MANTENIMIENTO, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA NUEVA VERSIÓN DE LOS MÓDULOS DE AGENDA PARA COMUNIDAD UIS, COMUNIDAD ESCUELA Y PORTAL DE GRUPOS, PROGRAMAS ACADÉMICOS, PLANES DE ESTUDIO Y CORREOS PARA ADMINISTRADOR.

WEYMAR ANDRES VARGAS CASTILLO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

MANTENIMIENTO, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA NUEVA VERSIÓN DE LOS MÓDULOS DE AGENDA PARA COMUNIDAD UIS, COMUNIDAD ESCUELA Y PORTAL DE GRUPOS, PROGRAMAS ACADÉMICOS, PLANES DE ESTUDIO Y CORREOS PARA ADMINISTRADOR.

WEYMAR ANDRES VARGAS CASTILLO

Trabajo de grado para optar al título de
Ingeniero de Sistemas

Director
Msc. LUIS IGNACIO GONZÁLEZ RAMÍREZ
Magíster en Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2017

AGRADECIMIENTOS

A mis padres que han sido parte importante en cada uno de mis logros, a ellos agradezco el apoyo incondicional que me han brindado a lo largo de este camino.
A mi hermano quien ha sido un gran ejemplo a seguir.
A mis profesores quienes me han compartido sus conocimientos y me han permitido aprender cada día un poco más.
A mi director de proyecto quien además de guiarme en el desarrollo del proyecto es una excelente persona y como un padre para todos los miembros del grupo.
A todos mis compañeros del grupo Calumet con los que compartí muchos buenos momentos y los cuales me enseñaron muchas cosas, y en general a todas las personas que estuvieron presentes a lo largo de este camino y quienes de una u otra forma me aportaron algo para lograr este sueño.

CONTENIDO

Lista de tablas

Lista de figuras

Lista de anexos

RESUMEN

TÍTULO: Mantenimiento, análisis, diseño, desarrollo e implementación de una nueva versión de los módulos de Agenda para comunidad UIS, comunidad escuela y portal de grupos, Programas Académicos, Planes de estudio y Correos para administrador. *

AUTOR: Weymar Andres Vargas Castillo**

PALABRAS CLAVE: Portal Web, Calendario, Comunidad, Correo, Administrador, Asignatura, Plan de Estudios, Programa Académico.

DESCRIPCIÓN

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

Parte esencial de la Comunidad en el desarrollo de la misión de la UIS son los calendarios de actividades relacionadas con los planes de estudio, los grupos de investigación, las charlas, seminarios, conferencias, y demás actividades relacionadas con la docencia, la investigación y la extensión. Muchos de estos calendarios rigen para varias escuelas, como el calendario académico de pregrado. Otros por su contenido son de interés de ciertos grupos y comunidades solamente. Por lo anterior, es que se hace necesario llevar a cabo una reingeniería del sistema de calendarios para que cubra todas las necesidades tanto de las comunidades de las escuelas, como de los grupos.

Actualmente se cuenta con el módulo de planes de estudio y programas académicos, en este módulo se crean los planes de estudio y asignaturas que hacen parte de cada pensum, el modulo no cuenta con el servicio de gestionar programas académicos (crear, editar y eliminar), por lo tanto, se hace necesario crear para los administradores nuevas funcionalidades e interfaces que permitan subir de manera sencilla la información correspondiente a los programas académicos.

Nueva información se ha agregado a nuestras bases de datos como es: promedios, créditos aprobados y cursados, puntos, nivel, año de egreso, sede uis, y otra. También se ha agregado información concerniente a los estudiantes de las sedes, se hace necesario agregar nuevos filtros a las búsquedas que permita acceder a los estudiantes por sede, profesores por sede, y otros

---

* Trabajo de grado

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

ABSTRACT

TITLE: Maintenance, analysis, design, development and implementation of a new version of the modules of Agenda for UIS community, community school and group portal, Academic Programs, Study Plans and Posts for administrator. *

AUTHOR: Weymar Andres Vargas Castillo**

KEY WORDS: Web Portal, Calendar, Community, Mail, Administrator, Subject, Curriculum, Academic program.

DESCRIPTION

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

An essential part of the Community in the development of the UIS mission is the calendar of activities related to curricula, research groups, lectures, seminars, conferences, and other activities related to teaching, research and extension. Many of these calendars apply to various schools, such as the undergraduate academic calendar. Others because of their content are of interest to certain groups and communities only. Therefore, it is necessary to carry out a re-engineering of the calendars system to cover all the needs of both school communities and groups.

At present there is a module of curricula and academic programs, in this module the curricula and subjects that are part of each pensum are created, the module does not have the service of managing academic programs (create, edit and eliminate) , Therefore, it is necessary to create for the administrators new functionalities and interfaces that allow to easily upload the information corresponding to the academic programs.

New information has been added to our databases such as: averages, credits approved and completed, points, level, year of discharge, seat uis, and other. Information has also been added concerning the students of the venues, it is necessary to add new filters to the searches that allow access to the students by headquarters, teachers by headquarters, and others

---

* Degree Work

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

INTRODUCCIÓN

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

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

En este documento se presenta un soporte teórico, metodológico y técnico del desarrollo realizado en el servicio de: calendarios comunidad escuela para facilitar la publicación de calendarios de eventos de interés común a varias escuelas, además de la mejora de la interfaz para la presentación de estos, calendarios de los grupos para que estos puedan hacer sus calendarios de actividades y los publiquen si así lo desean, planes de estudio y programas académicos para facilitar las tareas relacionadas con la creación, eliminación y edición de programas académicos y planes de estudio con sus respectivos pensum, correos uis y administrador para permitir segmentar mucho más la búsqueda del grupo de persona a quienes se les desean enviar correos.

1. PLANTEAMIENTO DEL PROYECTO

1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL

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

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

Los módulos de Calendarios comunidad Uis y escuela, Planes de estudio y programas académicos, Correos para el administrados y correos Uis deben ser actualizados para soportar nuevas necesidades de la comunidad, la actualización consta de cambio de interfaz e implementación de nuevas funcionalidades que permitan a los usuarios interactuar con los servicios Web de una mejor manera, el módulo de Calendarios Grupos se debe desarrollar e implementar de tal forma que los grupos puedan hacer públicas sus agendas de actividades.

1.2 JUSTIFICACIÓN

Parte esencial de la Comunidad en el desarrollo de la misión de la UIS son los calendarios de actividades relacionadas con los planes de estudio, los grupos de investigación, las charlas, seminarios, conferencias, y demás actividades relacionadas con la docencia, la investigación y la extensión. Muchos de estos

calendarios rigen para varias escuelas, como el calendario académico de pregrado. Otros por su contenido son de interés de ciertos grupos y comunidades solamente. Por lo anterior, es que se hace necesario llevar a cabo una reingeniería del sistema de calendarios para que cubra todas las necesidades tanto de las comunidades de las escuelas, como de los grupos.

En el nuevo módulo de calendarios se hace necesario soportar calendarios de que rigen o son de interés de varias escuelas, es decir, debe abrirse la posibilidad de publicar calendarios en varias escuelas a la vez. También es importante permitir a los grupos de investigación, y en general a todos, crear sus calendarios de actividades, que podrán compartir con las demás escuelas si lo desean. El módulo contará con nuevas interfaces tanto para el ingreso de datos, como un moderno formato para mostrarlos. Además, se seguirá soportando la posibilidad de subir los calendarios y sus eventos desde un archivo tipo csv, que hará más sencilla la alimentación del módulo por parte de los administradores tanto de las escuelas, como de los grupos.

Con la puesta en funcionamiento de los servicios Comunidad Uis, el módulo de soporte a los Procesos de Calidad de los programas y otros, se hace necesario crear para los administradores nuevas funcionalidades e interfaces que permitan subir de manera sencilla la información correspondiente a los programas académicos. Esta información consta del código del programa, nombre, tipo (pregrado, especialización, maestrías, doctorado, …), planes de estudio de este programa, créditos, contenidos de las materias. Además, se debe agregar la información que actualmente no se muestra de los programas.

El sistema de correos de nuestro portal representa una de las herramientas que fomenta la comunicación de manera sencilla entre los miembros de la Comunidad. Nueva información se ha agregado a nuestras bases de datos como es: promedios, créditos aprobados y cursados, puntos, nivel, año de egreso, sede uis, y otra. También se ha agregado información concerniente a los estudiantes de las sedes,

es así que hoy ya las escuelas pueden promover el registro de estos estudiantes, para tenerlos en cuenta en sus actividades. Todos estos cambios nos fuerzan a realizar constantemente reingenierías a nuestros servicios para que ellos puedan encontrar de una manera más precisa a los grupos de personas a las que deseamos enviar mensajes. Se hace necesario agregar nuevos filtros a las búsquedas que permita acceder a los estudiantes por sede, profesores por sede, y otros

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Realizar las funciones de mantenimiento, análisis, diseño, desarrollo e implementación de los servicios con el fin de hacer óptimos, eficientes y sostenibles los trámites dentro de las escuelas y el acceso a la información de manera más sencilla, ágil, óptima y eficiente.

1.3.2 OBJETIVOS ESPECÍFICOS

- Realizar reingeniería al servicio de calendarios de la comunidad escuela, que contemple mejora de la interface gráfica, así como de su funcionalidad. Debe permitir calendarios privados (de acceso solo al usuario), así como públicos que se puedan mostrar en una o varias escuelas. Debe, además, permitir agregar eventos desde un archivo de manera automática

- Diseñar y desarrollar el servicio de calendarios para el portal de los grupos que les permita dar a conocer a la comunidad de su Escuela o UIS las actividades que van a realizar.

- Realizar reingeniería para la creación y administración de programas académicos, planes de estudio y asignaturas con contenidos de las escuelas que contemple, entre otras, la mejora de la interface gráfica y su adecuación a las novedades que se presentan a la fecha.

- Realizar reingeniería al servicio de correos del administrador de manera que agregue nuevos filtros que permitan a los administradores segmentar más las búsquedas a las personas a las que desea enviar correos.

- Crear o modificar las ayudas o soportes que existen a los usuarios sobre los servicios creados o modificados. Estas ayudas deben quedar en el portal de ayudas y en archivos editables para su uso.

1.4 IMPACTO Y VIABILIDAD

1.4.1 Impacto.

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

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

1.4.2 Viabilidad.

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

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

2. MARCO TEÓRICO

2.1 ARQUITECTURA CLIENTE/SERVIDOR

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

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

Figura 1. Modelo Cliente/Servidor

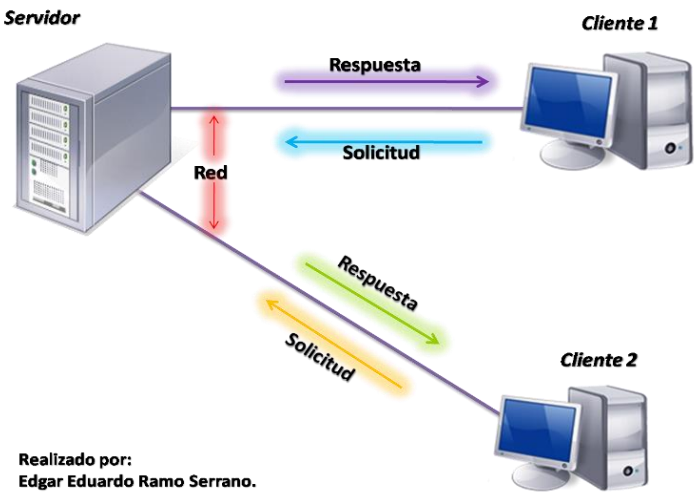

Figura 1. Modelo Cliente/Servidor

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

2.1.1 Características de la arquitectura Cliente/Servidor.

Las características básicas de una arquitectura Cliente / Servidor son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
- El proceso del cliente da la interface entre usuarios y el resto del sistema, maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades independientes.
- Las tareas del cliente y el servidor tienen diferentes requerimientos como: velocidad del procesador, memoria o capacidad del disco, por tanto, la

plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma y eso se conoce como ambiente heterogéneo.

