

**ANÁLISIS, MANTENIMIENTO, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DE NUEVAS FUNCIONALIDADES Y MEJORAS RELACIONADAS CON LOS
MÓDULOS DE PQRS, HOJA DE VIDA Y PORTAL WEB DE PROFESORES EN
LA PLATAFORMA COMA COMUNIDAD ACADÉMICA.**

CARLOS ANTONIO DÍAZ FLÓREZ

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

2017

**ANÁLISIS, MANTENIMIENTO, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DE NUEVAS FUNCIONALIDADES Y MEJORAS RELACIONADAS CON LOS
MÓDULOS DE PQRS, HOJA DE VIDA Y PORTAL WEB DE PROFESORES EN
LA PLATAFORMA COMA COMUNIDAD ACADÉMICA.**

CARLOS ANTONIO DÍAZ FLÓREZ

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

Director

**Msc. LUIS IGNACIO GONZÁLEZ RAMÍREZ
Magíster en Informática**

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

2017

AGRADECIMIENTOS

A Dios por estar siempre en cada paso que doy en la vida, mostrándome el camino a seguir, gracias por darme la fuerza para terminar este proyecto y hacer este sueño realidad.

A mis padres por el esfuerzo tan grande que han hecho para que pueda terminar mis estudios, el apoyo incondicional a lo largo de mi vida, y sobre todo en los momentos difíciles, son unos excelentes padres y un gran ejemplo a seguir.

A mi hermano que me dio su apoyo en los momentos que le he solicitado y ha respondido de la mejor manera.

A mis profesores que me han formado profesionalmente y compartido sus conocimientos para aprender cada día más.

Al profesor Luis Ignacio González quien fue mi director de proyecto, que no solo fue una guía sino también como un amigo quien confió en mi para llevar a cabo actividades del grupo. Por su manera de ser tan abierta y hablar de muchos temas tanto académicos como personales lo hace ser una excelente persona.

A mis compañeros del grupo Calumet con los cuales compartí momentos agradables de esparcimiento, a los que me orientaron en este proyecto y como funciona las diferentes herramientas de trabajo.

A mis amigos de la universidad por compartir grandes momentos de diversión y hacer que la vida universitaria fuera muy alegre.

CONTENIDO

INTRODUCCIÓN	13
1. PLANTEAMIENTO DEL PROYECTO	14
1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL	14
1.2 JUSTIFICACIÓN	14
1.3 OBJETIVOS	16
1.4 IMPACTO Y VIABILIDAD	18
2. MARCO TEÓRICO	19
2.1 ARQUITECTURA CLIENTE/SERVIDOR	19
2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS	23
2.3 BASES DE DATOS	26
2.4 NETBEANS	28
2.5 SISTEMA DE CONTROL DE VERSIONES	29
2.6 PROGRAMACIÓN UTILIZADA	30
2.7 SERVIDORES WEB	32
3. MARCO METODOLÓGICO	34
3.1 PROTOTIPO EVOLUTIVO	34
3.2 LENGUAJE DE MODELADO UNIFICADO	36
3.3 ESTÁNDARES DE PROGRAMACIÓN	40
4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO	42
4.1 PROTOTIPO ESPERADO	42

4.2 DIAGRAMAS DE CASOS DE USO	45
4.3 DOCUMENTACION DE CASOS DE USO DEL SISTEMA	51
4.4 DISEÑO Y ANÁLISIS.....	57
4.5 DESCRIPCIÓN DE LAS ENTIDADES.....	60
4.6 MODELO DE PROCESOS DEL SISTEMA.....	60
4.7 IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES.....	61
4.8 MANTENIMIENTO Y ADMINISTRACIÓN.....	62
5. PRUEBAS DEL SISTEMA	64
5.1 PRUEBAS DE VERIFICACIÓN	64
6. CONCLUSIONES	69
7. RECOMENDACIONES.....	70
BIBLIOGRAFÍA.....	71
ANEXOS	73

LISTA DE TABLAS

Tabla 1 Casos de uso: Editar hoja de vida de un profesor.	51
Tabla 2 Casos de uso: Buscadores de perfil e información de usuarios.....	51
Tabla 3 Casos de uso: Administración del portal por parte del profesor.....	52
Tabla 4 Casos de uso: Utilización del portal por usuario público o de la comunidad.	54
Tabla 5 Casos de uso: Administración del servicio de PQRS.....	54
Tabla 6 Casos de uso: Registro de una PQRS.....	55
Tabla 7 Casos de uso: Funcionamiento del foro entre el administrador y el usuario.	56
Tabla 8 Descripción de las entidades	60
Tabla 9 Pruebas Realizadas: Editar hoja de vida de un profesor.	64
Tabla 10 Pruebas Realizadas: Búsqueda de perfil	65
Tabla 11 Pruebas Realizadas: Búsqueda de información de un usuario	65
Tabla 12 Pruebas Realizadas: Administración del portal profesor.....	66
Tabla 13 Pruebas Realizadas: Utilización del portal por usuarios UIS y públicos..	67
Tabla 14 Pruebas Realizadas: Administración del servicio de PQRS.....	67
Tabla 15 Pruebas Realizadas: Registro de una PQRS.....	68

LISTA DE FIGURAS

Figura 1 Modelo Cliente/servidor	19
Figura 2 Modelo de acceso a JSP	25
Figura 3 Prototipo evolutivo	34
Figura 4 Diagramas de Casos de Uso	37
Figura 5 Diagrama de Secuencias.....	39
Figura 6 Diagrama de casos de uso: Editar hoja de vida de un profesor.....	46
Figura 7 Diagrama de casos de uso: Buscadores de perfil e información de usuarios.	46
Figura 8 Diagrama de casos de uso: Administración del portal por parte del profesor.....	47
Figura 9 Diagrama de casos de uso: Utilización del portal por usuario público o de la comunidad.....	48
Figura 10 Diagrama de casos de uso: Administración del servicio de PQRS	49
Figura 11 Diagrama de casos de uso: Registro de una PQRS	50
Figura 12 Diagrama de casos de uso: Funcionamiento del foro entre el administrador y el usuario.	50
Figura 13 Diagrama E/R: Editar hoja de vida de un profesor.....	57
Figura 14 Diagrama E/R: Editar hoja de vida de un profesor.....	58
Figura 15 Diagrama E/R: PQRS	59
Figura 16 Diagrama de secuencia: Editar perfil profesor	73
Figura 17 Diagrama de secuencia: Buscadores de perfil e información	74
Figura 18 Diagrama de secuencia: Crear menú del blog profesor.....	76
Figura 19 Diagrama de secuencia: Editar menú del blog profesor	77
Figura 20 Diagrama de secuencia: Eliminar menú blog profesor.....	78
Figura 21 Diagrama de secuencia: Crear Tipo de PQRS	79
Figura 22 Diagrama de secuencia: Modificar Tipo de PQRS.....	80
Figura 23 Diagrama de secuencia: Atender PQRS.....	81
Figura 24 Diagrama de secuencia: Crear PQRS	82

LISTA DE ANEXOS

Anexo A. Modelo de Procesos del Sistema	73
---	----

RESUMEN

TÍTULO: Análisis, mantenimiento, diseño, desarrollo e implementación de nuevas funcionalidades y mejoras relacionadas con los módulos de PQRS, Hoja de Vida y Portal Web de profesores en la plataforma COMA Comunidad Académica.*

AUTOR: Carlos Antonio Díaz Flórez**

PALABRAS CLAVE: Calumet, Portal Web, PQRS, Profesores.

DESCRIPCIÓN:

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

Parte importante del Portal Web es el módulo del portal de profesores en el cual se concentra toda la información y servicios que un profesor necesite para realizar sus labores académicas, algunos servicios son aula virtual, noticias, eventos, producción intelectual, enviar correos, etc. Debido a cambios de algunos servicios se hace necesario una actualización a los servicios vigentes atendiendo solicitudes de la comunidad.

Con el fin de mejorar la primera versión del módulo PQRS se hace necesario la creación de PQRS privadas las cuales serán visibles por miembros de la comunidad, además de una actualización de las interfaces y la opción de agregar más información en el chat entre usuario y el administrativo que atiende la solicitud.

El servicio de buscadores de perfil, estudiantes y egresados permiten a los miembros de la comunidad académica a encontrar información de los mismos por medio de nombre, código o materias entre otras opciones. Estos servicios se actualizaron para que muestre el nuevo formato de Hoja de Vida además de unas mejoras visuales.

Debido a solicitud de varias escuelas se crea un nuevo servicio para la edición de la Hoja de Vida de los profesores a cargo de un administrativo, los cambios realizados serán guardados con el nombre de quien lo editó en última instancia.

* Trabajo de grado

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

ABSTRACT

Title: Analysis, maintenance, design, development and implementation of new features and improvements related to modules PQRS, resume and teachers on the platform Web Portal COMA academic community.*

Author: Carlos Antonio Diaz Flórez.**

Keywords: Calumet, Web Portal, PQRS, teachers.

Now schools have an information system oriented to the Web that is responsible for the Administration and control of the various activities both academic, as administrative that are carried out within the schools and faculties, as well as the control of users and services that are provided to them.

An important part of the Web Portal is the module of the portal of professors which concentrates all the information and services that a teacher needs to perform their academic duties, some services are virtual classroom, news, events, intellectual production, send emails, etc. Due to changes in some services it is necessary an update to the services applicable to serving requests from the community.

In order to improve the first version of the module PQRS is made necessary the creation of private PQRS which are visible by members of the community, as well as an update of the interfaces and the option of adding more information in the chat between user and the administrative serving the request.

Service profile seekers, students, and alumni members of the academic community help find information by name, code or materials among other options. These services have been updated so that it shows the new resume format in addition to a few visual enhancements.

Due to several school's request creates a new service for the edition of the life sheet of the teachers in charge of an administrative, the changes will be saved with the name of who edited what ultimately.

* Degree Work

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

INTRODUCCIÓN

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

El grupo Calumet, grupo de desarrollo de software de la escuela de ingeniería de sistemas, se encarga de diseñar, desarrollar e implementar aplicaciones web y soluciones informáticas con el fin de ayudar y facilitar a las escuelas de la universidad a administrar su información, buscando unir a la comunidad universitaria y brindar seguridad en los procesos e integridad a las personas. Con el fin de cumplir el objetivo principal del grupo y proporcionar portales web con contenido dinámico, se cuenta con herramientas software de libre distribución como Netbeans, Github, JavaScript, JQuery, GNU Linux, entre otros.

En este documento se presenta un soporte teórico, metodológico y técnico del desarrollo realizado en los servicio de: Portal Web Profesores para facilitar el acceso a la información de los profesores, se mejora la interfaz actualizando los servicios y puesto en marcha otros, el módulo de PQRS se avanza en el desarrollo para que cumpla su finalidad además se agrega más información y se mejora interfaz de edición, los buscadores de perfiles los cuales se actualizan para que muestren el nuevo formato de Hoja de Vida y mejoras visuales, finalmente se crea un nuevo servicio para la edición de la Hoja de Vida de los profesores, permitiendo a un administrativo designado por cada escuela para que realice los cambios pertinentes

1. PLANTEAMIENTO DEL PROYECTO

1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL

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

Los módulos de PQRS, Portal Web Profesores y Hoja de vida deben ser actualizados para adaptarse a las nuevas necesidades de la comunidad, la actualización consta de mejoras funcionales , de interfaz e implementando nuevas funcionalidades que permitan a los usuarios interactuar con los servicios Web, y entre ellos mismos, de una mejor manera.

Además, se agrega un nuevo servicio en el portal de profesores en el cual puedan publicar temas de interés académico o según lo que el docente crea conveniente. Para las PQRS se agrega tipos privadas y la opción que el usuario no registrado pueda hacer seguimiento a la solicitud realizada, entre otros cambios funcionales que mejore el desempeño y la atención al usuario.

1.2 JUSTIFICACIÓN

El Portal Web Comunidad Académica (COMA), que desarrolla y mantiene el grupo Calumet, es una plataforma que provee lugares, adicionales a los ya existentes, de encuentro y comunicación a la comunidad de los programas de las diferentes escuelas. Esta plataforma contribuye a la realización de la misión en sus tres ejes: extensión, investigación y docencia. En ella, la Comunidad podrá mantener

actualizada su hoja de vida, publicar noticias, proponer eventos, consultar y crear agendas, compartir documentos, enviar correos, y otros servicios que podrán hacer más fácil la realización de la misión institucional. Los servicios de los portales web deben actualizarse constantemente, mejorar y adaptarse a los cambios que se presenten en su entorno, satisfaciendo las necesidades de los usuarios del sistema. Esto permite aumentar el tiempo de vida útil del software evitando que pase a ser obsoleto con la creación de nuevos servicios, y con el mantenimiento realizado a los existentes.

Actualmente el módulo de atención a PQRS, peticiones quejas reclamos y solicitudes por sus siglas, se encuentra en su primera versión. En esta versión un usuario, logueado o no puede hacer una PQRS y, de acuerdo al tipo, esta será direccionada a un cierto usuario administrativo para su respuesta. Cuando el usuario se encuentra logueado un sencillo chat promueve y conserva los mensajes entre el usuario PQRS y el administrativo que responde. Las PQRS podrán ser pasadas a estado Respondida y Finalizada por el administrativo. En este módulo, se hace necesario mejorar las interfaces y la información que se muestran a los usuarios y a los administrativos. Además, es necesario soportar PQRS para usuarios logueados, es decir, PQRS que solo podrán realizar usuarios registrados y que hayan ingresado con nombre de usuario y contraseña. Los tipos de PQRS se podrán crear por un administrativo que deberá de definir si es público o privado y, que además, deberá definir el usuario que va a atender este tipo de PQRS. Se debe garantizar que el usuario administrativo con máximo nivel (tipo PE1) pueda revisar todas las PQRS por estado y por tipo.

