de cada aula virtual
TB_Categoria	Contiene la información de todas las categorías de usuarios del aula.

**7.1.6 Modelo de Procesos del Sistema.** Para una mejor interpretación de los modelos de los procesos del sistema, se realizaron los diagramas de secuencias necesarios para cada caso de uso, en los que se explica con detalle los pasos para el funcionamiento de cada uno de los servicios. (Ver Anexo F).

- **Servicio de Consultas y Sugerencias.**

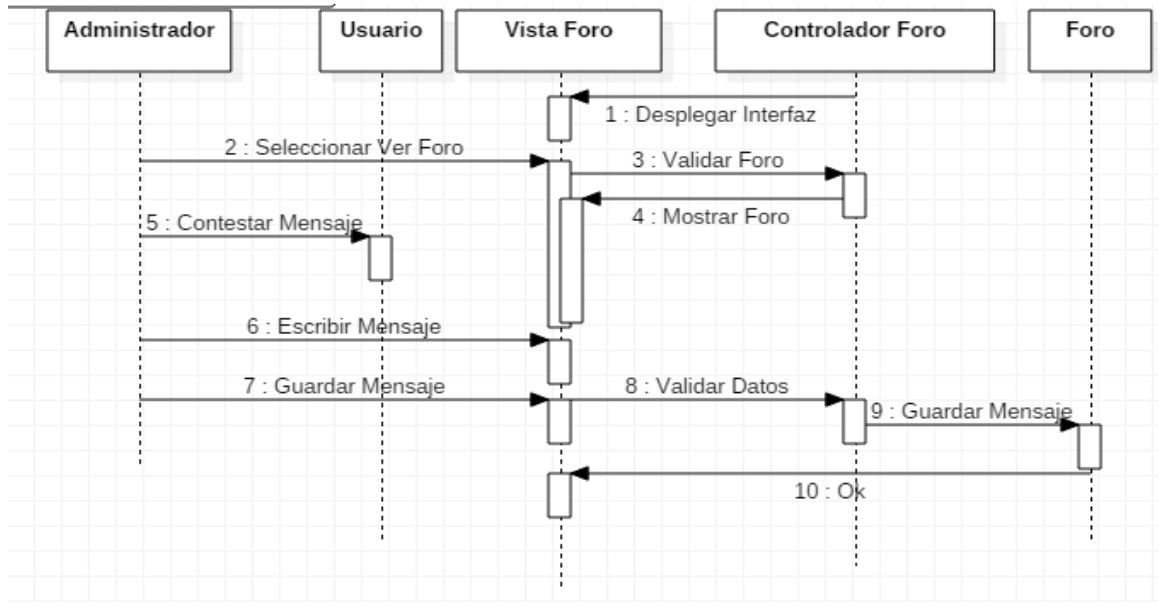
Figura 19. Diagrama de Secuencia: Crear PQRS en el servicio de Consultas y Sugerencias.



1. El controlador crear PQRs despliega la interfaz en vista crear PQRs.
2. El usuario selecciona un tipo en vista crear PQRs.
3. En el controlador crear PQRs se validan los datos ingresados del tipo de PQRs.
4. El usuario escribe el correo en la vista crear PQRs
5. Se valida el ingreso del correo en el controlador crear PQRs.
6. El usuario escribe la descripción de la PQRs en la vista crear PQRs.
7. Se valida la descripción de la PQRs en el controlador crear PQRs.
8. En la vista crear PQRs se guarda la PQRs.
9. Se valida los datos ingresados y el usuario en controlador crear PQRs.
10. El controlador crear PQRs guarda en la base de datos PQRs.

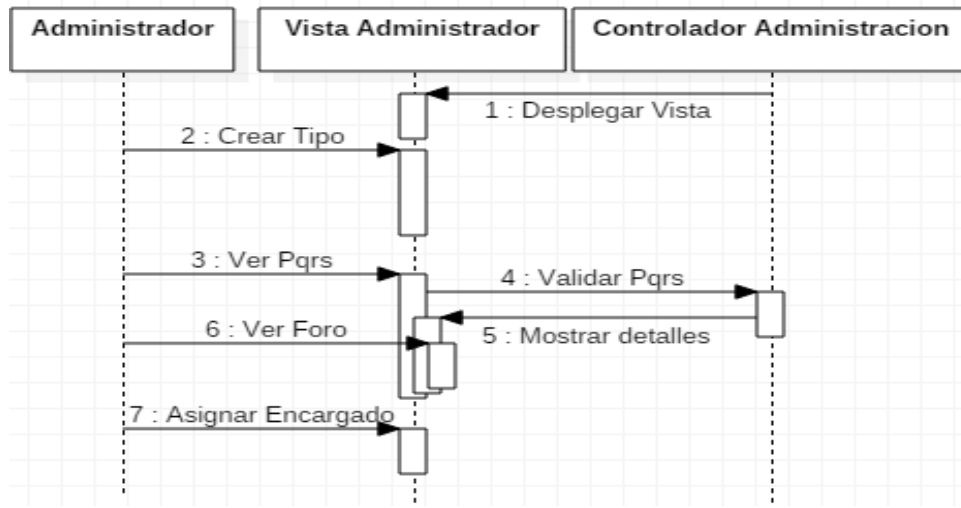
11. La base de datos retorna un mensaje de confirmación.

Figura 20. Diagrama de Secuencia: Enviar mensaje en el Foro del servicio de consultas y sugerencias



1. Se despliega la interfaz en la vista foro.
2. El administrador selecciona ver foro en la vista foro.
3. Se valida el evento al darle clic en ver foro.
4. El controlador mostrar foro muestra la vista foro.
5. El administrador contesta el mensaje al usuario.
6. El administrador escribe el mensaje en la vista foro.
7. El administrador guarda el mensaje en la vista foro.
8. Se valida el ingreso de los datos en el controlador foro.
9. Se guarda la información ingresada en la base de datos de foro.
10. Retorna un mensaje de confirmación en la vista de foro.

