

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Proyecto Tecnológico

SISTEMA BASADO EN SERVICIOS WEB PARA GENERAR Y
VISUALIZAR ESTADISTICAS SOBRE OPINIONES DE UEA's

César Ontiveros Herrera

205361398

Asesor:

Dra. Maricela Claudia Bravo Contreras

Profesor Asociado

Departamento de sistemas

Trimestre 2016 Invierno

21 de septiembre del 2016

Yo, **Maricela Claudia Bravo Contreras**, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Maricela Claudia Bravo Contreras

Yo, **César Ontiveros Herrera**, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



César Ontiveros Herrera

Resumen del Proyecto

- Diseño de la interfaz de usuario.
- Diseño e implementación de las bases de datos que contendrán la información.
- Diseño e implementación de los componentes de registro de opiniones en la base de datos en Java.
- Diseño e implementación de los componentes de consulta de opiniones en la base de datos desde Java.
- Diseño e implementación de los componentes de cálculos estadísticos de las opiniones registradas en la base de datos.
- Programación de los componentes software que contendrán los servicios web.
- Documentación y elaboración del reporte.

Índice de contenido	Pág.
Introducción.....	3
Objetivo general.....	3
Objetivos particulares.....	3
Marco teórico.....	4
Metodología empleada en el proyecto.....	4
Arquitectura del sistema.....	8
Desarrollo del proyecto.....	8
Requerimientos.....	8
Módulo de registro.....	8
Módulo mostrar.....	9
Módulo evaluar.....	9
Módulo de las estadísticas.....	9
Diseño.....	9
Actores del sistema.....	9
Casos de uso.....	11
Caso de uso: Registrar profesor.....	11
Caso de uso: Evaluar profesor.....	11
Caso de uso: Buscar profesor.....	11
Diagrama de secuencia.....	11
Base de Datos.....	14
Implementación.....	15
Ejecución de la aplicación.....	16
Conclusiones.....	20
Bibliografía.....	21

Índice de figuras	Pág.
Figura 1. Esquema de una aplicación MVC.....	5
Figura 2. Arquitectura básica de los servicios web.....	7
Figura 3. Arquitectura general del sistema.....	8
Figura 4. Diagrama de casos de uso general.....	10
Figura 5. Diagrama de secuencia Registrar Profesor.....	12
Figura 6. Diagrama de secuencia Evaluar Profesor.....	13
Figura 7. Diagrama Relacional de la Base de Datos.....	14
Figura 8. Esquema del paquete base de sistema.....	15
Figura 9. Página principal de la aplicación.....	16
Figura 10. Registrar profesor.....	17
Figura 11. Mostrar profesores registrados.....	18
Figura 12. Buscar profesor.....	19

Introducción

Existen muchas instituciones de nivel superior donde los alumnos inscriben las materias que van a tomar al inicio de cada periodo, en ese momento los alumnos deben decidir sobre la conveniencia de inscribirse a un curso con uno u otro profesor; por ello es importante para los alumnos contar con recomendaciones emitidas por alumnos acerca de la forma que cada profesor tiene para impartir y evaluar sus cursos. En la mayoría de estas instituciones no existe manera alguna para que los alumnos puedan conocer la forma de impartir el curso por parte de los profesores en las aulas, lo que provoca cierta desventaja para los alumnos ya que no cuentan con información suficiente antes de cursar cualquier materia con algún profesor.

Este proyecto de integración tiene como finalidad ofrecer un sistema a través del cual se puedan procesar y consultar las opiniones que los alumnos de la UAM Azcapotzalco tienen sobre los profesores de los diferentes departamentos de la misma unidad, de tal forma que la información recabada pueda ser de utilidad a los interesados, en este caso a los alumnos, para contar con mayor información antes de inscribirse a una UEA (Unidad de Enseñanza Aprendizaje).

Objetivo general

Diseñar y desarrollar un sistema basado en Servicios Web que permita registrar opiniones, generar y visualizar estadísticas sobre el desempeño de un profesor y las UEA (Unidad de Enseñanza-Aprendizaje) que imparte en relación a su facilidad para explicar, puntualidad, apego al programa de la UEA, asesoría extra-clase y forma de evaluar.

Objetivos particulares

1. Diseñar e implementar la base de datos para almacenar la información utilizando un sistema gestor de bases de datos relacional.
2. Diseñar e implementar los componentes Java que realizaran la lógica de la aplicación.
3. Modelar y programar los componentes de software para generar los Servicios Web.
4. Diseñar e implementar la interface de usuario que nos permita visualizar los resultados.

Marco teórico

En esta sección se presentan algunos conceptos generales sobre la metodología utilizada para el desarrollo del sistema de información, así como su arquitectura general. Se definen conceptos generales sobre los servicios web con el propósito de dar a conocer su significado y alcance.

Metodología empleada en el proyecto

En este proyecto se utilizó la arquitectura Modelo Vista Controlador (MVC) o Model View Controller, esta arquitectura es un patrón de diseño que especifica cómo debe ser estructurada una aplicación, las capas que van a componer la misma y la funcionalidad de cada una. El patrón MVC consiste en dividir las aplicaciones en tres capas:

- Controlador
- Vista
- Modelo

El **controlador** se puede decir que es el “cerebro” de la aplicación. Todas las peticiones a la capa intermedia que se realicen desde el cliente son dirigidas al Controlador, cuya misión es determinar las acciones a realizar para cada una de estas peticiones e invocar al resto de los componentes de la aplicación (Modelo y vista) para que realicen las acciones requeridas en cada caso, encargándose también de la coordinación de todo el proceso.

Tal y como se puede deducir de su nombre, la **Vista** es la encargada de generar las respuestas (habitualmente XHTML¹) que deben ser enviadas al cliente. Cuando estas respuestas tienen que incluir datos proporcionados por el Controlador, el código XHTML de la página no será fijo si no que deberá ser generado de forma dinámica, por lo que su implementación correrá a cargo de una página JSP².

En la arquitectura MVC la lógica de negocio de la aplicación, incluyendo el acceso a los datos y su manipulación, está encapsulada dentro del modelo. El **Modelo** lo forman una serie de componentes de negocio independientes del controlador y el desacoplamiento entre las capas.

¹ XHTML: Siglas del inglés eXtensible HyperText Markup Language. XHTML es básicamente HTML expresado como XML válido.

² JSP: JavaServer Pages. Tecnología para crear páginas web dinámicas basada en Java.

En una aplicación J2EE³ el modelo puede ser implementado mediante clases estándar Java o a través de Enterprise JavaBeans⁴.

En la Figura 1. Esquema de una aplicación MVC, se muestra esta arquitectura para el caso de una aplicación desarrollada con tecnologías J2EE. En ella podemos ver cómo se relacionan estos tres bloques funcionales entre sí, su interacción con el resto de las capas de la aplicación y la tecnología con la que están implementados.