Mantener actualizada la hoja de vida por parte de los profesores es una labor que se dificulta a estos por la cantidad de actividades que realizan para llevar a cabo la misión de nuestra Universidad. Por lo anterior, es que las Escuelas vienen solicitando se permita a un usuario administrativo ayudar en esta labor, para esto es necesario crear un servicio nuevo que permita seleccionar el docente y actualizar

sus datos. Este servicio será privado y solo se le habilitará a la persona que indique la Escuela. Con esto logramos hacer las labores del docente más sencilla.

Con la nueva versión del módulo de HV y Perfil en el portal se hace necesario actualizar los buscadores de miembros de la Comunidad por los diferentes servicios: Información Todo Usuario de los administradores, buscador público de perfiles, información de los estudiantes en el aula, y información de estudiantes y egresados de los profesores. En algunos casos es actualizar el buscador, y en otros, agregar a la información que se muestra el perfil u hoja de vida.

Con todos los cambios llevados a la fecha se hace necesario actualizar el Portal de los Profesores. Usar el ancho de la ventana, actualizar los menús de entrada y facilitar la actualización de los menús internos por parte de los administradores está dentro de las labores a realizar. Además, los profesores deben poder crear menús para publicar y/o opinar sobre temáticas de su interés. Otra de las solicitudes realizadas al grupo es abrir la posibilidad de usar una imagen personalizada de fondo. Reactivar el menú que permitía contactar a los profesores con estrategias para evitar que se envíen correos no deseados de manera masiva.

1.3 OBJETIVOS

1.3.1 Objetivo general. Realizar las funciones de mantenimiento, análisis, diseño, desarrollo e implementación de los servicios de PQRS, Hoja de Vida y Portal Web de Profesores con el fin de hacerlos óptimos, eficientes y sostenibles, permitiendo acceder a la información de manera más sencilla, ágil, óptima y eficiente.

1.3.2 OBJETIVOS ESPECÍFICOS

1.3.2.1. Crear un nuevo servicio que permita a un administrativo actualizar las hojas de vida de los profesores y guardar el último usuario que modificó el perfil.

1.3.2.2. Actualizar buscadores de perfil de usuarios con la nueva versión de HV y Perfil, entre los cuales están:

- Buscador información todo usuario, a la información que se muestra agregar la posibilidad de consultar la HV y el Perfil
- Buscador público de perfiles actualizar el buscador
- Información estudiantes y egresados, agregar a la información que se muestra la posibilidad de consultar la HV y el perfil.

1.3.2.3. Realizar actualización en el portal de profesores

- Actualizar el menú de la página agregando servicios nuevos y eliminando servicios en desuso
- Publicar la hoja de vida de acuerdo con los nuevos estándares que se tienen para el perfil y hoja de vida
- Realizar cambios sencillos en la interfaz gráfica, aumentar el tamaño de la interfaz del portal, a un tamaño semejante del portal de las escuelas.
- Agregar un captcha para el envío de correos masivos.
- Dar la posibilidad a los profesores de tener un servicio que permite publicar y/o opinar sobre temáticas de interés.

1.3.2.4. Realizar actualización y revisión en el módulo de PQRS

- Revisar funcionamiento e interfaces de este módulo.
- Continuar con el desarrollo del módulo para que cumpla su finalidad.
- Crear PQRS privados que solo se podrán hacer estando logueados.
- Agregar más información en las interfaces actuales.

1.4 IMPACTO Y VIABILIDAD

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

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

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

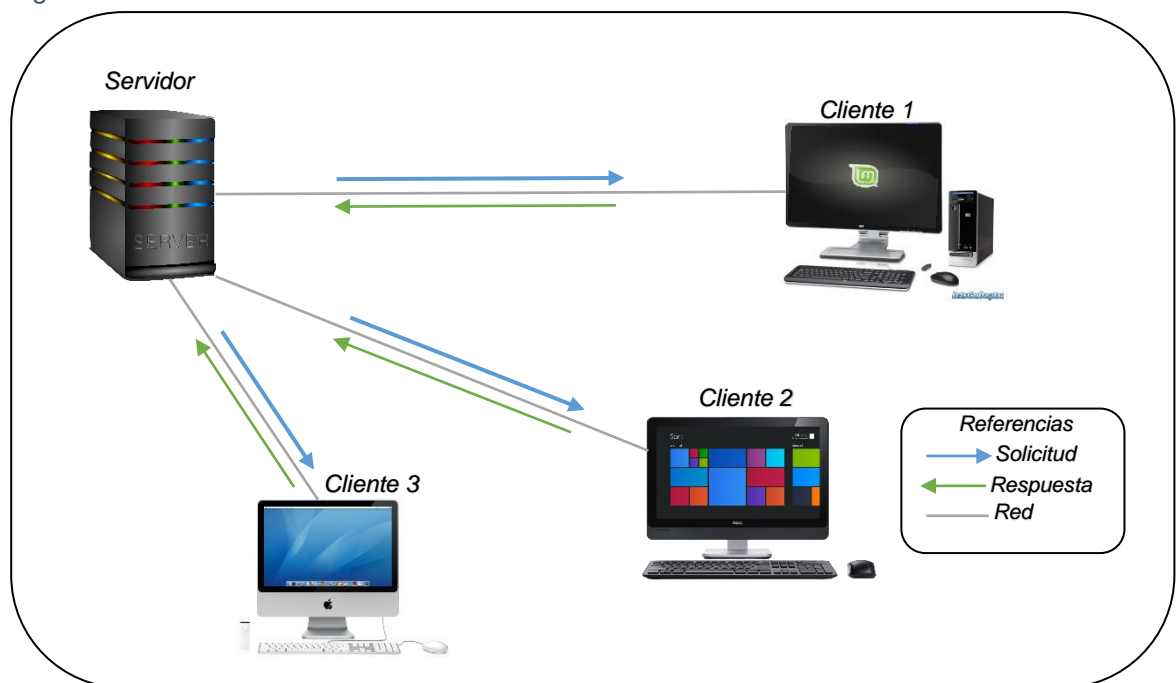
Además, se cuenta con la supervisión por parte del director del proyecto y la colaboración del equipo de trabajo CALUMET, agentes de gran apoyo en la realización de este proyecto.

2. MARCO TEÓRICO

2.1 ARQUITECTURA CLIENTE/SERVIDOR

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

Figura 1 Modelo Cliente/servidor



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

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

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

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

La siguiente es la clasificación de la arquitectura Cliente/Servidor:

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

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

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

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

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

- Capa de Cliente: Interfaz con el usuario, se usa un navegador web.

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

2.1.4 Ventajas del esquema Cliente/Servidor

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

2.1.5 Desventajas del esquema Cliente/Servidor

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

- El desempeño es un aspecto a tener en cuenta en el esquema C/S, problemas de este estilo pueden presentarse por congestión en la red.

2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

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

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

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

- JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite, crear ventanas, mostrar texto en movimiento y verificar las entradas a un formulario.
- Controles Activos: tecnología Microsoft que permite la creación de aplicaciones Windows, como pueden ser Visual Basic Script o Visual C. Es la respuesta de Microsoft a los Applets de Java.
- Java Applets: Programas escritos en lenguaje de programación Java, se incrustan en HTML y se ejecutan en el navegador gracias a la Máquina Virtual de Java (JVM) que lleva éste incorporado.

2.2.2 Código del Lado del Servidor (Server Side Scripts). Código que se ejecuta en el servidor. Para su actividad el programa ejecuta y procesa los datos o peticiones que el usuario envía desde su navegador, para luego enviar los

resultados del programa en una página HTML que el usuario verá normalmente en su navegador. Los más usados son:

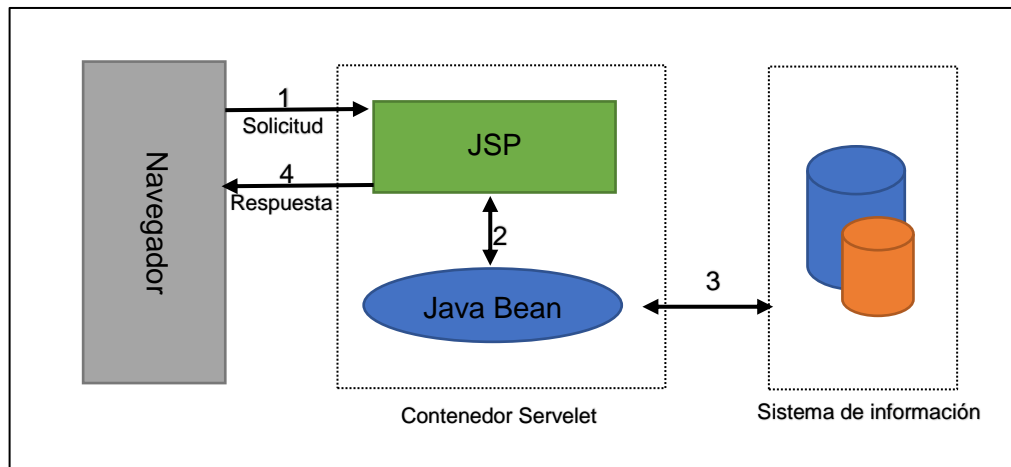
- ASP (Active Server Pages): Permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables, requiere de un computador configurado como Servidor Web de Microsoft (Microsoft WebServer), el navegador del cliente es indiferente pues el trabajo se realiza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

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

2.2.3.1 Modelo de acceso a JSP

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

Figura 2 Modelo de acceso a JSP



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

2.3 BASES DE DATOS

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, con una redundancia controlada y una estructura que refleja las interrelaciones y restricciones existentes en el mundo real. En la base de datos se almacena información considerada necesaria para una determinada organización o negocio. Existen modelos que describen la estructura de una base de datos (entidades, atributos y relaciones), la mayoría de los modelos de datos poseen un conjunto de operaciones básicas como consultar y actualizar y eliminar.

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

2.3.1.1 Base de Datos Jerárquica. 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. Está limitado por su incapacidad de representar eficientemente la redundancia de datos.

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

2.3.1.3 Base de Datos Relacional. Es el más utilizado para modelar problemas reales y administrar datos dinámicamente. Su fundamento es el uso de

"relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas de una tabla). Los datos pueden ser recuperados o almacenados mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

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

Las funciones principales de un DBMS son:

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

2.3.3 MySQL. Es un sistema de base de datos operacional considerado uno de los más importantes, utilizado por usuarios del medio para el diseño y programación de base de datos de tipo relacional. Además, MySQL es conocida por desarrollar alta velocidad de búsqueda de datos e información, a diferencia de sistemas anteriores.

2.3.4 Ventajas de MySQL

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

2.4 NETBEANS

Es un entorno de desarrollo integrado (IDE), siendo una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas escritos en JAVA, pero puede servir para cualquier otro tipo lenguaje de programación. NetBeans es un producto libre y gratuito sin restricciones de uso. El NetBeans es un entorno de desarrollo integrado de código abierto escrito completamente en Java usando la plataforma NetBeans, soporta desarrollo de

todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). La versión actual es NetBeans IDE 8.2, 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

Todas las funciones del IDE son provistas por módulos. Cada Módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permite al usuario comenzar a trabajar inmediatamente.

2.5 SISTEMA DE CONTROL DE VERSIONES

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

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

2.5.1 Subversión. Esto permite recuperar versiones antiguas de los datos o examinar cómo han ido evolucionando esto. Su objetivo es ser un sucesor prácticamente compatible del ampliamente usado Concurrent Version system (CVS).

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

2.6 PROGRAMACIÓN UTILIZADA

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

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

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

2.6.2 Objetos. Cualquier cosa real o abstracta, que posee atributos y un conjunto

de operaciones que manipulan esos atributos que da un comportamiento particular. Un objeto es una instancia de una clase, el estado del objeto se determina por el estado (valor) de sus propiedades o características (atributos).

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

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

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

2.6.6 Beneficios de la Programación Orientada a Objetos

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

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

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

El JDK es un software que provee herramientas de desarrollo para la creación de programas en java.