- La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar el rendimiento y la escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores. Se puede realizar independientemente cambios en las plataformas de los clientes o de los servidores, ya sea actualización o reemplazo tecnológico, de manera transparente para el usuario final.

2.1.2 Clasificación de las arquitecturas Cliente/Servidor.

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

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

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

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

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

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

2.1.2.1 Arquitectura Cliente/Servidor de dos capas.

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

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

2.1.2.2 Arquitectura Cliente/Servidor de tres capas

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

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

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

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

2.1.4 Ventajas del esquema Cliente/Servidor

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

2.1.5 Desventajas del esquema Cliente/Servidor

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

2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

Las páginas dinámicas aportan grandes beneficios porque permiten entrar a bases de datos para extraer información que pueda presentarse al usuario, dependiendo de algunos permisos y de la misma forma para almacenar información.
Existen diferentes tecnologías para el desarrollo de páginas dinámicas entre ellas están:

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

- JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis

semejante a la del lenguaje Java y el lenguaje C. Permite, crear ventanas, mostrar texto en movimiento y verificar las entradas a un formulario.

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

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

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

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

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

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

2.2.3.1 Modelo de acceso a JSP

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

Figura 2. Modelo de acceso a JSP

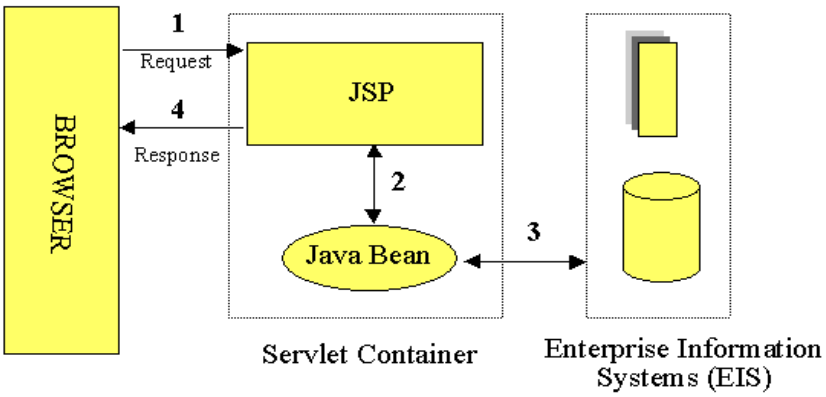

Figura 2. Modelo de acceso a JSP

Fuente: Internet: http://geneura.ugr.es/~jmerelo/JSP/

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

2.3 BASES DE DATOS

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

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

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

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

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

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

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

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

Las funciones principales de un DBMS son:

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

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

2.3.4 Ventajas de MySQL

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

2.4 NETBEANS

Es un entorno de desarrollo integrado (IDE), siendo una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas escritos en

JAVA, pero puede servir para cualquier otro tipo lenguaje de programación. Netbeans es un producto libre y gratuito sin restricciones de uso.

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

2.5 SISTEMA DE CONTROL DE VERSIONES

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

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

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

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

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

2.6 PROGRAMACIÓN UTILIZADA

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

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

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

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

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

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

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

2.6.6 Beneficios de la Programación Orientada a Objetos

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

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

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

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

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

2.7 SERVIDORES WEB

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

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

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

3. MARCO METODOLÓGICO

3.1 PROTOTIPO EVOLUTIVO

Para realizar los nuevos servicios para 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.

Figura 3. Prototipo Evolutivo

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

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

Procedimiento a seguir para la metodología planteada:

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

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

3.2 LENGUAJE DE MODELADO UNIFICADO

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

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

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

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

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

Figura 4. Diagramas de Casos de Uso

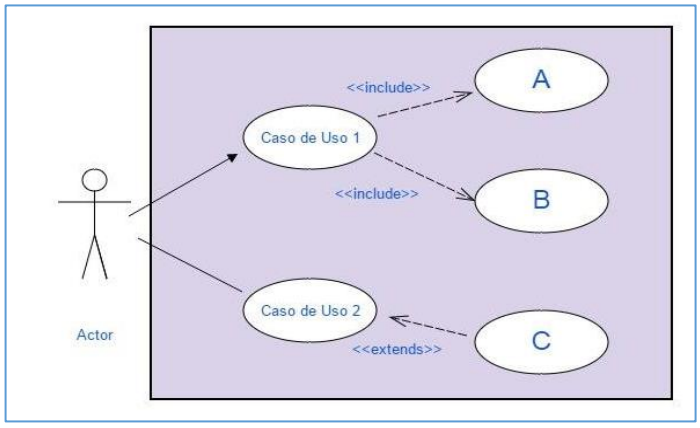

Figura 4. Diagramas de Casos de Uso

Actores: Un actor en un caso de uso representa un rol, que alguien o algo puede desempeñar dentro un sistema y no un alguien o algo específico.
En este proyecto se destacan tres clases de actores:

- **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios tienen un perfil de administrador, con el cual tiene permisos extras a los que tiene un usuario comúnmente dentro del sitio; alguno de estos son los auxiliares de administración del portal, los profesores, las secretarías de las escuelas, entre otras. Dentro de esta categoría se incluye también el súper administrador.
- **Súper Administrador:** Es el usuario que puede administrar, controlar y modificar los portales web de las escuelas, sus parámetros y sus usuarios.

- **Usuario Portal Web Comunidad Académica:** Es el tipo de usuario común de los portales web y a quien van dirigidas las páginas de servicio. Este usuario solo tiene control sobre sus servicios permitidos.

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

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

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

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

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

Tiempo: El diagrama representa al tiempo en dirección vertical. Inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.
GUI: (Siglas en Ingles) La interfaz gráfica de usuario; es la interfaz de interacción del usuario y en la que más interactividades se presentan con otros objetos.

Figura 5. Diagrama de Secuencias

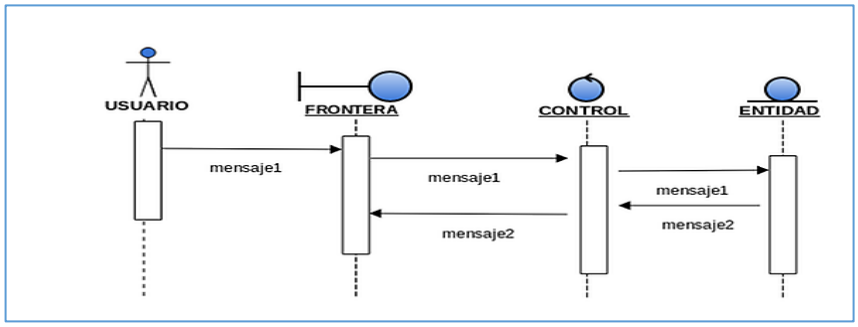

Figura 5. Diagrama de Secuencias

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 “CalendarioGrupo” de los calendarios es conocida como “TR_ CalendarioGrupo”.

- **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 “Calendarios”, es conocida como “TP_ Calendarios”.

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: Calendarios.java, Usuarios.java.

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, CrearCalendario.jsp, EnviarCorreosAdministrador.jsp.

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

- 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 “Calendarios”, “Administrador/Programas” y “Grupos/Calendario” 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.

Realizar reingeniería al servicio de calendarios de la comunidad escuela, que contemple mejora de la interface gráfica, así como de su funcionalidad. Debe permitir calendarios privados (de acceso solo al usuario), así como públicos que se puedan mostrar en una o varias escuelas. Debe, además, permitir agregar eventos desde un archivo de manera automática.

Objetivo inicial:

Permitir la creación de calendarios que puedan ser visibles a un solo usuario (creador), los usuarios de una o varias escuelas y público de la comunidad académica.

Requisitos finales del objetivo:

- Facilitar la publicación de calendarios en común a varias escuelas.
- Permitir la creación, edición y eliminación de calendarios tanto públicos como privados.
- Cambiar la interfaz de visualización de los eventos de los diferentes calendarios
- Permitir la creación, edición y eliminación de eventos asociados a diferentes calendarios
- Permitir la publicación de diferentes calendarios de eventos.
- Permitir subir eventos a diferentes calendarios por medio de un archivo externo (.CSV)
- Permitir a cualquier usuario crear sus calendarios privados.

Diseñar y desarrollar el servicio de calendarios para el portal de los grupos que les permita dar a conocer a la comunidad de su Escuela o UIS las actividades que van a realizar.

Objetivo inicial:
Desarrollar e implementar el servicio de calendarios en el portal de los grupos.

Requisitos finales del objetivo:

- Crear interfaz que permita la creación, edición y eliminación de calendarios de los grupos.
- Permitir que los usuarios de los grupos creen y publiquen calendarios ya sea solo al grupo como a la comunidad académica en general.
- Permitir la creación, edición y eliminación de eventos asociados a diferentes calendarios
- Permitir la publicación de diferentes calendarios de eventos.
- Permitir a los miembros de un grupo crear sus propios calendarios privados

Realizar reingeniería para la creación y administración de programas académicos, planes de estudio y asignaturas con contenidos de las escuelas que contemple, entre otras, la mejora de la interface gráfica y su adecuación a las novedades que se presentan a la fecha.

Objetivo inicial:
Implementar una interfaz que permita la creación de programas académicos y planes de estudio con mayor facilidad.

Requisitos finales del Portal de Producción Intelectual:

- Cambiar la interfaz de gestionar asignaturas.
- Permitir modificar y actualizar los datos y contenidos de las asignaturas.

- Permitir la eliminación de asignaturas.
- Implementar interfaz que permita la administración (creación, edición y eliminación) de programas académicos.
- Cambiar la interfaz de gestionar planes de estudio.
- Cambiar la interfaz de creación y edición de pensum para los planes de estudio.

Realizar reingeniería al servicio de correos del administrador de manera que agregue nuevos filtros que permitan a los administradores segmentar más las búsquedas a las personas a las que desea enviar correos.

Objetivo inicial

Implementar nuevos filtros a la búsqueda de personas para enviar correos.

Requisitos finales del objetivo:

- Cambiar interfaz de envío de correos del administrador.
- Implementar filtro de estudiantes por nivel para el envió de correos del lado del administrador.
- Agregar filtro de búsqueda por estado de usuario para el envió de correos del lado del administrador.
- Agregar filtro de estudiantes de pregrado por sedes al servicio de Correos Uis.

Crear o modificar las ayudas o soportes que existan a los usuarios sobre los servicios creados o modificados. Estas ayudas deben quedar en el Portal de Ayudas y en archivos editables para su uso, y posible envió.

Objetivo inicial:

Actualizar el material que se encuentra en el Portal de Ayudas, para brindar ayuda o soporte sobre los cambios realizados en el Portal Web Comunidad Académica.

Requisitos finales del objetivo:

- Actualización del material que se encuentra en el Portal de Ayudas del portal Web Comunidad Académica.
- Dar orientación a los integrantes del grupo sobre los cambios realizados en Portal para que ellos puedan brindar un soporte eficaz a solicitudes que pueda realizar el usuario.

