

**MANTENIMIENTO, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DE SERVICIOS NUEVOS PARA REALIZAR ENCUESTAS A LA COMUNIDAD
DE LA ESCUELA Y LA EVALUACIÓN DOCENTE EN LAS ASIGNATURAS DE
POSGRADOS**

CINDY JULIETH GÓMEZ CHACÓN

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

2019

**MANTENIMIENTO, ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DE SERVICIOS NUEVOS PARA REALIZAR ENCUESTAS A LA COMUNIDAD
DE LA ESCUELA Y LA EVALUACIÓN DOCENTE EN LAS ASIGNATURAS DE
POSGRADOS**

CINDY JULIETH GÓMEZ CHACÓN

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

LUIS IGNACIO GONZÁLEZ RAMÍREZ

Magíster en Informática

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

2019

AGRADECIMIENTOS

En primer lugar le doy infinitas gracias a Dios, que siempre me ha dado las fuerzas necesarias para salir adelante en los momentos más difíciles.

Gracias mamá y papá por ser mi más grande apoyo, porque aunque lejos físicamente siempre están conmigo dándome ánimo y fuerzas para continuar. Esto es de ustedes.

A mi hijo Samuel porque es mi fuerza, mi motor, mi razón de ser.

A mis hermanos que nunca dejaron que me rindiera en el camino.

A mi tía Stella y mi nona Carmen, por acompañarme siempre en cada paso de mi vida y en este proceso, por todo su amor y sus valiosos consejos.

A mi gran amiga Paty, gracias por la paciencia, por estar ahí siempre que la necesité, por ser esa voz de aliento en todo momento, por todo lo que me enseñó y me ayudó todo este tiempo, pero sobre todo por nuestra amistad.

A mi maravillosa familia ENLACE, tías, tíos, primas, primos, a los que están lejos, a todos los que se nos adelantaron en el camino, a todos y cada uno ¡GRACIAS!, por la fortuna de hacer parte de la mejor familia, por su ejemplo y cada una de sus enseñanzas, por sus oraciones y por estar presentes en los buenos y malos momentos.

Al profesor Luis Ignacio, por permitirme hacer parte del grupo CALUMET, por motivarme a continuar, por la paciencia, por el tiempo dedicado y por enseñarnos a trabajar en comunidad.

A todos los amigos y personas que de una u otra manera fueron partícipes de este arduo camino, dejando grabados los mejores recuerdos.

Cindy Julieth Gómez Chacón

CONTENIDO

	Pág.
INTRODUCCIÓN	14
1. PLANTEAMIENTO DEL PROYECTO.....	15
1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL	15
1.2 JUSTIFICACIÓN	15
1.3 OBJETIVOS.....	17
1.3.1 Objetivo General.....	17
1.3.2 Objetivos Específicos.....	17
1.4 IMPACTO Y VIABILIDAD.....	18
1.4.1 Impacto.....	18
1.4.2 Viabilidad.....	18
2. MARCO TEÓRICO	20
2.1 ARQUITECTURA CLIENTE SERVIDOR	20
2.1.1 Características de la arquitectura Cliente/Servidor.....	21
2.1.2 Clasificación de las arquitecturas Cliente/Servidor.....	21
2.1.2.1 Arquitectura Cliente/Servidor de dos capas.....	22
2.1.2.2 Arquitectura Cliente/Servidor de tres capas.....	22
2.1.3 Arquitectura Cliente/Servidor aplicada.....	22
2.1.4 Ventajas del esquema Cliente/Servidor	23
2.1.5 Desventajas del esquema Cliente/Servidor	23
2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS.....	24
2.2.1 Código del Lado del Cliente (Client Side Scripts).....	24
2.2.2 Código del Lado del Servidor (Server Side Scripts).....	25
2.2.3 Tecnología aplicada.....	25
2.2.3.1 Modelo de acceso a JSP	26
2.3 BASES DE DATOS.....	27

2.3.1 Modelos de Bases de Datos.	28
2.3.1.1 Base de Datos Jerárquica.	28
2.3.1.2 Base de Datos de Red.	28
2.3.1.3 Base de Datos Relacional.	28
2.3.2 Manejadores o Gestores de Bases de Datos.	29
2.3.3 MySQL.	30
2.3.4 Ventajas de MySQL.	30
2.4 NETBEANS.	30
2.5 SISTEMA DE CONTROL DE VERSIONES.	31
2.5.1 Subversion.	32
2.6 PROGRAMACIÓN UTILIZADA.	32
2.6.1 Clases.	33
2.6.2 Objetos.	33
2.6.3 Atributos.	33
2.6.4 Métodos.	33
2.6.5 Herencia.	33
2.6.6 Beneficios de la Programación Orientada a Objetos.	34
2.6.7 Java y JDK (Java Development Kit).	34
2.7 SERVIDORES WEB.	35
2.7.1 Servidor Jakarta Tomcat.	35
3. MARCO METODOLÓGICO.	36
3.1 PROTOTIPO EVOLUTIVO.	36
3.2 LENGUAJE DE MODELADO UNIFICADO.	38
3.2.1 Diagramas de UML.	38
3.2.2 Diagramas de casos de uso.	39
3.2.3 Diagramas de secuencias.	40
3.3 ESTÁNDARES DE PROGRAMACIÓN.	41
3.3.1 Modelo de datos.	41
3.3.2 Clases.	42
3.3.3 Páginas JSP.	42

3.3.4 Organización de Directorios.....	43
4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO.....	44
4.1 PROTOTIPO ESPERADO.....	44
4.1.1 Diagramas de casos de uso.....	48
4.1.1.1 Servicio de Encuestas Escuela.....	48
4.1.1.2 Servicio de Evaluación Docente.....	49
4.1.2 Documentación de Casos de Uso del Sistema.....	49
4.1.2.1 Servicio de Encuestas Escuela.....	49
4.1.2.2 Servicio de Evaluación Docente.....	50
4.1.3 Diseño y Análisis.....	51
4.1.4 Modelo de Procesos del Sistema.....	54
4.1.5 Implementación, Implantación y Pruebas Generales.....	63
5. PRUEBAS DEL SISTEMA.....	65
5.1 PRUEBAS DE VERIFICACIÓN.....	65
5.1.1 Pruebas por componente.....	65
5.1.1.1 Servicio de Encuestas.....	66
5.1.1.2 Servicio de Evaluación Docente.....	66
6. CONCLUSIONES.....	68
7. RECOMENDACIONES.....	69
BIBLIOGRAFÍA.....	70

LISTA DE FIGURAS

	Pág.
Figura 1. Modelo Cliente / Servidor.....	20
Figura 2. Modelo de acceso a JSP	27
Figura 3. Prototipo Evolutivo	36
Figura 4. Diagramas de Casos de Uso	39
Figura 5. Diagrama de Secuencias.....	41
Figura 6. Diagrama de casos de uso: Encuestas Escuela.....	48
Figura 7. Diagrama de casos de uso: Evaluación Docente	49
Figura 8. Diagrama E / R: Encuestas Escuela.....	51
Figura 9. Diagrama E / R: Evaluación Docente.....	52
Figura 10. Diagrama de secuencia: Gestionar Encuestas – Crear Encuesta	54
Figura 11. Diagrama de secuencia: Gestionar Encuestas – Editar Encuesta.....	55
Figura 12. Diagrama de secuencia: Gestionar Encuestas – Ver Resultados Encuesta.....	56
Figura 13. Diagrama de secuencia: Gestionar Encuestas – Duplicar Encuesta	56
Figura 14. Diagrama de secuencia: Gestionar Encuestas – Responder Encuesta	58
Figura 15. Diagrama de secuencia: Gestionar Evaluación Docente – Crear Evaluación Docente	59
Figura 16. Diagrama de secuencia: Gestionar Evaluación Docente – Editar Evaluación Docente	60
Figura 17. Diagrama de secuencia: Gestionar Evaluación Docente – Ver Resultados Evaluación Docente	61
Figura 18. Diagrama de secuencia: Gestionar Evaluación Docente – Duplicar Evaluación Docente	62
Figura 19. Diagrama de secuencia: Gestionar Evaluación Docente – Responder Evaluación Docente	63

LISTA DE TABLAS

	Pág.
Tabla 1. Casos de uso: Encuesta Escuela	49
Tabla 2. Casos de Uso: Evaluación Docente.....	50
Tabla 3. Descripción de las Entidades.....	52
Tabla 4. Pruebas realizadas: Encuestas.....	66
Tabla 5. Pruebas realizadas: Evaluación Docente.....	66

RESUMEN

TÍTULO: Mantenimiento, análisis, diseño, desarrollo e implementación de servicios nuevos para realizar encuestas a la comunidad de la escuela y la evaluación docente en las asignaturas de posgrados*.

AUTOR: Cindy Julieth Gómez Chacón**

PALABRAS CLAVE: Portal Web, Encuesta, Evaluación Docente, Escuela de Ingeniería de Petróleos.

DESCRIPCIÓN

Actualmente los portales web de las escuelas de la Universidad Industrial de Santander cuentan con los servicios de comunicación, los cuales permiten la interacción entre los miembros de la comunidad estudiantil como docentes, estudiantes, administrativos, entre otros.

Entre los servicios de comunicación, se cuenta con el servicio de Encuestas UIS, en donde los usuarios pueden publicar noticias a su portal grupo, a su escuela y otras escuelas. Este mismo servicio estará dirigido a los miembros de la escuela, lo cual permitirá la recopilación de información necesaria para la realización de diversas actividades en la escuela.

En la escuela de Ingeniería de Petróleos se realizan evaluaciones a los docentes, por parte de los estudiantes, al finalizar cada ciclo de estudio de los programas de posgrados. Estas evaluaciones son realizadas en papel y los resultados son tratados en documentos de Excel. Para facilitar la realización de dichas evaluaciones y el tratamiento de sus resultados, se necesita un servicio de Evaluación Docente que permita crear, editar, eliminar y organizar las evaluaciones, crear diferentes tipos de preguntas y mostrar los resultados de una forma dinámica y de fácil interpretación. De la misma forma es necesario crear un servicio para que los estudiantes puedan ver y responder las evaluaciones correspondientes a las asignaturas cursadas durante el ciclo.

* Trabajo de grado Modalidad Trabajo de Investigación

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

ABSTRACT

TITLE: Maintenance, analysis, design, development and implementation of new services to do surveys for school community and teacher evaluation in postgraduate subjects*.