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

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

2.7 SERVIDORES WEB

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

2.7.1 Servidor Apache Tomcat. Servidor de aplicaciones Java basado en los estándares definidos por Sun Microsystems. Tomcat es desarrollado como parte del

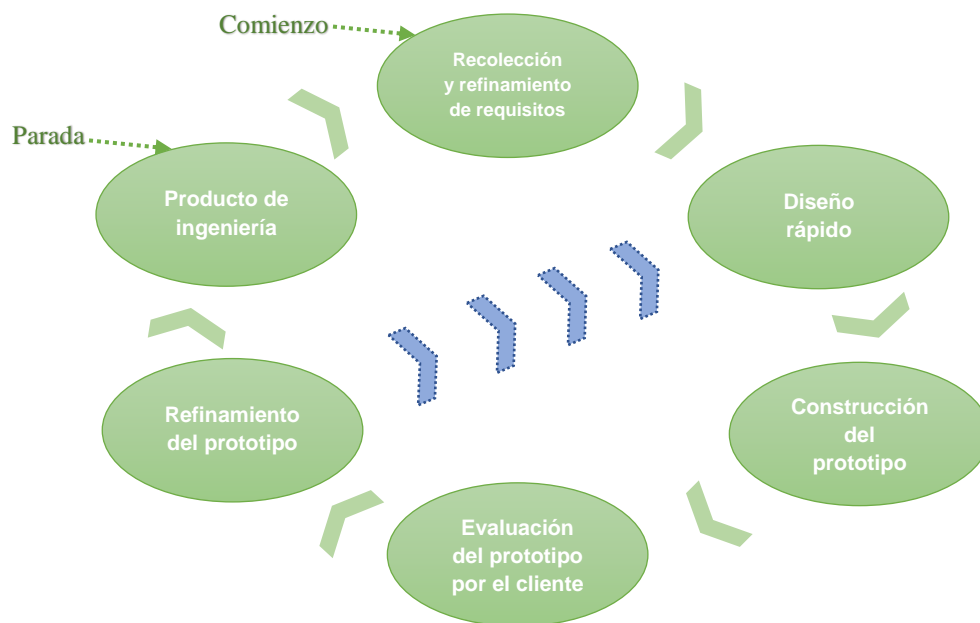
proyecto de código abierto de la fundación de software Apache y es uno de los servidores de aplicaciones Java más utilizados, en especial porque es liviano, cumple con todos los estándares, sencillo de instalar, tiene muy buena documentación y es gratuito, además por ser escrito en Java funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java (JVM).

3. MARCO METODOLÓGICO

3.1 PROTOTIPO EVOLUTIVO

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

Figura 3 Prototipo evolutivo



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

- Es deseable tener un bosquejo de lo que se desee mejorar o crear para poder incorporar sugerencias de cambio por parte de los usuarios del portal de las escuelas en las etapas tempranas del desarrollo.

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

Procedimiento que seguir para la metodología planteada:

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

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

3.2 LENGUAJE DE MODELADO UNIFICADO

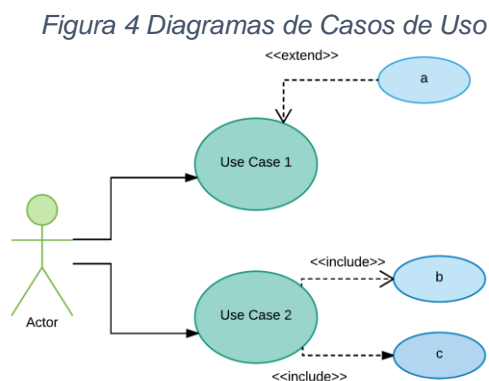
El Lenguaje de Modelado Unificado o Unified Modeling Language (UML), es el más utilizado en la actualidad. Es un lenguaje gráfico estándar para visualizar, especificar, construir y documentar un sistema para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. UML no es un método de desarrollo porque no indica los pasos a seguir para llegar al código, es decir, no especifica como pasar del análisis al diseño y de este al código. Al no ser un método de desarrollo resulta ser independiente del ciclo de desarrollo que se siga, puede encajar en un ciclo en cascada, evolutivo, espiral o en métodos ágiles de desarrollo.

3.2.1 Diagramas de UML. Los diagramas UML utilizados en el desarrollo de este proyecto fueron diagramas de casos de uso y diagramas de secuencias.

Las principales razones por las cuales se prefirió UML como el lenguaje de modelado son:

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

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



3.2.2.1 Actores. Un actor en un caso de uso representa un rol, que alguien o

algo puede desempeñar dentro un sistema y no un alguien o algo específico.

En este proyecto se destacan tres clases de actores:

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

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

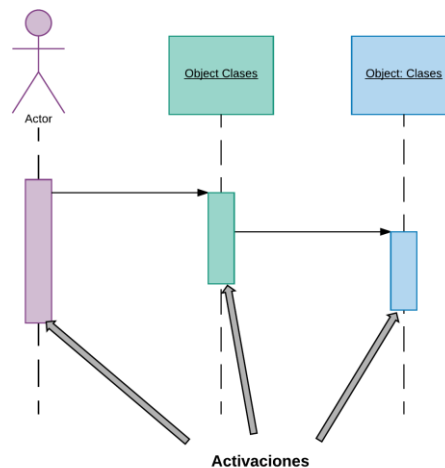
3.2.2.3 Extensión (extend). Es otra forma de interacción, una extensión representa una variación de un caso de uso a otro, es decir, una dependencia específica entre los casos de uso, a través de la cual un caso de uso puede extender a otro.

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

representado como una progresión vertical.

- **Objetos:** Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la línea de vida de un objeto. Con la línea de vida se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto.
- **Mensaje:** Un mensaje que va de un objeto a otro pasa la línea de vida de un objeto a otro. Un objeto puede enviarse un mensaje a sí mismo. Un mensaje puede ser simple, sincrónico o asincrónico.
- **Tiempo:** El diagrama representa al tiempo en dirección vertical. Inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.
- (Siglas en Ingles) La interfaz gráfica de usuario; es la interfaz de interacción del usuario y en la que más interactividades se presentan con otros objetos.

Figura 5 Diagrama de Secuencias



3.3 ESTÁNDARES DE PROGRAMACIÓN

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

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

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

- **Tabla básica:** Aquella cuyos registros son necesarios para el correcto funcionamiento de la base de datos. Estas tablas no experimentan muchos cambios en los datos. El prefijo a anteponer a los nombres de estas tablas es “TB_”, es decir la tabla que almacena las categorías de clasificación de los usuarios del portal web es llamada “TB_Categorias”, por ejemplo.
- **Tabla de Relación:** Surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deben ser siempre descriptivos para cada relación. El prefijo a anteponer a los nombres de estas tablas es “TR_”, por ejemplo, la tabla “MiembrosConsejo” es conocida como “TR_ MiembrosConsejo”.

- Tabla Principal: Aquella cuyo número de registros tiende a crecer en gran cantidad y que además no es posible clasificar como tabla básica o de relación. Un ejemplo de tabla principal es la tabla que almacena los usuarios del portal EISIWeb. El prefijo a anteponer a los nombres de estas tablas es “TP_”, es decir la tabla “Usuarios”, es conocida como “TP_ Usuarios”.

3.3.3 Clases. Los nombres de las clases deben ser sustantivos en plural, la primera letra de cada palabra debe ser mayúscula. Estos deben ser simples, descriptivos como, por ejemplo: TipoPlan.java, Usuarios.java, NotasCartelera.java, etc.

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

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

- El sitio cuenta con un directorio llamado “images”; en éste se encuentran almacenados todos los archivos .jpg, .gif, .png.
- Para el desarrollo de este proyecto se crearon los directorios “/eisi/PerfilProfesores”, “/eisi/images/BlogProfesor”, los cuales almacenan los archivos correspondientes al funcionamiento de estos servicios

4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

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

4.1 PROTOTIPO ESPERADO

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

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

4.1.1 Nuevo servicio para la edición de la Hoja de Vida de los profesores

Objetivo inicial:

- Crear un nuevo servicio que permita a un administrativo modificar la hoja de vida de un profesor.

Requisitos finales del objetivo:

- Listar los profesores por las categorías planta y cátedra en dos tablas de acuerdo al estándar Elise*.
- Una vez listado los profesores , se permite la selección de un profesor para acceder a la hoja de vida y realizar la edición.
- Al momento de guardar los cambios se guardará el nombre del usuario quien modificó la hoja de vida y la fecha .

4.1.2 Actualizar buscadores de perfil de usuarios con la nueva versión de Hoja de Vida y perfil.

Objetivo inicial:

- Se debe actualizar los buscadores: información todo usuario, público de perfiles e información estudiantes y egresados para que muestren la nueva hoja de vida.

Requisitos finales del objetivo:

- Modificar el servicio de la hoja de vida y perfil, para que muestre los datos personales completos solo al propietario y administrador con perfil PE1
- Agregar un enlace a la nueva hoja de vida y perfil en la información que se muestra en el buscador información todo usuario.
- Actualizar el buscador público de perfiles con los nuevos estándares de Calumet haciéndolo óptimo, además de cambiar la forma en que se muestra la información.

- Corregir errores al filtrar los usuarios en el buscador de estudiantes y egresados , actualizar a los nuevos estándares de Calumet mostrando la nueva hoja de vida y perfil a la información que actualmente se muestra.

4.1.3 Actualización en el portal de los profesores

Objetivo inicial:

- Actualizar el portal profesores en el que se incluyan los servicios actuales y mejorar la interfaz gráfica.

Requisitos finales del objetivo:

- Eliminar los menús de cartelera, eventos y asignaturas virtuales ya que tienen poca utilidad y están en desuso
- La página inicial mostrará el nuevo formato de perfil, si el profesor creó noticias o eventos se podrán visualizar.
- Mejoras visuales en el horario del profesor
- Reactivar el menú de contacto con el profesor implementando un captcha para evitar el envío de correos masivos.
- Agrandar el ancho del portal profesor a un tamaño similar que el de las escuelas además de realizar cambios sencillos en la interfaz gráfica.
- Agregar la opción de subir fondos en el menú de personalizar
- En la parte de administrativa del portal añadir un botón que actualiza las rutas de los servicios.
- Crear un nuevo servicio en el cual al profesor le será permitido crear menús de su preferencia con contenido editable para que se muestre en el portal profesor, tendrá la posibilidad de administrarlos (crear, editar, eliminar, cambiar de estado) tanto el portal del profesor como en el portal de la escuela.

4.1.4 Actualización en el módulo de PQRS

Objetivo inicial:

- Realizar una revisión del estado del módulo y continuar su desarrollo para que cumpla su finalidad.

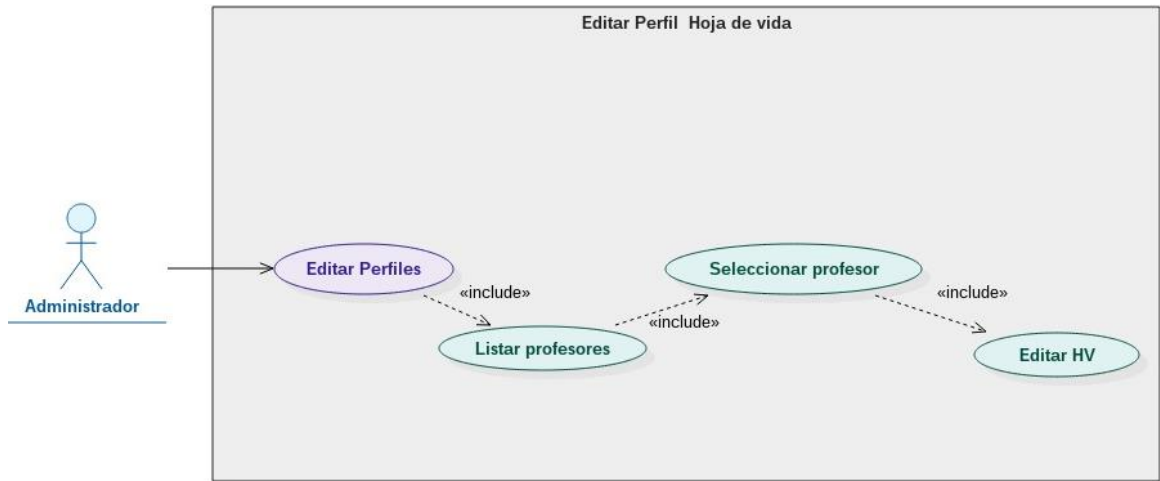
Requisitos finales del objetivo:

- Soportar PQRS para usuarios logueados, es decir, PQRS que solo podrán realizar usuarios registrados y que hayan ingresado con nombre de usuario y contraseña.
- Los tipos de PQRS se podrán crear por un administrativo que deberá de definir si es público o privado y, que además, deberá definir el usuario que va a atender este tipo de PQRS
- Se debe garantizar que el usuario administrativo con máximo nivel (tipo PE1) pueda revisar todas las PQRS
- Los usuarios públicos podrán consultar el estado de su solicitud por medio de la cédula registrada y el id de PQRS enviada al correo.
- Agregar el ckeditor de Elise en el chat entre el usuario y el administrador del PQRS
- Mejorar la función de edición de los tipos de PQRS para que puedan cambiar de estado o de privacidad.