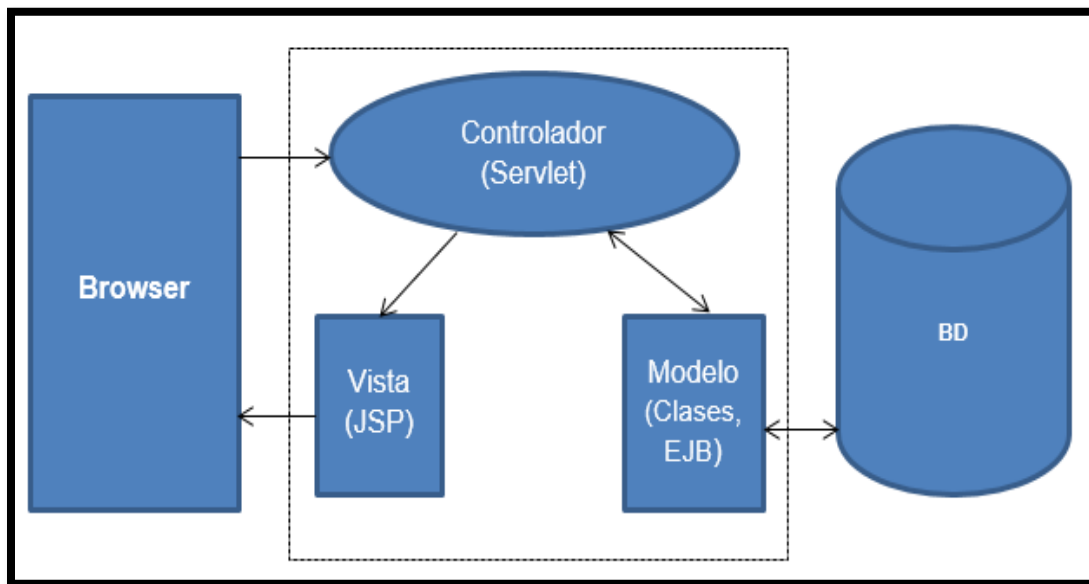


Figura 1. Esquema de una aplicación MVC [2]

³ J2EE: Java Enterprise Edition. Java 2 Edición Empresarial.

⁴ Enterprise JavaBeans: son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE.

Para la implementación de este proyecto se utilizaron servicios web (SW), los cuales se describen a continuación:

Según Gustavo Alonso [1] una definición precisa de SW es la emitida por la W3C⁵: “una aplicación software identificada por una URI⁶, cuyas interfaces y vinculaciones son capaces de ser definidas, descritas y descubiertas como artefactos XML⁷. Un SW soporta la interacción con otros agentes software mediante el intercambio de mensajes basado en XML a través de protocolos basados en Internet”.

La arquitectura de los servicios Web toma lugar entre la relación de varios componentes y las tecnologías que comprenden un Servicio Web. La arquitectura básica incluye:

- Intercambiar mensajes.
- Describir Servicios Web.
- Publicar y descubrir las descripciones de los Servicios Web.

La arquitectura básica de servicios Web define la interacción entre los agentes de software como un intercambio de mensajes entre los solicitantes de servicios y los proveedores de servicios. Los solicitantes son agentes de software que solicitan la ejecución de un servicio. Los proveedores son agentes de software que proporcionan un servicio. Los agentes pueden ser tanto los solicitantes de servicios y proveedores. Los proveedores son responsables de la publicación de la descripción de los servicios que proporcionan. Los solicitantes deben ser capaces de encontrar las descripciones de los servicios. Los modelos de las interacciones de la arquitectura básica de los servicios web entre los tres papeles son los siguientes:

- Solicitante del servicio: solicita la ejecución de un servicio web.
- Proveedor de servicios: procesa una solicitud de servicio web.
- Agencia descubierta: agencia mediante la cual una descripción del servicio Web se publica y está visible

⁵W3E: es una comunidad internacional donde las organizaciones miembros, trabajan conjuntamente para desarrollar estándares web.

⁶ URI: Uniform Resource Identifier, Identificador de recursos uniforme.

⁷ XML: Extensible Markup Lenguaje, es un metalenguaje que nos permite definir lenguajes de marcado adecuados a sus determinados.

La Figura 2. Arquitectura básica de los servicios web, ilustra la arquitectura básica de los Servicios Web, en la cual un servicio solicitante y un servicio proveedor interactúan basados en la descripción del servicio de información publicado por el proveedor y descubierto por el solicitante a través de una forma de agencia descubierta. Los servicios solicitante y proveedor interactúan intercambiando mensajes, los cuales pueden ser agregados para formar MEPs (Patrones de intercambio de mensajes).

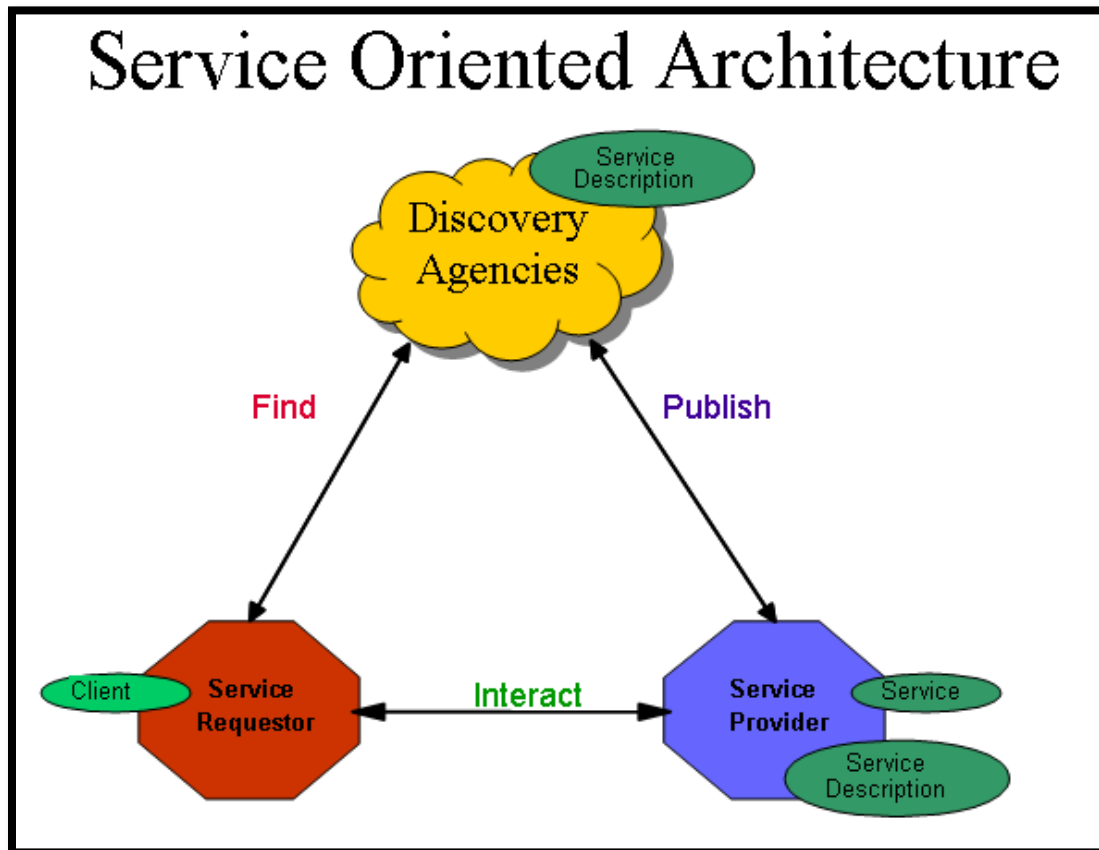


Figura 2. Arquitectura básica de los Servicios Web [8].