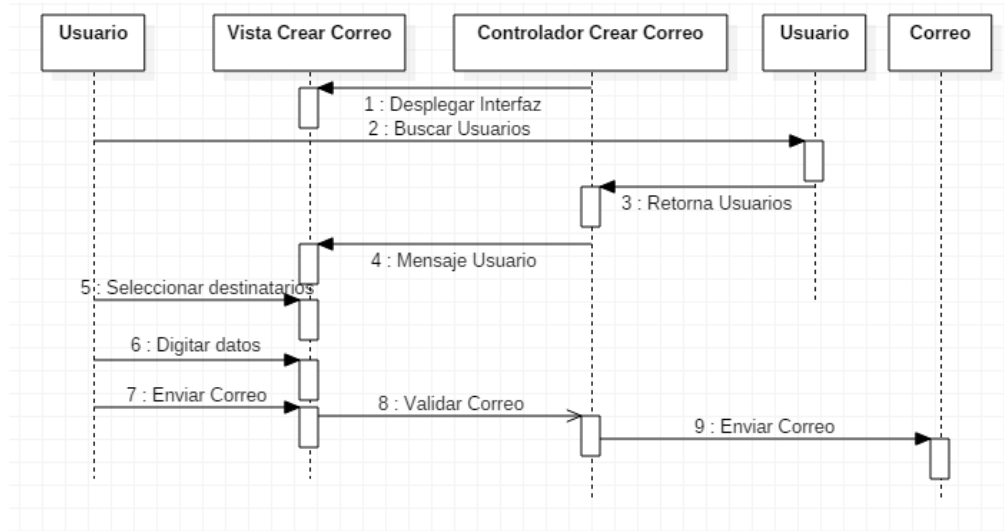
Figura 21. Diagrama de Secuencia: Administración de PQRS.



1. Despliega la interfaz en la vista administrador.
2. El administrador da clic en el evento crear tipo que se muestra en la vista administrador.
3. El administrador da clic en el evento ver PQRs en la vista administrador.
4. Se valida la información ingresada en el controlador administrador.
5. Muestra los detalles en la vista administrador.
6. El administrador da clic en el evento ver foro que se despliega en la vista administrador.
7. El administrador asigna los encargados en la vista administrador.

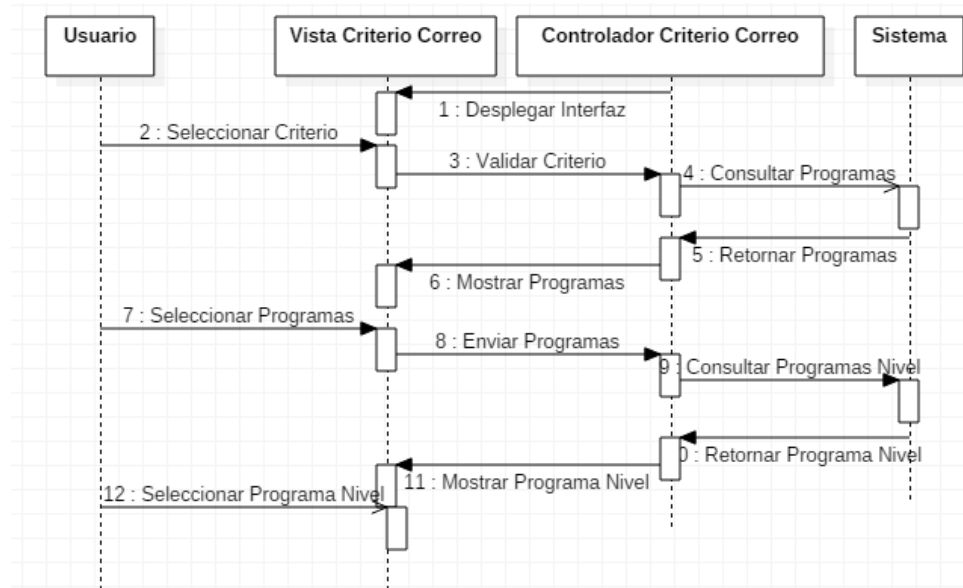
- **Correo Comunidad Escuela.**

Figura 22. Diagrama de Secuencia: Crear Correo.



1. Se despliega la interfaz en vista crear correo.
2. El usuario busca las personas en la base de datos Usuario.
3. La base de datos Usuario retorna los usuarios en el controlador crear correo.
4. El controlador despliega la información solicitada por el usuario en la vista crear correo.
5. El usuario selecciona los destinatarios en la vista crear correo.
6. El usuario digita los datos indicados en la vista crear correo.
7. El usuario da clic en el evento enviar correo que se despliega en la vista crear correo.
8. El controlador crear correo valida el correo que se va a enviar.
9. El controlador envía a la base de datos correo la información ingresada.

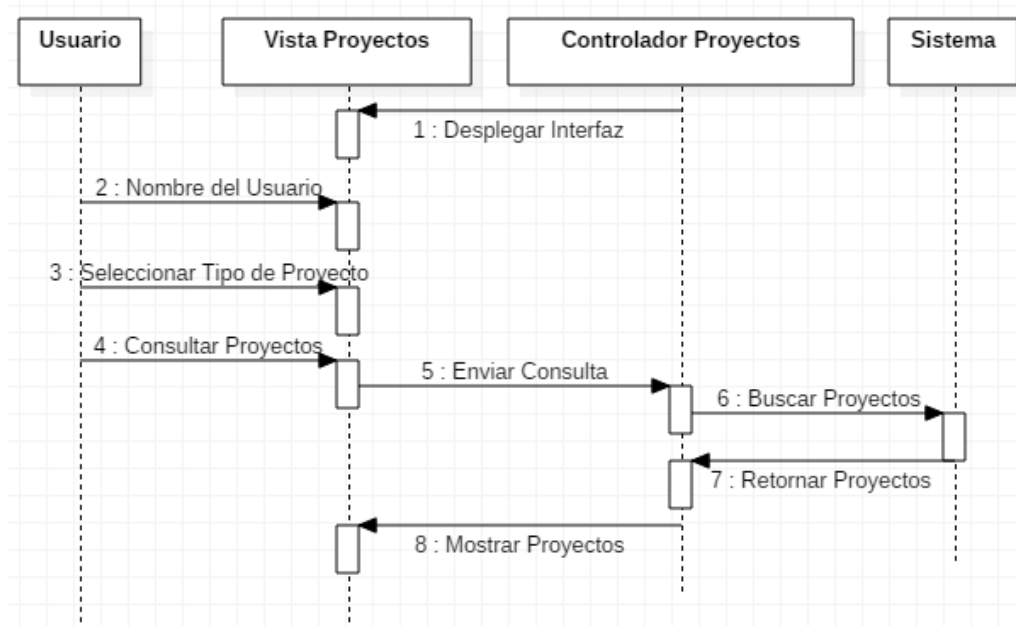
Figura 23. Diagrama de Secuencia: Selección criterio de correo.