4.2 DIAGRAMAS DE CASOS DE USO

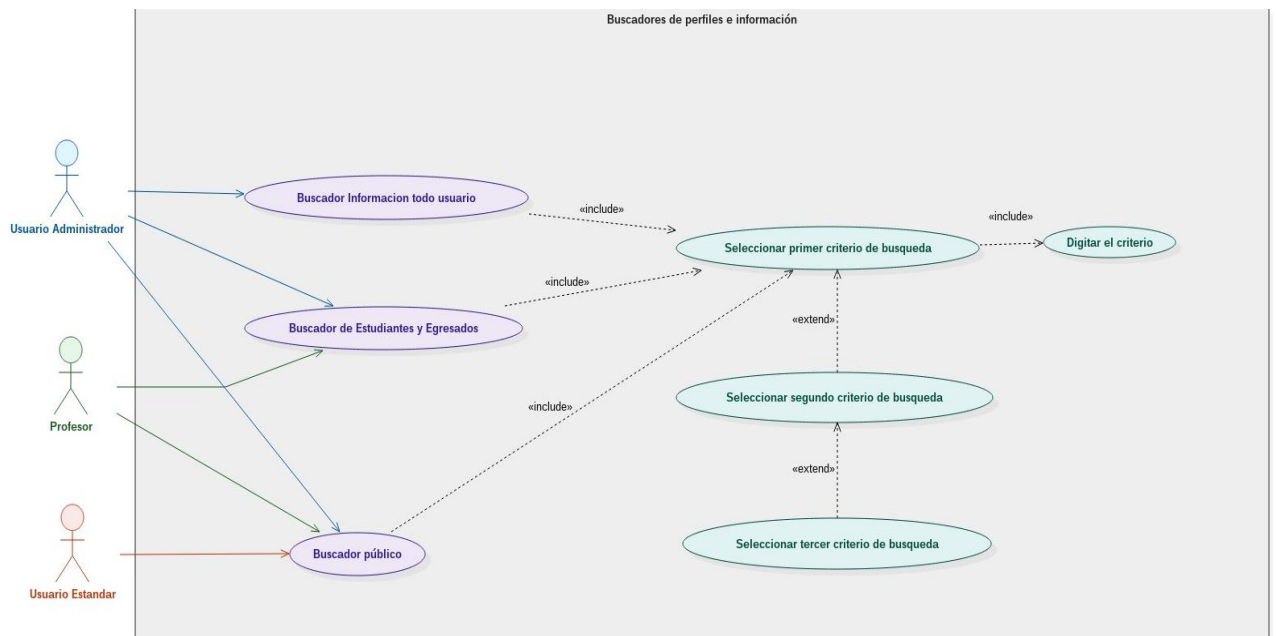
4.2.1 Servicio para la edición de la Hoja de Vida de los profesores

Figura 6 Diagrama de casos de uso: Editar hoja de vida de un profesor.



4.2.2 Servicios buscadores de perfil e información

Figura 7 Diagrama de casos de uso: Buscadores de perfil e información de usuarios.



4.2.3 Servicio Portal Profesor

Figura 8 Diagrama de casos de uso: Administración del portal por parte del profesor

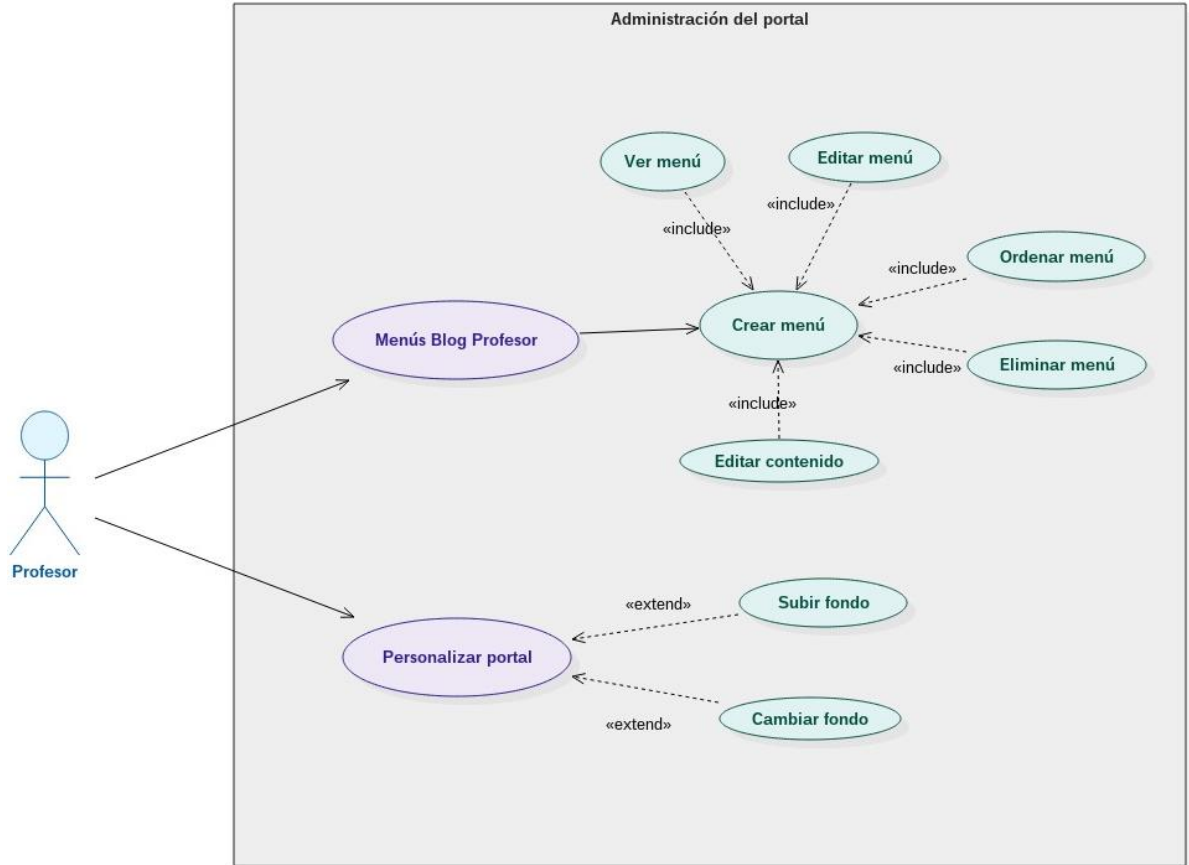
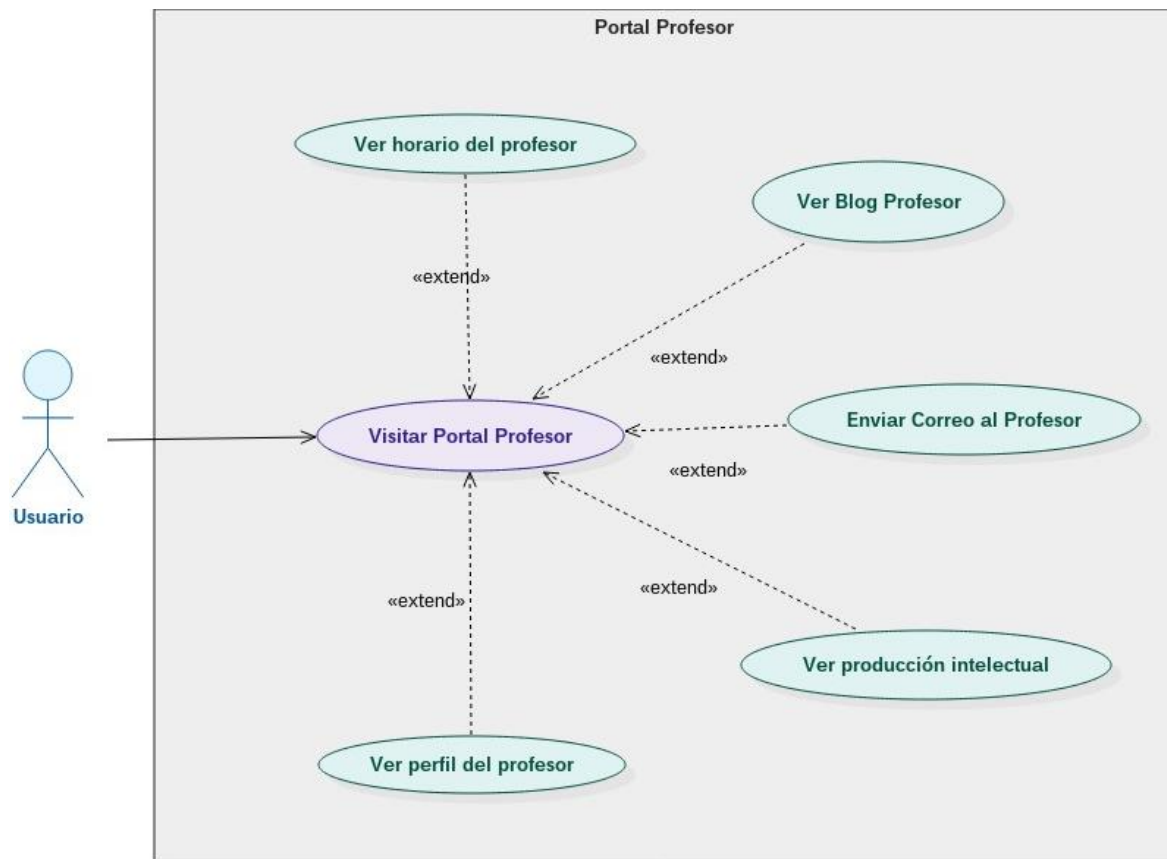


Figura 9 Diagrama de casos de uso: Utilización del portal por usuario público o de la comunidad.



