



Universidad Autónoma del Estado de México
Centro Universitario UAEM Zumpango

"Sistema Informático de Encuestas a los Alumnos Aceptados
del PE de Contaduría del Centro Universitario UAEM
Zumpango"

TESINA

Que para obtener el título de:

INGENIERO EN COMPUTACIÓN

presenta:

Christian Abraham Flores Lara

Asesor:

M. en C. Asdrúbal López Chau

Junio, 2013

Dedicatorias

A mis padres

Este trabajo está elaborado gracias al apoyo y oportunidad que me brindaron mis padres en todo este trayecto, no hay forma de expresar lo que siento con palabras, sin embargo, les doy un agradecimiento muy grande y espero se sientan orgullosos por lo que he logrado hasta el momento, con esto me comprometo a ser una persona de bien y ser mejor cada día. Estoy consciente de lo que ustedes tuvieron que pasar en su vida, no tuvieron las mismas oportunidades ni comodidades que a mí me brindaron, y aun así, con todas esas carencias que tuvieron que pasar gracias a su esfuerzo nunca me faltó el apoyo y la motivación para que tuviera una profesión, son un ejemplo para mí, muchas gracias.

A la Universidad Autónoma del Estado de México

Un agradecimiento a la Universidad Autónoma del Estado de México por haberme aceptado y darme la oportunidad de cursar la carrera de Ingeniero en Computación. Es un orgullo decir que soy egresado de ella.

A mis profesores

Un agradecimiento a los profesores quienes me compartieron sus conocimientos, experiencias y motivación de seguir aprendiendo. Personalmente creo que no es fácil ser un docente, y mucho menos un docente con vocación y gusto de enseñar, les pido sigan preparándose y nunca pierdan el gusto y motivación de enseñar y transmitir sus conocimientos, muchos estamos y estarán agradecidos de ustedes y serán recordados con respeto y admiración.

A mis amigos

No podía faltar el agradecimiento especial a mis amigos con quienes compartí muchas experiencias en esta etapa ya concluida, gracias por el apoyo, por convivir todo este tiempo, compartimos buenos y malos momentos, aprendimos muchas cosas que nos hizo crecer como personas y profesionistas, cuando surgieron obstáculos siempre nos brindamos el apoyo, algunos ya nos veremos como antes, sin embargo, les deseo lo mejor, que construyan y alcancen sus metas.

Resumen

Actualmente las aplicaciones Web son comunes en diversas áreas, no es difícil encontrarse con aplicaciones Web de contabilidad, de ventas o educativas. En este trabajo se desarrolló un Sistema de Información Web para Aplicación de Encuestas (SIWAE) para estudiantes de Licenciatura en Contabilidad del Centro Universitario UAEM Zumpango. El proyecto se derivó de la necesidad de contar con un sistema capaz de almacenar y analizar datos acerca de alumnos de primer año de Contabilidad. Antes de este proyecto los cuestionarios eran manualmente aplicados usando formatos impresos y las respuestas eran capturadas después en hojas de cálculo electrónicas. Es bien sabido que los procesos realizados manualmente no son recomendados especialmente si la cantidad de información crece rápidamente. La mayor contribución de este trabajo es el desarrollo de SIWAE el cual automatiza la tarea de capturar respuestas, ejecuta validaciones sobre ellas y es capaz de generar reportes básicos que pueden ayudar a analizar los datos en la base de datos del sistema. La aplicación ofrece algunas opciones para administración como respaldar la base de datos, modificar los datos de usuarios, cambiar permisos y eliminar usuarios. Permite también realizar algunos reportes y gráficas de los resultados de las encuestas.

Abstract

Web applications are common in many areas, it is not difficult to run into accounting, sales or educational applications that are Web ones. In this work a Web Information System for Application of Questionnaires (SIWAE for its Spanish Acronym) to bachelor accounting students of Centro Universitario UAEM Zumpango was developed. The project was derived from the need for a system able to store and also analyze the data about freshman bachelor accounting students. Before this project, the questionnaires were manually applied using printed forms and then the answers were captured in electronic spreadsheets. It is well known that manual processes are not recommended specially if the amount of information rampantly grows. The main contribution of this work is the development of SIWEA, which automates the task of capture of answers, executes some validations on them and is able to generate basic reports that can help to analyze the data stored in the data base of the system, the developed application also generates reports and graphs of results of surveys.