1. El controlador criterio correo despliega la interfaz en vista criterio correo.
2. El usuario selecciona el criterio de la vista criterio controlador.
3. Se valida el criterio en el controlador criterio correo.
4. Se consulta en la base de datos los programas.
5. La base de datos retorna los programas los cuales llegan al controlador criterio correo.
6. El controlador criterio correo muestra los programas en la vista criterio correo.
7. El usuario selecciona e programa en la vista criterio correo.
8. La vista envía el programa a el controlador criterio correo para que sea validad la solicitud.
9. El controlador envía a la base de datos Sistema la consulta de los programas y niveles.
10. La base de datos retorna los programas y niveles en el controlador criterio correo.
11. El controlador criterio correo despliega en la vista criterio correo los programas con nivel.
12. El usuario selecciona los programas con nivel deseados.

- **Buscador de proyectos.**

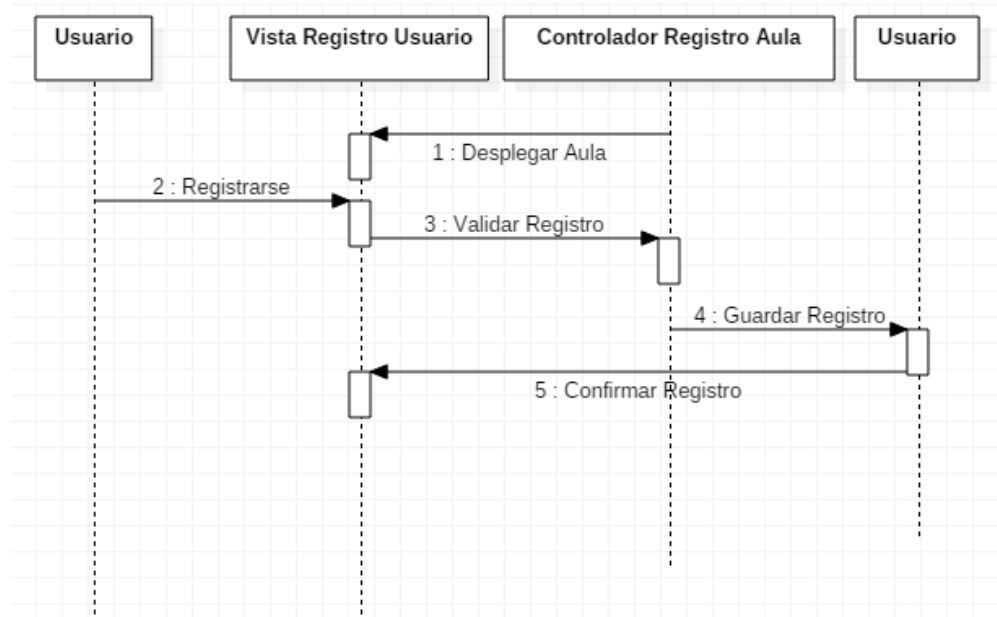
Figura 24. Diagrama de Secuencia: Consultar proyectos.



1. Se despliega la interfaz en la vista proyectos.
2. El usuario digita el nombre al cual desea buscar.
3. El usuario selecciona el tipo de proyecto que desea buscar en la vista proyectos.
4. El usuario da clic en el evento consultar proyectos en la vista proyectos.
5. La vista envía al controlador proyectos la consulta.
6. El controlador proyectos envía al Sistema la búsqueda en la base de datos según los criterio solicitados.
7. La base de datos retorna los resultados encontrados de los proyectos en el controlador proyectos.
8. El controlador proyectos muestra en la vista proyectos el resultado de la consulta.

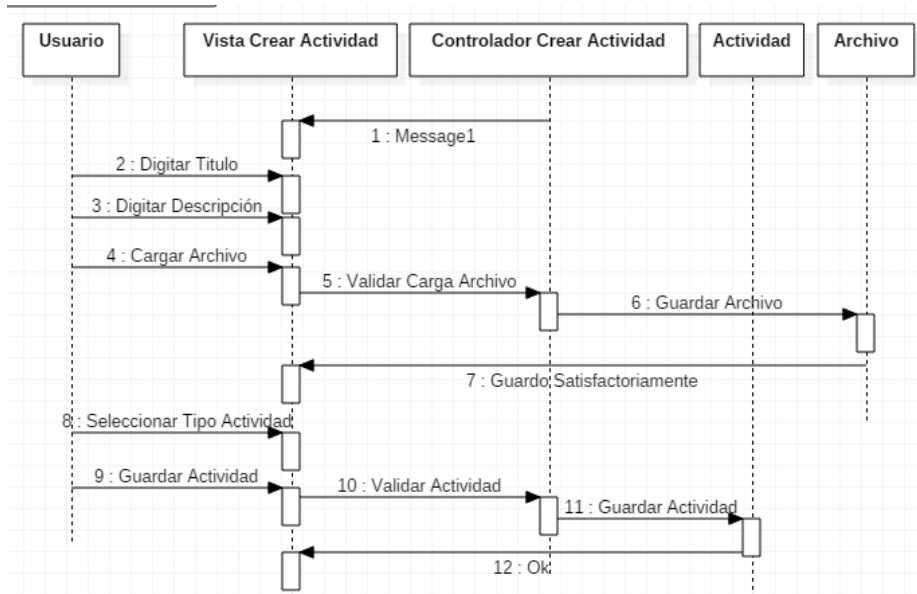
- **Aula virtual.**

Figura 25. Diagrama de Secuencia: Registrar estudiantes en el Aula.



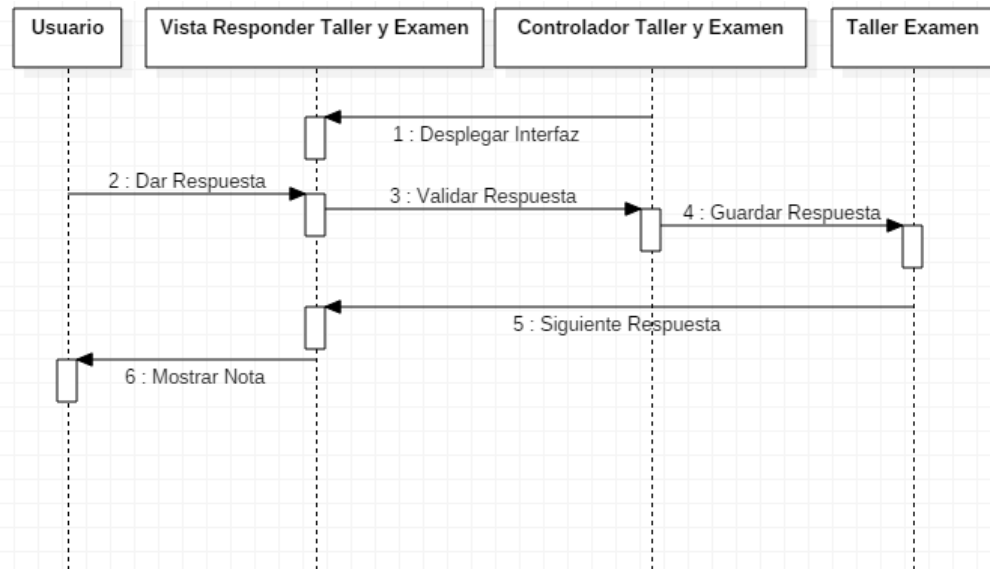
1. El controlador registro aula despliega la interfaz en la vista registro usuario.
2. El usuario da clic en el evento registrarse en la vista de registro usuario.
3. La vista registro usuario envía la validación al controlador registro aula.
4. El controlador envía a la base de datos la información ingresada para que se guarde en usuario.
5. La base de datos despliega un mensaje de confirmación en la vista registro usuario.