Arquitectura del sistema

La Figura 3, muestra la arquitectura del sistema y su funcionamiento general.

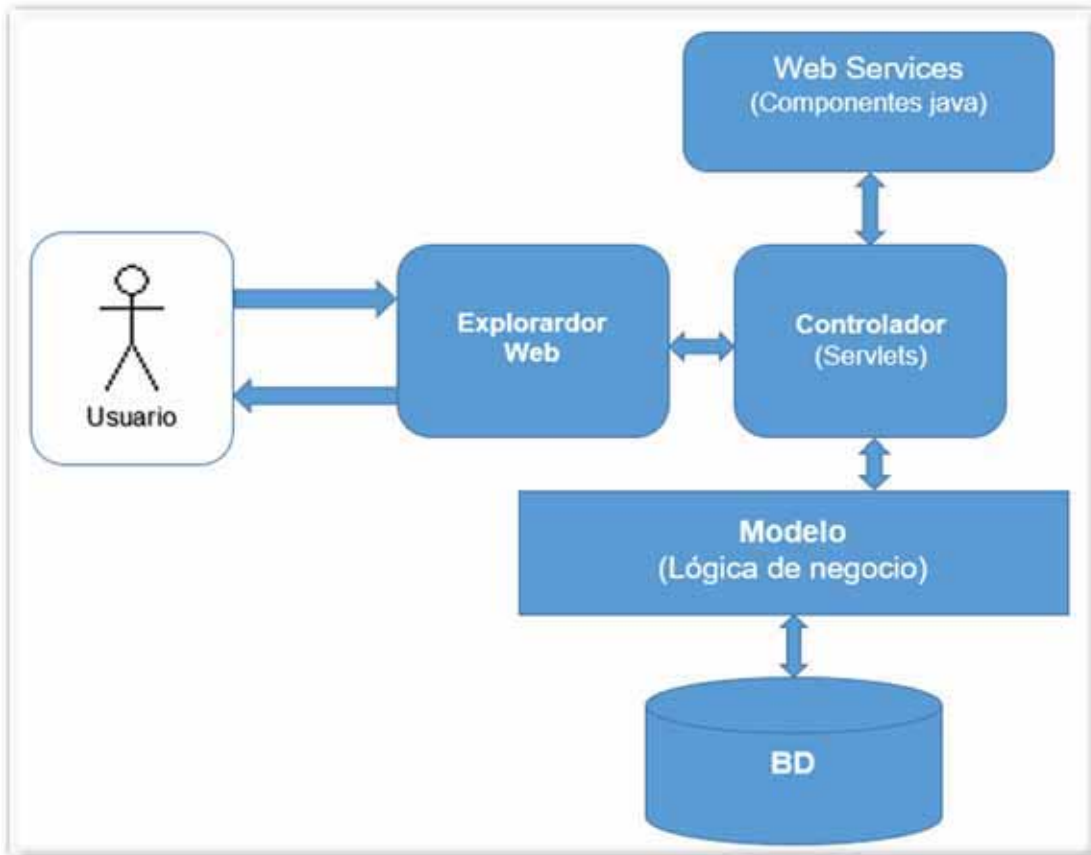


Figura 3. Arquitectura general del sistema

Desarrollo del proyecto

Requerimientos

A continuación, se da una descripción de las principales funciones que realiza el sistema de información.

Módulo de registro

Este módulo se encarga principalmente del registro en el sistema de los profesores que serán posteriormente evaluados.

Módulo mostrar

Este módulo permite al usuario listar todos los profesores registrados en la base de datos, además de consultar las evaluaciones y estadísticas de cada uno de los profesores listados siempre y cuando el número de evaluaciones sea mayor a 0.

Modulo evaluar

Este módulo es el más importante del sistema, ya que permita al usuario evaluar a un profesor previamente selección de la lista de profesores disponibles.

Módulo de las estadísticas

La única función de este módulo es la de realizar los cálculos necesarios para generar las evaluaciones y estadísticas de cada profesor.

Diseño

Esta etapa consta de los diagramas de casos de uso que componen el Sistema, así como de los actores que participan en el mismo.

Actores del sistema

Los actores del Sistema de Información son los siguientes:

Actor	Usuario
Casos de uso	Registrar Profesor, Mostrar Profesor registrados, Evaluar Profesor, Ver Evaluaciones, Buscar Profesor.
Tipo	Primario
Descripción	Es el actor principal y representa a cualquier alumno que utiliza el sistema para evaluar el desempeño de un profesor.

Actor	Base de datos
Casos de uso	Registrar Profesor, Evaluar Profesor, Buscar Profesor, Mostrar Profesor.
Tipo	Secundario
Descripción	Es el actor secundario y representa la base de datos donde se guarda la información relacionada a los profesores y su evaluación.

La Figura 4, muestra el diagrama de caso de uso general señalando las acciones que el sistema es capaz de realizar, a saber: registrar profesor, listar profesores registrados, buscar profesor, así como evaluar a un profesor y mostrar las evaluaciones de un profesor.