AUTHOR: Cindy Julieth Gómez Chacón**

KEY WORDS: Web Portal, Survey, Teacher Evaluation, Petroleum Engineering School.

DESCRIPTION

Currently the web portals of the schools of the Universidad Industrial de Santander have communication services, which allow interaction among the members of the student community as teachers, students, administrators and others.

Among the communication services, there is the UIS Surveys service, where users can post news to their web portal, their school and other schools. This same service will be addressed to the members of the school, which will allow the collection of information necessary to carry out different activities in the school.

At the Petroleum Engineering School, evaluations are made to the teachers by the students, at the end of each study cycle of the graduate programs. These evaluations are done on paper and the results are treated in Excel documents. To facilitate the performance of such evaluations and the treatment of their results, a Teaching Evaluation service is needed to create, edit, eliminate and organize the evaluations, create different types of questions and show the results in a dynamic and easy to interpret way. In the same way it is necessary to create a service which students can see and respond to the evaluations corresponding to the subjects studied during the cycle.

* Trabajo de grado Modalidad Trabajo de Investigación

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

INTRODUCCIÓN

Los portales web Comunidad Académica son los primordiales medios de comunicación e interacción entre los miembros de las diferentes escuelas y en general de la comunidad UIS. Actualmente muchas escuelas de la comunidad universitaria se han vinculado a los servicios ofrecidos por estos portales web, y debido a su gran aceptación es importante mantenerlos y mejorarlos, además de crear nuevos servicios que satisfagan las necesidades de todos los usuarios.

El grupo de desarrollo de software CALUMET de la escuela de Ingeniería de Sistemas e Informática, es el encargado de mantener, actualizar y crear continuamente nuevos servicios que cumplan con la demanda de los usuarios de la comunidad. Para cumplir con estos objetivos, se cuentan con herramientas software de libre distribución como NetBeans, Github, Sqlyog, entre otras.

Este documento presenta un soporte teórico, metodológico y técnico del desarrollo del módulo de encuestas escuela que permita mayor interacción y comunicación entre los miembros de cada escuela; el módulo de evaluación docente que contribuya a la mejoría de la aplicación de dicha evaluación y el tratamiento e interpretación de los resultados para la toma de decisiones.

1. PLANTEAMIENTO DEL PROYECTO

1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL

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

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

1.2 JUSTIFICACIÓN

Los Directivos de las escuelas, y en general toda la Comunidad, requieren de herramientas que les permitan de una manera más acertada, tomar decisiones de tipo administrativo y/o académico. Un sistema de encuestas dirigido a los miembros de la escuela, podría de manera sencilla recopilar información para una mejor toma de decisiones. Esta herramienta, no solo aportará lo anterior, sino además permitirá a la Comunidad conocerse mejor.

Este sistema permitirá crear, modificar, eliminar y organizar encuestas con preguntas tipo única y múltiple respuesta, o preguntas abiertas según se requiera. Se debe contar con interface de resultados con diferentes opciones tanto para quien creó, como respondió la encuesta.

Actualmente, en la escuela de Ingeniería de Petróleos, en el área de posgrados, se realiza una evaluación docente al finalizar cada asignatura para obtener realimentación, para el análisis del docente y del comité asesor de posgrados. Dichas evaluaciones se llevan a cabo por medio de preguntas, que son realizadas en papel y digitadas manualmente en plantillas de Excel para posteriormente emitir los resultados de éstas. El proceso es bastante complicado, tedioso y demorado lo que conlleva a dificultades para tomar decisiones a tiempo.

Se hace necesaria la creación de la evaluación docente de posgrados en el portal web de la escuela de Ingeniería de Petróleos, con el fin de agilizar dicho proceso, evitando que las evaluaciones realizadas se tengan que tabular a mano para obtener los resultados. El módulo de evaluación docente de los posgrados permitirá, de manera muy sencilla, al coordinador de programa crear, modificar, eliminar y organizar encuestas para los estudiantes de posgrados con preguntas tipo única y múltiple respuesta o preguntas abiertas. Así mismo, el coordinador tendrá la posibilidad de notificar a los estudiantes de posgrados, cuando estos deban contestar alguna evaluación.

En consecuencia, se debe crear un servicio donde los estudiantes puedan ver y diligenciar las preguntas creadas por el coordinador de posgrados, el cual solo estará disponible para aquellos usuarios asignados por el coordinador.

La Escuela de Ingeniería de Petróleos requiere no solamente poder generar evaluaciones docentes de un modo más práctico, también se hace necesario poder visualizar de una manera más completa los resultados obtenidos de dichas

evaluaciones, para poder tomar las medidas necesarias con respecto a los docentes o materias.

Se hace necesaria la creación de un servicio de resultados para el docente, donde encontrará los resultados obtenidos de las últimas evaluaciones, un histórico, o dichos resultados agrupados por ciclo o materia.

Los resultados de las evaluaciones docentes son de vital interés para el comité asesor de posgrados, por lo que es necesario crear un servicio para el comité, donde podrán visualizarlos agrupados por ciclo, materia o históricos por cada docente.

1.3 OBJETIVOS

1.3.1 Objetivo General. Realizar las funciones de análisis, mantenimiento, diseño, desarrollo e implementación de los servicios para hacer eficientes y sostenibles los trámites dentro de las escuelas y el acceso a la información.

1.3.2 Objetivos Específicos. Análisis, diseño, desarrollo e implementación de nuevos servicios.

- Crear el servicio de encuestas dirigidas a la comunidad de la escuela, donde se podrán crear encuestas y obtener los resultados de éstas por medio de estadísticas y gráficos, que faciliten su interpretación. Se presentará una vista de los resultados en diferentes pestañas, de acuerdo a los permisos de los usuarios.
- Agregar un servicio especial para crear la evaluación docente de posgrados, el cual estará habilitado para los coordinadores, de manera que puedan crear,

modificar o eliminar preguntas, generarla y enviarla a los estudiantes de posgrados.

- Implementar el servicio para responder la evaluación docente, el cual estará habilitado únicamente para los usuarios interesados, y donde podrán ver y diligenciar la encuesta creada por el coordinador de posgrados.
- Crear el servicio de resultados por docente, donde cada docente encontrará sus resultados de las últimas evaluaciones docentes realizadas, un histórico de resultados de todas las evaluaciones, o dichos resultados agrupados por ciclo o materia.
- Agregar el servicio de resultados para el comité asesor de posgrados, donde podrán visualizar dichos resultados agrupados por ciclo, o históricos por cada docente.
- 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 envío.

1.4 IMPACTO Y VIABILIDAD

1.4.1 Impacto. Los portales web de las escuelas y facultades han sido una herramienta útil para el manejo de la información. Por esta razón es indispensable realizar trabajos de administración y mantenimiento, para brindar al usuario la confiabilidad en los servicios que requieran.

Se pretende dar respuesta a dichos requerimientos, con el fin de que los procesos llevados a cabo en las escuelas y facultades de la universidad, cada día sean más eficientes, seguros y confiables, y se ajusten a las necesidades de los usuarios.

1.4.2 Viabilidad. El presente proyecto es viable, pues hay disponibilidad de recursos tecnológicos y personal calificado para llevarlo a cabo, los cuales permiten

el desarrollo de nuevos servicios, el soporte a los usuarios, la administración y el mantenimiento de los mismos.

2. MARCO TEÓRICO

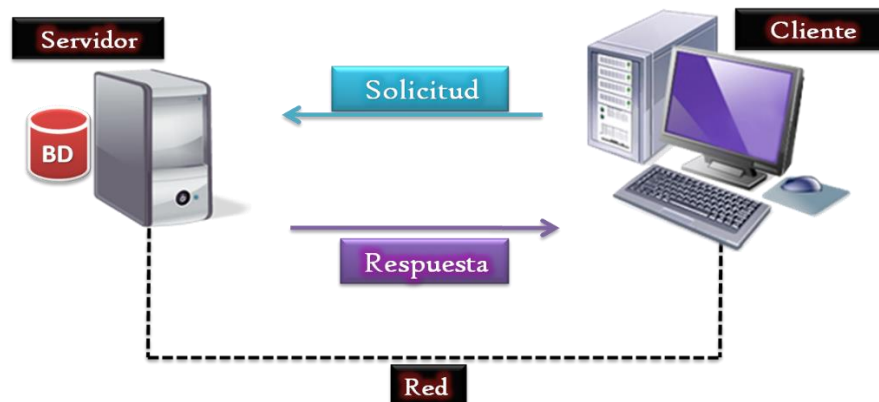
2.1 ARQUITECTURA CLIENTE SERVIDOR

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

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

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

Figura 1. Modelo Cliente / Servidor



Fuente: CHARLIEDAW2236 Arquitectura cliente servidor
<http://charliedaw2236.blogspot.com.co/p/arquitectura-cliente-servidor.html>

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

- El proceso del cliente da la interface entre usuarios y el resto del sistema, 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:

- 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 páginas JSP y servlets se ejecutan en la Máquina Virtual de Java, lo cual permite que se puedan usar en cualquier tipo de computador, siempre y cuando este instalada la Máquina Virtual de Java. Cada JSP se ejecuta en su propio contexto (llamado también hilo o hebra); pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma

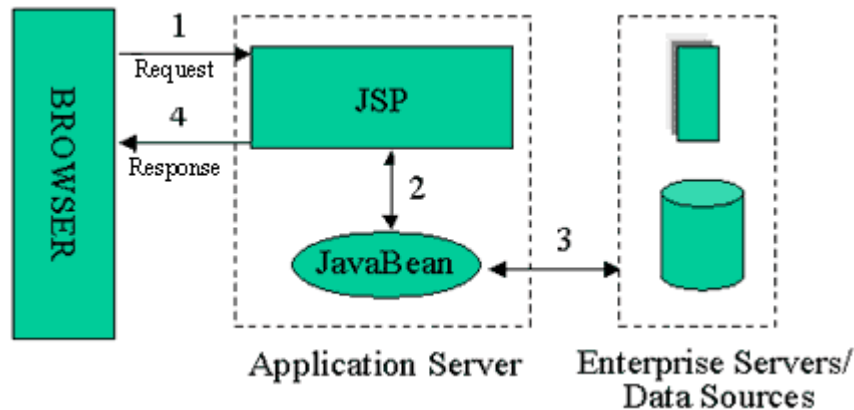
que no se pierde tiempo en invocarlo. Su persistencia permite hacer cosas de forma más eficiente como la conexión a bases de datos y manejo de sesiones.