4.1.1 Diagramas de Casos de Uso

Servicio de Calendarios comunidad escuela y grupos

Figura 6. Diagrama de Casos de Uso: Calendarios comunidad escuela y grupos

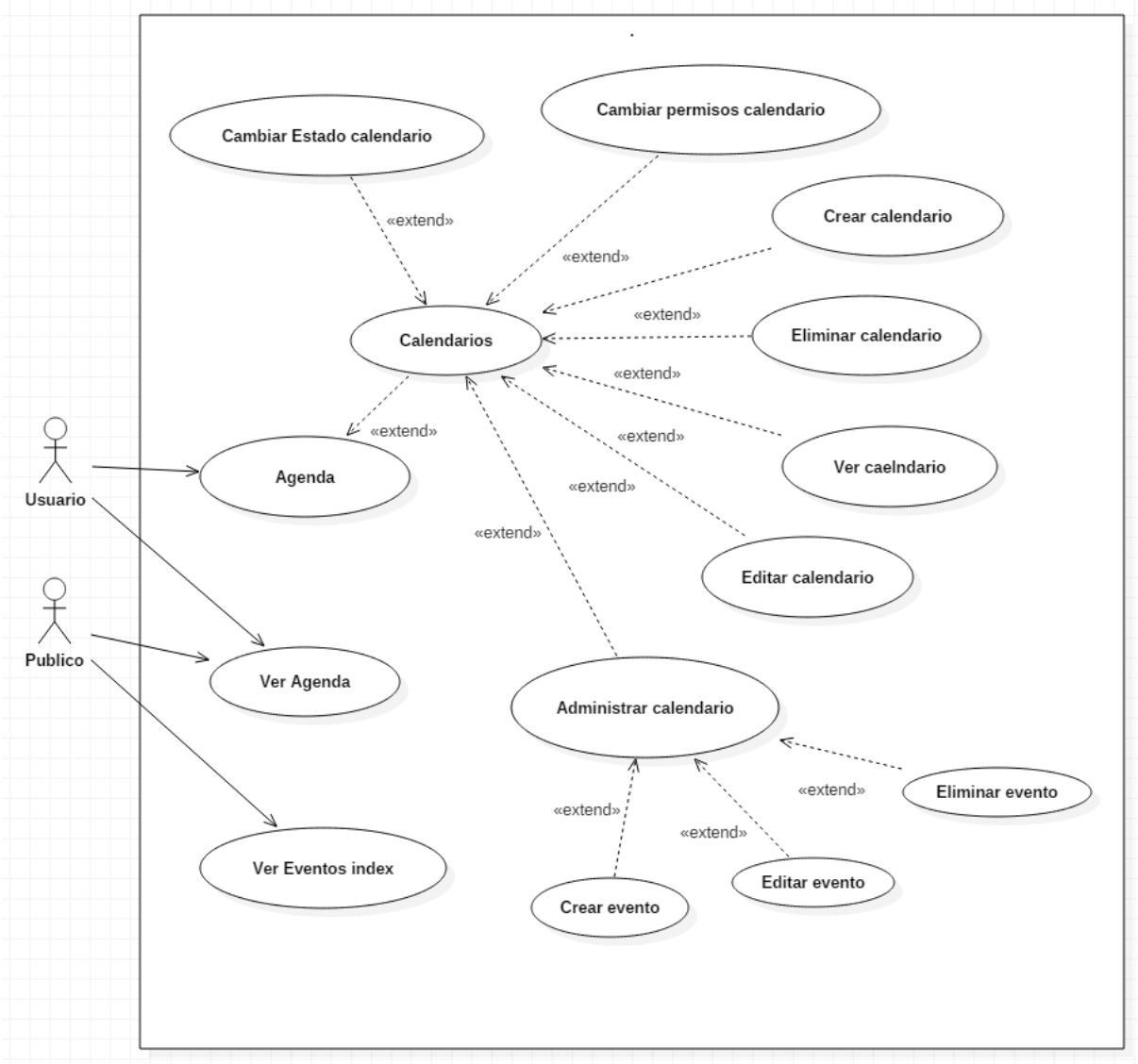

Figura 6. Diagrama de Casos de Uso: Calendarios comunidad escuela y grupos

Servicio de Planes de estudio y programas académicos

Figura 7. Diagrama de Casos de Uso: Planes de estudio y programas académicos

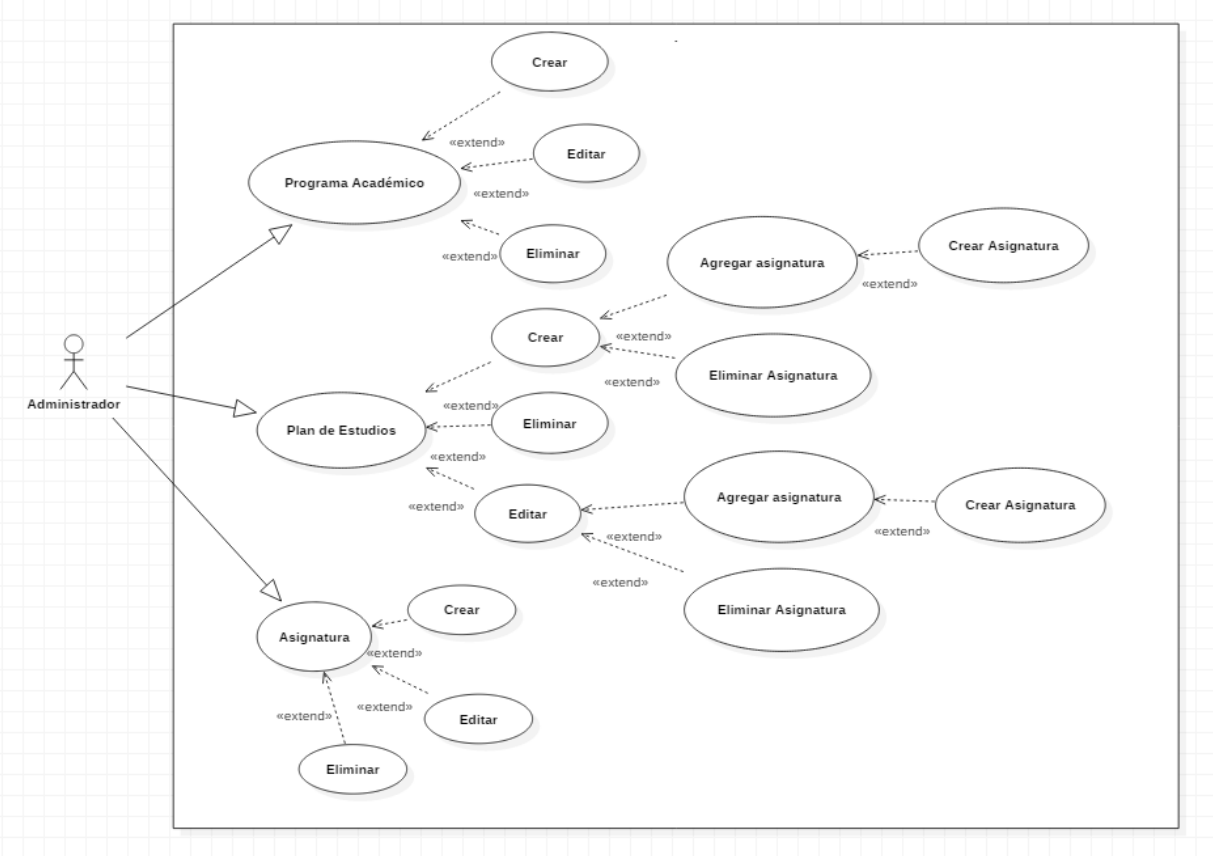

Figura 7. Diagrama de Casos de Uso: Planes de estudio y programas académicos

Figura 8. Diagrama de Casos de Uso: Correos UIS y Administrador

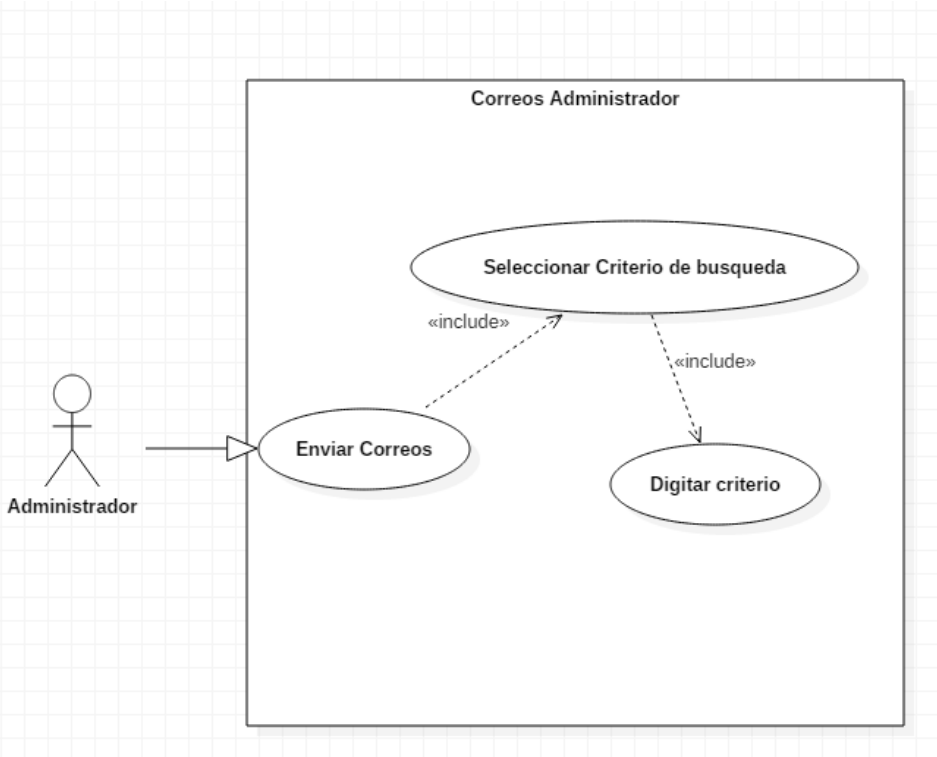

Figura 8. Diagrama de Casos de Uso: Correos UIS y Administrador

4.1.2 Documentación de Casos de Uso del Sistema

4.1.2.1 Servicios de Calendarios comunidad uis, escuela y grupos

Tabla 1. Casos de uso: Calendarios comunidad uis, escuela y grupos

TÍTULO	FUNCIONES PRIMARIAS
Ver Eventos índex	Ver los eventos más próximos a la fecha actual en el índex del portal web
Ver Agenda	Ver todos los eventos de los diferentes

	calendarios públicos
Calendarios	Acceder a servicio que muestra los eventos de los calendarios tanto públicos como privados
Calendarios-Cambiar Estado	Cambiar el estado de un calendario (Activo o Inactivo)
Calendarios-Cambiar permisos	Cambiar lo permisos de visibilidad de un calendario (Solo para mí, Publico, Usuarios Portal)
Calendarios-Crear	Creación de un calendario con la información mínima requerida
Calendarios-Eliminar	Eliminación de un calendario y todos sus eventos asociados
Calendarios-Ver calendario	Permite la visualización de todos los eventos de un calendario
Calendarios-Editar	Modificar y actualizar los datos mínimos de un calendario
Calendarios-Administrar calendario	Acceder a la administración de un calendario donde se muestran todos los eventos de dicho calendario
Calendarios-Administrar calendario-Crear Evento	Crear un evento con los datos mínimos requeridos.
Calendarios-Administrar calendario-Editar Evento	Modificar y actualizar los datos mínimos requeridos de un evento
Calendarios-Administrar calendario-Eliminar Evento	Eliminación de un evento.