4.2.4 Servicio Peticiones, quejas, reclamos y sugerencias

Figura 10 Diagrama de casos de uso: Administración del servicio de PQRS

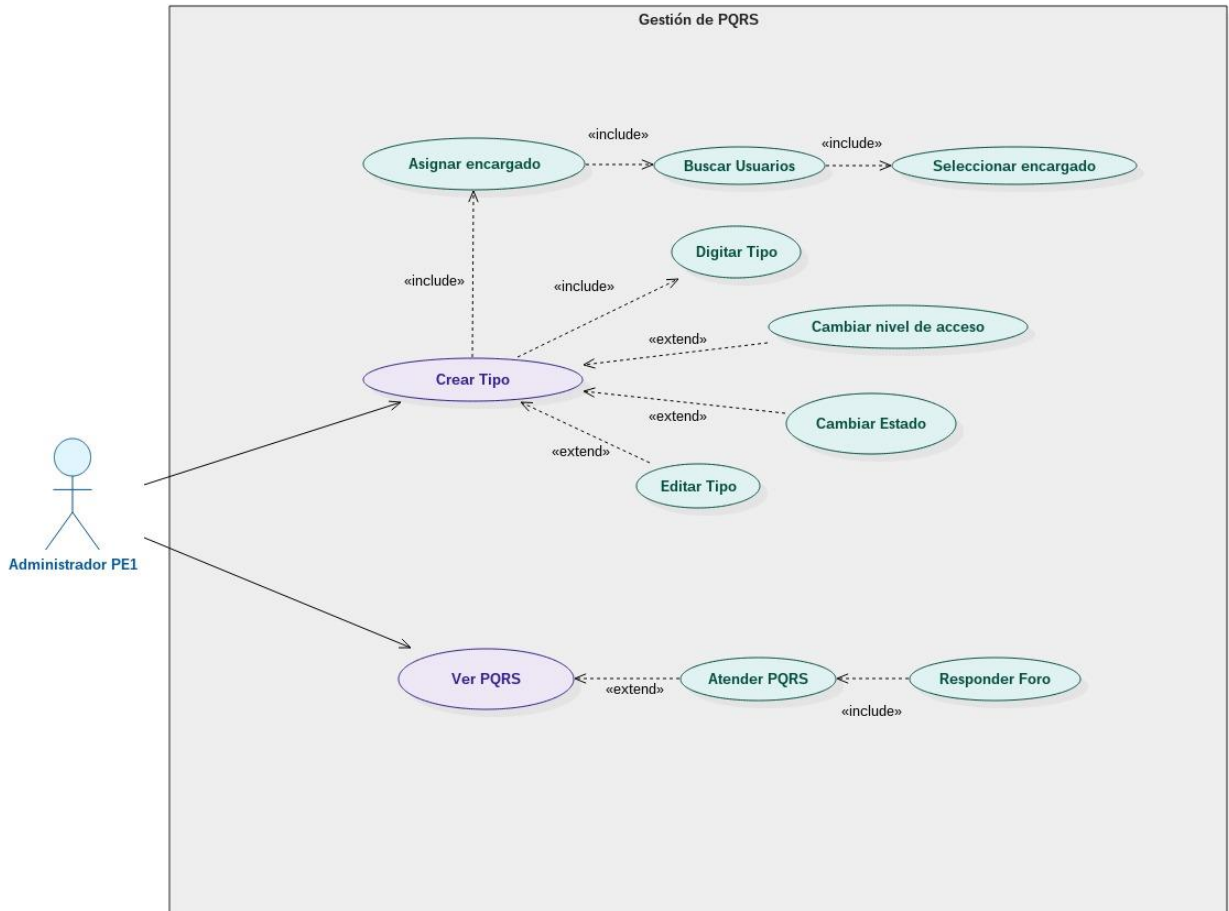


Figura 11 Diagrama de casos de uso: Registro de una PQRS

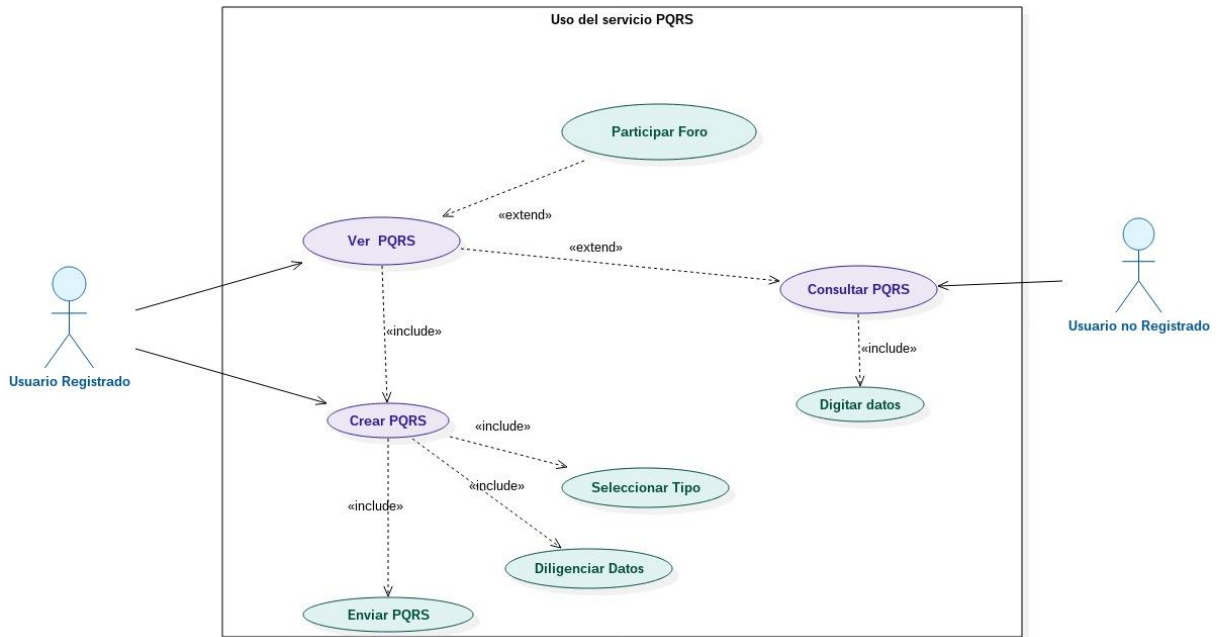
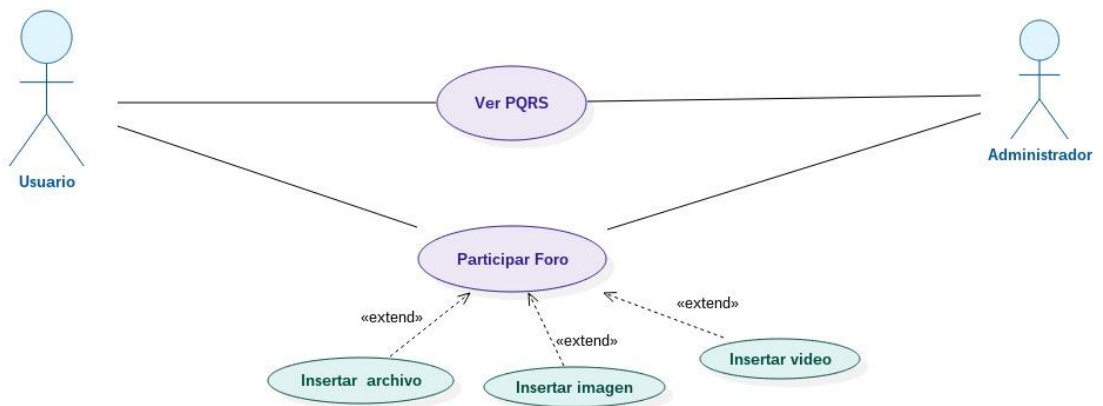


Figura 12 Diagrama de casos de uso: Funcionamiento del foro entre el administrador y el usuario.



4.3 DOCUMENTACION DE CASOS DE USO DEL SISTEMA

4.3.1 Servicio para la edición de la Hoja de Vida de los profesores

Tabla 1 Casos de uso: Editar hoja de vida de un profesor.

TÍTULO	FUNCIONES PRIMARIAS
Editar Perfiles	Este servicio agrega o cambia la información del perfil y HV de un profesor, solo podrá acceder a este servicio el administrativo que designe cada escuela
Listar profesores	Se listan todos los profesores de la escuela por categoría planta y catedra
Seleccionar Profesor	Se selecciona el profesor a editar
Editar HV	El administrativo cambia o edita los campos pertinentes de la HV

4.3.2 Servicios buscadores de perfil e información

Tabla 2 Casos de uso: Buscadores de perfil e información de usuarios.

TÍTULO	FUNCIONES PRIMARIAS
Buscador Información todo usuario	Este servicio solo está habilitado para el administrador del portal, en el cual muestra información académica, datos personales y la nueva HV de todos los usuarios
Buscador de Estudiantes y Egresados	Este servicio está habilitado para que los profesores consulten la información

	académica y HV de estudiantes y egresados
Buscador público	Este servicio esta habilitado para la comunidad en general en el cual se muestra la HV de todos los usuarios.
Seleccionar primer criterio de búsqueda	Entre los criterios se encuentran buscar por nombre, código (estudiante o profesor), cedula, categoría, materia y grupos de investigación
Seleccionar segundo criterio de búsqueda	Según el primer criterio seleccionado se podrá seleccionar entre materias, categorías o grupos de investigación
Seleccionar tercer criterio de búsqueda	Se selecciona el grupo de la materia seleccionada anteriormente.
Digitar criterio	Se utiliza cuando el primer criterio es nombre, cedula y código

4.3.3 Servicio Portal Profesor

Tabla 3 Casos de uso: Administración del portal por parte del profesor

TÍTULO	FUNCIONES PRIMARIAS
Menús blog profesor	Servicio en el cual al profesor le será permitido crear menús de su preferencia con contenido editable para que se muestre en el portal profesor





Crear Menú	Asignar un nombre, descripción, el estado y si quiere que tenga contenido en caso de que sea un menú principal.
Ver Menú	Se visualiza el contenido del menú al dar clic sobre él.
Editar Menú	Si es necesario el profesor puede cambiar el estado del menú para que se muestre o no el menú en el portal, cambiar el nombre y descripción
Ordenar Menú	Por medio de los botones     Se ordenan los menús principales más hacia la izquierda o derecha y los submenús hacia arriba o abajo
Eliminar Menú	Elimina el menú principal o submenú con su contenido, para eliminar un menú principal debe haber eliminado todos los submenús.
Editar contenido	Después de haber creado el menú se procede a escribir su contenido, el cual tiene la posibilidad de cargar imágenes, insertar video y agregar vínculos
Personalizar portal	El profesor puede cambiar el aspecto del portal a su gusto.
Subir fondo	El profesor tiene la posibilidad de subir fondos si quiere más de los predeterminados.
Cambiar fondo	Guarda el fondo seleccionado.

Tabla 4 Casos de uso: Utilización del portal por usuario público o de la comunidad.

TÍTULO	FUNCIONES PRIMARIAS
Visitar portal profesor	Este servicio brinda al público en general información del profesor y poder tener un contacto con él.
Ver el horario del profesor	Indica los horarios de clase, actividades y de atención.
Ver blog profesor	Se pueden ver los temas de interés del profesor.
Enviar correo al profesor	Servicio para contactar al profesor
Ver producción intelectual	Se visualiza los proyectos, ponencias, artículos, etc. que el profesor haya realizado.
Ver perfil del profesor	Se encuentra la información que el profesor cree conveniente se muestre en su perfil, además de las noticias y eventos que él haya creado.

4.3.4 Servicio Peticiones, quejas, reclamos y sugerencias

Tabla 5 Casos de uso: Administración del servicio de PQRS

TÍTULO	FUNCIONES PRIMARIAS
Crear Tipo	Servicio para crear un tipo de pqrs ya sea privada o pública y asignar a una persona que responda las solicitudes
Asignar Encargado	Buscar a una persona(s) que se encarguen de un tipo de pqrs
Buscar usuarios	Se listan los usuarios según el criterio de búsqueda.
Digitar Tipo	Asigna un nombre al tipo de pqrs
Seleccionar encargado	De la lista generada se selecciona el o los encargados.

Cambiar nivel de acceso	Por defecto es público, pero se puede cambiar a privado un tipo de pqr
Cambiar estado	Por defecto está activo, el administrador puede cambiar a inactivo sino quiere que se muestre
Editar tipo	Después de creados los tipos de pqr se pueden cambiar todos sus campos, incluso los encargados.
Ver PQRS	Los administradores PE1 pueden ver todas las pqr por tipo o estado. Los encargados solo visualizan los tipos de pqr asignados.
Atender PQRS	Se visualiza las pqr por atender, en atención y las finalizadas
Responder Foro	El encargado responde al usuario la solicitud realizada y queda abierto el foro hasta que el usuario tenga una respuesta satisfactoria.

Tabla 6 Casos de uso: Registro de una PQRS

TÍTULO	FUNCIONES PRIMARIAS
Consultar PQRS	Para los usuarios no registrados existe una interfaz para consultar las pqr registradas.
Digitar Datos	Los datos solicitados a los usuarios no registrados son el ID de pqr y la cédula para realizar la consulta.
Ver PQRS	Se visualiza el estado y la opción de participar en el foro
Participar en el foro	El usuario visualiza las respuestas del encargado y puede comentar si la solicitud no se ha respondió correctamente o no este muy clara.
Crear PQRS	Para los usuarios registrados, como para los públicos se datos importantes para llevar a cabo la solicitud requerida.
Seleccionar Tipo	Los usuarios podrán escoger diversos tipos de pqr, para los usuarios registrados podrán seleccionar tipos adicionales.

Enviar PQRS	Cuando el usuario registre una pqr se enviará un correo con la solicitud realizada y el identificador de esta misma.
-------------	--

Tabla 7 Casos de uso: Funcionamiento del foro entre el administrador y el usuario.

TÍTULO	FUNCIONES PRIMARIAS
Ver PQRS	Se visualiza el estado y la opción de participar en el foro
Participar en el foro	El usuario visualiza las respuestas del encargado y puede comentar si la solicitud no se ha respondió correctamente o no este muy clara.
Insertar video, imagen y archivos	En el foro tanto administrador como usuario pueden anexar fotos, videos y archivos para mejorar la comunicación

4.4 DISEÑO Y ANÁLISIS

Diagramas Entidad/Relación de los servicios desarrollados

4.4.1 Servicio para la edición de la Hoja de Vida de los profesores

Figura 13 Diagrama E/R: Editar hoja de vida de un profesor



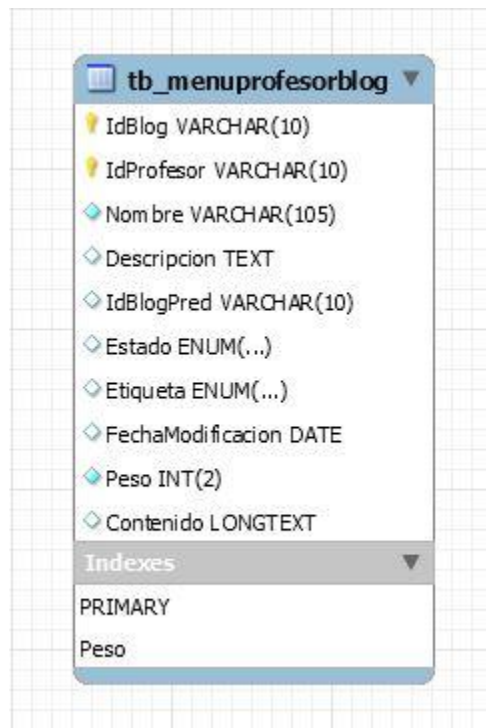
The image shows a screenshot of a database schema viewer for a table named 'tp_usuarios'. The table contains the following fields:

Field Name	Data Type
IdEsc	VARCHAR(10)
IdUsr	VARCHAR(12)
PrimNomUsr	VARCHAR(25)
SegNomUsr	VARCHAR(25)
PrimApeUsr	VARCHAR(25)
SegApeUsr	VARCHAR(25)
Genero	ENUM(...)
Foto	VARCHAR(400)
CorreoUsr	VARCHAR(100)
CodEst	VARCHAR(7)
Identif	BIGINT(20)
Estado	ENUM(...)
TipoIdentif	ENUM(...)
Domicilio	VARCHAR(200)
Correspondencia	VARCHAR(200)
Telefono1	VARCHAR(100)
Telefono2	VARCHAR(100)
PublicarMiPerfil	ENUM('SI', 'NC')
FechaModMiPerfil	VARCHAR(50)
CiudadResidencia	VARCHAR(200)
LugarNacimiento	VARCHAR(200)
FechaNacimiento	VARCHAR(10)
Descripcion	TEXT
Edad	INT(3)
NumLibretaMilitar	VARCHAR(30)
LoginUsr	VARCHAR(30)
ClaveUsr	VARCHAR(32)
IdUsrModifica	VARCHAR(12)

At the bottom of the viewer, there is a section labeled 'Indexes' with a right-pointing arrow.