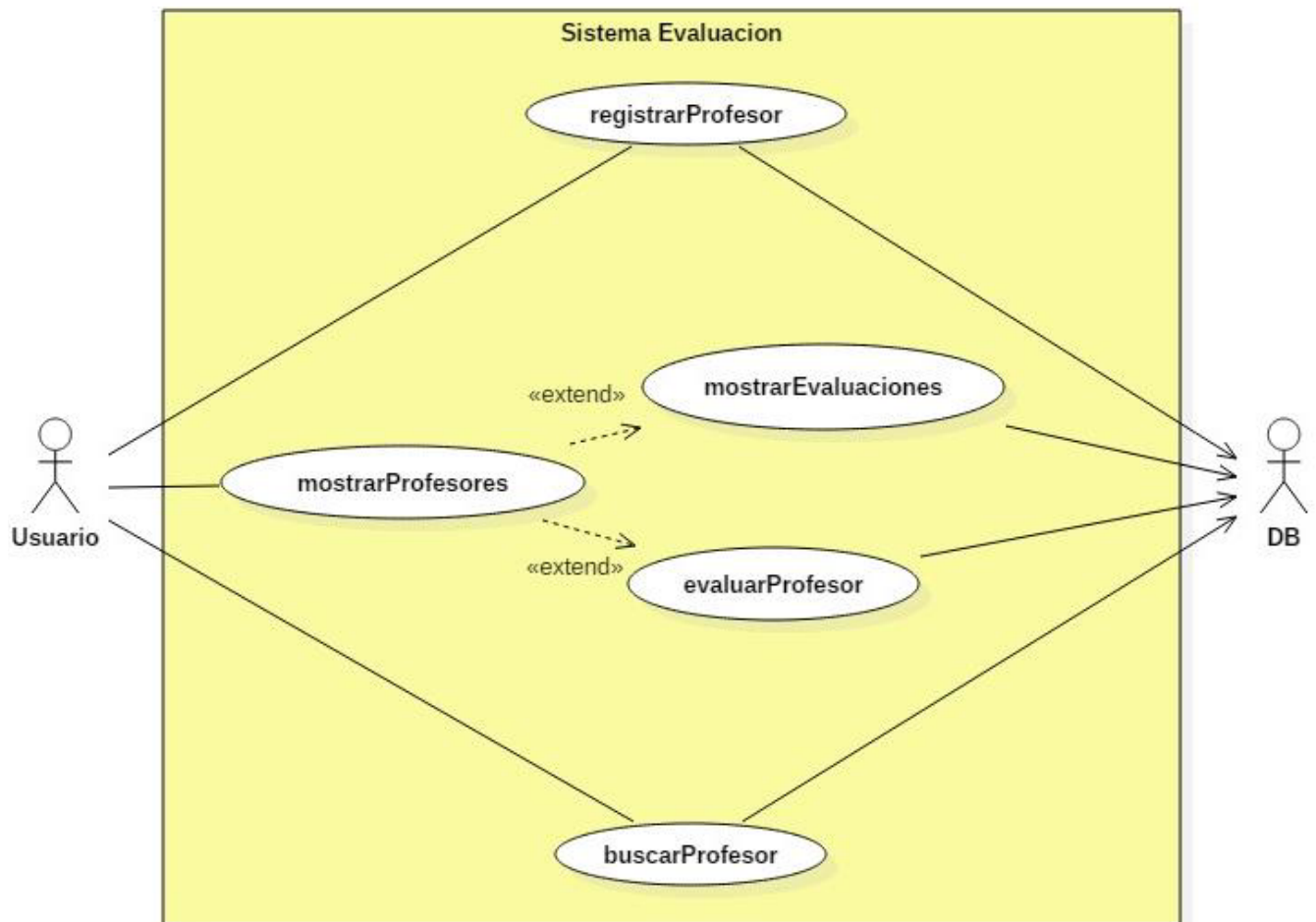


Figura 4. Diagrama de casos de uso general.

Casos de uso

El usuario tendrá la facultad de registrar un profesor en caso de que no exista, así como consultar los profesores registrados, evaluar a un profesor, consultar las evaluaciones y estadísticas de un profesor y finalmente buscar a un determinado profesor.

Caso de uso: Registrar profesor

Casos de uso	Registrar Profesor
Actores	Usuario, Base de Datos
Descripción	El usuario tendrá la posibilidad de registrar un profesor en caso de que no exista en la lista.

Caso de uso: Mostrar profesor

Casos de uso	Mostrar Profesor
Actores	Usuario, Base de Datos
Descripción	El usuario podrá consultar los profesores registrados. En esta opción también podrá evaluar o consultar las evaluaciones de algún profesor.

Caso de uso: Buscar profesor

Casos de uso	Buscar Profesor
Actores	Usuario, Base de Datos.
Descripción	El usuario podrá realizar una consulta para saber si determinado profesor ya se encuentra registrado, en caso contrario podrá registrar a dicho profesor.

Diagramas de secuencia

A continuación, se explican los diagramas de secuencia correspondientes a los casos de uso descritos anteriormente.

La Figura 5, muestra la operación registrar profesor por parte del usuario, al cual se le mostrara una pantalla que solicitara los datos del profesor para ser registrado en el sistema.

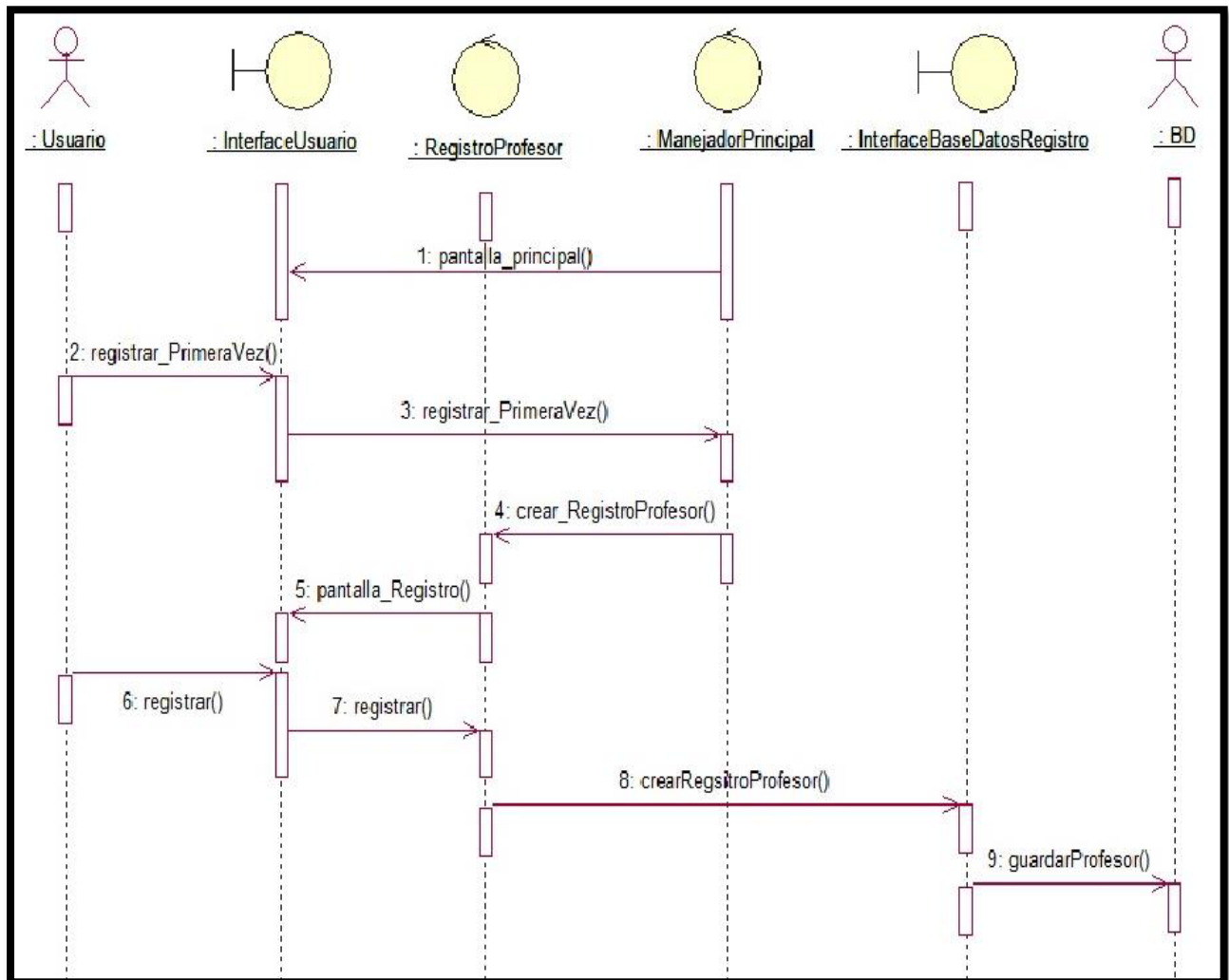


Figura 5. Diagrama de secuencia Registrar Profesor.