Figura 26. Diagrama de Secuencia: Crear actividad.



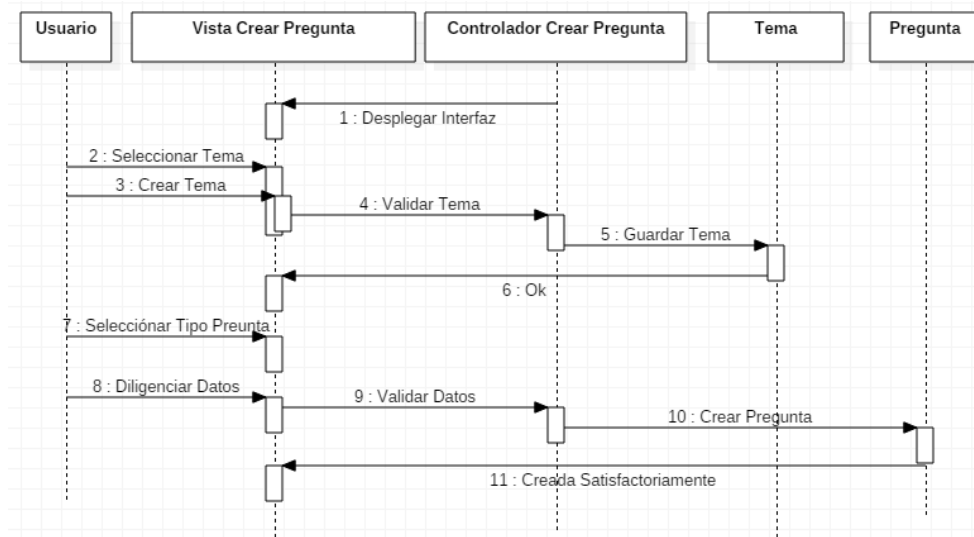
1. El controlador crear actividad despliega la interfaz en la vista crear actividad.
2. El usuario digita el título de la actividad en la vista crear actividad.
3. El usuario digita la descripción de la actividad.
4. El usuario carga los archivos deseados en la vista crear actividad.
5. El controlador crear actividad valida la carga de archivos.
6. El controlador crear actividad envía los archivos a guardar en la base de datos archivo.
7. La base de datos archivo retorna un mensaje de confirmación en la vista crear actividad.
8. El usuario selecciona los tipos de actividades en la vista crear actividad.
9. El usuario da clic en el evento guardar actividad que está en la vista crear actividad.
10. El controlador crear actividad valida los datos ingresados en la vista crear actividad.
11. El controlador crear actividad guarda la información en la base de datos.
12. La base de datos actividad devuelve un mensaje de confirmación.

Figura 27. Diagrama de Secuencia: Responder taller y examen.



1. El controlador despliega la interfaz en la vista responder taller y examen.
2. El usuario da clic en dar respuesta en la vista responder taller y examen.
3. El cotrolador taller yexamen valida la respuesta que el usuario digito en la vista.
4. El controlador envia a la base de datos taller y examen la respuesta para que sea guardada.
5. La base de datos taller y examenes envia un mensaje de confirmacion.
6. La vista responder taller y examen muestra la nota al usuario.

Figura 28. Diagrama de Secuencia: Crear pregunta.



1. El controlador crear pregunta despliega la interfaz en la vista crear pregunta.
2. El usuario de clic en el evento seleccionar tema en la vista crear pregunta.
3. El usuario da clic en crear tema en la vista vrear pregunta.
4. El controlador valida el tema que ingreso el usuario.
5. El controlador envia la base de datos tema el dato que a ingresado a guardar.
6. La base de datos envia un mensaje de confirmacion en la vista crear pregunta.
7. El usuario selecciona el tipo de pregunta en la vista crear pregunta.
8. El usuario digita los datos en la vista crear pregunta.
9. La vista crear pregunta envia al controlador los datos ingresados par que valide la informacion.
- 10.El controlador envia a la base de datos pregunta la informaicon para que sea ingresada.
- 11.La base de datos envia un mensaje de confirmacion ala vista de crear pregunta.

**7.1.7 Implementación, Implantación y Pruebas Generales.** Para la implementación de los servicios se utilizaron las siguientes herramientas:

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

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

Para el desarrollo se trabajó con la base de datos "Diamante" ubicada en todos los servidores de los portales de las escuelas, y con la base de datos centralizada "Poseidón" ubicada en el servidor del portal web de la 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.

## **7.2 MANTENIMIENTO Y ADMINISTRACIÓN**

**7.2.1 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

**7.2.2 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.

## 8. PRUEBAS DEL SISTEMA

Para garantizar el correcto desarrollo de los servicios realizados, se realizaron las siguientes pruebas:

### 8.1 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.

**8.1.1 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 G).

### 8.2 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.

### **8.3 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.

## 9. CONCLUSIONES

- El proyecto realizado ha contribuido de manera favorable, dando soporte a las escuelas gracias al mejoramiento de los módulos a través a la implementación de los nuevos servicios que han dado respuesta a las necesidades de los usuarios.
- La implementación del sistema de Consultas y Sugerencias, ha contribuido a que los usuarios se integren y puedan junto a personas asignadas resolver las diferentes PQRS que puedan realizar y con la nueva herramienta de foros el usuario podrá conversar con el administrador.
- La implementación de nuevas opciones en el servicio de Correos comunidad Escuela ayudara a los diferentes usuario poder enviar correos a toda la comunidad teniendo en cuenta filtros como estudiantes por carrera y nivel entre otros.
- La implementación del buscador de proyectos tanto dirigidos, evaluados y calificados, ayudará a los profesores y directores de proyecto consultar los proyectos de los cuales están asignado
- Implementar dentro del aula virtual la creación de actividades evaluables durante el curso.
- La implementación de crear actividades evaluables dentro del aula virtual y la posibilidad de poder asignar a varios profesores a la misma aula facilitara el manejo de la misma y la unión de la comunidad.