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

2.2.3.1 Modelo de acceso a JSP

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

Figura 2. Modelo de acceso a JSP



Fuente: JAVAWORLD Modelo de acceso a JSP [en línea] disponible en: <http://www.javaworld.com/article/2076557/java-web-development/understanding-javascript-pages-model-2-architecture.html>

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 los cuales pueden utilizarse para describir un conjunto de datos y las operaciones para manipularlos.

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. Los sistemas manejadores de base de datos (SGBD), son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

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

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

2.3.4 Ventajas de MySQL

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

2.4 NETBEANS

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

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

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

2.5 SISTEMA DE CONTROL DE VERSIONES

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

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

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

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

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

2.6 PROGRAMACIÓN UTILIZADA

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

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

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

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

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

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

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

2.6.6 Beneficios de la Programación Orientada a Objetos

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

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

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

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

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

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

2.7 SERVIDORES WEB

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

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

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

3. MARCO METODOLÓGICO

3.1 PROTOTIPO EVOLUTIVO

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

Figura 3. Prototipo Evolutivo



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

- Los usuarios de los portales web de la comunidad académica detectan constantemente nuevas necesidades, por lo tanto se requiere el desarrollo de nuevos servicios y así mismo la mejora de los ya existentes. Debido a que el

sistema no es considerado un producto final, es necesaria su continua actualización.

- El desarrollo del aspecto visual del sistema y su usabilidad pueden estar sujetas a cambios por requerimiento de los usuarios de los portales web de la comunidad académica, incluso durante el desarrollo del proyecto.
- Durante la primera etapa de los nuevos servicios es necesario tener una clara interpretación de los requerimientos de los usuarios y escuelas, aunque en repetidas ocasiones los usuarios no tienen una idea clara de lo que necesitan. Por esta razón, los prototipos nos dan la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado.
- Otra ventaja es tener la posibilidad de realizar cambios en etapas tempranas y crear varios prototipos evaluables durante el desarrollo, y así obtener una metodología integral para el proceso de evaluación del programa.
- Esta metodología favorece la realimentación constante y autocrítica del sistema, lo que conlleva a que se produzca muchas pruebas antes de dar un nuevo prototipo, así como mejoras rápidas a problemas que puedan surgir durante su uso.

Procedimiento a seguir para la metodología planteada:

- Se hace un análisis de requerimientos para la construcción de los prototipos.
- Se especifican los objetivos globales para conocer en detalle el software que se va 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.
- Posteriormente se presenta 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.
- El prototipo es evaluado por el usuario y se utiliza para filtrar los requisitos del software a desarrollar.

- Se produce un proceso interactivo en el que el prototipo es depurado para satisfacer necesidades del usuario, de igual forma el desarrollador obtiene una mejor comprensión de lo que hay que hacer para la entrega del producto final de ingeniería requerido por el usuario.

3.2 LENGUAJE DE MODELADO UNIFICADO

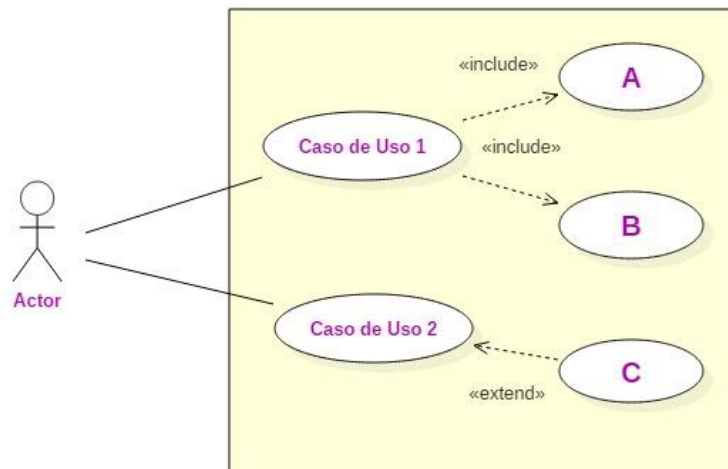
El Lenguaje de Modelado Unificado o Unified Modeling Language (UML), es el más utilizado en la actualidad. Es un lenguaje gráfico estándar para visualizar, especificar, construir y documentar un sistema para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio,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

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

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

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

Figura 4. Diagramas de Casos de Uso



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

En este proyecto se destacan los siguientes actores:

- **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios, tienen un perfil de administrador, con el cual tienen permisos extra a los que tiene un usuario normalmente dentro del sitio. Algunos de estos son los auxiliares de administración del portal, los docentes, las secretarías de las escuelas, entre otros. 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 usuarios.

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

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

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

3.2.3 Diagramas de secuencias. Los diagramas de secuencia muestran la forma en que un grupo de objetos se comunican o interactúan entre sí a lo largo de un tiempo y facilitan comprender la ejecución de un proceso. Constan de objetos que se representan de modo usual: rectángulo con nombre, mensajes entre los objetos representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

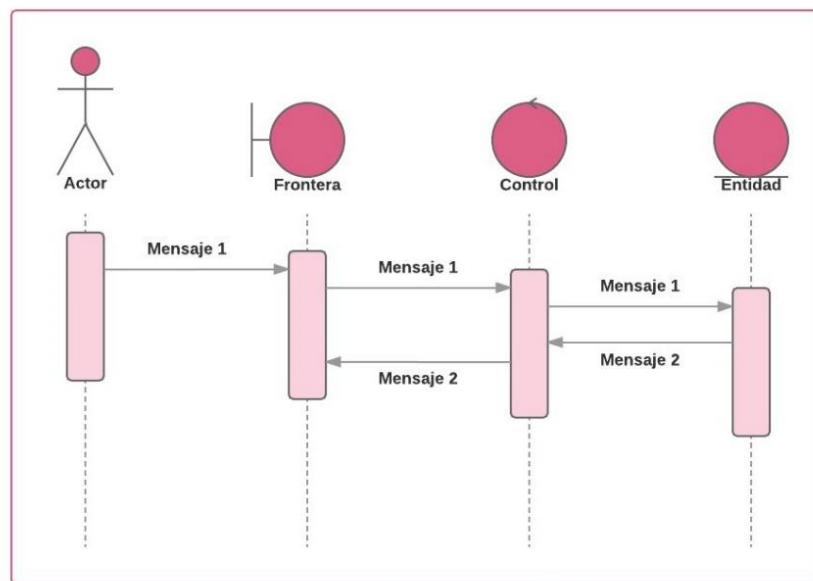
Objetos: Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la *línea de vida* de un objeto. Con la línea de vida se encuentra un pequeño rectángulo conocido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



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).

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

3.3.2 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: Encuesta.java, Votante.java.

3.3.3 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, VerCronograma.jsp, GenerarCronograma.jsp.

3.3.4 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: EncuestasEscuela, EvaluacionDocente.

- El sitio cuenta con un directorio llamado “images”; en éste se encuentran almacenados todos los archivos .jpg, .gif, .png.
- El sitio cuenta con un directorio llamado Script, donde están todos los archivos de JavaScript “.js” necesarios para el buen funcionamiento del portal.

Para el desarrollo de este proyecto se crearon los directorios “EncuestasEscuela” y “EvaluacionDocente”, los cuales almacenan los archivos correspondientes al funcionamiento de dichos servicios.

4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

Como se mencionó anteriormente para el desarrollo de este proyecto se siguió la metodología de prototipo evolutivo.

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

4.1 PROTOTIPO ESPERADO

Al inicio el proyecto no se tenía una concepción clara de cómo sería el producto final, sin embargo durante el desarrollo y evolución de los prototipos, las pruebas y análisis del sistema se pudo comprobar que se estaba acercando a los requerimientos iniciales, esto con el fin de enfocar exitosamente el desarrollo a la solución de las necesidades de los usuarios.

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

Crear el servicio de Encuestas a nivel de escuela, de manera que los miembros de la escuela se les permitan crear, editar, eliminar encuestas y dirigirlas a

otros integrantes de la misma. Se permite ver los resultados y estadísticas con sus gráficos de la encuesta, de acuerdo a los permisos otorgados al crearla, para su respectiva interpretación y análisis.

Objetivo inicial:

Crear el servicio de encuestas dirigidas a la comunidad de la escuela, donde se podrán crear encuestas y obtener los resultados de éstas por medio de estadísticas y gráficos, que faciliten la interpretación de los mismos.

Requisitos finales del objetivo:

- Usuarios con permisos de administrador pueden gestionar encuestas.
- Permite crear encuestas, en las que se asigna un título y descripción, el rango de fechas de participación en la encuesta por parte de los usuarios, se agregan preguntas personalizadas que los usuarios encuestados deberán responder, además es posible dar permisos a los usuarios, de manera que puedan ver resultados y/o listado de votantes o no.
- Los usuarios que crean la encuesta pueden hacerla pública o privada y, en este último caso deberán indicar a cuales categorías de usuario va dirigida, por ejemplo egresados, estudiantes de pregrado, docentes, etc. Puede escoger una o varias categorías.
- Es posible agregar diferentes campos personalizados para realizar las preguntas de la encuesta, por ejemplo campos tipo texto, párrafo, selección múltiple, entre otros.
- Permite modificar el orden de los campos, así como editarlos y eliminarlos.
- Permite editar, duplicar y eliminar encuestas.
- Se tiene una vista de resultados a través de pestañas, las cuales están activas de acuerdo a los permisos del administrador de la encuesta. El administrador tiene acceso a todas las pestañas de resultados.

- Se muestran estadísticas de las respuestas. En el caso de preguntas de selección múltiple se presentan gráficos para facilitar su interpretación.

Agregar un servicio especial para la escuela de Ingeniería de Petróleos, para crear la evaluación docente de posgrados, de modo que un administrador pueda agregar una descripción, un intervalo de fechas de participación de la evaluación, crear, modificar o eliminar preguntas de dicha evaluación y enviarla a los estudiantes de posgrados que correspondan.

Objetivo inicial:

Crear el nuevo servicio de evaluación docente para posgrados de Ingeniería de Petróleos, que permita gestionar las preguntas y dirigir dicha evaluación a los estudiantes de posgrados.