La Figura 6, muestra el diagrama de secuencia Evaluar Profesor, el usuario podrá seleccionar a uno de los profesores registrados en el sistema para asignarle una evaluación.

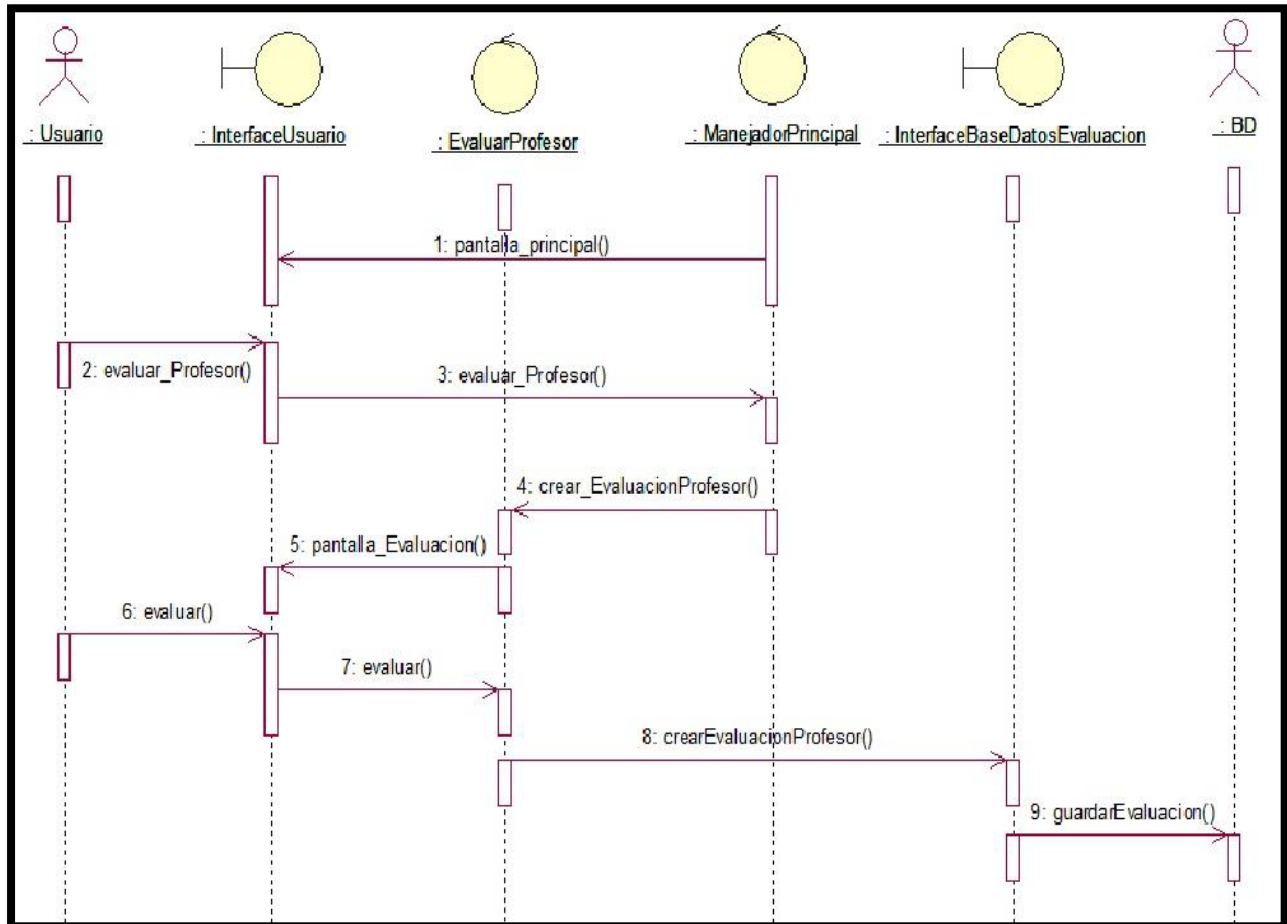


Figura 6. Diagrama de secuencia Evaluar Profesor.

Base de datos

La Figura 7, muestra el diagrama entidad relación de la base de datos que se utiliza para almacenar la información relacionada del sistema de información. La figura muestra las tablas que se requieren para almacenar la información, así con los campos de cada tabla en donde se almacenan tanto los datos del profesor como la evaluación que le corresponde a cada uno de los profesores.