## 10. RECOMENDACIONES

- Para mantener una buena comunicación con los usuarios de los portales, es fundamental dar pronta y precisa respuesta a las sugerencias de los mismos. De esta manera se logra dar soporte oportuno a sus inquietudes.
- Se recomienda capacitar a los usuarios para que utilicen de una manera óptima los nuevos servicios, y con ellos ahorren tiempo y agilicen sus tareas relacionadas a los mismos.
- Se recomienda dar continuidad y promocionar el portal de producción intelectual, el cual es muy útil para mostrar a las personas interesadas en las diferentes producciones generadas en la universidad, y poder visualizarlas dependiendo de cada escuela, profesor o grupo de investigación. Debido a que esto generará un reconocimiento al compromiso y entrega de los implicados en el desarrollo de dichas producciones.

## BIBLIOGRAFÍA

GISSET. Información acerca de la Arquitectura Cliente Servidor. [en línea]. <http://g701giadar.wikispaces.com/page/diff/Arquitctura+Cliente+Servidor/229968658> [citado el 01 Junio de 2014]

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.

ORACLE. Api de Java [en línea]. <http://docs.oracle.com/javase/7/docs/api/> [citado el 01 Junio de 2014]

ORACLE CORPORATION. MySQL Documentation: MySQL Reference Manuals.[en línea]. <http://dev.mysql.com/doc/> [citado el 06 Febrero de 2014]

PIATTINI, Mario, CALVO-MANZANO, José A., CERVERA, Joaquín, FERNANDEZ, Luis. Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Alfaomega, 2000.

PONCE DE LEÓN, Diego. Tutoriales HTML. [en línea]. <http://www.htmlquick.com/es/tutorials.html> [citado el 01 Mayo de 2014]

RICH, Clark, et al. Html element index [online]. Available from Internet: <http://html5doctor.com/element-index/> [cited el 03 Abril de 2014]

SCHMULLER, Joseph. Aprendiendo UML en 24 horas. 1ª ed. México: Alhambra Mexicana S.A, 2000.

STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005.

STUMPF, Robert, TEAGUE, Lavette. Object-Oriented Systems Analysis and Design with UML. Prentice Hall. 2004.

THE JQUERY FOUNDATION. Api de la librería jQuery. [en línea]. <http://api.jquery.com/> [citado el 18 Mayo de 2014]

UNIVERSIDAD INDUSTRIAL DE SANTANDER. Grupo Desarrollador Calumet. Algunas herramientas de recursividad y utilidad para cuando se codifica JavaScript. [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 07 Mar. 2014]. Disponible en Internet: <http://cormoran.uis.edu.co/eisi/Estandar/tools.jsp#js>

----- . ¿Cómo extender y profundizar en herramientas para crear aplicaciones web front-end avanzadas? [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 12 Feb. 2014]. Disponible en Internet: <http://cormoran.uis.edu.co/eisi/Estandar/tools.jsp#liberias>

----- . Componentes de interfaz de usuario dinámica e interactiva para crear aplicaciones. [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre

----- . Estilos y diseños básicos y complementarios para crear componentes de la Librería Elise. [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 20 Feb. 2014]. Disponible en Internet: <http://cormoran.uis.edu.co/eisi/Estandar/design.jsp>

----- . Guía de documentación de selectores, eventos, manipulación y procedimientos de jQuery. [en línea] [Bucaramanga, Colombia]: Elise Library V2.

Noviembre 2014. [Citado 10 Nov. 2014]. Disponible en Internet:  
<http://www.oscarotero.com/jquery/>

----- . Herramientas adicionales en el proceso de testear y codificar servicios y aplicaciones. [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 10 Nov. 2014]. Disponible en Internet:  
<http://cormoran.uis.edu.co/eisi/Estandar/tools.jsp#otros>

----- . Herramientas JavaScript y JQuery para la creación de procedimientos para aplicaciones [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 16 Mar. 2014]. Disponible en Internet:  
<http://cormoran.uis.edu.co/eisi/Estandar/codes.jsp>

----- . Herramientas para facilitar y agilizar la creación de diseños para objetos HTML con CSS. [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 12 Feb. 2014]. Disponible en Internet:  
<http://cormoran.uis.edu.co/eisi/Estandar/tools.jsp#librerias>

----- . Plantillas estándar para crear servicios, aplicaciones o cualquier proyecto para Calumet. [en línea] [Bucaramanga, Colombia]: Elise Library V2. Noviembre 2014. [Citado 05 Feb. 2014]. Disponible en Internet:  
<http://cormoran.uis.edu.co/eisi/Estandar/templates.jsp>

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

# **ANEXOS**

## Anexo A. Arquitectura Cliente/Servidor

### 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.

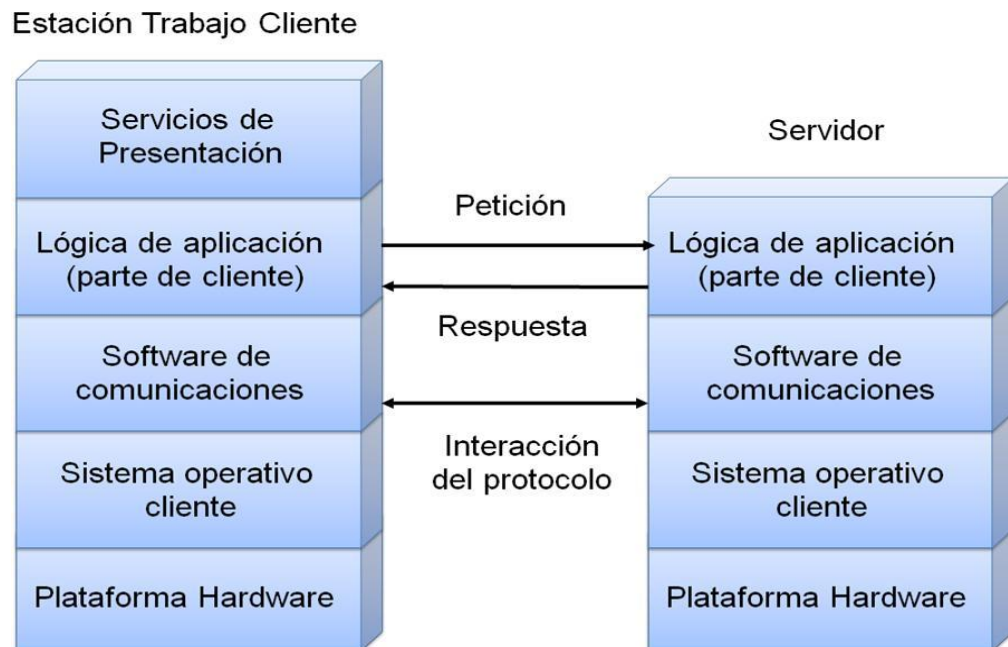
A continuación se muestra la clasificación de los sistemas Cliente / Servidor:

## 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.

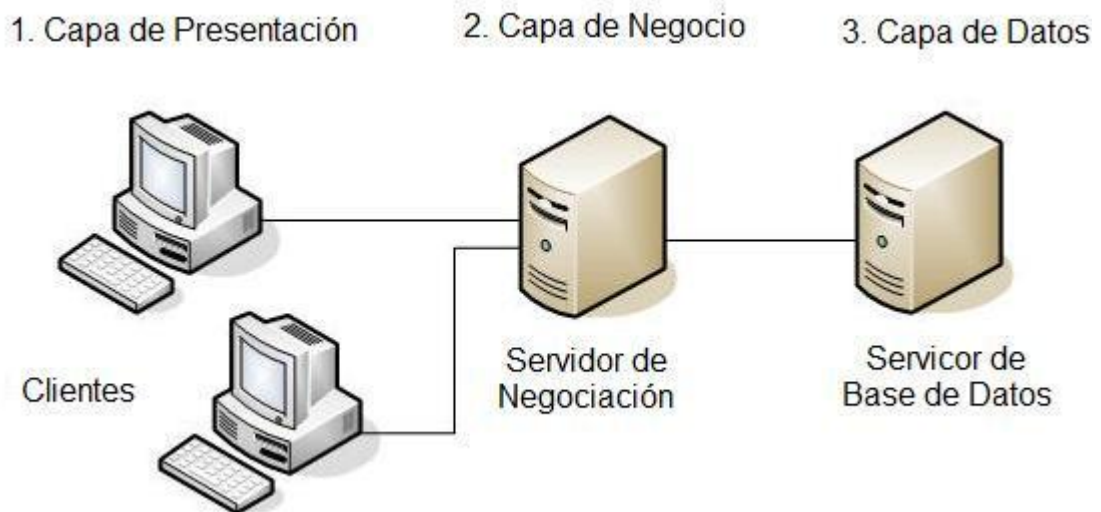


Arquitectura Cliente/Servidor de Dos Capas.

## 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.



Arquitectura Cliente/Servidor de Tres Capas

## **Anexo B. Tecnologías de desarrollo de páginas web dinámicas**

### **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**

### **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:

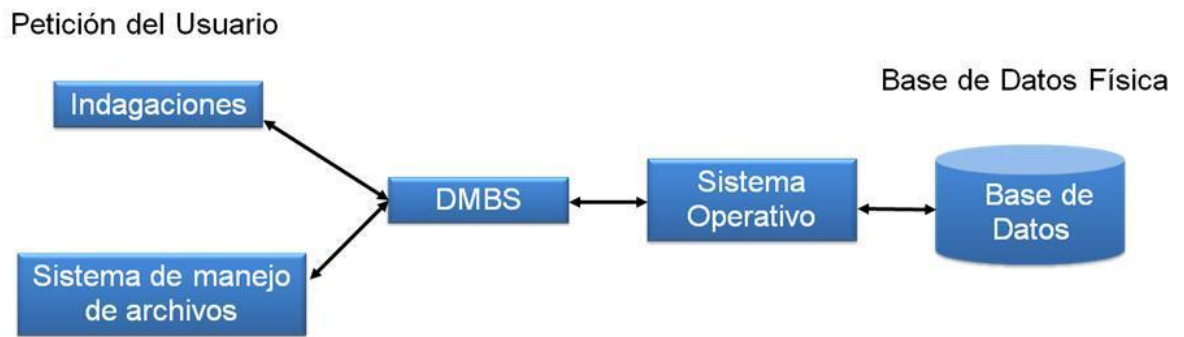
- Cargar un controlador JDBC.
- Utilizar el controlador para abrir una conexión con la base de datos.
- Formular instrucciones SQL a través de la conexión.
- Procesar los conjuntos de resultados devueltos por las operaciones SQL.

#### **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.



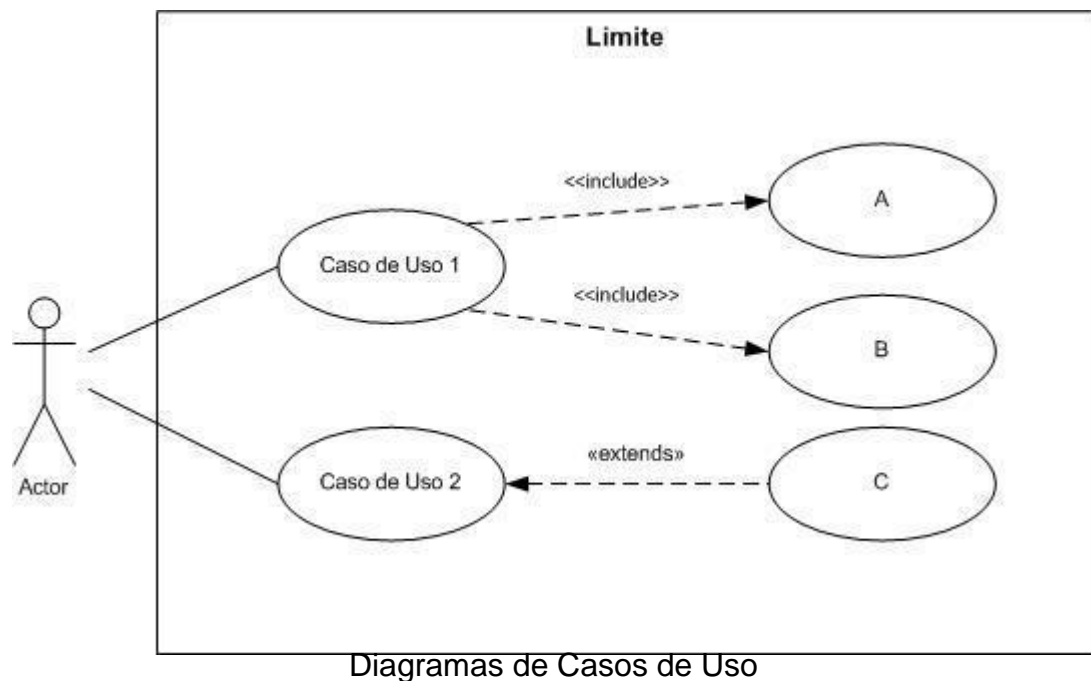
Función del DBMS

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)

### LEGUAJE DE MODELADO UNIFICADO (UML)

#### 1. DIAGRAMAS DE CASOS DE USO



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

En este proyecto se destacan las siguientes 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 EIMTWEB, FIFMEWEB:** Es el tipo de usuario común de los portales

EIMTWEB y FIFMEWEB y a quien van dirigidos los servicios. Este usuario solo tiene control sobre sus privilegios.