Tabla 1. Casos de uso: Calendarios comunidad uis, escuela y grupos

4.1.2.2 Servicio de Planes de Estudio y Programas Académicos

Tabla 2. Casos de uso: Planes de Estudio y Programas Académicos

TÍTULO	FUNCIONES PRIMARIAS
Programa Académico	Acceder al servicio que muestra la información correspondiente a los programas académicos.
Programa Académico-Crear	Crear un programa académico agregando la información mínima obligatoria
Programa Académico-Editar	Modificar y actualizar la información correspondiente a un programa académico
Programa Académico-Eliminar	Eliminar un programa académico
Planes de Estudio	Acceder al servicio que muestra la información correspondiente a los planes de estudio
Planes de Estudio-Crear	Permite la creación de un plan de estudios agregando la información mínima requerida
Planes de Estudio-Crear-Agregar Asignatura	Agregar asignatura al pensum de un plan de estudios cuando se está creando.
Planes de Estudio-Crear-Eliminar Asignatura	Eliminar asignatura del pensum de un plan de estudios cuando se está creando
Planes de Estudio-Eliminar	Eliminar un plan de estudios incluyendo todas las asignaturas del pensum.
Planes de Estudio-Editar Pensum	Modificar pensum de un plan de estudios (Agregar y eliminar asignaturas de los niveles)

Planes de Estudio-Editar Pensum-Agregar Asignatura	Agregar asignatura al pensum de un plan de estudios cuando se está editando el pensum
Planes de Estudio-Editar Pensum-Eliminar Asignatura	Eliminar asignatura del pensum de un plan de estudios cuando se está editando el pensum.
Planes de Estudio-Editar	Modificar y actualizar la información mínima de un plan de estudios.
Asignatura	Acceder al servicio que muestra la información correspondiente a las asignaturas
Asignatura-Crear	Crear asignatura agregando la información mínima obligatoria
Asignatura-Editar	Modificar y actualizar la información mínima requerida de una asignatura
Asignatura-Eliminar	Eliminación de una asignatura

Tabla 2. Casos de uso: Planes de Estudio y Programas Académicos Planes de Estudio

4.1.2.3 Servicio de Correos UIS y Administrador

Tabla 3. Casos de uso: Correos UIS y Administrador

TÍTULO	FUNCIONES PRIMARIAS
Enviar Correos	Permite al administrador del sistema enviar correos a diferentes tipos de usuarios.
Seleccionar criterio de búsqueda	Permite filtrar la búsqueda por distintos criterios para buscar el grupo o usuario a quien se desea enviar un correo.
Digitar criterio	Después de seleccionar un criterio de búsqueda el usuario debe digitar algún dato relacionado con ese criterio.