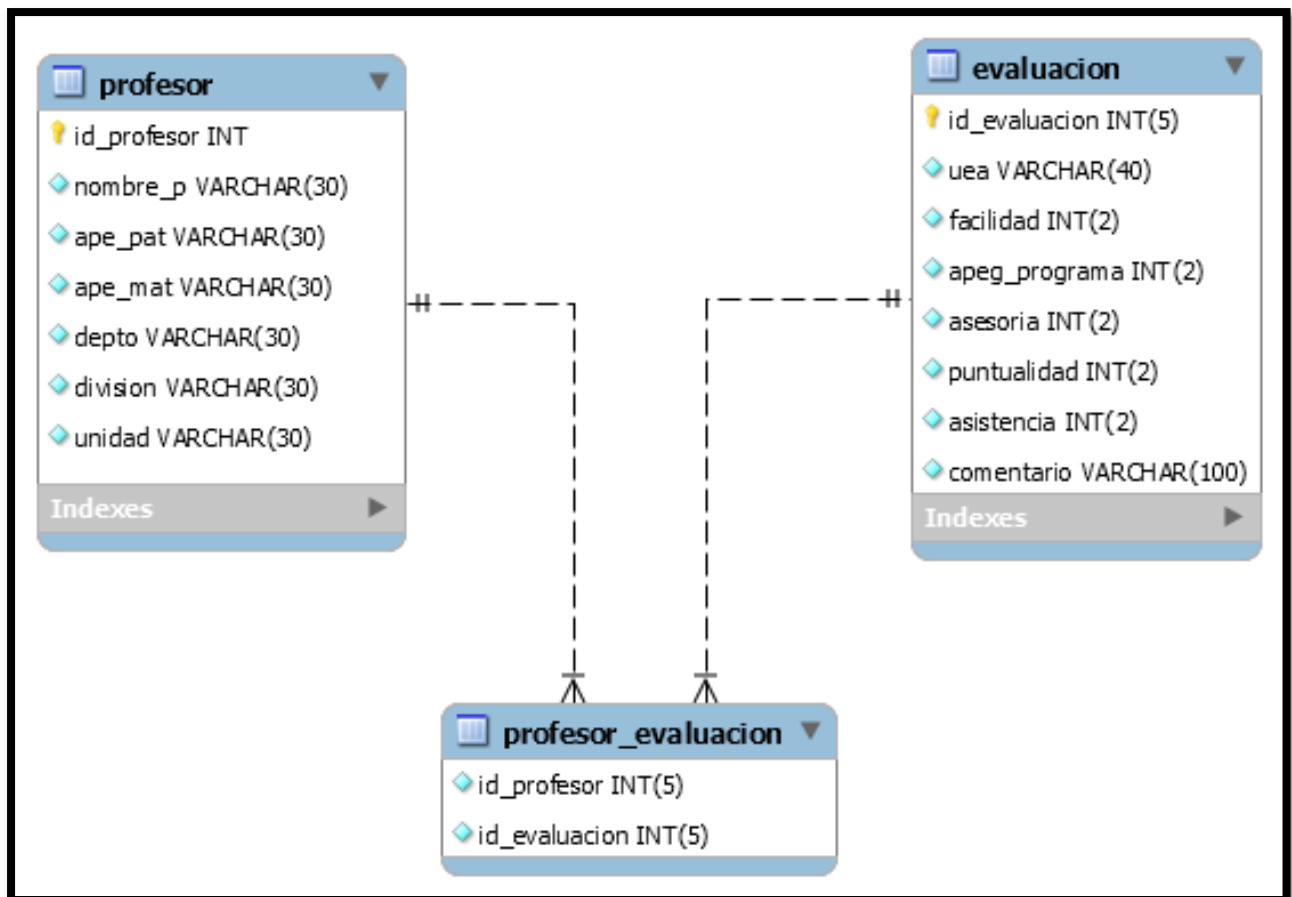


Figura 7. Diagrama Relacional de la Base de Datos.

Implementación

Para la implementación del sistema se utilizó el lenguaje de programación java EE, el cual nos permite escribir aplicaciones distribuidas basadas en componentes. Un componente J2EE es una unidad de software independiente que es ensamblado en la aplicación junto con sus ficheros y clases. En nuestro caso solo utilizamos los componentes Servlets y JSP.

El sistema tiene un paquete base mostrado en la Figura 8, el cual se compone de:

1. El dominio del servidor donde residirá la aplicación, por ejemplo: mx.uam.proyectoEvaluacion.
2. El nombre del proyecto, por ejemplo: proyectoEvaluación.

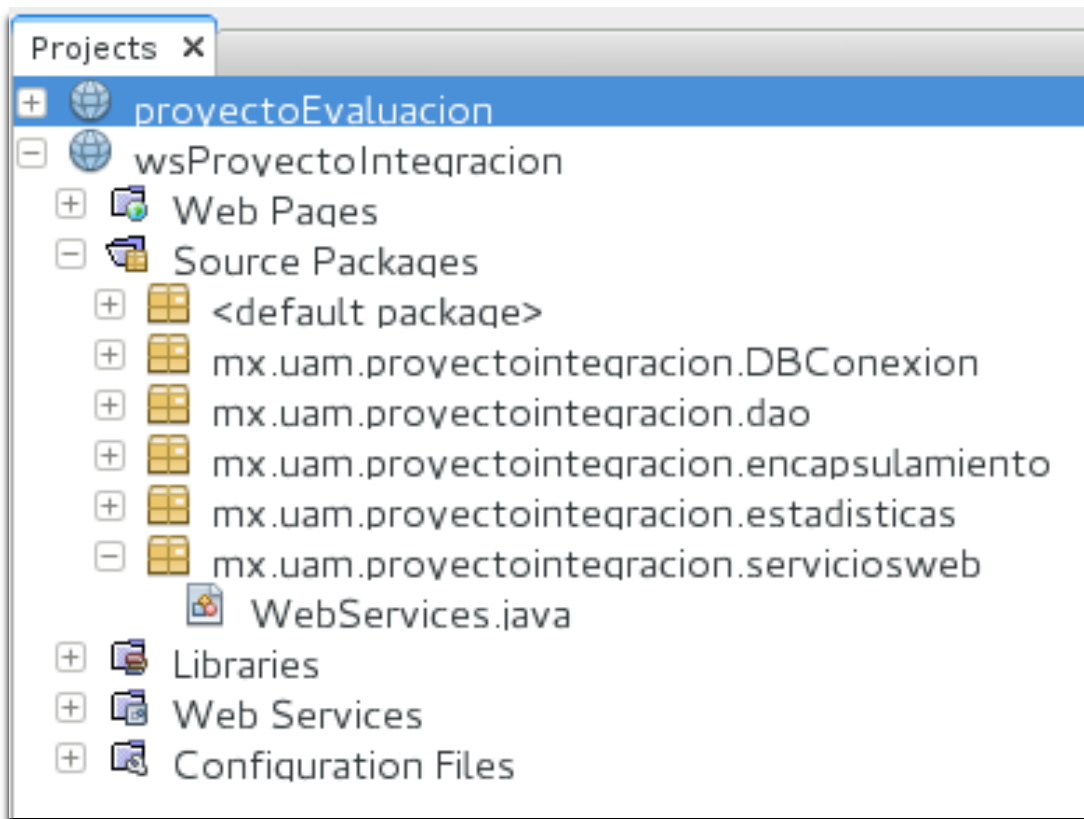


Figura 8. Esquema del paquete base del sistema.

Ejecución de la aplicación

En la Figura 9, se muestra la pantalla principal de la aplicación, desde la cual se pueden realizar distintas operaciones dependiendo de la acción seleccionada.