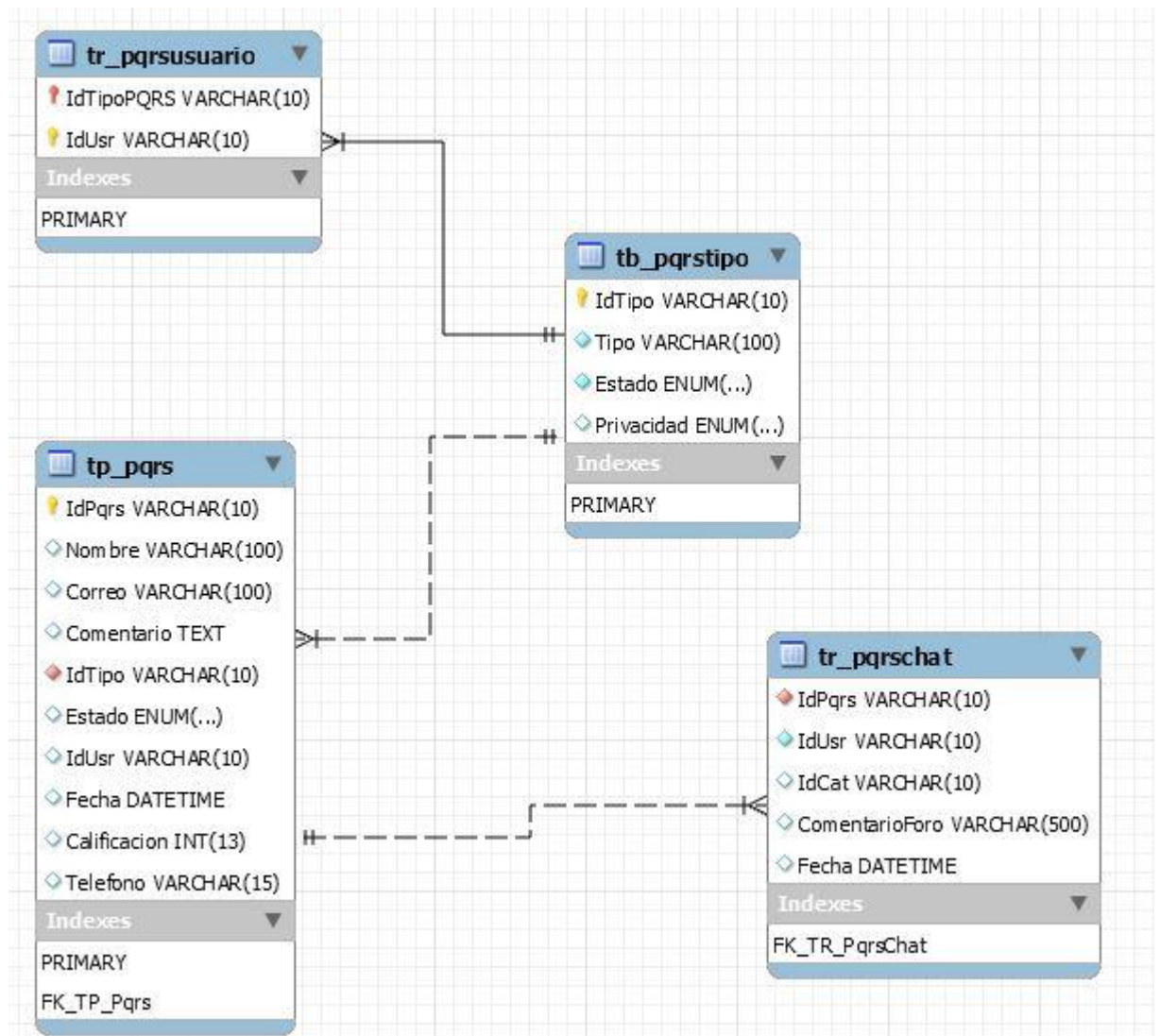
4.4.2 Servicio Menú blog profesor

Figura 14 Diagrama E/R: Editar hoja de vida de un



4.4.3 Servicio Peticiones, quejas, reclamos y sugerencias

Figura 15 Diagrama E/R: PQRS



4.5 DESCRIPCIÓN DE LAS ENTIDADES

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

Tabla 8 Descripción de las entidades

ENTIDAD	DESCRIPCIÓN
TP_Usuarios	Contiene la información personal de los usuarios de todas las escuelas.
TB_MenuProfesorBlog	Contiene los nombres de los menús y submenús creados por cada profesor, así como el contenido de estos.
TP_Pqrs	Contiene las PQRS realizadas por los usuarios.
TB_PqrsTipo	Contiene los tipos de PQRS, su estado y nivel de acceso.
TR_PqrsUsuario	Contiene las personas encargadas de un tipo de pqrs.
TR_PqrsChat	Contiene las conversaciones realizadas en los foros de cada PQRS entre el usuario y el encargado de resolverla.

4.6 MODELO DE PROCESOS DEL SISTEMA

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

4.7 IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES

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

- Lenguaje de marcado para la elaboración de páginas web, HTML.
- Lenguaje de programación orientado a la web, JSP.
- Lenguaje de programación interpretado orientado a objetos, JavaScript.
- Hojas de estilo en cascada, CSS.
- Lenguaje Java.
- NetBeans, IDE para desarrollar las clases de Java y los archivos JSP.
- Servidor Apache Tomcat.
- Manejador de Base de datos, MySQL 5.7.

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

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

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

4.8 MANTENIMIENTO Y ADMINISTRACIÓN

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

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

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

4.8.3 Actividades de Administración. Dentro de las actividades realizadas por los miembros del Grupo Calumet, se encuentra la tarea de administrar los servidores de los portales de las escuelas, cada semestre se asignan nuevos administradores, los cuales cuentan con un usuario dentro de uno de los portales

a los cuales presta soporte el grupo. Como administrador las tareas que se realizan comúnmente son:

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

5. PRUEBAS DEL SISTEMA

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

5.1 PRUEBAS DE VERIFICACIÓN

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

Se efectúa ejecutando paso a paso el proceso del servicio de manera que se explora cada funcionalidad que tiene el módulo desarrollado, realizando verificaciones de validación, los campos que son obligatorios no pueden quedar vacíos, por ejemplo. Las siguientes tablas describen las pruebas de cada caso de uso de los servicios que fueron desarrollados.

5.1.1 Pruebas por componente. Esta prueba se realizó para cada caso de uso de cada servicio desarrollado y descrito anteriormente:

5.1.1.1 Servicio para la edición de la Hoja de Vida de los profesores

Tabla 9 Pruebas Realizadas: Editar hoja de vida de un profesor.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Listar profesores	Ingresar al servicio y ver lista de profesores planta y catedra	✓
Editar Profesor	Acceder a la hoja de vida de un profesor.	✓

Modificar criterio	Modificar un estudio, herramientas, idiomas, experiencias, etc.	✓
--------------------	---	---

5.1.1.2 Servicios buscadores de perfil e información

Tabla 10 Pruebas Realizadas: Búsqueda de perfil

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Buscar por nombre	Digitar el nombre de un usuario y consultar el perfil	✓
Buscar por documento	Digitar documento de identificación de un usuario y consultar el perfil.	✓
Buscar estudiante por código	Digitar el código de un estudiante y consultar el perfil	✓
Buscar profesor por el código	Digitar el código de un profesor y consultar el perfil	✓
Buscar por categoría	Seleccionar categoría por categoría y ver el perfil de un usuario.	✓
Buscar por materia	Seleccionar una materia y ver el perfil de un usuario por grupo.	✓
Buscar por grupo de investigación	Seleccionar un grupo de investigación y ver el perfil de un usuario	✓

Tabla 11 Pruebas Realizadas: Búsqueda de información de un usuario

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Buscar por nombre	Digitar el nombre de un usuario y consultar la información del usuario	✓
Buscar por documento	Digitar documento de identificación de un	✓

	usuario y consultar la información del usuario	
Buscar estudiante por código	Digitar el código de un estudiante y consultar la información del usuario	✓
Buscar profesor por el código	Digitar el código de un profesor y consultar la información del usuario	✓
Buscar por categoría	Seleccionar categoría por categoría, seleccionar un usuario ver la información.	✓
Buscar por materia	Seleccionar una materia y ver la información del usuario de un grupo de clase.	✓
Buscar por grupo de investigación	Seleccionar un grupo de investigación y ver la información de un usuario	✓

5.1.1.3 Servicio Portal Profesor

Tabla 12 Pruebas Realizadas: Administración del portal profesor

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Blog profesor	Visitar módulo de administración del blog profesor.	✓
Crear menú	Crear menú editable para el portal profesor.	✓
Editar contenido	Luego de crear un menú se agrega contenido deseado.	✓
Verificar menú en el portal	Visitar en portal profesor en sesión de invitado y comprobar visibilidad del menú.	✓
Cambiar Visibilidad al menú	Editar el menú para que no se muestre en el portal profesor	✓

Crear submenú	A un menú principal crear un submenú	✓
Ordenar menús	Cambiar el orden de los menús y verificar el cambio.	✓
Eliminar submenú	Eliminar Submenú y verificar que no se muestre.	✓
Eliminar menú	Eliminar menú y verificar que no se muestre.	✓
Subir Fondo	Subir una imagen al portal del profesor como fondo.	✓
Cambiar fondo	Aplicar la imagen subida como nuevo fondo.	✓

Tabla 13 Pruebas Realizadas: Utilización del portal por usuarios UIS y públicos

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ver perfil profesor	Visitar un portal profesor para ver el perfil junto con noticias y eventos.	✓
Ver el horario	Consultar el horario el horario del profesor.	✓
Enviar correo	Enviar correo al profesor y verificar el funcionamiento del captcha.	✓
Ver producción intelectual	Ver los diferentes proyectos, artículos y demás productos académicos.	✓
Ver Blog profesor	Ver los menús editables del profesor.	✓

5.1.1.4 Servicio Preguntas, quejas, reclamos y sugerencias

Tabla 14 Pruebas Realizadas: Administración del servicio de PQRS

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Crear un tipo de PQRS	Ingresar en modo super Administrador y crear	✓

	un tipo de PQRS publica y otra privada	
Editar nombre de tipo de PQRS	Cambiar el nombre del tipo de PQRS	✓
Cambiar encargado	Eliminar un encargado de un tipo de pqrs y agregar uno nuevo	✓
Cambiar privacidad de tipo de PQRS	Cambiar un tipo de PQRS de privada a publica y viceversa.	✓
Atender una PQRS	Responder una PQRS en el foro.	✓
Subir una imagen en el foro	Atender una PQRS con una imagen.	✓
Subir un documento en el foro	Atender una PQRS con un documento.	✓
Finalizar una PQRS	Cerrar una PQRS	✓

Tabla 15 Pruebas Realizadas: Registro de una PQRS

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Crear PQRS pública	Llenar el formulario de la PQRS y enviarla	✓
Consultar PQRS pública	Digitar los campos requeridos y consultar la PQRS	✓
Crear PQRS privada	Ingresar al portal Web y crear una PQRS de tipo privada	✓
Participar en el foro	Enviar una respuesta al encargado de la PQRS	✓
Subir una imagen en el foro	Responder en el foro con una imagen.	✓
Subir un documento en el foro	Responder en el foro con un documento	✓
Calificar al encargado	Calificar de 1 a 5 estrellas la atención de la PQRS	✓

6. CONCLUSIONES

1. La creación del servicio para editar la hoja de vida y perfil del profesor por parte de un administrativo facilita la actualización de esta misma sin afectar el factor tiempo de los profesores.
2. La actualización al buscador público y al de estudiantes-egresados optimizo el código e implementó los nuevos estándares de Calumet, esto permitió darles mayor seguridad a los portales Web.
3. El nuevo servicio de Blog Profesor da mayor libertad al profesor de publicar contenido que el crea conveniente y organizarlos.
4. La actualización al portal profesor brinda a los usuarios información más reciente y una forma de contactar al profesor de manera fácil. También optimiza la actualización de rutas de los servicios del portal profesor.
5. La actualización al servicio de PQRS brinda a los usuarios públicos tener una conversación directa con el encargado de atender las solicitudes, para los usuarios privados se crearon tipos de PQRS exclusivas. Se facilita la comprensión entre encargado y usuarios al poder subir material multimedia que ayude a explicar sus respuestas.

7. RECOMENDACIONES

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

Capacitar a los profesores sobre los cambios realizados en el servicio de portal profesor para que así puedan hacer uso de manera óptima de este servicio.

Capacitar a los profesores sobre el nuevo servicio de blog profesor para que puedan hacer uso de este mismo sin ningún problema.

Capacitar a los usuarios sobre las mejoras realizadas en el servicio de PQRS para que puedan hacer buen uso del mismo y ahorren tiempo al ejecutar tareas relacionadas con este.

BIBLIOGRAFÍA

JAVASCRIPT-YA. Tutoriales del lenguaje de programación JavaScript. [En línea]. (Recuperado en 01 octubre 2017) Disponible en <http://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=0>

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.

MANUALES. Manuales básicos de JSP. [En línea]. (Recuperado en 30 septiembre 2017). Disponible en <http://www.desarrolloweb.com/manuales/73/>

PROGRAMACIÓN. En este sitio se encuentran un completo curso para aprender java. [En línea]. (Recuperado en 01 octubre 2017). Disponible en http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188

PROGRAMACIÓN FÁCIL. Manual de programación para Java Jsp (Java Server Pages). [En línea]. (Recuperado en 01 octubre 2017). Disponible en http://www.programacionfacil.com/java_servlets/start.html

STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005. Este libro se ocupa de los conceptos completos de las características de los sistemas operativos.

STUMPF, Robert, TEAGUE, Lavette. Object-Oriented Systems Analysis and Design with UML. Prentice Hall. 2004. Este libro introduce los conceptos y métodos del análisis y diseño de sistemas orientados a objetos.

SQL. En este sitio se encuentra una buena documentación con ejemplos de sql. [En línea]. (Recuperado en 07 octubre 2017). Disponible en <http://www.1keydata.com/es/sql/>

WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. Thomson International, 2003. En este libro se encuentra información sobre desarrollo de software orientado a objetos.