Tabla 3. Casos de uso: Correos UIS y Administrador

4.1.3 Diseño y Análisis

Diagrama Entidad/Relación de los servicios desarrollados

Servicio Calendarios Comunidad Escuela y Grupos

Figura 9. Diagrama E/R: Calendarios comunidad escuela y grupos

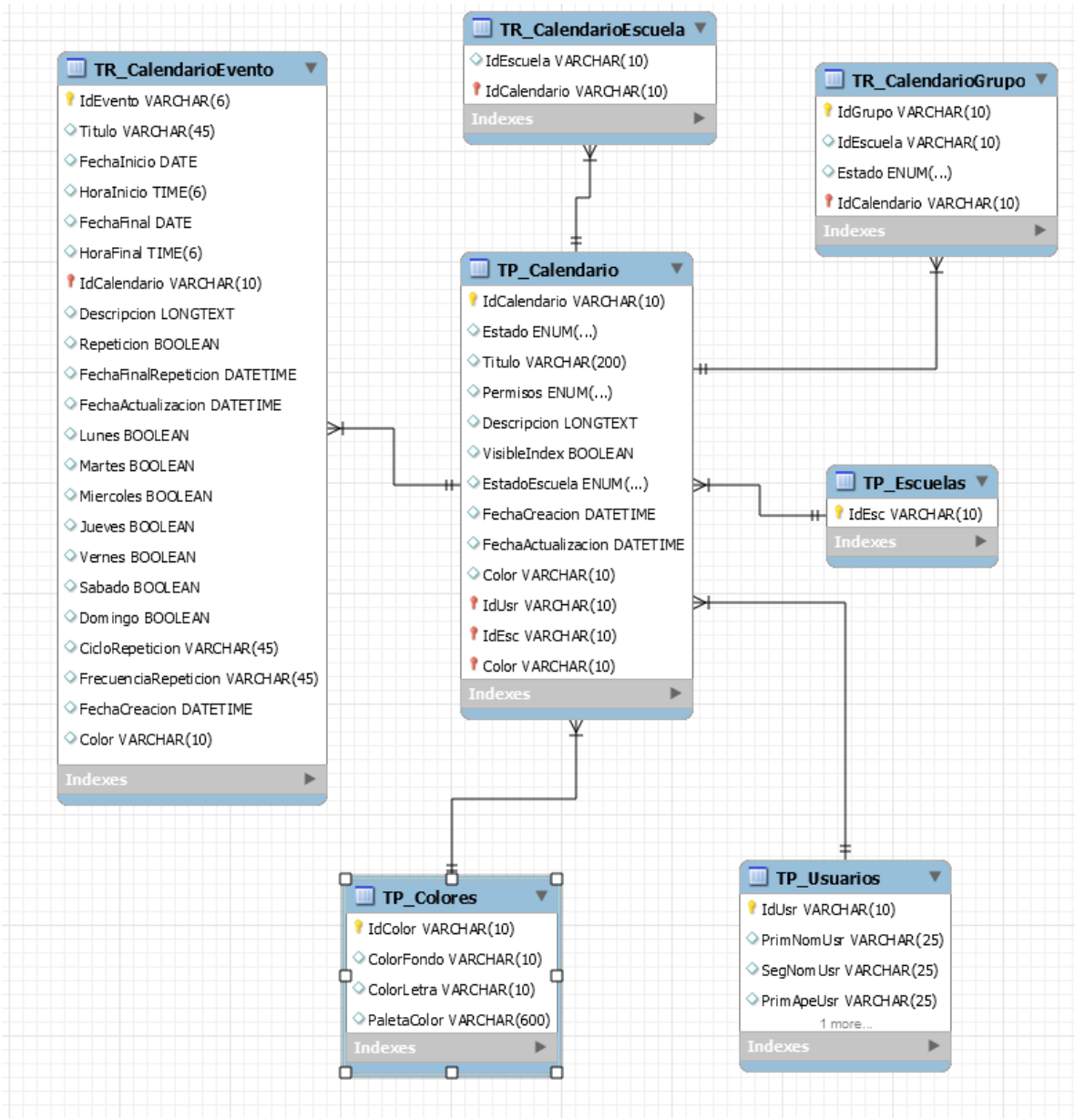

Figura9. Diagrama E/R: Calendarios comunidad escuela y grupos

Servicio Planes de Estudio y Programas Académicos

Figura 10. Diagrama E/R: Planes de Estudio y Programas Académicos

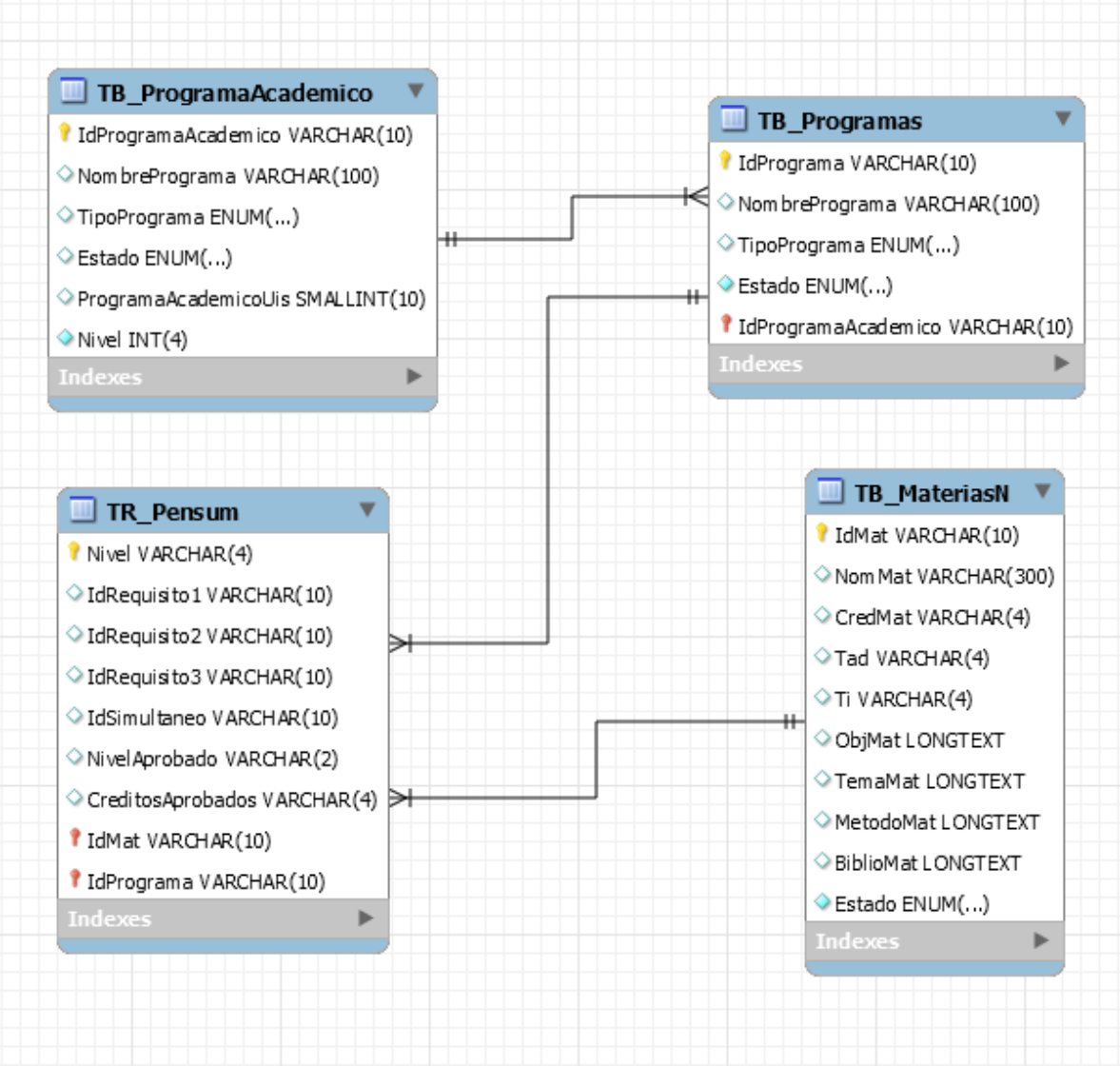

Figura 10. Diagrama E/R: Planes de Estudio y Programas Académicos

Servicio Correos Uis

Figura 11. Diagrama E/R: Correos Uis

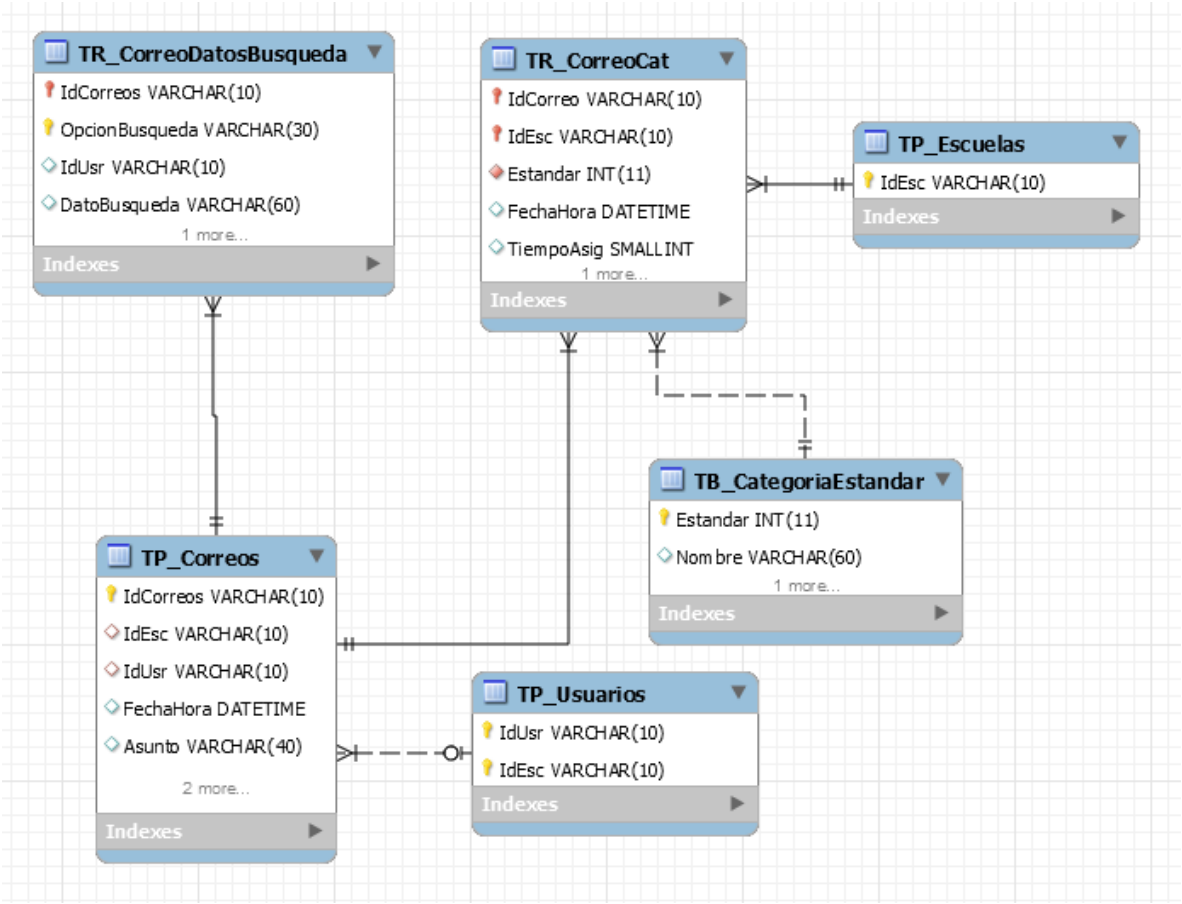

Figura 11. Diagrama E/R: Correos Uis

Descripción de las Entidades

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

Tabla 4. Descripción de las Entidades

ENTIDAD	DESCRIPCION
TP_Usuarios	Contiene la información de todos los usuarios registrados en el portal.
TP_Calendario	Contiene la información de los diferentes calendarios creados por los usuarios
TR_CalendarioGrupo	Relaciona los grupos con los calendarios creados.
TR_CalendarioEscuela	Relaciona los calendarios con las escuelas a las cuales va dirigido.
TR_CalendarioEvento	Contiene la información de los eventos asociados a los diferentes calendarios.
TP_Colores	Contiene la información de los colores y paleta de colores disponible para los calendarios y sus eventos.
TP_Escuelas	Contiene la información de las diferentes escuelas.
TP_Correos	Contiene la información de los correos.
TR_CorreoCat	Relaciona los correos con las categorías de usuarios
TR_CorreoDatosBusqueda	Relaciona los correos con las opciones de búsqueda de los usuarios cuando la categoría es especial (0).
TB_CategoriaEstandar	Contiene la información relacionada con las diferentes categorías de usuarios.
TB_ProgramaAcademico	Contiene la información relacionada con los programas académicos.
TB_Programas	Contiene la información relacionada con los planes de estudio.
TB_MateriasN	Contiene la información relacionada con

	las materias.
TR_Pensum	Relaciona los planes de estudio con las materias y sus requisitos.

Tabla 4. Descripción de las Entidades

4.1.4 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).

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

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

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

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

4.2 MANTENIMIENTO Y ADMINISTRACIÓN

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

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

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

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

- Realizar periódicamente copias de las bases de datos Diamante y División.
- 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 de Calendarios comunidad escuela y grupos

Tabla 5. Pruebas Realizadas: Calendarios comunidad escuela y grupos

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ver Eventos índex	Ver los eventos más próximos de los calendarios públicos.	
Ver Agenda	Ver todos los eventos de los diferentes calendarios públicos	
Calendarios	Al acceder al servicio se muestran los eventos de los calendarios tanto públicos como privados	
Calendarios-Cambiar Estado	Permite cambiar el estado de un calendario (Activo o Inactivo)	
Calendarios-Cambiar permisos	Permite cambiar lo permisos de visibilidad de un calendario (Solo para mí, Publico, Usuarios Portal)	
Calendarios-Crear	Al crear el calendario valida los campos obligatorios y almacena correctamente.	
Calendarios-Eliminar	Elimina el calendario seleccionado y borra el registro del mismo como de los eventos relacionados a este.	
Calendarios-Ver calendario	Visualización correcta de los eventos del calendario seleccionado.	
Calendarios-Editar	Al editar la información básica de un calendario se actualiza correctamente la información.	
Calendarios-Administrar calendario	Al acceder al servicio se muestran los eventos del calendario seleccionado.	
Calendarios-		

Administrar calendario-Crear Evento	Al crear un evento se validan lo campos obligatorios y se almacena correctamente.	
Calendarios-Administrar calendario-Editar Evento	Al editar la información básica de un evento se actualiza correctamente la información.	
Calendarios-Administrar calendario-Eliminar Evento	Elimina el evento seleccionado y borra su registro de base de datos.	

Tabla 5. Pruebas Realizadas: Calendarios comunidad escuela y grupos

5.1.1.2 Servicio de Planes de estudio y programas académicos

Tabla 6. Pruebas Realizadas: Producción Intelectual

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Programa Académico	Al acceder al servicio se listan correctamente los programas académicos.	
Programa Académico-Crear	Al crear un programa académico se validan los campos obligatorios y se almacena correctamente.	
Programa Académico-Editar	Al editar el programa académico se actualiza correctamente la información.	
Programa Académico-Eliminar	Al eliminar el programa académico se borran los registros que corresponden a este y sus planes de estudio asociados	
Planes de Estudio	Al acceder al servicio lista correctamente todos los planes de estudio.	
Planes de Estudio-Crear	Al crear un plan de estudios se validan los campos obligatorios y se almacena correctamente.	
Planes de Estudio-Crear-Agregar Asignatura	Al agregar una asignatura se almacena correctamente.	
Planes de Estudio-Crear-Eliminar Asignatura	Se elimina correctamente a la asignatura y borra su registro.	

Planes de Estudio-Eliminar	Elimina el plan de estudios y sus asignaturas, borra los registros exitosamente.	
Planes de Estudio-Editar Pensum	Al editar un pensum de un plan de estudios actualiza correctamente los datos relacionados con las asignaturas.	
Planes de Estudio-Editar Pensum-Agregar Asignatura	Al agregar una asignatura se almacena correctamente.	
Planes de Estudio-Editar Pensum-Eliminar Asignatura	Se elimina correctamente a la asignatura y borra su registro.	
Planes de Estudio-Editar	Al editar un plan de estudios se actualizan correctamente los datos	
Asignatura	Al acceder al servicio se listan todas asignaturas	
Asignatura-Crear	Al crear una asignatura se validan los campos obligatorios y almacena correctamente	
Asignatura-Editar	Al editar una asignatura se actualizan correctamente los datos	
Asignatura-Eliminar	Elimina la asignatura y borra su registro correctamente.	

Tabla 6. Pruebas Realizadas: Planes de estudio y programas académicos

5.1.1.3 Servicio de correos uis y administrador.

Tabla 7. Pruebas Realizadas: Correos uis y administrador

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Enviar Correos	Carga correctamente las opciones para el filtrado y búsqueda de usuarios.	
Seleccionar criterio de búsqueda	Carga correctamente los criterios de búsqueda y al seleccionar alguno hace bien el filtrado	
Digitar criterio	Después de seleccionar un criterio de busca al digitar lo que se quiere buscar, carga los usuarios que cumplen con el criterio de búsqueda perfectamente.	

Tabla 7. Pruebas Realizadas: Correos uis y administrador

6. CONCLUSIONES

- La reingeniería al servicio de calendarios comunidad escuela facilita la publicación de calendarios de común interés a varias escuelas, además de contar con una interfaz mucho más agradable que la versión anterior.

- El desarrollo e implementación del servicio de calendarios para los grupos permite a estos publicar su calendario de actividades ya sea a los miembros de los mismos como a la comunidad en general.

- La reingeniería del servicio planes de estudio permite la creación de programas académicos de una manera mucho más sencilla, además permite la creación y edición del pensum de cada plan de estudios de forma más cómoda e intuitiva.

- La reingeniería al servicio de correos uis y para el administrador permite hacer un filtro más detallado del grupo de persona a las cuales se les desee enviar algún correo.

- La orientación a los nuevos integrantes del grupo calumet, su familiarización con el sito, las herramientas y tecnologías utilizadas en el desarrollo y mantenimiento de los portales y la dinámica de trabajo de los miembros de éste, permiten la continuidad de la labor realizada por el grupo, en pro de una mejora continua de los servicios.

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 usuarios sobre los cambios realizados en el servicio de calendarios para que así puedan hacer uso de manera óptima de este servicio.
- Capacitar a los usuarios de los grupos sobre el nuevo servicio implementado para que puedan hacer buen uso del mismo.
- Capacitar a los usuarios sobre las mejoras realizadas en el servicio de planes de estudio para que puedan hacer buen uso del mismo y ahorren tiempo al ejecutar tareas relacionadas con el mismo.

BIBLIOGRAFÍA

JAVASCRIPT-YA. Tutoriales del lenguaje de programación JavaScript. [En línea]. [Citado 26 enero, 2017]. <URL: http://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=0>.

W3SCHOOLS. Tutoriales de cómo programar en html de gran ayuda en detalles básicos y sencillos. [En línea]. [Citado 20 enero, 2017]. <URL: http://www.w3schools.com/html/>.

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.

SQL. En este sitio se encuentra una buena documentación con ejemplos de sql. [En linea]. [Citado 20 enero, 2017]. <URL: http://www.1keydata.com/es/sql/ >

MANUALES. Manuales básicos de JSP. [En línea]. [Citado 18 enero, 2017]. Disponible en Internet: <URL: http://www.desarrolloweb.com/manuales/73/ >.

PIATTINI, Mario, CALVO-MANZANO, José A., CERVERA, Joaquín, FERNANDEZ, Luis. Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Alfaomega, 2000. Este libro contiene información sobre técnicas para el buen modelado de aplicaciones informáticas.

PROGRAMACIÓN. En este sitio se encuentran un completo curso para aprender java. [En línea]. [Citado 23 enero, 2017]. <URL: 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]. [Citado 16 Septiembre, 2016]. <URL:http://www.programacionfacil.com/programacion:manual_java_jsp>.

SCHMULLER, JOSEPH. Aprendiendo UML en 24 horas. 1ª ed. México: Alhambra Mexicana S.A, 2000. En este libro se encuentra una guía muy práctica que permite conocer y entender sobre UML.

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.

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.

WIKISPACES. Información acerca de la Arquitectura Cliente Servidor. [En línea]. [Citado 13 enero, 2017]. Disponible en Internet: <URL:http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor>.

YOUBLISHER. Tutoriales básicos sobre UML. [En línea]. [Citado 18 enero, 2017]. <URL: http://www.tutorialspoint.com/uml/ >

ANEXOS

Anexo A. Modelo de Procesos del Sistema

Servicio Calendarios comunidad escuela y grupos

Figura 12. Diagrama de secuencia: Crear calendario

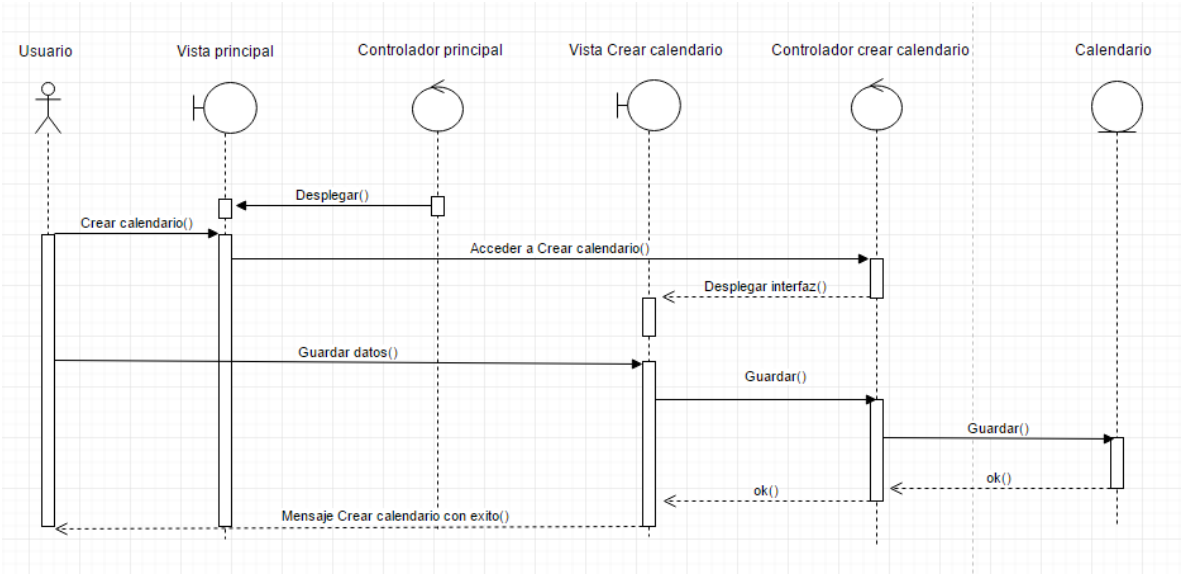

Figura 12. Diagrama de secuencia: Crear calendario.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crear un calendario en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear calendario al Controlador.
4. El controlador Crear calendario recibe la petición y despliega la interfaz en la vista crear.
5. El Usuario quiere guardar los datos del calendario.
6. La Vista crear calendario envía la petición al Controlador crear y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador Crear calendario y la Vista crear calendario muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.