Requisitos finales del objetivo:

- Usuarios con permisos de administrador pueden gestionar evaluaciones.
- Permite crear evaluaciones, en las que se asigna un título y descripción, el rango de fechas de participación en la evaluación por parte de los estudiantes de posgrados. Se agregan preguntas personalizadas que los usuarios encuestados deberán responder.
- El administrador escoge en una tabla de cronogramas vigentes, los docentes a los cuales se evaluarán, según la asignatura y el grupo correspondientes.
- Es posible agregar diferentes campos personalizados para realizar las preguntas de la evaluación, por ejemplo campos tipo texto, párrafo, selección múltiple, entre otros.
- Permite modificar el orden de los campos, así como editarlos y eliminarlos.
- Permite editar, duplicar y eliminar evaluaciones.

Implementar servicio que permita a los estudiantes de posgrados de la escuela de Ingeniería de Petróleos, ver y diligenciar las evaluaciones docente que le correspondan según las asignaturas que haya cursado.

Objetivo inicial:

Crear servicio que permita al estudiante responder las evaluaciones docente que tenga activas, según le corresponda.

Requisitos finales del objetivo:

- El administrador de la evaluación la habilitará para los estudiantes de posgrado, según el docente y el grupo que les corresponda.
- Los estudiantes de posgrados podrán ver las evaluaciones que tengan vigentes por responder y proceder a diligenciarlas.

Agregar el servicio de resultados, el cual estará disponible tanto para docentes como para el comité asesor de posgrados. Dichos resultados serán mostrados por materia – grupo de cada docente, dando un promedio final por cada ítem evaluado y en general por cada asignatura – grupo.

Objetivo inicial:

Agregar el servicio de resultados para el comité asesor de posgrados y para los docentes a evaluar.

Requisitos finales del objetivo:

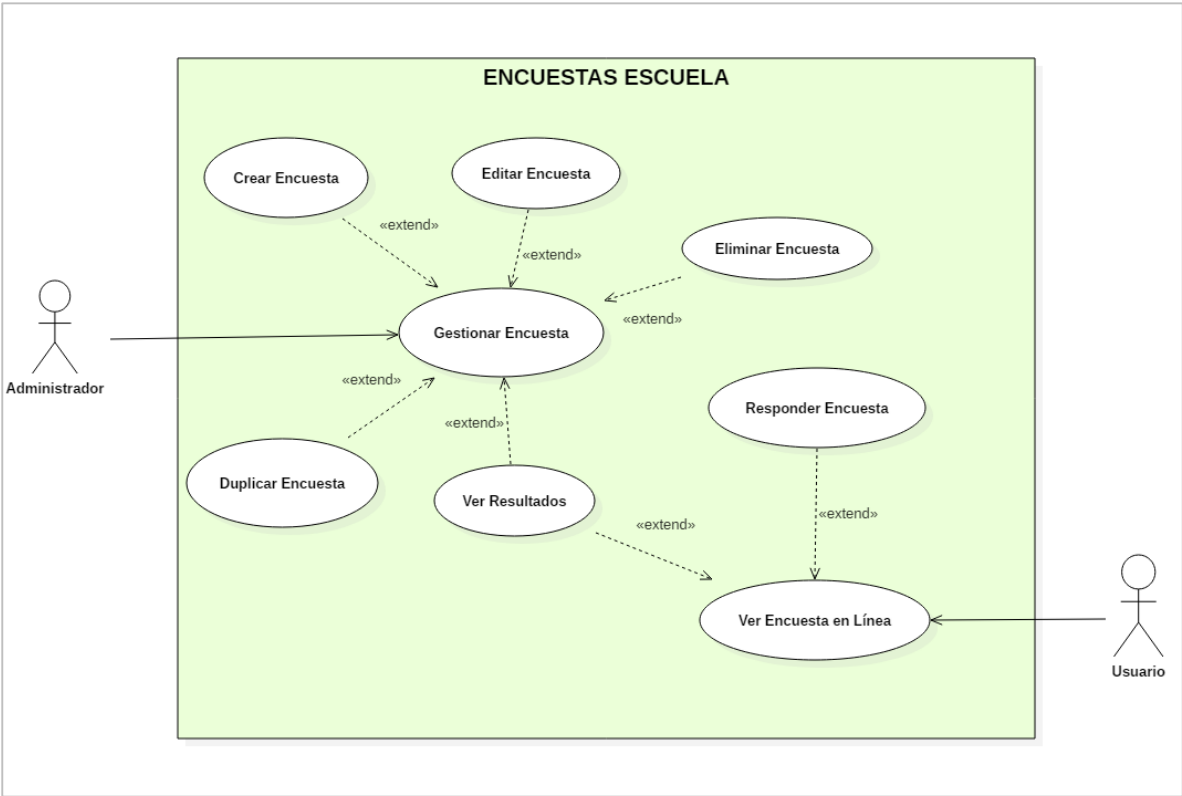
- Solo los administradores, los miembros del comité asesor de posgrados y los docentes evaluados pueden ver los resultados. El comité podrá ver los

resultados de todos los docentes, mientras que estos solo podrán ver sus resultados de manera individual, según sus asignaturas a cargo.

4.1.1 Diagramas de casos de uso

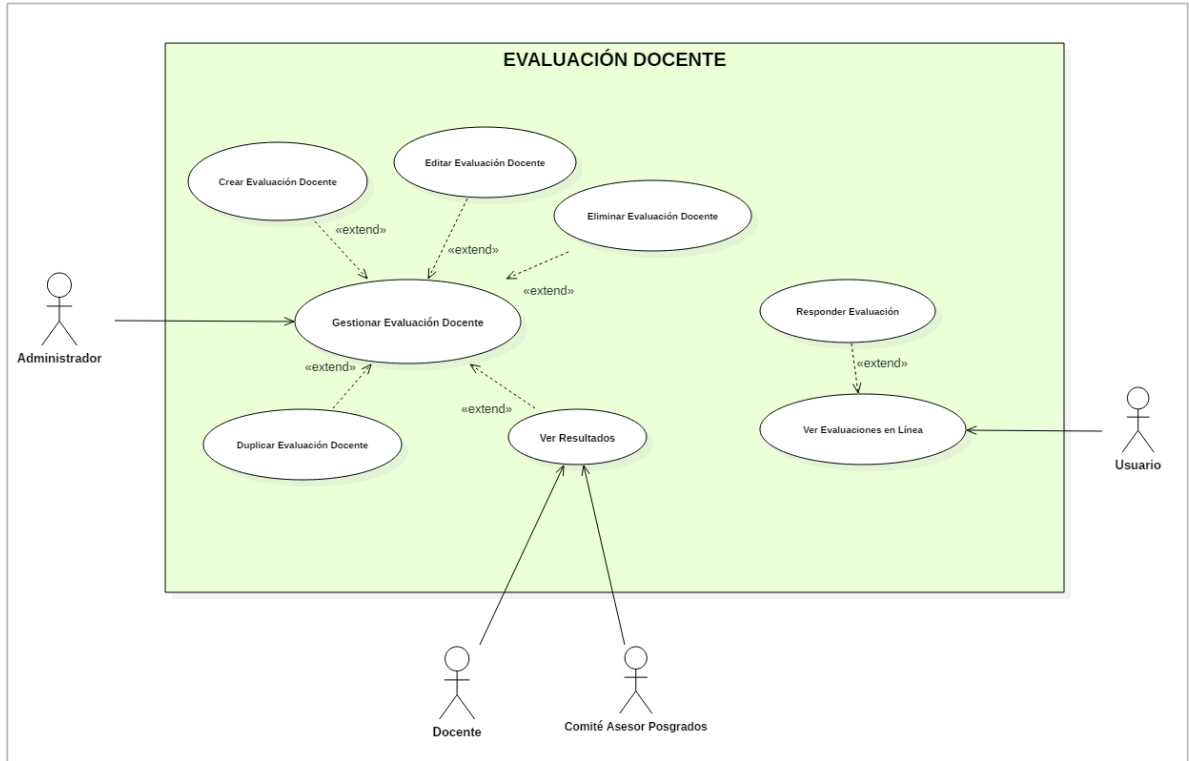
4.1.1.1 Servicio de Encuestas Escuela

Figura 6. Diagrama de casos de uso: Encuestas Escuela



4.1.1.2 Servicio de Evaluación Docente

Figura 7. Diagrama de casos de uso: Evaluación Docente



4.1.2 Documentación de Casos de Uso del Sistema

4.1.2.1 Servicio de Encuestas Escuela

Tabla 1. Casos de uso: Encuesta Escuela

TÍTULO	FUNCIONES PRIMARIAS
Gestionar Encuesta	Acceder al listado de las encuestas que han sido creadas hasta el momento por el usuario.
Crear Encuesta	Crear una encuesta con la debida información, estableciendo fechas de vigencia y con las preguntas elegidas por el usuario.

TÍTULO	FUNCIONES PRIMARIAS
Editar Encuesta	Modificar y actualizar la información de una encuesta antes creada.
Eliminar Encuesta	Borrar una encuesta creada anteriormente.
Duplicar Encuesta	Crea una encuesta igual a otra ya creada pero con estado inactivo.
Ver Resultados	Acceder a la vista de los resultados de la encuesta. El administrador puede ver todos los resultados en detalle y el listado de los votantes. El usuario participante solo puede ver lo que el administrador permita.
Ver Encuestas en Línea	Acceder al listado de las encuestas en las que el usuario está habilitado para participar.
Responder Encuesta	Participar en una encuesta en línea.

4.1.2.2 Servicio de Evaluación Docente

Tabla 2. Casos de Uso: Evaluación Docente

TÍTULO	FUNCIONES PRIMARIAS
Gestionar Evaluación Docente	Acceder al listado de las evaluaciones docente que han sido creadas hasta el momento por el usuario, tanto las vigentes como las inactivas y vencidas.
Crear Evaluación Docente	Crear una evaluación con la debida información, estableciendo fechas de vigencia, seleccionando de los cronogramas la materia y el grupo a los cuales se aplicarán, con las preguntas elegidas por el administrador.
Editar Evaluación Docente	Modificar y actualizar la información de una evaluación docente antes creada.
Eliminar Evaluación Docente	Borrar una evaluación docente creada anteriormente.