Índice general

Índice general	IX
Índice de figuras	XII
1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Justificación	1
1.3. Objetivo general	2
1.3.1. Objetivos específicos	2
1.4. Alcance del trabajo	2
1.5. Estructura de la tesina	3
2. Fundamentos Tecnológicos	5
2.1. World Wide Web	5
2.1.1. Funcionamiento de World Wide Web	5
2.1.2. Aplicación Web	6
2.2. Framework Web	7
2.2.1. Struts	7
2.2.2. Tapestry	8
2.2.3. Java Server Faces	8
2.2.4. ASP.NET	9
2.2.5. Ruby on Rails	9
2.3. Patrón de Diseño MVC	9
2.4. Base de Datos	10
2.4.1. Gestor de Base de Datos	10
2.4.2. PostgreSQL	10
2.4.3. FireBird	11
2.4.4. SQLite	11
2.4.5. Oracle	12
2.4.6. MySQL (Community Server)	12
2.5. Servidores <i>Web</i>	13
2.5.1. Apache Tomcat	13
2.5.2. IIS (Internet Information Services)	13
2.5.3. GlashFish	13
2.5.4. WAS (Websphere)	14
2.6. UML	14
2.6.1. Diagramas de comportamiento	14
2.6.2. Diagramas de Estructura	15

2.7.	Javascript	16
2.8.	JQuery	16
2.9.	JQueryUI	16
2.10.	CSS	16
2.11.	IDE	17
2.11.1.	Netbeans	17
2.11.2.	Eclipse	17
2.12.	Java	18
2.12.1.	JPA	18
2.12.2.	Tiles	18
3.	Análisis y Diseño del Sistema	19
3.1.	Análisis del Sistema	19
3.1.1.	Casos de uso del Sistema	20
3.2.	Diseño del sistema	22
3.2.1.	Diagramas de interacción	23
3.2.2.	Diagrama de actividades	27
4.	Desarrollo del Sistema	33
4.1.	Arquitectura Modelo-Vista-Controlador	33
4.2.	Descripción del Sistema	34
4.2.1.	Modelo	34
4.3.	Descripción de la Vista	43
4.3.1.	Vista - Result	44
4.3.2.	Etiquetas Struts 2	44
4.3.3.	Tiles	45
4.3.4.	Validación	46
4.3.5.	JQuery	47
4.3.6.	CSS	49
4.3.7.	Paginación	50
4.3.8.	Reportes (Ireport y Jasper Reports)	51
4.4.	Descripción del Controlador	53
4.4.1.	Struts 2	53
4.4.2.	Sesión	54
4.4.3.	Interceptores	55
4.4.4.	Subir archivos al servidor	56
4.4.5.	Internacionalización y localización (i18n)	57
4.4.6.	Log4j	58
5.	Pruebas realizadas al SIWAE	60
6.	Conclusiones, Recomendaciones y Trabajo Futuro	68
6.1.	Conclusiones	68
6.2.	Recomendaciones	69
6.3.	Trabajo futuro	70
	Bibliography	71
	Anexo	73

A. Glosario	73
B. Manual de Usuario	77
B.1. Registro de un usuario	77
B.2. Autenticación en el sistema	81
B.3. Cerrar Sesión	83
B.4. Modificar mis datos personales	83
B.5. Contestar Encuesta	84
C. Manual del administrador	86
C.1. Buscar Usuario	86
C.2. Modificar datos de los usuarios	87
C.3. Activar usuarios	88
C.4. Ver usuarios	89
C.5. Eliminar usuarios	90
C.6. Modificar permisos usuarios	91
C.7. Enviar correo electrónico	92
C.8. Enviar correo electrónico individual	93
C.9. Enviar correo electrónico a todos	94
C.10. Resultados encuestas	95
C.11. Generar reportes	97
C.12. Crear respaldo	98
C.13. Ver respaldos	100
C.14. Cargar o subir respaldo	100
C.15. Descargar respaldo	101
C.16. Restaurar respaldos	103
C.17. Eliminar respaldos	104
C.18. Eliminar Datos	105
C.19. Configurar sistema	106
C.20. Agregar licenciatura, generación, semestre, grupo y escuelas	107
C.21. Modificar licenciatura, generación, semestre, grupo y escuelas	108
C.22. Eliminar licenciatura, generación, semestre, grupo y escuelas	109
C.23. Configurar servidor	111