W3SCHOOLS. Tutoriales de cómo programar en html de gran ayuda en detalles básicos y sencillos. [En línea]. (Recuperado en 07 octubre 2017). Disponible en <http://www.w3schools.com/html/>

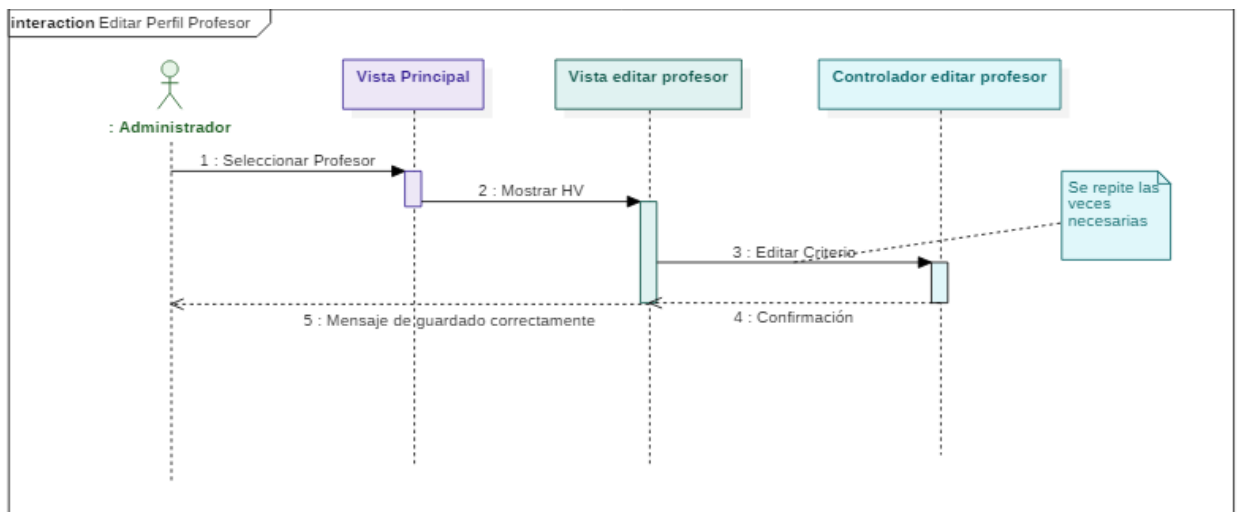
WIKISPACES. Información acerca de la Arquitectura Cliente Servidor. [En línea]. (Recuperado en 02 octubre 2017). Disponible en [URL:http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor](http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor)

ANEXOS

Anexo A. Modelo de Procesos del Sistema

Servicio para la edición de la Hoja de Vida de los profesores

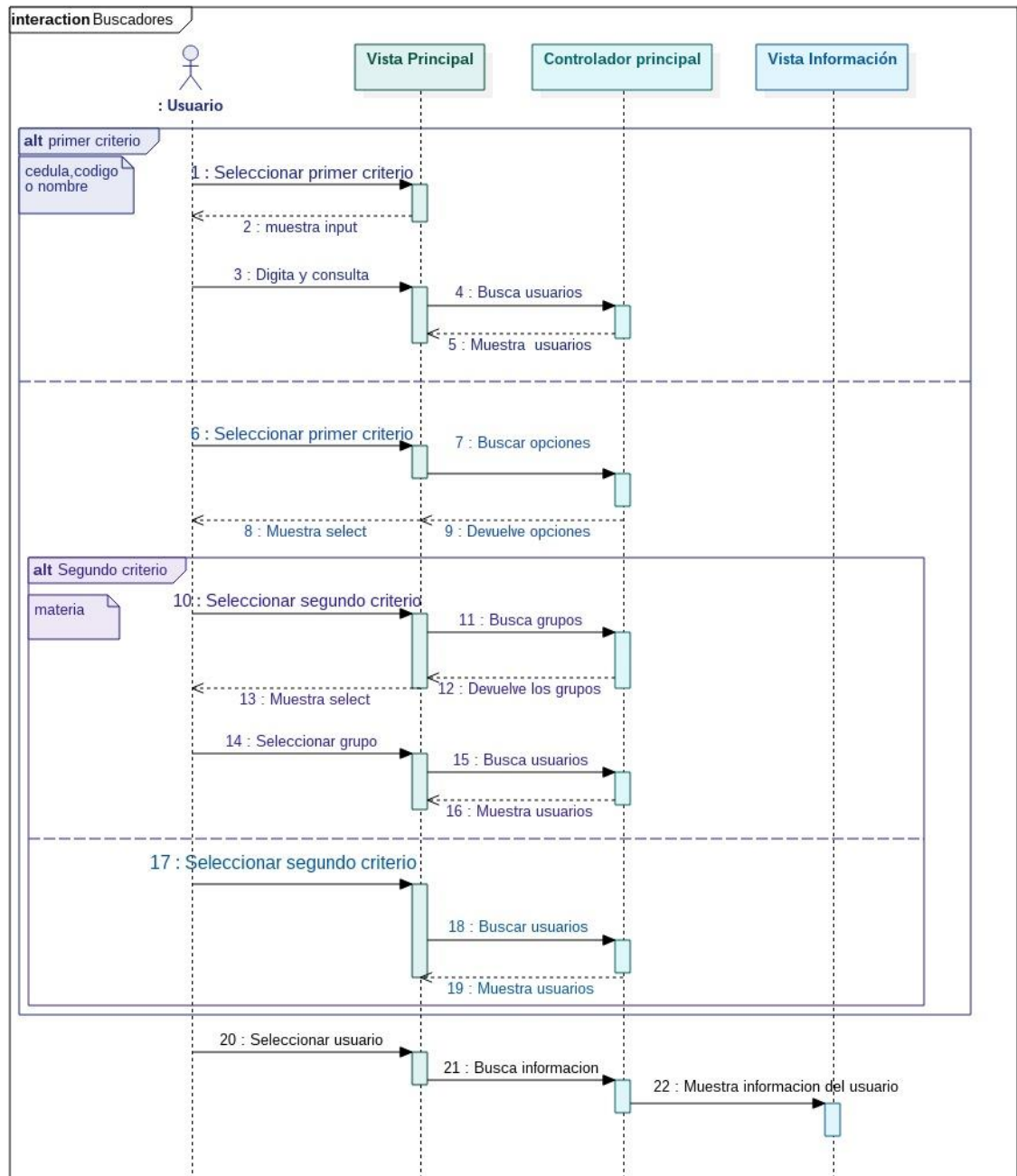
Figura 16 Diagrama de secuencia: Editar perfil profesor



1. Del listado de profesores seleccionar el profesor a editar.
2. La vista principal muestra la HV en la vista de editar profesor.
3. Desde la vista editar profesor se edita un o los criterios necesarios.
4. El controlador guarda los cambios en bases de datos y envía una confirmación a la vista de editar profesor
5. La vista de editar profesor muestra al administrador un mensaje “guardado correctamente”.

Servicio de buscadores de perfil e información

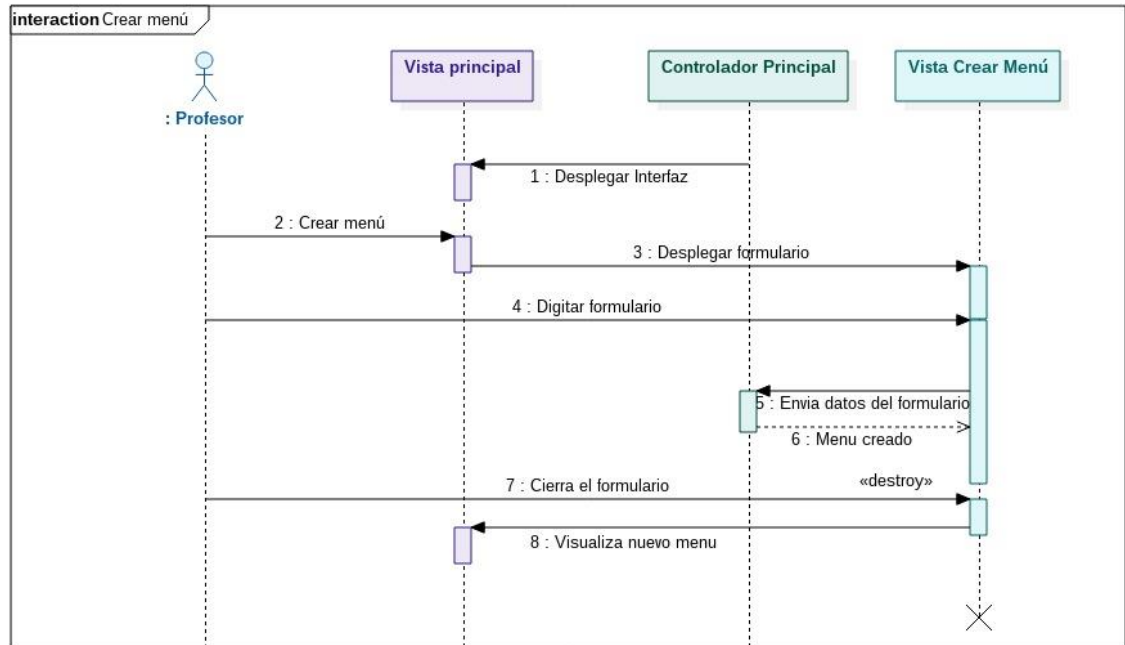
Figura 17 Diagrama de secuencia: Buscadores de perfil e información



1. El usuario selecciona el primer criterio para este caso selecciona un nombre, código o documento.
2. La vista principal muestra al usuario un input para escribir el criterio.
3. El usuario digita el criterio y da clic en consultar.
4. La vista principal envía al controlador el criterio digitado.
5. El controlador principal busca en bases de datos a los usuarios que coincidan con el criterio y devuelve la información.
6. En caso de que no sean los criterios de (1), la vista principal envía al controlador un id con el cual buscar opciones.
7. El controlador principal consulta en bases de datos las opciones posibles
8. Devuelve las opciones encontradas por el controlador a la vista
9. La vista principal muestra un select con las posibles opciones al usuario.
10. Seleccionar segundo criterio, para este caso cuando es una materia
11. La vista principal envía un id de grupo al controlador principal
12. El controlador devuelve los grupos encontrados
13. La vista principal muestra un select con las opciones de grupos de la materia
14. El usuario selecciona un grupo de la vista principal.
15. La vista principal envía los id para la realización de la búsqueda de usuarios.
16. El controlador busca en la base de datos y envía a la vista los usuarios que cumplan con la búsqueda.
17. Cuando no se cumple la condición de (10)
18. La vista principal envía al controlador los id para realizar la búsqueda
19. El controlador busca en la base de datos y envía a la vista los usuarios que cumplan con la búsqueda.
20. El usuario selecciona al usuario del que desea la información.
21. La vista principal envía al controlador el id de usuario para realizar la búsqueda.
22. El controlador principal realiza la búsqueda en las bases de datos y muestra la información en la vista de información.

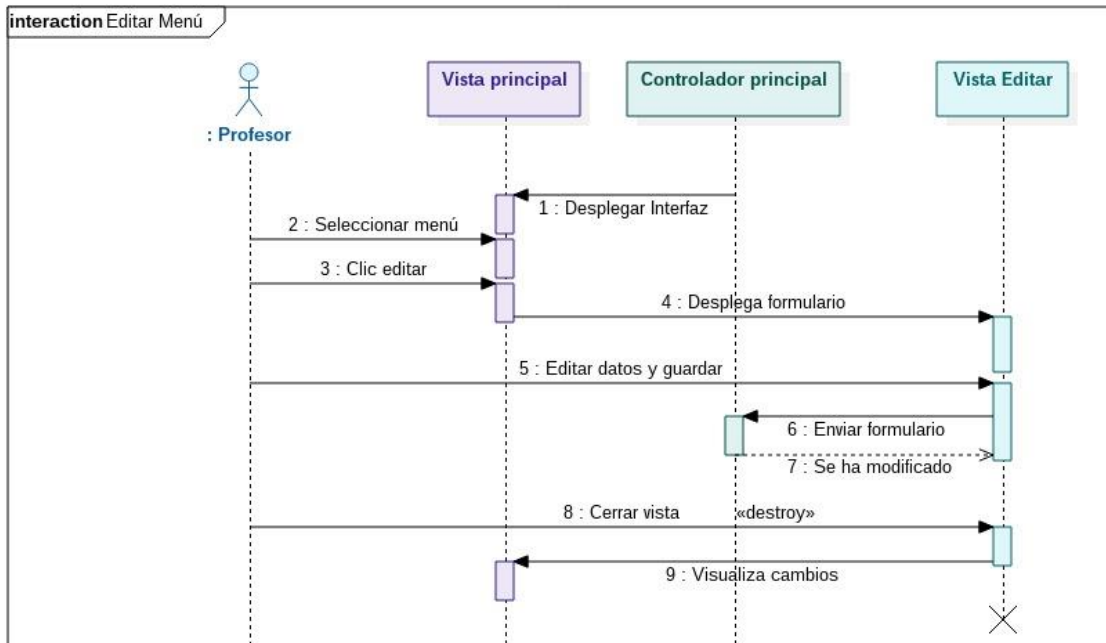
Servicio Portal Profesor

Figura 18 Diagrama de secuencia: Crear menú del blog profesor



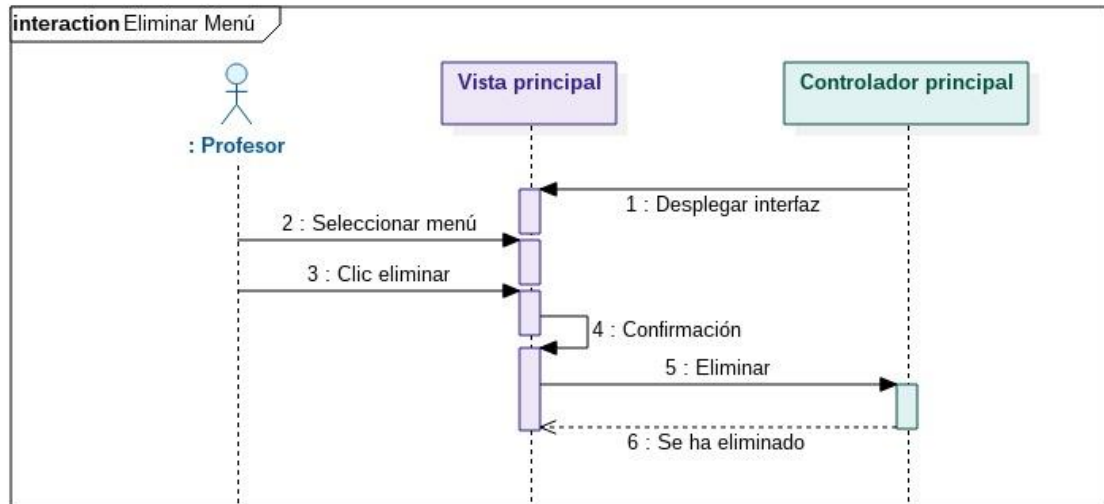
1. El controlador principal despliega la interfaz gráfica de la vista principal.
2. El profesor da clic en crear menú.
3. La vista principal despliega un formulario en la vista crear menú.
4. El profesor digita el formulario y lo envía.
5. La vista crear menú envía los datos al controlador principal.
6. El controlador principal guarda los datos del nuevo menú en bases de datos y devuelve un mensaje a la vista crear menú de éxito.
7. El profesor cierra la vista de crear menú.
8. La vista de crear menú refresca la vista principal para mostrar el nuevo menú antes de cerrarse.