- **Sistema:** Es el sistema encargado de realizar determinadas funciones, que le solicita el usuario.

**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.

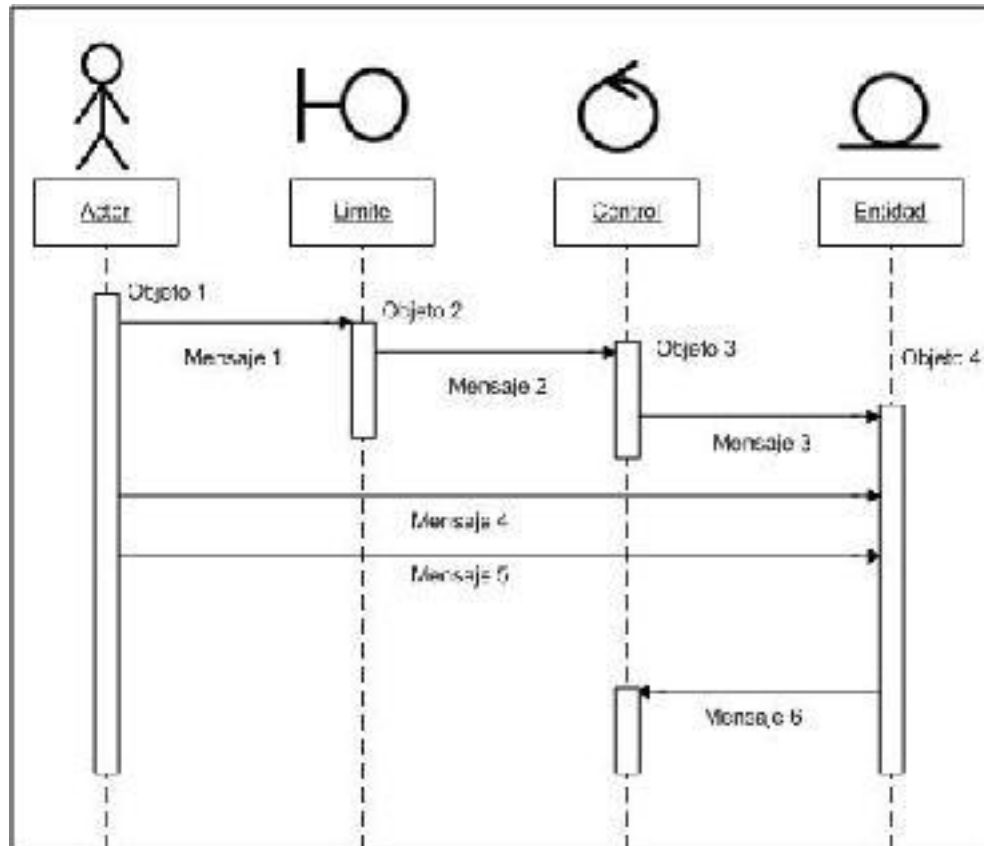


Diagrama de Secuencias

## Anexo E. Documentación de casos de uso del sistema

### DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA

#### 1. SERVICIO DE CONSULTAS Y SUGERENCIAS

Casos de uso: Crear PQRS en el servicio de Consultas y Sugerencias

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Crear PQRS	Despliega todas las opciones para crear las PQRs.
Seleccionar Tipo	Permite seleccionar el tipo de PQRs las cuales se despliegan para su selección.
Diligenciar Datos	Permite al usuario definir los datos de la PQRs que desea diligenciar.
Guardar PQRs	Permite al usuario guardar la PQRs con todas las características definidas.

Enviar mensaje en el Foro del servicio de consultas y sugerencias

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Ver Foro	En esa sección se muestra el foro con todas las respuestas tanto del administrador como del mismo usuario.
Contestar mensaje	Permite responder a un mensaje que fue dejado por el administrador.

Casos de uso: Administración de PQRS

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Crear Tipo	Permite crear un tipo de consulta.
Ver PQRS	Permite ver la PQRs que fue creada.
Asignar Encargado	Permite asignar un usuario encargado de responder un tipo específico de PQRs.
Responder Foro	Permite al usuario dar respuesta en el foro.

## 2. CORREO COMUNIDAD ESCUELA

Casos de uso: Crear Correo

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Crear correo	Despliega todas las opciones para crear un correo.
Seleccionar destinatario	Permite seleccionar los usuarios a los cuales se les va a enviar el correo.
Digital asunto y mensaje	Permite digitar el asunto y el mensaje.
Enviar correo	Permite enviar el correo validando todos los datos ingresados.

Casos de uso: Selección criterio de correo

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Seleccionar criterio	Despliega todas las opciones de envío de correo.
Seleccionar programa nivel	Permite seleccionar los programas a los cuales se desea enviar correo.
Seleccionar programa	Permite seleccionar la Carrera a la cual desea enviar el correo.
Consultar programa	Permite hacer la consulta de usuarios según su especificación de búsqueda.
Mostrar programa	Muestra todos los programas.

## 3. BUSCADOR DE PROYECTOS

Casos de uso: Consultar proyectos

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Consultar proyectos	Despliega todas las opciones de consulta de correos.
Criterio de búsqueda	Permite seleccionar los diferentes tipo de consultas
Dirigidos.	Muestra a los proyectos de los cuales dirige.
Evalutados	Muestra los proyectos de los cuales es evaluador.
Calificados	Muestra los proyectos de los cuales son calificadores.
Mostrar proyectos	Muestra todos los proyectos según el criterio de búsqueda

#### 4. AULA VIRTUAL

##### Casos de uso: Registrar estudiantes en el Aula

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Registrar usuario aula	Permite al usuario registrarse en un guion ya creado.
Validar usuario	Valida los datos que el usuario está registrando y define si el usuario existe o no.
Enviar datos usuario	Enviar los datos que el usuario diligencia en el registro.

##### Casos de uso: Crear actividad

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Crear actividad	Permite crear una actividad de clase de acuerdo a las necesidades que el usuario tenga.
Seleccionar tipo actividad	Permite al usuario escoger entre qué tipo de actividad desea crear.
Diligenciar datos actividad	Permite al usuario definir los detalles de la actividad.
Guardar actividad	Permite al usuario guardar la actividad que ha creado con todas las especificaciones.