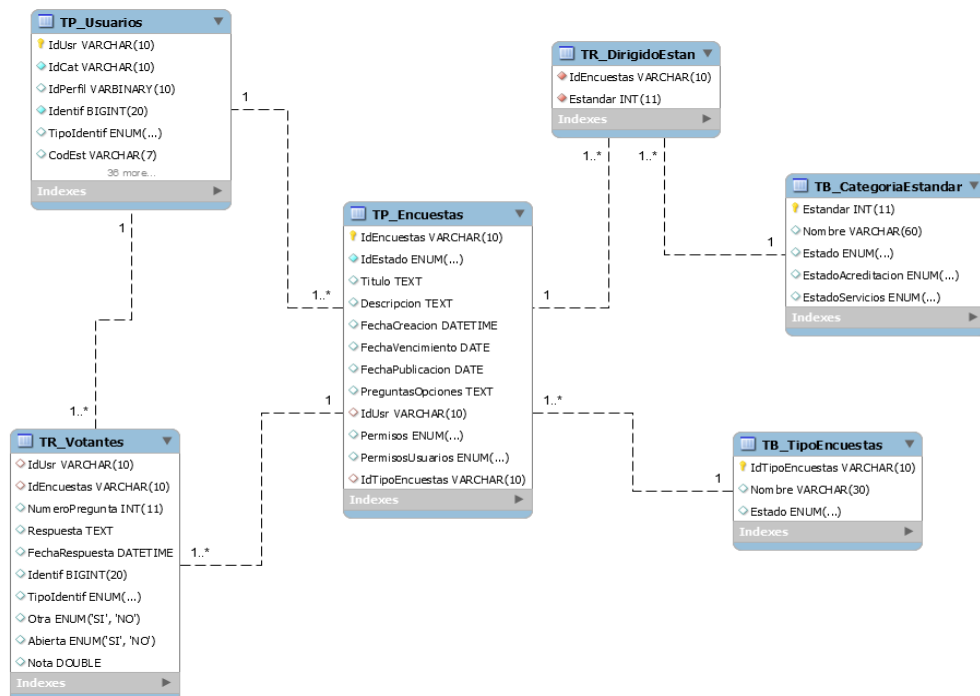
TÍTULO	FUNCIONES PRIMARIAS
Duplicar Evaluación Docente	Crea una evaluación docente igual a otra ya creada pero con estado inactivo.
Ver Resultados	Acceder a la vista de los resultados de la evaluación docente. Dichos resultados estarán disponibles únicamente para el administrador de la evaluación, docentes evaluados y comité asesor de posgrados.
Ver Evaluaciones en Línea	Acceder al listado de las evaluaciones en las que el usuario está habilitado para participar.
Responder Evaluación	Participar en una evaluación en línea.

4.1.3 Diseño y Análisis

Diagrama Entidad / Relación de los servicios desarrollados

Servicio Encuestas Escuela

Figura 8. Diagrama E / R: Encuestas Escuela

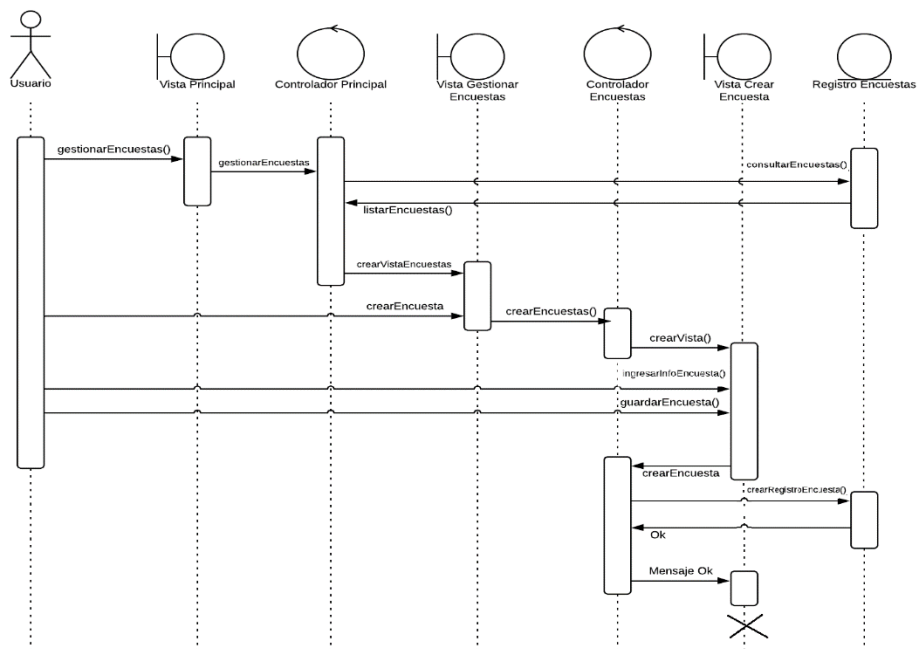


ENTIDAD	DESCRIPCIÓN
TB_CategoriaEstandar	Contiene todas las categorías que puede tener un usuario.
TR_DirigidoEstan	Relaciona las encuestas con las categorías estándar.
TB_TipoEncuestas	Contiene los diferentes tipos de encuestas que existen.
TR_EvalDocenteEncues	Contiene la información básica de una evaluación docente.
TR_Alumnos	Relaciona a los alumnos con la evaluación docente.
TB_MateriasN	Contiene la información básica de las asignaturas.
TR_Pensum	Relaciona los programas académicos con las asignaturas.
TP_CronogramaAcademico	Contiene la información básica de los cronogramas académicos vigentes.
TB_Programas	Contiene la información básica de los programas académicos.
TR_CronogramaMateria	Relaciona los cronogramas académicos con las asignaturas.
TB_Ciudades	Contiene las distintas ciudades donde se desarrollan los programas académicos.
TP_Recurso	Contiene información de los recursos solicitados para el desarrollo de los cronogramas.
TR_DocenteMateriaCronograma	Relaciona los docentes con el cronograma y materias a las que pertenecen.
TB_TipoRecurso	Contiene la información básica de los tipos de recursos disponibles para asignar a los cronogramas.
TR_CronogramaMateriaUsoRecurso	Relaciona cronogramas y materias y las fechas asignadas para el uso de los recursos.
TR_UsoRecurso	Relaciona los recursos con la disponibilidad de fechas para su uso.

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 secuencias necesarios para cada caso de uso, en los que se explica con detalle los pasos para el funcionamiento de cada uno de los servicios.

Servicio de Encuestas Escuela

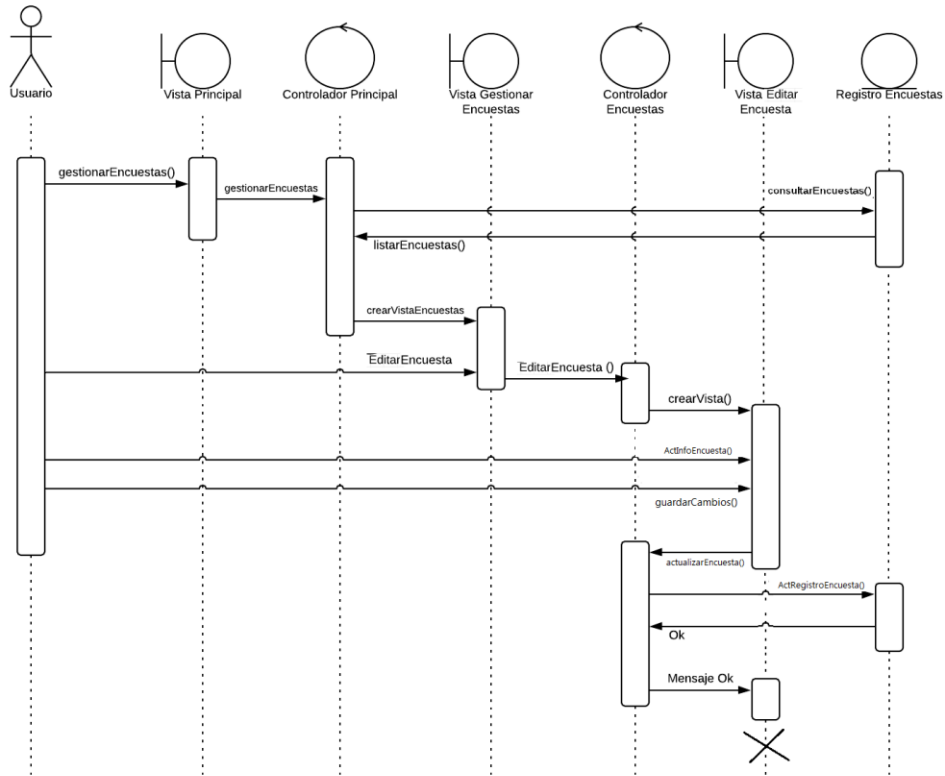
Figura 10. Diagrama de secuencia: Gestionar Encuestas – Crear Encuesta



1. El usuario solicita Gestionar Encuestas en la vista principal.
2. El controlador ControladorPrincipal recibe la petición, consulta las encuestas en la base de datos y despliega la vista VistaGestionarEncuestas.
3. El usuario solicita crear una encuesta en la vista VistaGestionarEncuestas.
4. El controlador ControladorEncuesta recibe la petición de crearEncuesta, crea la encuesta en la base de datos y retorna un mensaje de Ok al usuario en la VistaCrearEncuestas.