Figura 13. Diagrama de secuencia: Editar Calendario

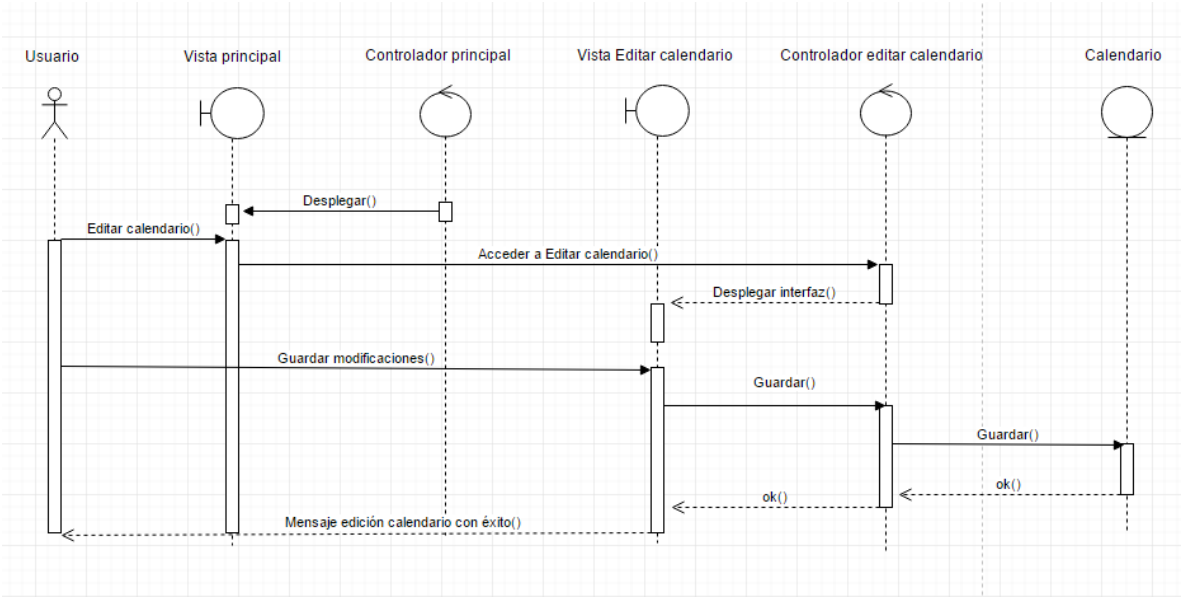

Figura 13. Diagrama de secuencia: Editar Calendario.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita Editar un calendario en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de editar calendario al Controlador.
4. El controlador Editar calendario recibe la petición y despliega la interfaz en la vista Editar.
5. El Usuario desea guardar los datos modificados del calendario.
6. La Vista editar calendario envía la petición al Controlador editar y el controlador hace la petición de guardar a la base de datos.
7. La base de datos actualiza la información y da una respuesta de OK al Controlador editar calendario y la Vista editar calendario muestra un mensaje.
8. El Usuario ve el mensaje de modificado con éxito y oprime aceptar para terminar.

Figura 14. Diagrama de secuencia: Eliminar Calendario

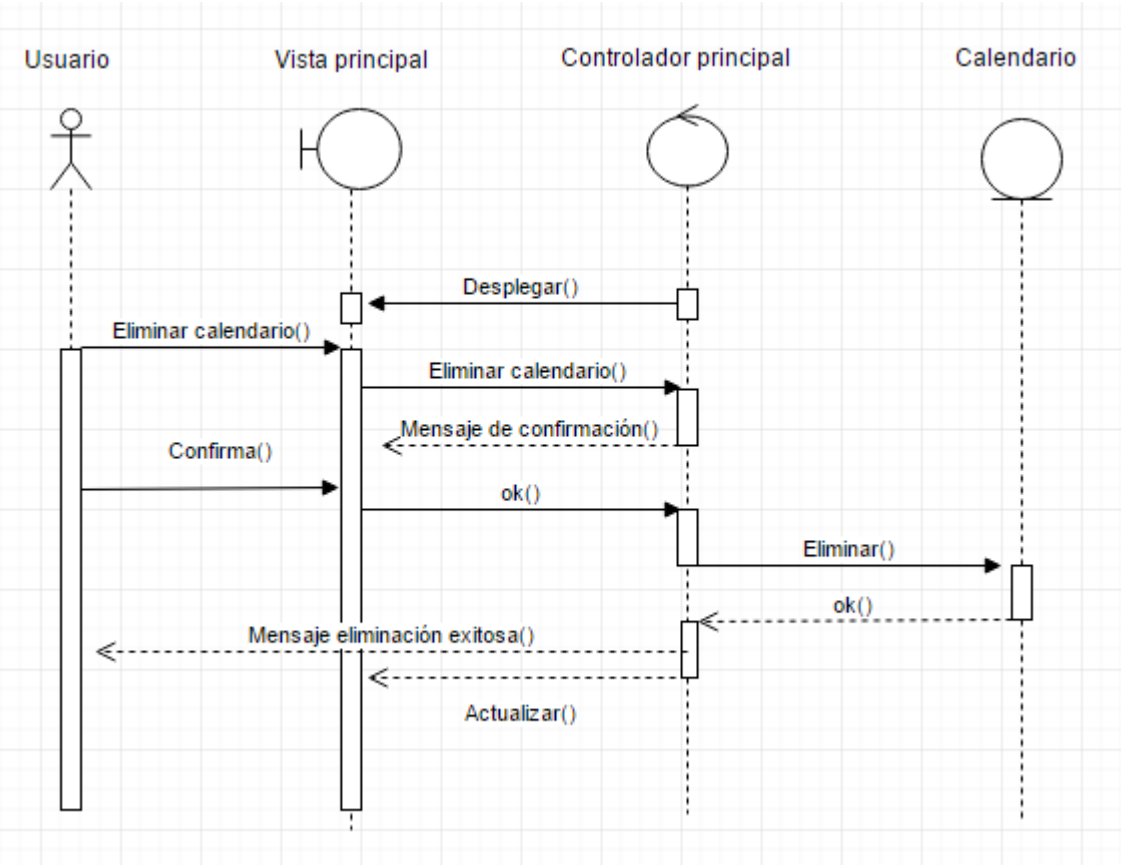

Figura 14. Diagrama de secuencia: Eliminar Calendario.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El usuario solicita eliminar un calendario.
3. El Controlador Principal envía un mensaje de confirmación por seguridad.
4. El Usuario confirma que quiere eliminar el calendario.
5. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
6. El Controlador Principal solicita a la base de datos eliminar el calendario.
7. La base de datos confirma que se eliminó el calendario.
8. El controlador confirma que se eliminó y actualiza Vista Principal

Figura 15. Diagrama de secuencia: Crear evento

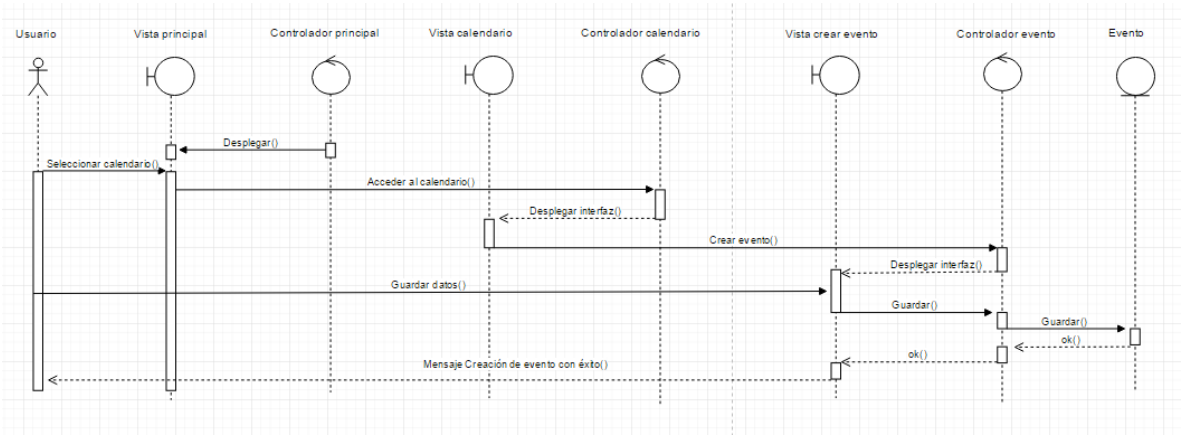

Figura 15. Diagrama de secuencia: Crear evento

.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario selecciona un calendario en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de calendario al Controlador.
4. El Usuario solicita crear un evento en la Vista Calendario.
5. La Vista calendario le envía la solicitud de acceder a la interfaz de crear evento al Controlador.
6. El controlador Crear evento recibe la petición y despliega la interfaz en la vista crear
7. El controlador calendario recibe la petición y despliega la interfaz en la vista calendario.
8. El Usuario quiere guardar los datos del evento.
9. La Vista crear evento envía la petición al Controlador crear y el controlador hace la petición de guardar a la base de datos.
10. La base de datos guarda la información y da una respuesta de OK al Controlador Crear evento y la Vista crear evento muestra un mensaje.
11. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.

Figura 16. Diagrama de secuencia: Editar evento

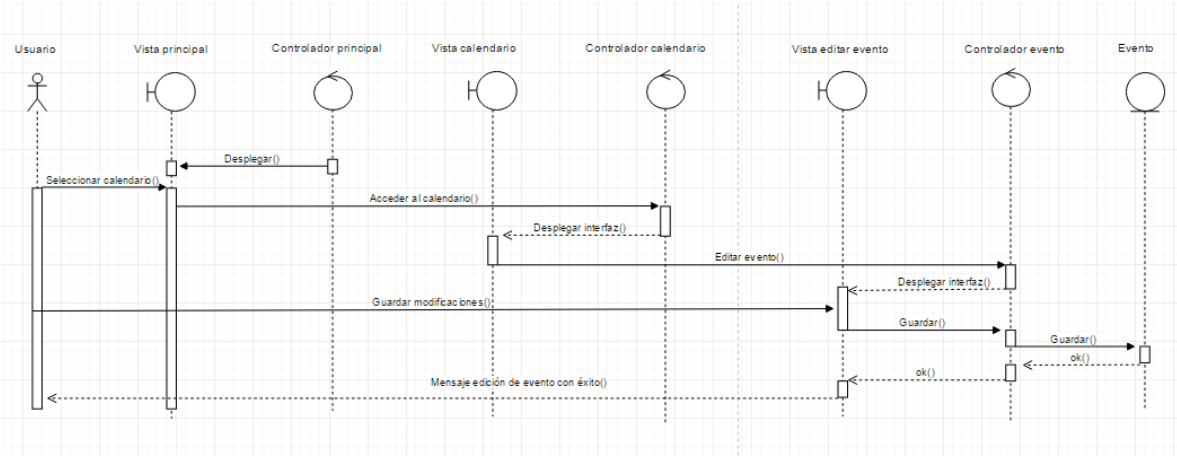

Figura 16. Diagrama de secuencia: Editar evento

.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario selecciona un calendario en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de calendario al Controlador.
4. El Usuario solicita editar un evento en la Vista Calendario.
5. La Vista calendario le envía la solicitud de acceder a la interfaz de editar evento al Controlador.
6. El controlador editar evento recibe la petición y despliega la interfaz en la vista crear
7. El controlador calendario recibe la petición y despliega la interfaz en la vista calendario.
8. El Usuario quiere guardar los datos modificados del evento.
9. La Vista editar evento envía la petición al Controlador editar y el controlador hace la petición de guardar a la base de datos.
10. La base de datos guarda la información y da una respuesta de OK al Controlador editar evento y la Vista editar evento muestra un mensaje.
11. El Usuario ve el mensaje de modificado con éxito y oprime aceptar para terminar.