##### Casos de uso: Responder taller y examen

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Responder taller y examen	Permite al usuario responder los talleres y exámenes.
Enviar respuesta	Permite al estudiante enviar y guardar la respuesta que da para cada pregunta y examen.
Dar respuesta	El usuario da respuesta para cada pregunta del taller o examen.

Casos de uso: Crear pregunta

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Crear pregunta	Permite al usuario crear una pregunta de acuerdo a las necesidades.
Seleccionar tema	Permite al usuario escoger entre todos los temas de la pregunta.
Diligenciar datos pregunta	Permite al usuario definir los detalles de la pregunta.
Guardar pregunta	Permite al usuario publicar y guardar las preguntas que fueron creadas.

## 5. MANTENIMIENTO Y ADMINISTRACIÓN DE LOS PORTALES EIMTWEB Y FIFMEWEB

Caso de Uso: Mantenimiento y Administración de los portales EIMTWEB y FIFMEWEB.

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
Administrar Servicios	Crear, modificar y eliminar servicios del portal. Establecer parámetros.
Actualizar Base de Datos	Actualizar tablas de datos de División y Diamante. Ver contadores de registros.
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 visitar 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 Cambiar clave Autorizar servicios a categorías y perfiles Crear, modificar, eliminar categorías y perfiles Enviar correos

<b>TÍTULO</b>	<b>FUNCIONES PRIMARIAS</b>
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 material.
Administrar Actualizaciones	Ver historial y estadísticas de cambios del portal web.

## Anexo F. Pruebas del sistema

### PRUEBAS DEL SISTEMA

#### 1. SISTEMA DE NOTIFICACIONES PARA EVENTOS Y FOROS

Pruebas Realizadas: Crear PQRS en el servicio de Consultas y Sugerencias.

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Crear PQRS	Verifica que el enlace sea enviado correctamente y que despliegue las opciones indicadas	✓
Seleccionar Tipo	Se verifica que se desplieguen todas las opciones indicadas y que se puedan seleccionar.	✓
Diligenciar Datos	Se verifica que se validen todos los campos y que concuerden con la información ingresada.	✓
Guardar PQRS	Se verificó que se validaran todos los datos y se guarde la PQRS correctamente.	✓

Pruebas Realizadas: Enviar mensaje en el Foro del servicio de consultas y sugerencias

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Ver Foro	Se comprobó que mostrara la PQRS indicada junto con los mensajes del foro.	✓
Contestar mensaje	Se valida el envío del mensaje en el foro y se verifica de que el correo haya llegado.	✓

Pruebas Realizadas: Administración de PQRS.

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Crear Tipo	Al crear un tipo se valida de que se digiten todos los campos.	✓
Ver PQRS	Se valida de que muestre la PQRS que sea indicada.	✓
Asignar	Realiza la búsqueda indicada mostrando los	✓

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Encargado	usuarios a los cuales puede asignar, escogiéndolos y guardando en la base de datos la información.	
Responder Foro	Valida la información ingresada guardándola en la base de datos y mostrándola en el foro.	✓

## 2. CORREO COMUNIDAD ESCUELA

Pruebas Realizadas: Crear Correo.

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Crear correo	Permite crear correos y validando todos los campos	✓
Seleccionar destinatario	Permite seleccionar el (los) destinatario(s) para poder enviar los correos.	✓
Digitar asunto y mensaje	Validar todos los campos para que se puedan digitar valores establecidos.	✓
Enviar correo	Envía el correo al destinatario y se verifica que el correo llego correctamente.	✓

Pruebas Realizadas: Selección criterio de correo.

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Seleccionar criterio	Se realiza la búsqueda correctamente en la base de datos y se verifica que se puedan seleccionar los criterios correctamente.	✓
Seleccionar programa nivel	Se verifica que liste los programas son su nivel específico y que sea el nivel según el criterio seleccionado.	✓
Seleccionar programa	Permite seleccionar la Carrera a la cual desea enviar el correo.	✓
Consultar programa	Se verifica que consulte la base de datos los programas que se seleccionaron.	✓
Mostrar programa	Se verifica que se listen todos los programas	✓

### 3. BUSCADOR DE PROYECTOS

Pruebas Realizadas: Consultar proyectos

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Consultar proyectos	Se verifica que se consulte en la base de datos los proyectos con los criterios seleccionados.	✓
Criterio de búsqueda	Se comprobó que consultara a la base de datos según el criterio de búsqueda.	✓
Dirigidos.	Se verifica que los datos que muestra sean de los proyectos dirigidos.	✓
Evaluados	Se verifica que los datos que muestra sean de los proyectos evaluados.	✓
Calificados	Se verifica que los datos que muestra sean de los proyectos calificados.	✓
Mostrar proyectos	Se verifica que se consulte la base de datos correctamente y liste todos los proyectos.	✓

### 4. AULA VIRTUAL

Pruebas Realizadas: Registrar estudiantes en el Aula

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Registrar usuario aula	Se registran diferentes tipos de usuario correctamente, verificando que pertenezcan al aula	✓
Validar usuario	Valida correctamente el usuario, si es un usuario Nuevo guarda satisfactoriamente todos los datos.	✓
Enviar datos usuario	Se envían satisfactoriamente los datos de registro.	✓

Pruebas Realizadas: Crear actividad

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Crear actividad	Se verifica de que se ingrese correctamente la información y se abran los jsp adecuados	✓
Seleccionar tipo actividad	Se verifica que se pueda seleccionar los tipo de actividades y que le usuario pueda seleccionarlas.	✓
Diligenciar datos actividad	Se validan correctamente los datos de la actividad.	✓
Guardar actividad	Se confirma que la actividad se guardó correctamente.	✓

Pruebas Realizadas: Responder taller y examen

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Responder taller y examen	Se confirma que se pueda responder los talleres y exámenes correctamente y guardándolos en la base de datos.	✓
Enviar respuesta	Se envía la respuesta de forma correcta y se verifica que la información enviada pueda ser vista por el profesor.	✓
Dar respuesta	Se confirma que el usuario logueado es el director del aula y la respuesta sea enviada con satisfacción.	✓

Pruebas Realizadas: Crear pregunta

<b>CASO DE USO</b>	<b>PRUEBA REALIZADA</b>	<b>RESULTADO</b>
Crear pregunta	Se verifica de que el usuario que esta logueado es el director del aula y pueda ingresar a crear la pregunta correctamente.	✓
Seleccionar tema	Se verifica que se pueda seleccionar el tema correctamente.	✓
Diligenciar datos pregunta	Se validan todos los datos ingresados por parte de usuario.	✓
Guardar pregunta	Se verifica que los datos ingresado se hayan guardado correctamente y que se le haya enviado un mensaje al director del curso.	✓