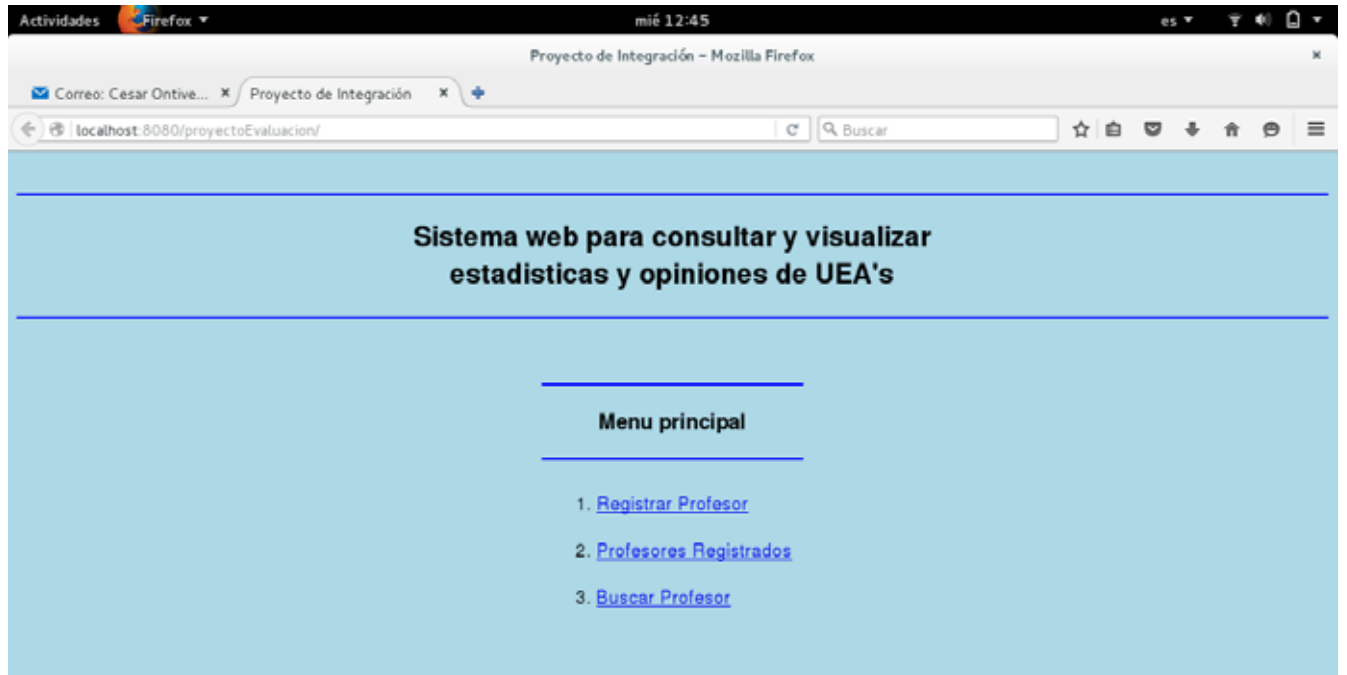
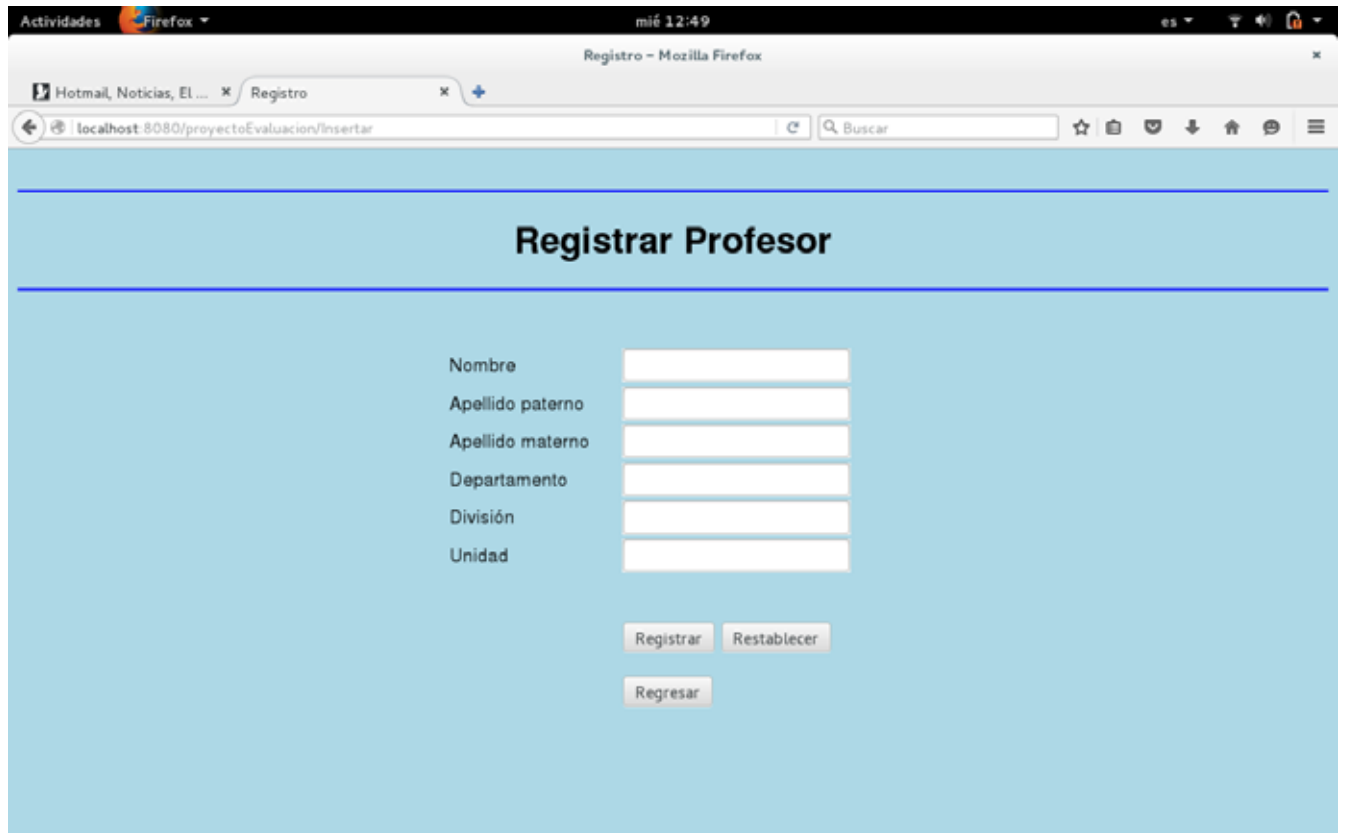


Figura 9. Página principal de la aplicación.

En la Figura 10, se muestra la pantalla que le permite al usuario registrar un nuevo profesor, después de seleccionar la primera opción del menú principal.



The screenshot shows a web browser window with the title "Registro - Mozilla Firefox". The address bar displays "localhost:8080/proyectoEvaluacion/insertar". The page content features a light blue background with a dark blue horizontal line at the top. Below this line, the heading "Registrar Profesor" is centered in a bold, black font. Another dark blue horizontal line is positioned below the heading. The registration form consists of six text input fields, each preceded by a label: "Nombre", "Apellido paterno", "Apellido materno", "Departamento", "División", and "Unidad". Below the input fields, there are three buttons: "Registrar" and "Restablecer" are positioned side-by-side, and "Regresar" is centered below them.

Figura 10. Registrar profesor.

La Figura 11, muestra la pantalla que se despliega cuando se seleccón la segunda opción del menú principal la cual muestra a todos los profesores registrados en el sistema. Desde esta misma pantalla el usuario tiene la posibilidad de seleccionar una de las dos opciones que se le presentan para cada profesor registrado, es decir, podrá evaluar a un profesor y/o consultar las evaluaciones de algún profesor.