Figura 17. Diagrama de secuencia: Eliminar evento

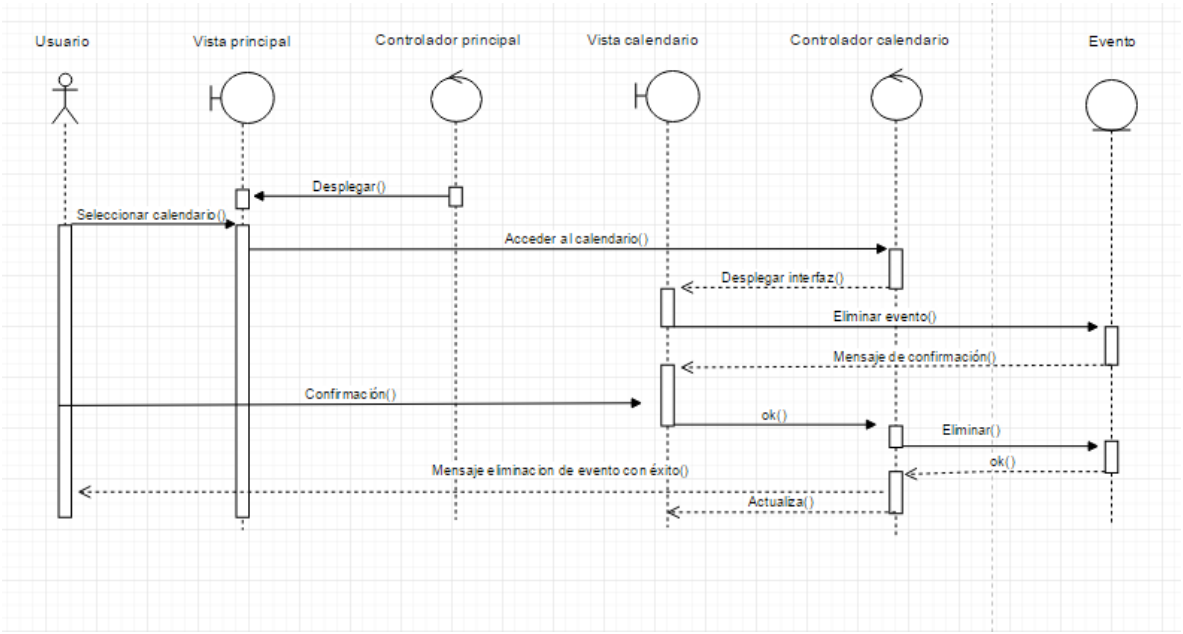

Figura 17. Diagrama de secuencia: Eliminar evento

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario selecciona un calendario en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de calendario al Controlador.
4. El usuario solicita eliminar un evento.
5. El Controlador evento envía un mensaje de confirmación por seguridad.
6. El Usuario confirma que quiere eliminar el evento.
7. La Vista calendario recibe la confirmación y la pasa al Controlador Calendario.
8. El Controlador Calendario solicita a la base de datos eliminar el evento.
9. La base de datos confirma que se eliminó el evento.
10. El controlador confirma que se eliminó y actualiza Vista calendario

Servicio Planes de estudio y programas académicos

Figura 18. Diagrama de secuencia: Crear – Editar – Eliminar, un Programa académico

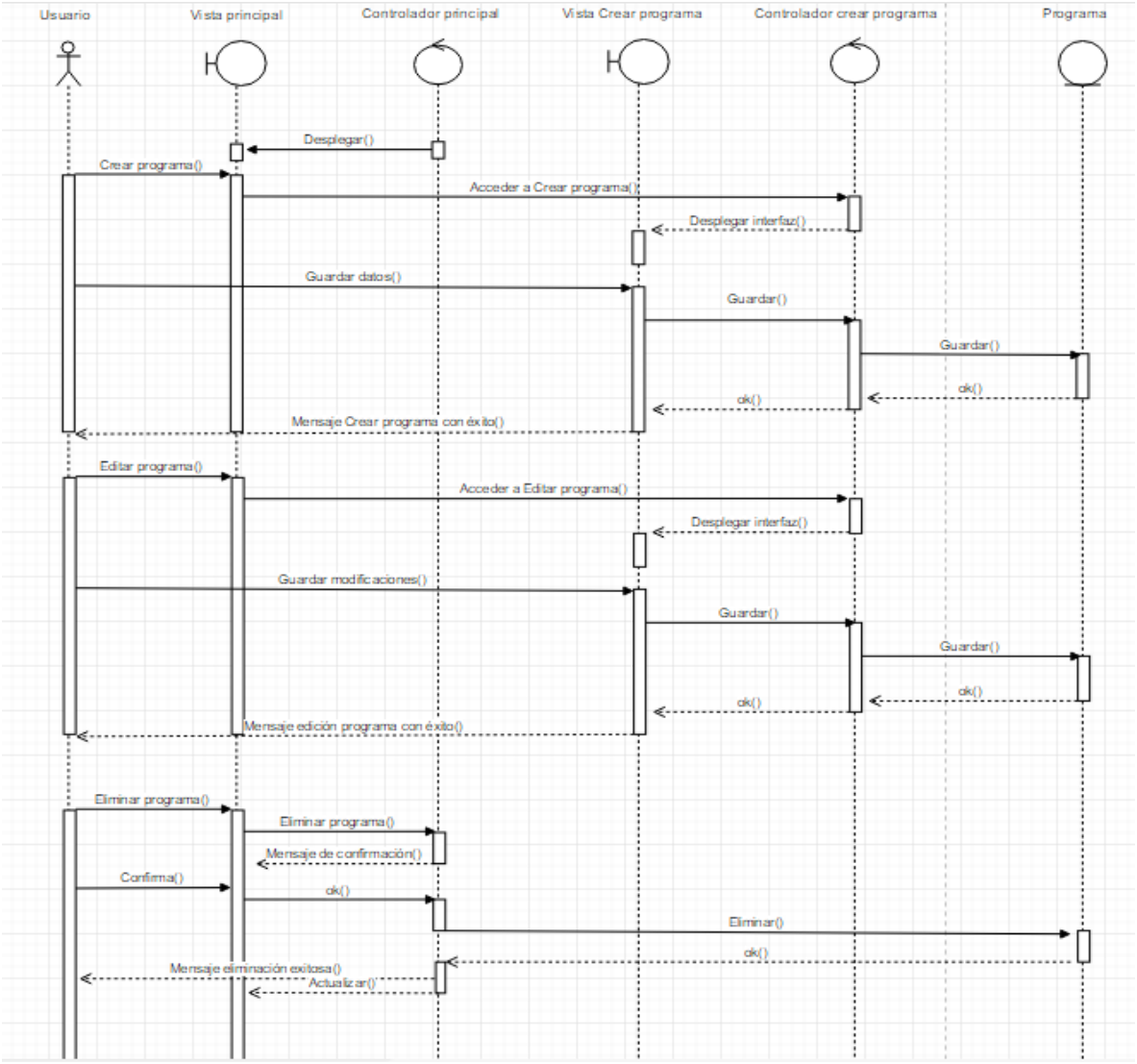

Figura 18. Diagrama de secuencia: Crear – Editar – Eliminar, un Programa académico.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crear un programa académico en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear un programa académico al Controlado.

4. El Controlador recibe la petición y despliega la interfaz en la Vista Crear programa.
5. El Usuario desea guardar los datos del programa académico que ingreso.
6. La Vista crear programa envía la petición al Controlador y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador y la Vista muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.
9. El Usuario solicita editar un programa académico
10. La Vista Principal solicita al Controlador editar poder acceder a editar un programa.
11. El controlador envía la información para visualizar la interfaz de modificar.
12. El Usuario realiza los cambios al programa y hace la petición de guardar las modificaciones.
13. La Vista recibe la solicitud y la envía al Controlador que hace la petición a la base de datos de guardar las modificaciones.
14. La base de datos guarda la información y da una respuesta de OK al Controlador y la Vista muestra un mensaje.
15. El Usuario ve el mensaje de que se guardaron las modificaciones y oprime aceptar para terminar.
16. El Usuario solicita eliminar un programa.
17. La Vista Principal toma la solicitud de eliminar y la pasa al Controlador Principal.
18. El Controlador Principal envía un mensaje de confirmación por seguridad.
19. El Usuario confirma que quiere eliminar el programa.
20. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
21. El Controlador Principal solicita a la base de datos eliminar el programa
22. La base de datos confirma que se eliminó el programa.
23. El controlador confirma que se eliminó y genera la vista actualiza en la Vista Principal.