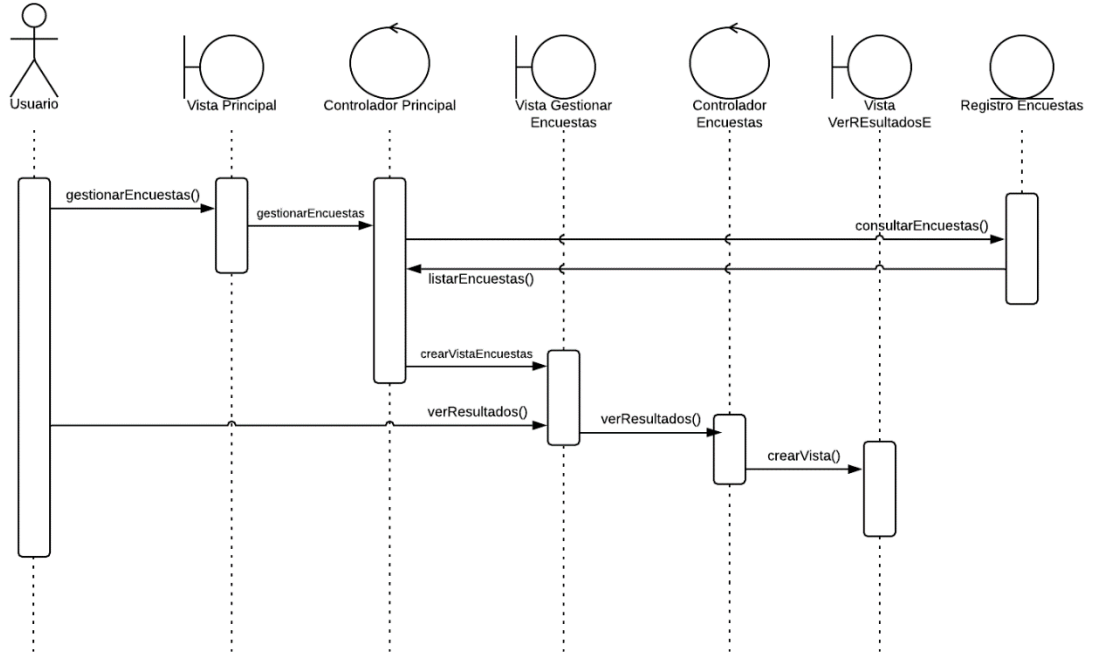
5. Se destruye la vista VistaCrearEncuesta.

Figura 11. Diagrama de secuencia: Gestionar Encuestas – Editar Encuesta



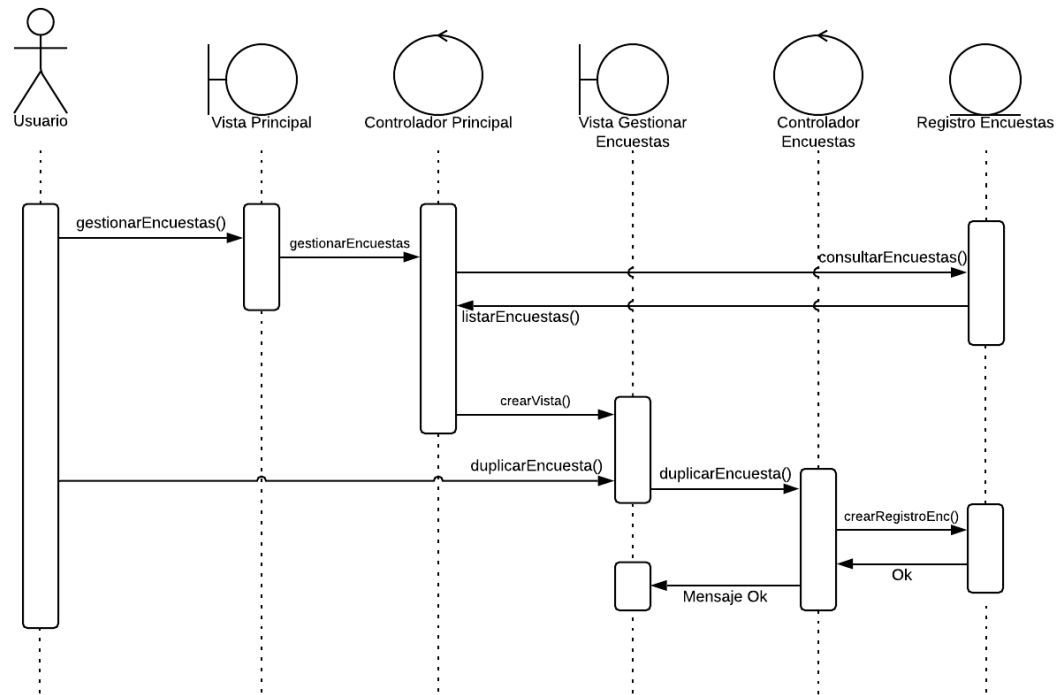
1. El usuario solicita Gestionar Encuestas en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las encuestas en la base de datos y despliega la vista VistaGestionarEncuestas.
3. El usuario solicita editar una encuesta en la vista VistaGestionarEncuestas.
4. El controlador ControladorEncuesta recibe la petición de editarEncuesta, actualiza la encuesta en la base de datos y retorna un mensaje de Ok al usuario en la VistaCrearEncuestas.
5. Se destruye la vista VistaCrearEncuesta.

Figura 12. Diagrama de secuencia: Gestionar Encuestas – Ver Resultados Encuesta



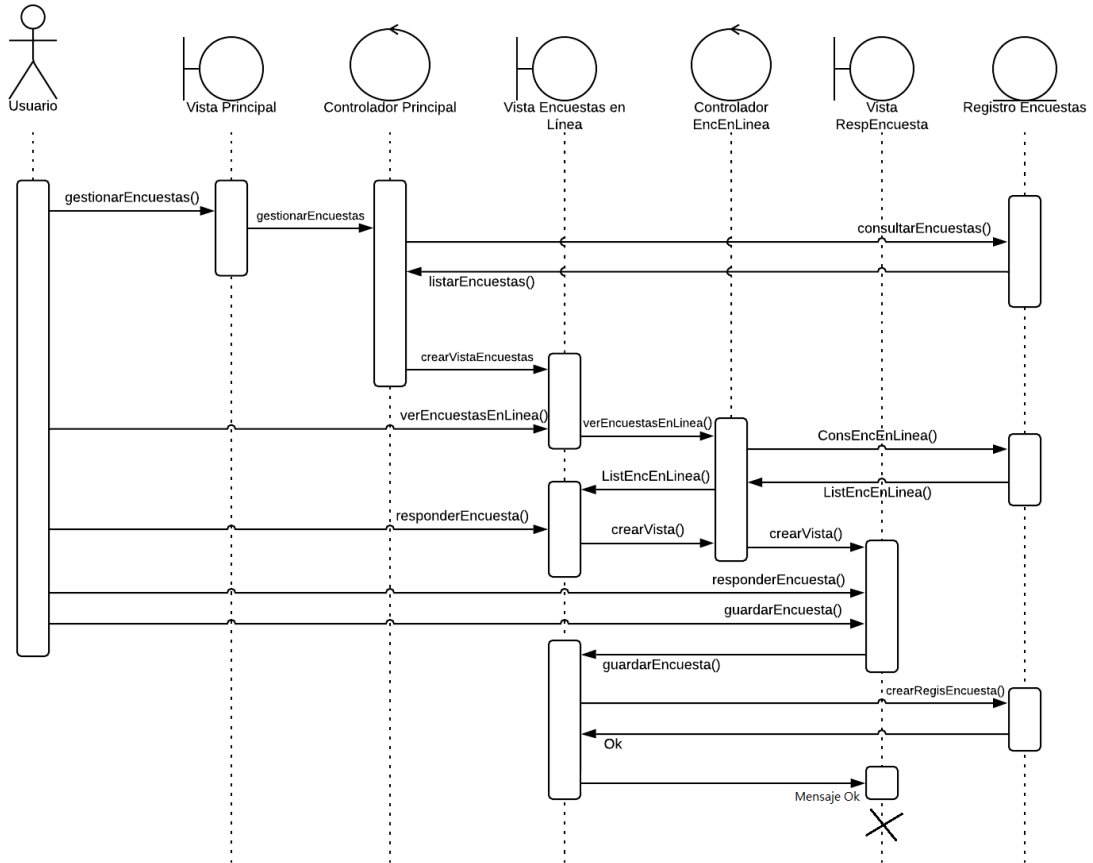
1. El usuario solicita Gestionar Encuestas en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las encuestas en la base de datos y despliega la vista VistaGestionarEncuestas.
3. El usuario solicita Ver Resultados en una encuesta en la vista VistaGestionarEncuestas.
4. El controlador ControladorEncuesta recibe la petición de verResultados y crea la vista vistaVerResultadosEncuesta.

Figura 13. Diagrama de secuencia: Gestionar Encuestas – Duplicar Encuesta



1. El usuario solicita Gestionar Encuestas en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las encuestas en la base de datos y despliega la vista VistaGestionarEncuestas.
3. El usuario solicita duplicarEncuesta en una encuesta en la vista VistaGestionarEncuestas.
4. El controlador ControladorEncuesta recibe la petición de duplicarEncuesta y devuelve un mensaje de Ok en la vista VistaGestionarEncuestas.