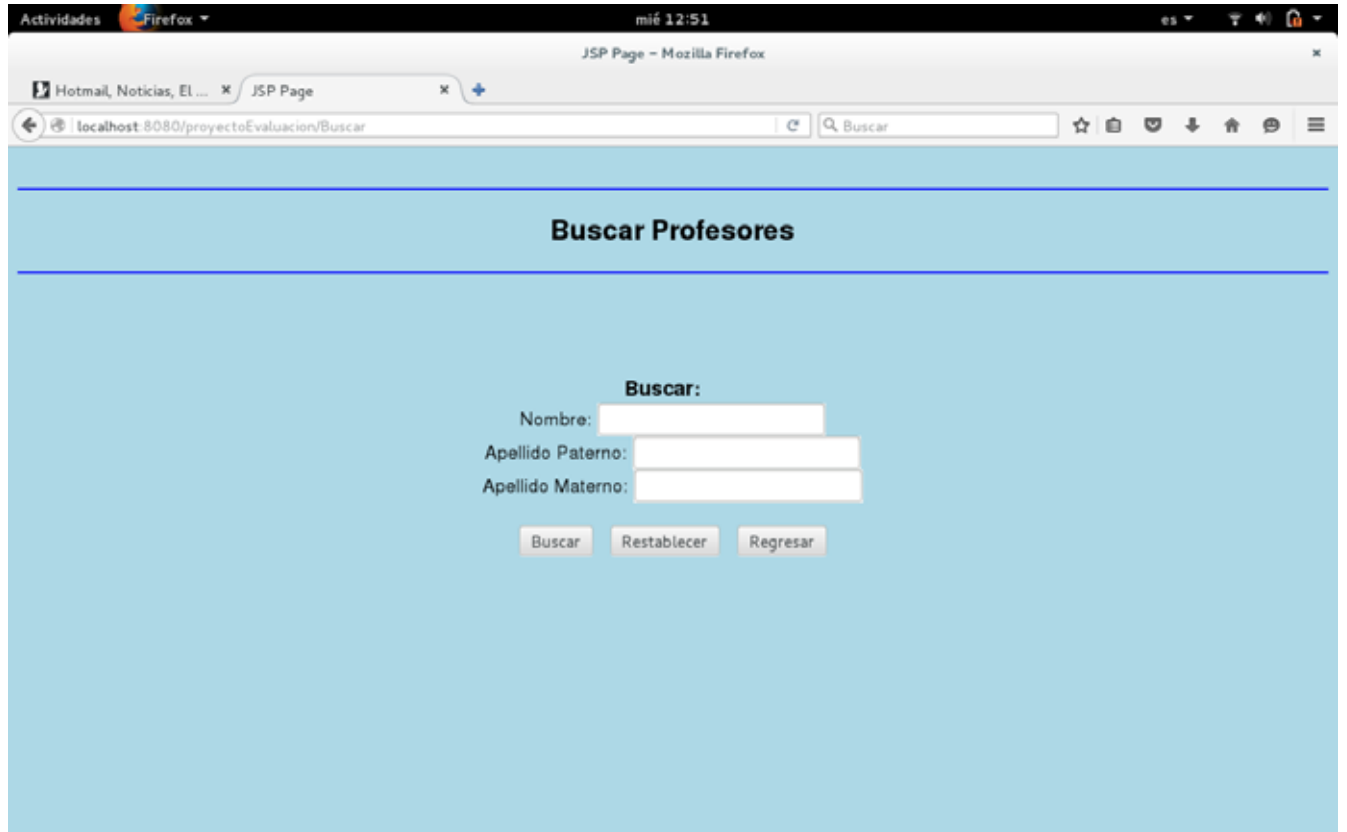
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/proyectoEvaluacion/Mostrar'. The page title is 'Profesores Registrados'. Below the title is a table with 9 columns: NUMERO, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, DEPARTAMENTO, DIVISION, UNIDAD, and two empty columns. The table contains 6 rows of data. Below the table is a 'Regresar' button.

NUMERO	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	DEPARTAMENTO	DIVISION	UNIDAD		
1	Diego	Jimenez	Alonso	Ciencias Basicas	CBI	Azcapotzalco	Evaluar	Evaluaciones
2	Sonia	Blanquez	Blanco	Sistemas	CBI	Azcapotzalco	Evaluar	Evaluaciones
3	Juan	Ferrara	Perales	Economia	CSH	Azcapotzalco	Evaluar	Evaluaciones
4	JosA©	Barriga	Ajenjo	Energia	CyAD	Azcapotzalco	Evaluar	Evaluaciones
5	Pablo	Cruz	Alcantarilla	Derecho	CSH	Azcapotzalco	Evaluar	Evaluaciones
6	Maricela	Bravo	Contreras	Sistemas	CBI	Azcapotzalco	Evaluar	Evaluaciones

[Regresar](#)

Figura 11. Mostrar profesores registrados.

Al ser seleccionada la opción buscar profesor del menú principal se muestra la Figura 12, que le permite al usuario buscar a un determinado profesor.



The image shows a screenshot of a web browser window. The browser is Mozilla Firefox, and the page title is "JSP Page - Mozilla Firefox". The address bar shows the URL "localhost:8080/proyectoEvaluacion/Buscar". The page content is a search form titled "Buscar Profesores". The form has three input fields: "Nombre:", "Apellido Paterno:", and "Apellido Materno:". Below the input fields are three buttons: "Buscar", "Restablecer", and "Regresar".

Figura 12. Buscar Profesor.

Conclusiones

Este sistema de evaluación estará habilitado, en primera instancia, para evaluar a profesores de los distintos departamentos que conforman la UAM-Azcapotzalco. Es decir, existe la posibilidad de extender dicho sistema para evaluar a los profesores de otras unidades académicas, más aún de otras universidades.

La siguiente tabla muestra los objetivos específicos, así como el alcance que se obtuvo con el desarrollo de este proyecto terminal.

Objetivos	Descripción
Diseñar e implementar la base de datos para almacenar la información utilizando un sistema gestor de bases de datos relacional.	Se cumplió con el diseño de la base de datos utilizando el modelo relacional.
Diseñar e implementar los componentes Java que realizaran la lógica de la aplicación.	Los componentes que realizaron la lógica del sistema fueron diseñados e implementados satisfactoriamente, ubicando a cada uno en su respectivo caso de uso y diagrama de secuencia.
Modelar y programar los componentes de software para generar los Servicios Web.	Los servicios web se desarrollaron correctamente.
Diseñar e implementar la interface de usuario que nos permita visualizar los resultados.	Se cumplieron los objetivos, tanto en el diseño como en la implementación de la base de datos, esta última utilizando la tecnología JSP.

Tabla 1. Objetivos alcanzados.

La realización de este proyecto de integración me permitió reafirmar mis conocimientos sobre el lenguaje de programación Java, así como adquirir conocimiento sobre otras tecnologías basadas en el mismo lenguaje de programación como son Servlet y Jps's, así como el uso de los Servicios Web.

El avance de este proyecto fue sustancial; sin embargo, se puede agregar otros módulos al sistema para hacerlo más íntegro y funcional.

Referencias bibliográficas

- [1] Alonso, G., Casati, F., Kuno, H. y Machiraju, V., *Web Services. Concepts, Architectures and Applications*, Carey, M.J. y Ceri, S. (Eds.). 2004, Berlin: Springer. Num. Páginas: 354.
- [2] Ceballos, Francisco Javier, *Java 2 Interfaces gráficas y aplicaciones para internet*, 3ª Ed. México, DF, Alfaomega, 2008.
- [3] Martin Sierra, Antonio. *Struts*, 1ª Ed. México, DF, Alfaomega, 2008.
- [4] Campderrich Falgueras, Benet, *Ingeniería del software*, 1a ed. Barcelona, ES:UOC, 2003.
- [5] J. X. Garcia Zavala, D. Martinez Vazquez y D. J. M. Rivera Corona. “*Desarrollo de un directorio usando un servicio web*”, tesis, Instituto Politécnico Nacional, México, 2009.
- [6] W3E. *Web Services Architecture*. (30, agosto, 2016) [En línea] Disponible en: <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>
- [7] Grady Booch, James Rumbaugh, Ivar Jacoboson. *El lenguaje unificado de modelado*, Editorial: Addison-Wesley, Madrid.
- [8] Weitzenfeld, Alfredo, *Ingeniería de software orientada a objetos con UML, Java e Internet*, Editorial: Thomson, 2005.