Figura 19. Diagrama de secuencia: Crear-Editar-Eliminar Plan de Estudios

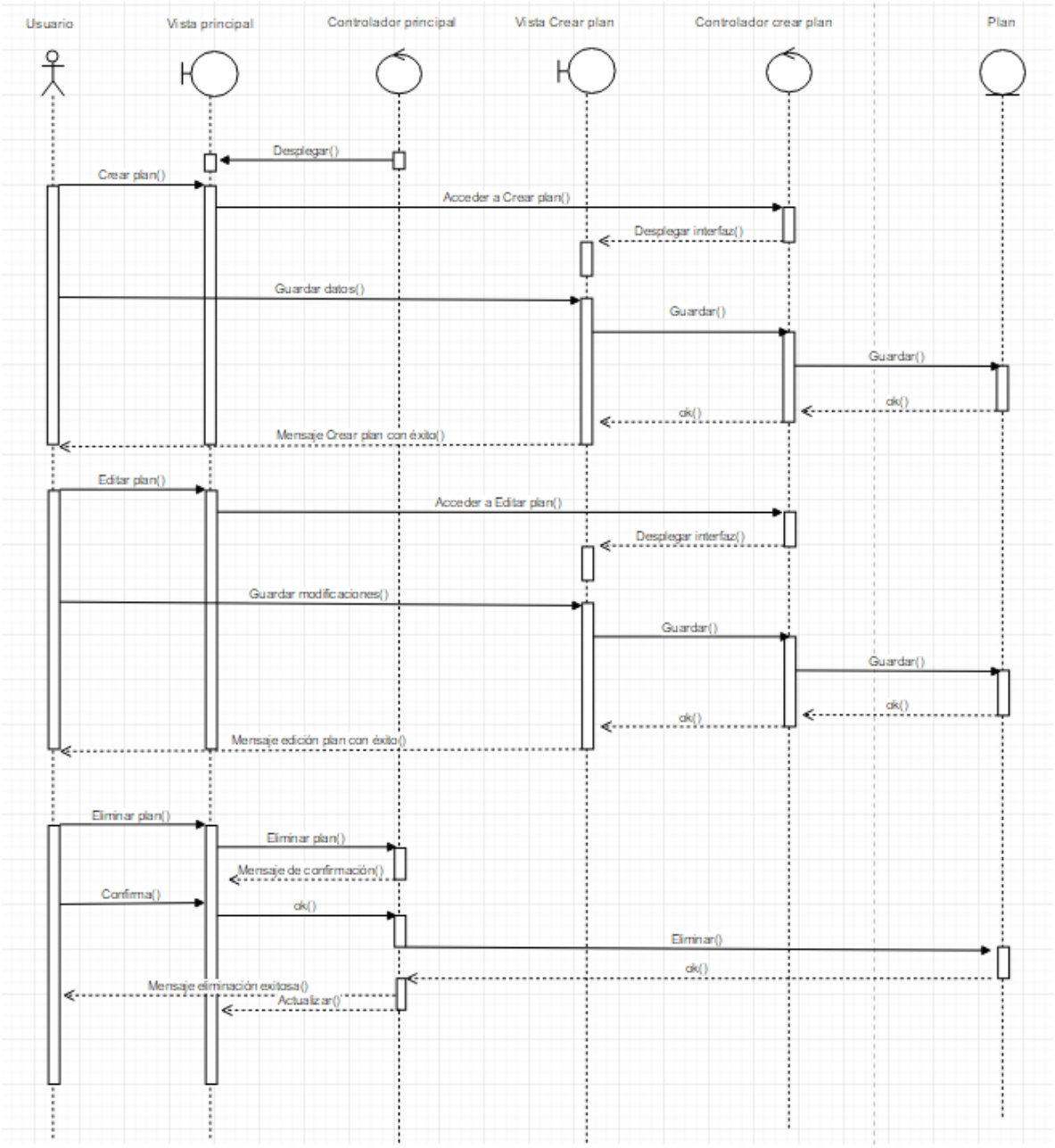

Figura 19. Diagrama de secuencia: Crear-Editar-Eliminar Plan de Estudios.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crear un plan de estudios en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear un plan de estudios al Controlador.

4. El Controlador recibe la petición y despliega la interfaz en la Vista Crear plan de estudios.
5. El Usuario desea guardar los datos plan de estudios que ingreso.
6. La Vista plan de estudios envía la petición al Controlador y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador y la Vista muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.
9. El Usuario solicita editar un plan de estudios
10. La Vista Principal solicita al Controlador editar poder acceder a editar un plan de estudios.
11. El controlador envía la información para visualizar la interfaz de modificar.
12. El Usuario realiza los cambios al plan de estudios y hace la petición de guardar las modificaciones.
13. La Vista recibe la solicitud y la envía al Controlador que hace la petición a la base de datos de guardar las modificaciones.
14. La base de datos guarda la información y da una respuesta de OK al Controlador y la Vista muestra un mensaje.
15. El Usuario ve el mensaje de que se guardaron las modificaciones y oprime aceptar para terminar.
16. El Usuario solicita eliminar un plan de estudios.
17. La Vista Principal toma la solicitud de eliminar y la pasa al Controlador Principal.
18. El Controlador Principal envía un mensaje de confirmación por seguridad.
19. El Usuario confirma que quiere eliminar el plan de estudios.
20. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
21. El Controlador Principal solicita a la base de datos eliminar el plan de estudios
22. La base de datos confirma que se eliminó el plan de estudios.
23. El controlador confirma que se eliminó y genera la vista actualiza en la Vista Principal.

Figura 20. Diagrama de secuencia: Crear-Editar-Eliminar Asignatura

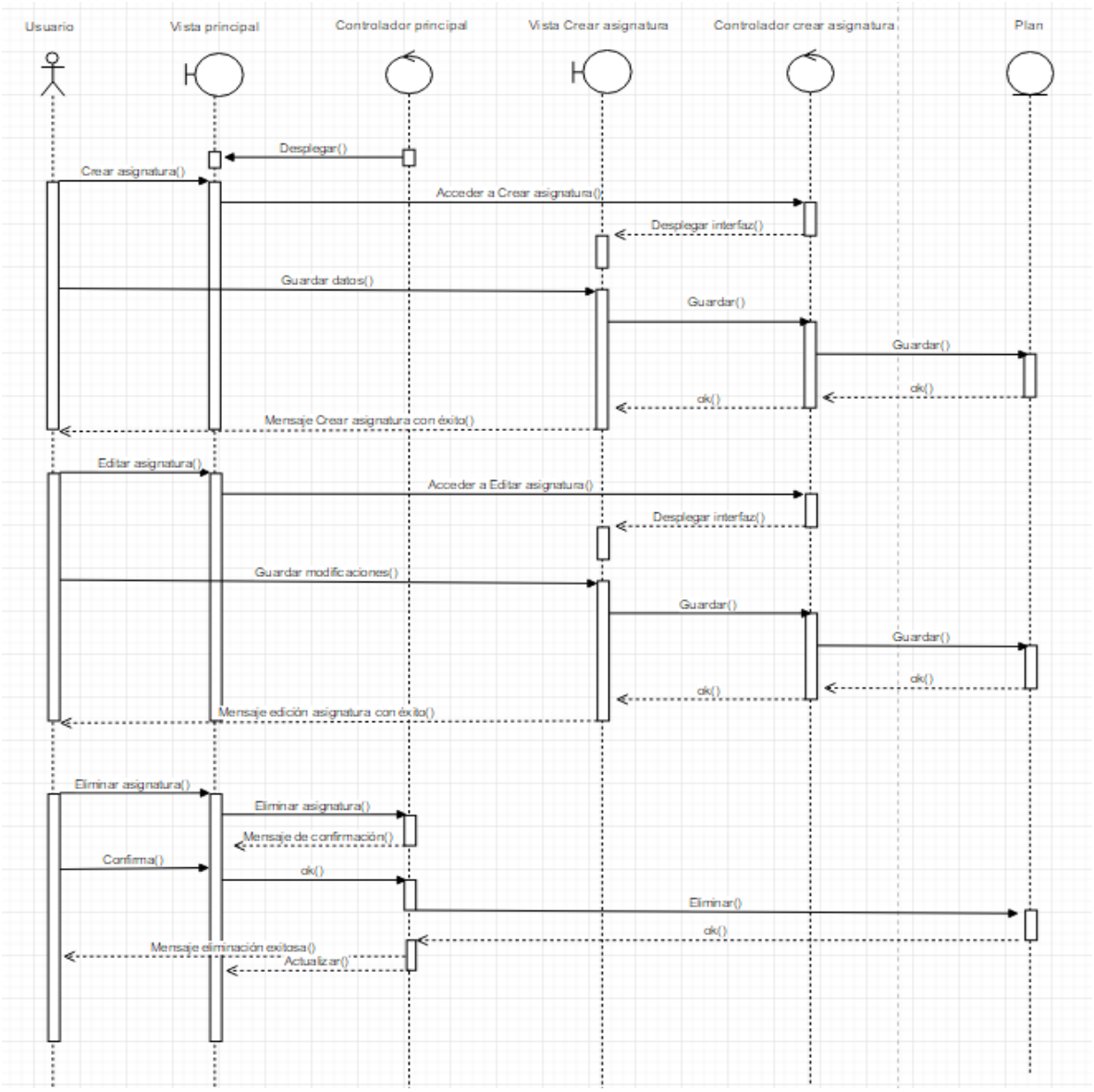

Figura 20. Diagrama de secuencia: Crear-Editar-Eliminar asignatura.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crear una asignatura en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear una asignatura al Controlador.

4. El Controlador recibe la petición y despliega la interfaz en la Vista Crear asignatura.
5. El Usuario desea guardar los datos asignatura que ingreso.
6. La Vista asignatura envía la petición al Controlador y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador y la Vista muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.
9. El Usuario solicita editar una asignatura
10. La Vista Principal solicita al Controlador editar poder acceder a editar una asignatura.
11. El controlador envía la información para visualizar la interfaz de modificar.
12. El Usuario realiza los cambios a la asignatura y hace la petición de guardar las modificaciones.
13. La Vista recibe la solicitud y la envía al Controlador que hace la petición a la base de datos de guardar las modificaciones.
14. La base de datos guarda la información y da una respuesta de OK al Controlador y la Vista muestra un mensaje.
15. El Usuario ve el mensaje de que se guardaron las modificaciones y oprime aceptar para terminar.
16. El Usuario solicita eliminar una asignatura.
17. La Vista Principal toma la solicitud de eliminar y la pasa al Controlador Principal.
18. El Controlador Principal envía un mensaje de confirmación por seguridad.
19. El Usuario confirma que quiere eliminar la asignatura.
20. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
21. El Controlador Principal solicita a la base de datos eliminar la asignatura
22. La base de datos confirma que se eliminó la asignatura.
23. El controlador confirma que se eliminó y genera la vista actualiza en la Vista Principal.

Servicio Correos

Figura 21. Diagrama de secuencia: Enviar Correo

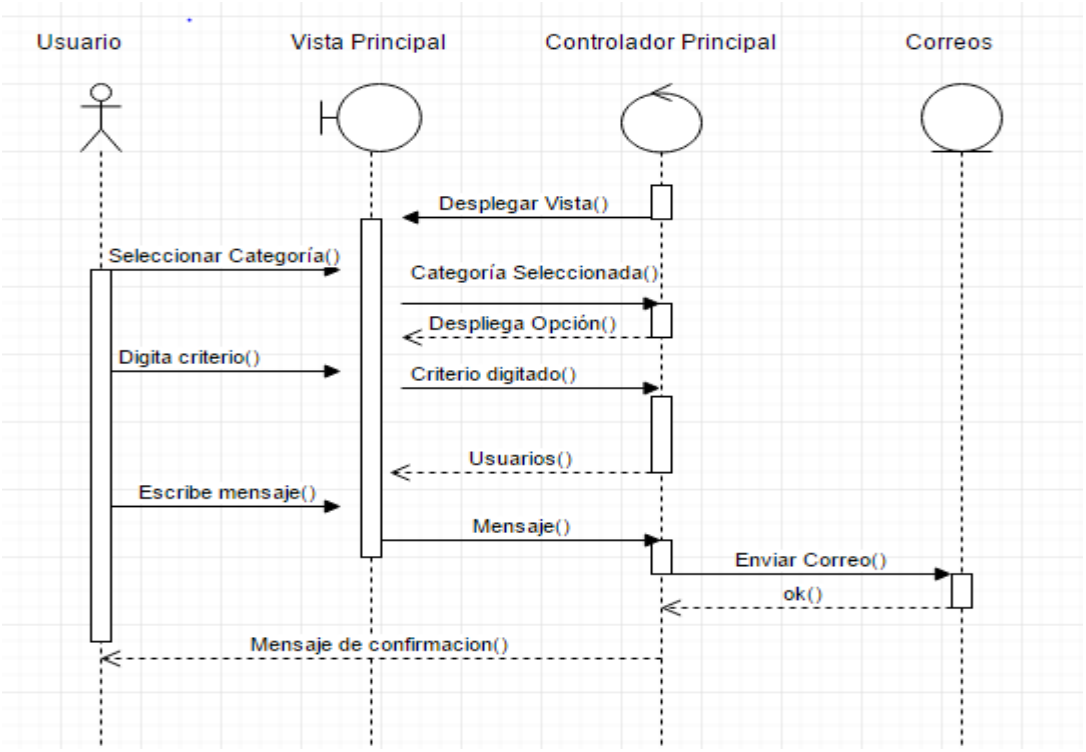

Figura 21. Diagrama de secuencia: Enviar Correo.

1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario selecciona una categoría de usuario.
3. La Vista Principal le envía la solicitud de selección a al Controlador.
4. El controlador despliega la opción de digitar criterio de búsqueda.}
5. El usuario digita el criterio de búsqueda.
6. La vista principal envía al controlador el criterio.
7. El controlador retorno los usuarios que coinciden con el criterio digitado.
8. El usuario escribe el mensaje a enviar
9. El controlador hace la petición de enviar correo a la base de datos
10. La base de datos guarda la información y responde con OK
11. El controlador responde con un mensaje de confirmación de envió.
12. El usuario ve el mensaje y oprime aceptar para terminar