Figura 14. Diagrama de secuencia: Gestionar Encuestas – Responder Encuesta

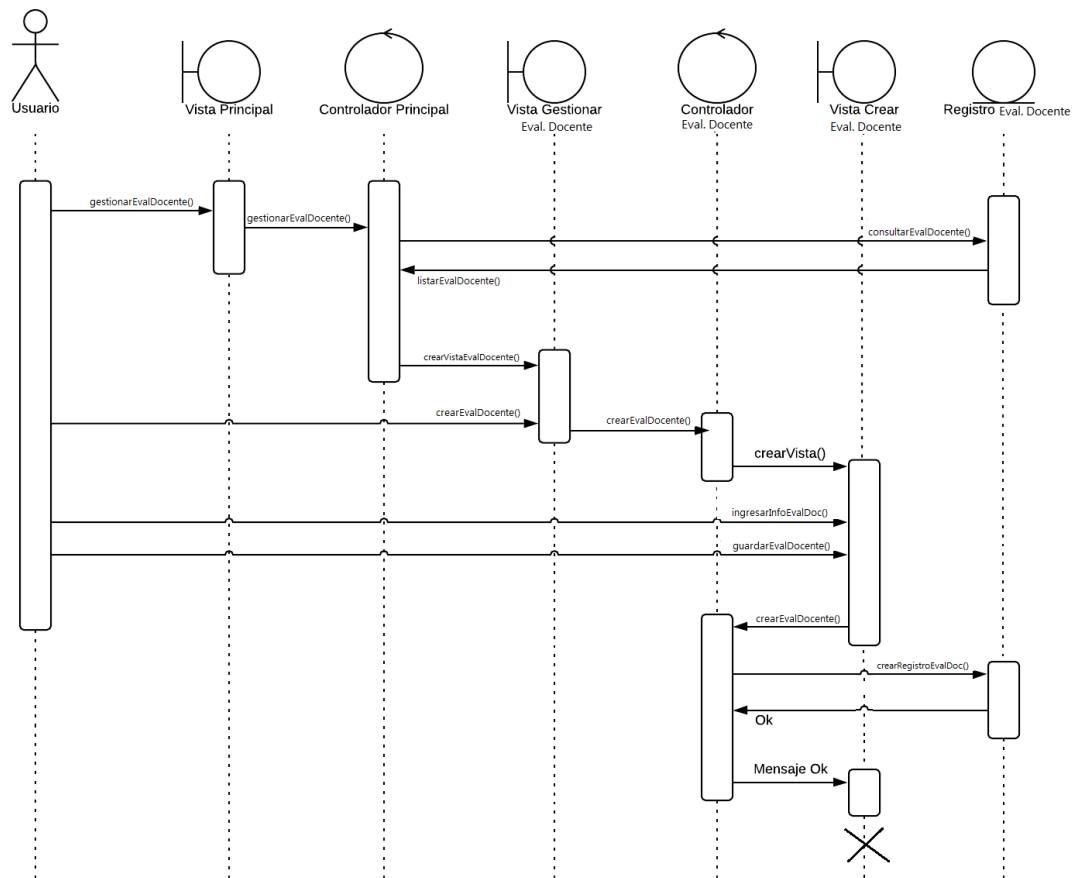


1. El usuario solicita Gestionar Encuestas en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las encuestas en la base de datos y despliega la vista VistaGestionarEncuestas.
3. El usuario solicita verEncuestaEnLinea en la vista VistaGestionarEncuestas y selecciona en cual encuesta que esté en línea va a participar.
4. El controlador ControladorEncuesta recibe la petición de responderEncuesta y crea la vistaResponderEncuesta.
5. El usuario diligencia los campos y guarda la encuesta.

6. El controlador ControladorEncuesta recibe la petición de guardarEncuesta, actualiza el registro de encuestas y devuelve un mensaje de Ok en la vista vistaResponderEncuestas.

Servicio de Evaluación Docente

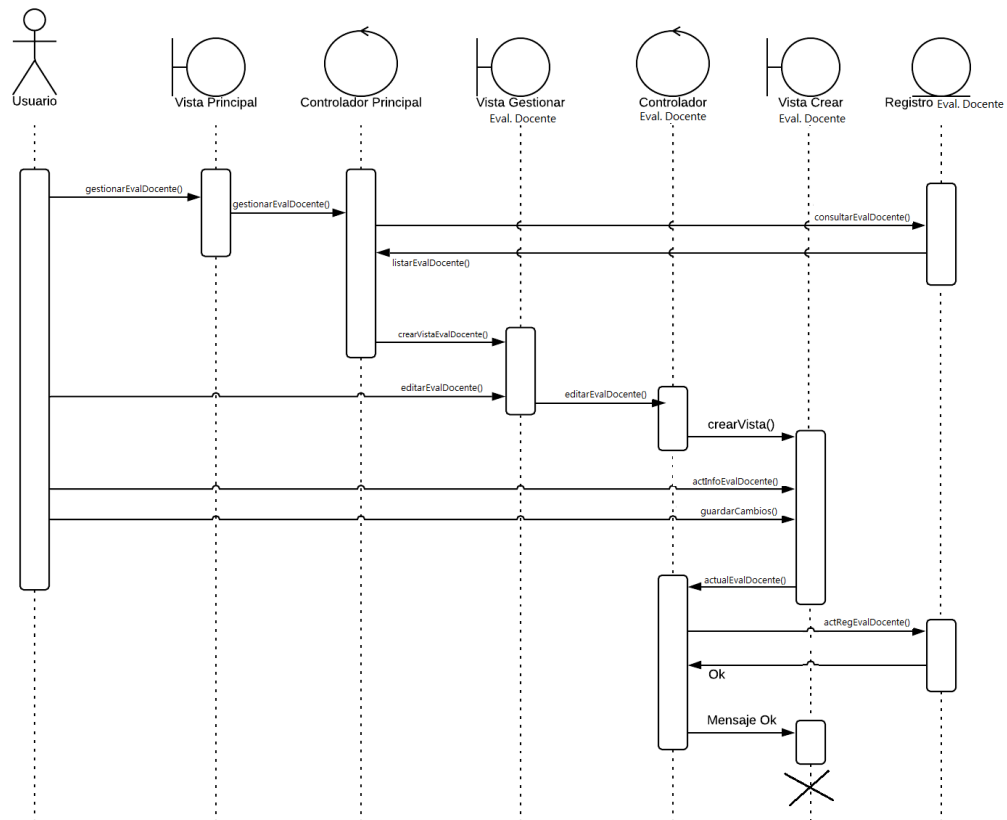
Figura 15. Diagrama de secuencia: Gestionar Evaluación Docente – Crear Evaluación Docente



1. El usuario solicita Gestionar EvalDocente en la vista principal.
2. El controlador ControladorPrincipal recibe la petición, consulta las evaluaciones en la base de datos y despliega la vista VistaGestionarEvalDocente.

3. El usuario solicita crear una encuesta en la vista VistaGestionarEvalDocente.
4. El controlador ControladorEvalDocente recibe la petición de crearEvalDocente, crea la evaluación en la base de datos y retorna un mensaje de Ok al usuario en la VistaCrearEvalDocente.
5. Se destruye la vista VistaCrearEvalDocente.

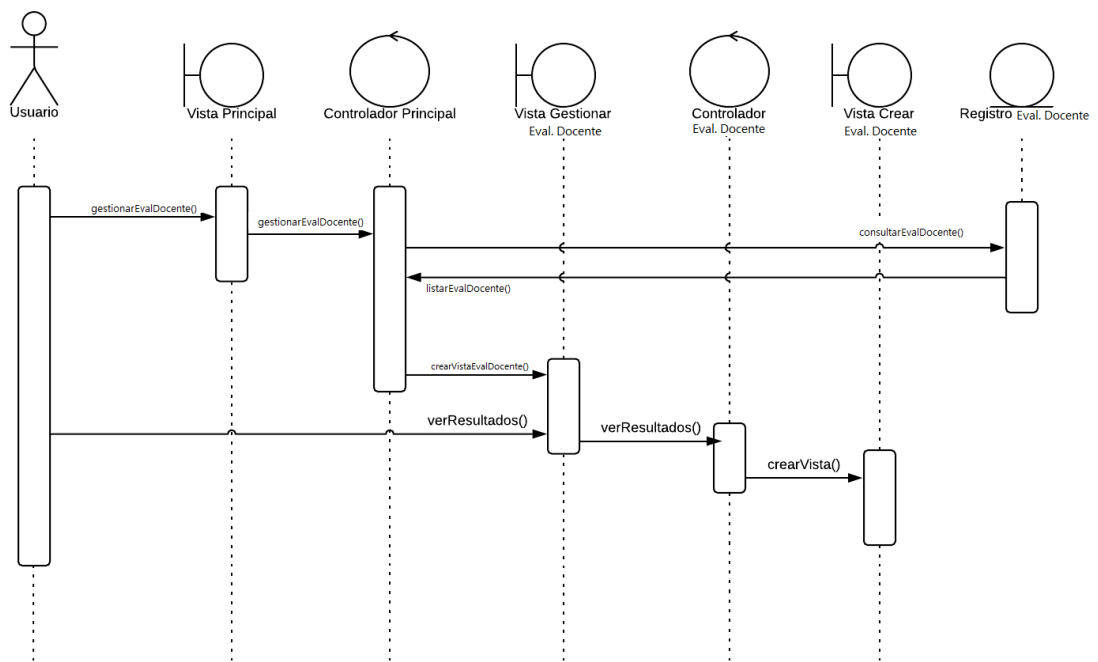
Figura 16. Diagrama de secuencia: Gestionar Evaluación Docente – Editar Evaluación Docente



1. El usuario solicita Gestionar EvalDocente en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las evaluaciones en la base de datos y despliega la vista VistaGestionarEvalDocente.
3. El usuario solicita editar una encuesta en la vista VistaGestionarEvalDocente.

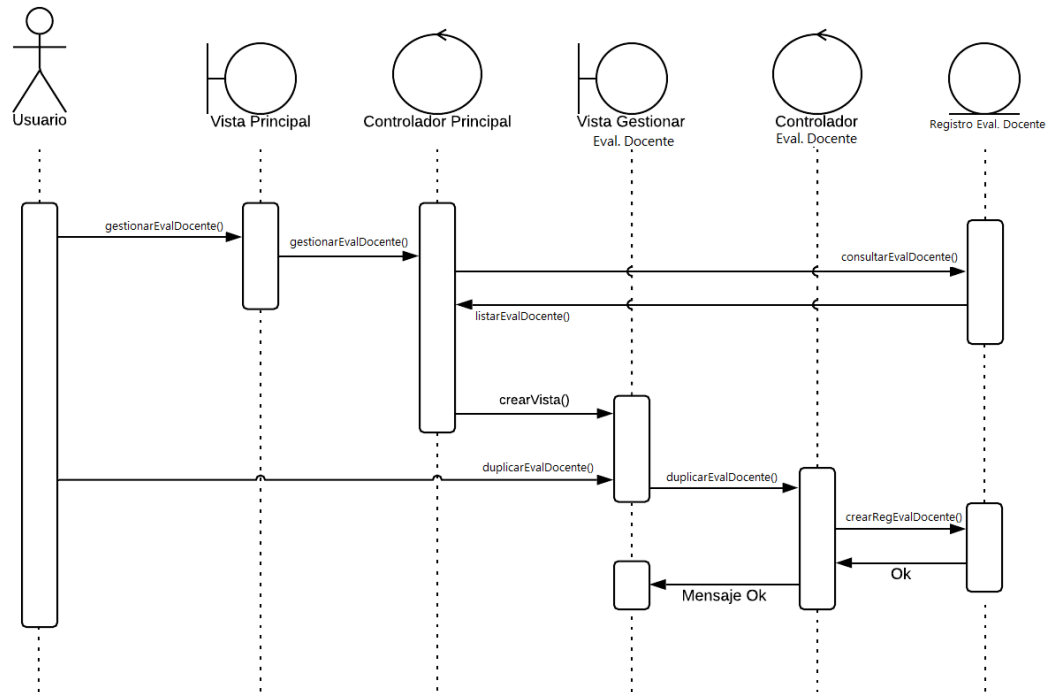
4. El controlador ControladorEvalDocente recibe la petición de editarEvalDocente, actualiza la evaluación en la base de datos y retorna un mensaje de Ok al usuario en la VistaEditarEvalDocente.
5. Se destruye la vista VistaEditarEvalDocente.