Figura 19 Diagrama de secuencia: Editar menú del blog profesor



1. El controlador principal despliega la interfaz gráfica de la vista principal.
2. El profesor selecciona un menú.
3. El profesor da clic en editar menú.
4. La vista principal despliega un formulario en la vista editar menú.
5. El profesor digita el formulario y lo envía.
6. La vista crear menú envía los datos al controlador principal.
7. El controlador principal guarda los datos del menú en bases de datos y devuelve un mensaje a la vista editar menú de éxito.
8. El profesor cierra la vista de editar menú.
9. La vista de editar menú refresca la vista principal para mostrar el nuevo menú antes de cerrarse.

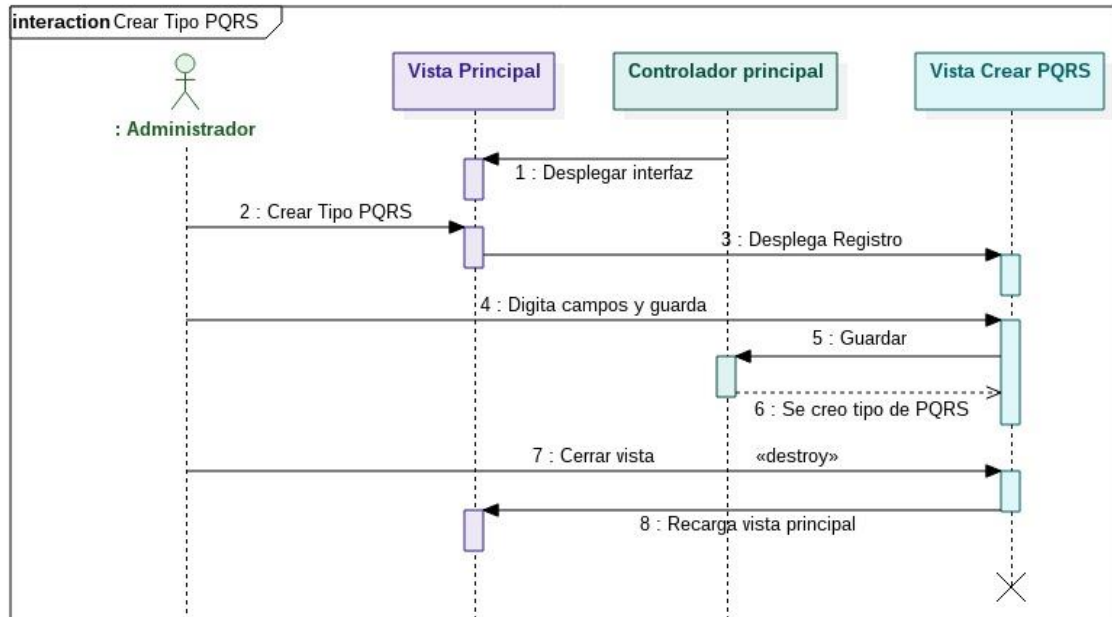
Figura 20 Diagrama de secuencia: Eliminar menú blog profesor



1. El controlador principal despliega la interfaz gráfica de la vista principal.
2. El profesor selecciona un menú.
3. El profesor da clic en eliminar menú.
4. La vista principal muestra un mensaje confirmación.
5. El profesor confirma y se procede a eliminar.
6. El controlador principal elimina de la base de datos el menú y envía un mensaje "Se ha eliminado".

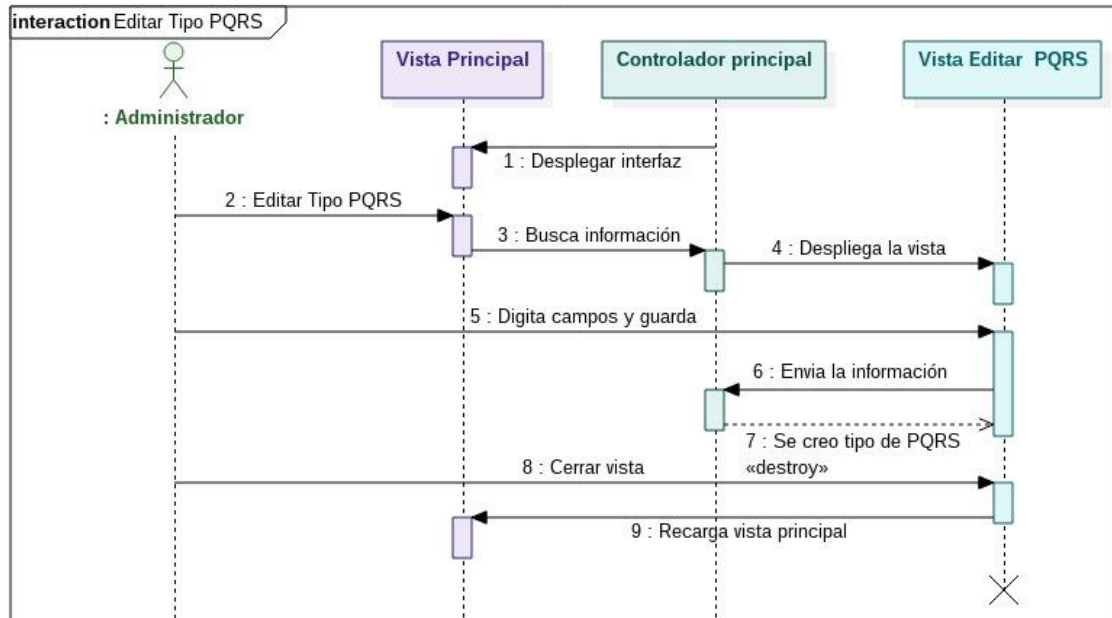
Servicio Peticiones, Quejas, Reclamos y sugerencias

Figura 21 Diagrama de secuencia: Crear Tipo de PQRS



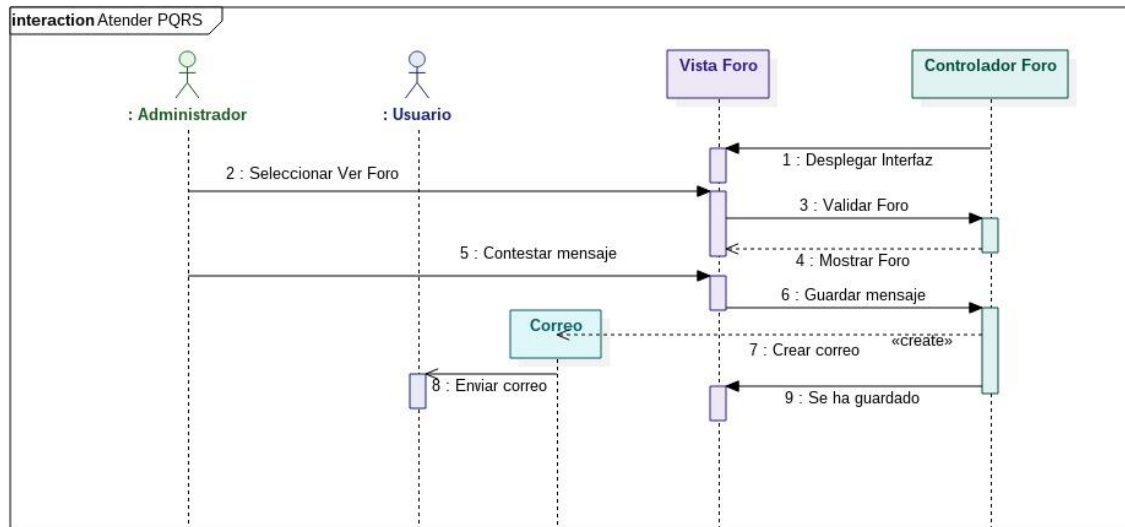
1. El controlador principal despliega la interfaz gráfica de la vista principal.
2. El administrador da clic en el botón crear tipo de PQRS en la vista principal
3. La vista principal despliega un registro en la vista de crear PQRS.
4. El administrador digita los campos de nombre, estado, privacidad y asigna encargados y envía el registro.
5. La vista Crear PQRS envía al controlador principal la información del tipo a guardar
6. El controlador principal guarda en bases de datos la información y responde a la vista crear PQRS "Se creo tipo".
7. El administrador cierra la vista crear PQRS
8. Antes de cerrarse la vista crear PQRS actualiza la vista principal.

Figura 22 Diagrama de secuencia: Modificar Tipo de PQRS



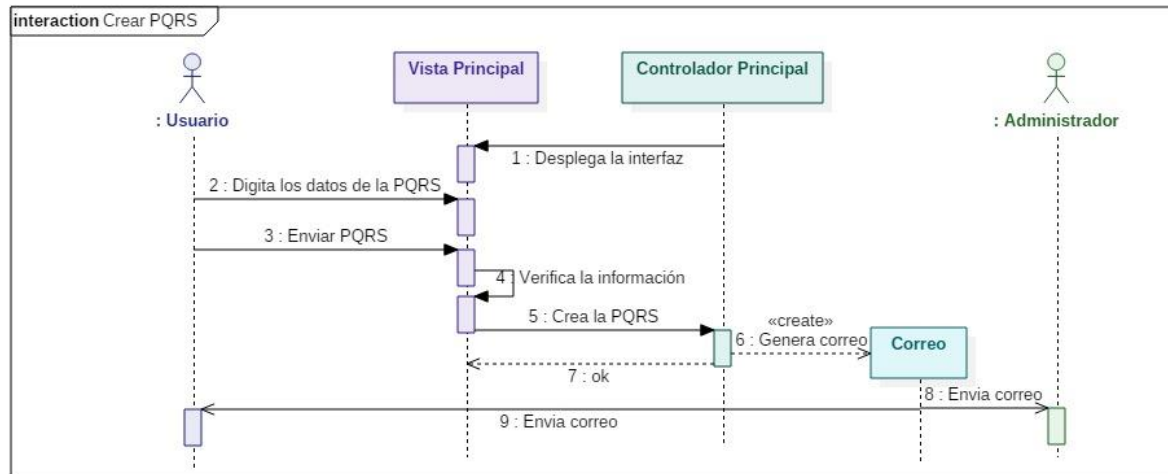
1. El controlador principal despliega la interfaz gráfica de la vista principal.
2. El administrador da clic en el botón editar tipo de PQRS en la vista principal
3. La vista principal despliega un registro en la vista de editar PQRS.
4. El administrador modifica los campos pertinentes y envía el registro.
5. La vista editar PQRS envía al controlador principal la información del tipo a modificar.
6. El controlador principal guarda en bases de datos la información y responde a la vista editar PQRS "Se modifico tipo".
7. El administrador cierra la vista crear PQRS
8. Antes de cerrarse la vista crear PQRS actualiza la vista principal.

Figura 23 Diagrama de secuencia: Atender PQRS



1. El controlador de foro despliega la interfaz de vista del foro
2. El administrador selecciona la PQRS por atender en la vista del foro.
3. La vista del foro solicita al controlador validar la información del foro.
4. El controlador del foro valida la información y muestra el foro en la vista foro
5. El administrador responde por medio de un comentario la PQRS.
6. La vista del foro envía al controlador la información del mensaje.
7. El controlador guarda la respuesta en base de datos y crea un correo para ser enviado al usuario con la notificación que se ha respondido la solicitud.
8. Correo envía al usuario de manera asincrónica la información de la PQRS y como entrar al foro para ver la respuesta.

Figura 24 Diagrama de secuencia: Crear PQRS



1. El controlador principal despliega la interfaz en la vista principal.
2. El usuario digita los campos requeridos para la PQRS
3. El usuario envía la PQRS
4. La vista principal verifica si la información es correcta.
5. Si la información es correcta procede a enviarla al controlador principal.
6. El controlador principal guarda en la base de datos la PQRS y crea un correo
7. Correo envía notificaciones tanto al usuario como administrador que la PQRS ha sido creada.