Figura 17. Diagrama de secuencia: Gestionar Evaluación Docente – Ver Resultados Evaluación Docente



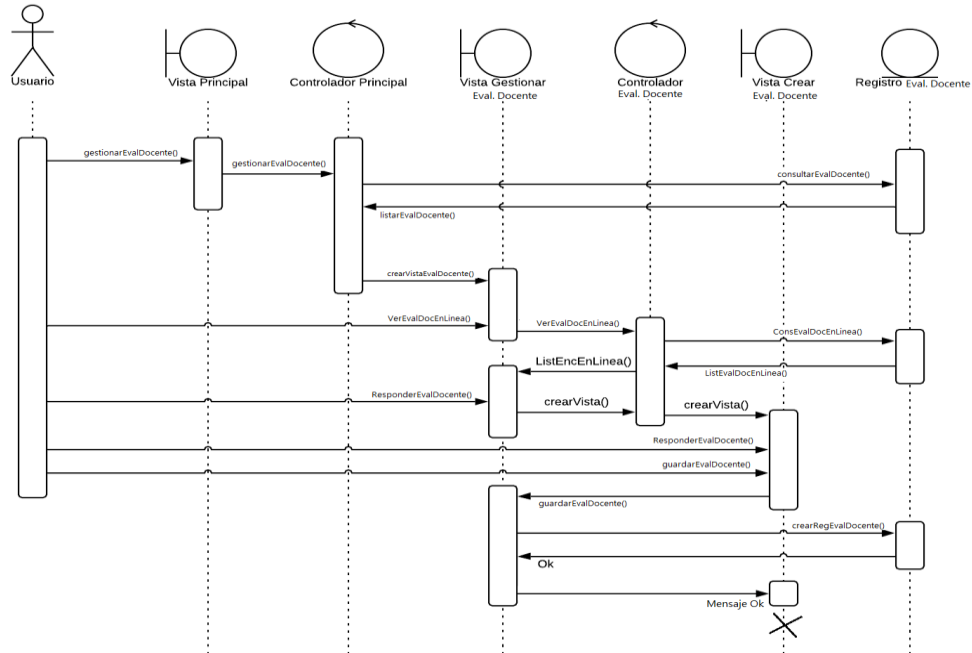
1. El usuario solicita Gestionar EvalDocente en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las evaluaciones en la base de datos y despliega la vista VistaGestionarEvalDocente.
3. El usuario solicita Ver Resultados en una evaluación en la vista VistaGestionarEvalDocente.
4. El controlador ControladorEvalDocente recibe la petición de verResultados y crea la vista vistaVerResultadosEvalDocente.

Figura 18. Diagrama de secuencia: Gestionar Evaluación Docente – Duplicar Evaluación Docente



1. El usuario solicita Gestionar EvalDocente en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las evaluaciones en la base de datos y despliega la vista VistaGestionarEvalDocente.
3. El usuario solicita duplicarEvalDocente en una evaluación en la vista VistaGestionarEvalDocente.
4. El controlador ControladorEvalDocente recibe la petición de duplicarEvalDocente y devuelve un mensaje de Ok en la vista VistaGestionarEvalDocente.

Figura 19. Diagrama de secuencia: Gestionar Evaluación Docente – Responder Evaluación Docente



1. El usuario solicita Gestionar EvalDocente en la vista VistaPrincipal.
2. El controlador ControladorPrincipal recibe la petición, consulta las evaluaciones en la base de datos y despliega la vista VistaGestionarEvalDocente.
3. El usuario solicita verEvalDocenteEnLinea en la vista VistaGestionarEvalDocente.y selecciona en cual encuesta que esté en línea va a participar.
4. El controlador ControladorEvalDocente recibe la petición de responderEvalDocente y crea la vistaResponderEvalDocente.
5. El usuario diligencia los campos y guarda la evaluación.

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 tuvieron en cuenta las sugerencias de personal administrativo de la escuela de Ingeniería de Petróleos (principales interesados en el servicio de evaluación docente), para realizar un posterior refinamiento a las interfaces.

Para el desarrollo del proyecto se trabajó con la base de datos “Diamante”, ubicada en todos los servidores de los portales de las escuelas, donde se crearon y modificaron las tablas necesarias para el funcionamiento de los servicios.

Se llevaron a cabo pruebas, verificando que los resultados correspondieran con lo esperado. De esta forma se evidencio el correcto funcionamiento en la captura de datos, selección de ítems y almacenamiento de la información.

La implantación de los servicios se llevó a cabo en primer lugar en el portal web de pruebas del grupo Calumet, donde se verifican todos los componentes. Finalmente se implementan en todos los portales web a los cuales el grupo calumet presta soporte.

5. PRUEBAS DEL SISTEMA

Para garantizar el correcto desarrollo de los nuevos servicios creados, 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 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 Encuestas

Tabla 4. Pruebas realizadas: Encuestas.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Gestionar Encuesta	Muestra correctamente las encuestas en línea, fuera de línea y vencidas.	<input type="checkbox"/>
Crear Encuesta	Crea y guarda correctamente una encuesta.	<input type="checkbox"/>
Editar Encuesta	Actualiza y guarda correctamente una encuesta editada.	<input type="checkbox"/>
Eliminar Encuesta	Elimina correctamente una encuesta no deseada.	<input type="checkbox"/>
Duplicar Encuesta	Se crea una encuesta igual a la anterior con estado inactivo y diferente id.	<input type="checkbox"/>
Ver Resultados	Muestra correctamente los resultados de una encuesta seleccionada que contenga respuestas.	<input type="checkbox"/>
Ver Encuesta En Línea	Muestra correctamente un listado de las encuestas en las que puede participar determinado usuario.	<input type="checkbox"/>

5.1.1.2 Servicio de Evaluación Docente

Tabla 5. Pruebas realizadas: Evaluación Docente.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Gestionar Evaluación Docente	Muestra correctamente las evaluaciones docente en línea, fuera de línea y vencidas.	<input type="checkbox"/>
Crear Evaluación Docente	Crea y guarda correctamente una evaluación docente.	<input type="checkbox"/>
Editar Evaluación Docente	Actualiza y guarda correctamente una evaluación docente editada.	<input type="checkbox"/>
Eliminar Evaluación Docente	Elimina correctamente una evaluación docente no deseada.	<input type="checkbox"/>

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Duplicar Evaluación Docente	Se crea una evaluación docente igual a la anterior con estado inactivo y diferente id.	<input type="checkbox"/>
Ver Resultados	Muestra correctamente los resultados de una evaluación docente seleccionada que haya sido respondida.	<input type="checkbox"/>
Ver Evaluación Docente en Línea	Muestra correctamente un listado de las evaluaciones docente en las que puede participar determinado usuario.	<input type="checkbox"/>

6. CONCLUSIONES

- La creación del servicio de encuestas a nivel de escuela, permite que se pueda obtener información importante sobre diversos temas que competen a la comunidad de cada escuela en particular, ayuda a la interacción y comunicación entre sus miembros y facilita la interpretación de los resultados, mostrándolos de una manera más dinámica.
- El desarrollo del servicio para la creación de evaluaciones docente de posgrados de la escuela de Ingeniería de Petróleos, facilitará la realización de la misma, evitando implementarla en papel, permitirá de manera sencilla la creación, edición o eliminación de preguntas al igual que la gestión de las evaluaciones docente que serán necesarias en la finalización de cada ciclo. De la misma manera permitirá habilitar estas evaluaciones a los alumnos que les corresponda en el momento.
- El servicio que permite participar en las evaluaciones docentes, facilita a los estudiantes acceder a estas en cualquier momento (después de ser activadas por el administrador), mostrándoles únicamente las evaluaciones que deberán responder según la asignatura y el docente responsable de la misma, de una manera rápida y sencilla y accediendo a todas ellas en el mismo lugar.
- El servicio de resultados de las evaluaciones docentes es de gran ayuda para los miembros del comité asesor de posgrados de la escuela de Ingeniería de Petróleos, pues de esta manera se podrá visualizar el rendimiento de los docentes en cada asignatura, permitiéndole al comité decidir sobre la continuidad de estos en los programas académicos. De igual manera los docentes podrán ver estos resultados que los impulsarán a continuar trabajando en pro de la mejoría de sus asignaturas y de los programas académicos en la misma medida.

7. RECOMENDACIONES

- Se sugiere agregar al servicio de encuestas y de evaluación docente, al momento de crear las preguntas, añadir la función de duplicar pregunta, debido a que en muchas ocasiones son muy similares y solo requieren de pequeños ajustes que se pueden realizar mediante la edición.
- Permitir la eliminación de dos o más encuestas o evaluaciones al mismo tiempo, ya que solo se puede realizar individualmente.
- Continuar mejorando las funciones de la evaluación docente, de manera que permita realizar un mayor manejo de los resultados, comparándolos a través del tiempo con el fin de facilitar la toma de decisiones.

BIBLIOGRAFÍA

APRENDER- INFORMÁTICA Tutoriales del lenguaje de programación JavaScript. [en línea] disponible en: <http://www.aprender-informatica.com/>. (n.d).

DGSCA Manuales básicos de JSP. [en línea] disponible en: <http://manuales.dgsca.unam.mx/jsp>.

HTMLQUICK tutoriales de cómo programar en html de gran ayuda en detalles básicos y sencillos. [en línea] disponible en: <http://www.htmlquick.com/es/tutorials.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.

MySQL, [en línea] disponible en: <http://www.mysql.com>. Sitio oficial de

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.

PROGRAMACION FACIL manual de programación para Java Jsp (Java Server Pages) [en línea] disponible en: http://www.programacionfacil.com/programacion:manual_java_jsp,

PROGRAMACION JAVA [en línea] disponible en: <http://www.programacion.com/java/tutorial>.

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.<http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor>. Información acerca de la Arquitectura Cliente Servidor.

YOUBLISHER Manual de conceptos básicos sobre [en línea] disponible en: UML.<http://www.youblisher.com/p/155105-Aprendiendo-UML/>.