Índice de figuras

2.1. Modelo Cliente-Servidor	6
2.2. MVC	10
3.1. Diagrama de dominio para el SIWAE	20
3.2. Casos de uso del SIWAE	21
3.3. Caso de Uso - Registrarse en el sistema	21
3.4. Caso de uso - Ingresar al sistema	22
3.5. Caso de uso - Contestar encuesta	22
3.6. Diagrama de interacción “Registrar usuario”	23
3.7. Diagrama de interacción “Ingresar al sistema”	23
3.8. Diagrama de interacción “Modificar mis datos”	24
3.9. Diagrama de interacción “Buscar usuarios”	24
3.10. Diagrama de interacción “Activar usuario”	25
3.11. Diagrama de interacción “Configurar servidor”	25
3.12. Diagrama de interacción “Eliminar usuario”	26
3.13. Diagrama de interacción “Enviar email”	26
3.14. Diagrama de interacción “Modificar permisos”	27
3.15. Diagrama de interacción “Cerrar sesión”	27
3.16. Diagrama de actividad “Registro de usuario”	28
3.17. Diagrama de actividad “Autenticación del usuario”	28
3.18. Diagrama de actividad “Modificar datos”	29
3.19. Diagrama de actividad “Eliminar usuarios”	29
3.20. Diagrama de actividad “Modificar permisos”	30
3.21. Diagrama de actividad “Enviar email”	30
3.22. Diagrama de actividad “Contestar encuesta”	31
4.1. Arquitectura Modelo-Vista-Controlador	34
4.2. Representación de la capa modelo	34
4.3. Diagrama de clases lógica de negocio	35
4.4. Clases Action para las actividades de registro activación de cuenta, acceso, salir, cancelar, buscar usuario y permisos	36
4.5. Clases Action para las encuestas socio económica, valores, aptitudes y afirmaciones	36
4.6. Clases Action para respaldo de datos, edición y reportes	37
4.7. Clases Action para configurar al servidor y al sistema, enviar email y eliminar usuarios	38
4.8. Mapeo con <i>Anotaciones</i> de una tabla con <i>JPA</i>	39
4.9. Archivo de configuración <i>JPA</i> con EclipseLink (<i>persistence.xml</i>)	39
4.11. Métodos <code>get()</code> y <code>set()</code> para interfaz con <i>JPA</i>	40

4.10. Consulta con lenguaje <i>JPQL</i>	40
4.12. Persistencia de un objeto Usuarios	41
4.13. Diagrama de clases lógica de datos	41
4.14. Base de datos mapeada con JPA parte 1	42
4.15. Base de datos mapeada con JPA parte 2	43
4.16. Clase para manipular la base de datos con JPA	43
4.17. etiquetas de <i>Struts 2</i> y equivalencia en <i>HTML</i>	45
4.18. Instalación y configuración del <i>Framework Tiles</i> para <i>Struts 2</i>	45
4.19. Plantilla de <i>Tiles</i>	45
4.20. Plantilla de <i>Tiles</i> en una <i>JSP</i>	46
4.21. Archivos necesarios para ejecutar <i>JQuery</i>	47
4.22. Tooltip con <i>JQuery</i>	47
4.23. Calendario con <i>JQuery (Datepicker)</i> utilizado en la aplicación	48
4.24. Checkbox y Radiobuttons con <i>Ezmark</i>	48
4.25. Checkbox y Radiobuttons sin <i>Ezmark</i>	49
4.26. Editor de texto enriquecido	49
4.27. Ejemplo de código en <i>CSS</i>	50
4.28. Tabla con <i>JMesa</i>	51
4.29. Funciones <i>Javascript</i> de <i>JMesa</i>	51
4.30. Etiquetas de <i>JMesa</i>	51
4.31. Plantilla <i>Ireport</i>	52
4.32. Código <i>XML</i> de una platilla <i>Ireport</i>	52
4.33. Compilación de un archivo <i>.jrxml</i>	53
4.34. Gráficas de pastel en <i>Ireport</i>	53
4.35. Filtro de <i>Struts 2</i>	54
4.36. Sesión en el archivo <i>Web.xml</i>	54
4.37. Sesión y <i>Action LoginAction</i>	55
4.38. Interceptor <i>AutenticacionInterceptor</i>	56
4.39. Propiedad <i>enctype</i>	56
4.40. Tamaño máximo del archivo a subir en el servidor	57
4.41. Archivo <i>.properties</i> de la aplicación	57
4.42. Archivos <i>.properties</i>	58
4.43. Configuración <i>Log4j</i>	58
4.44. Código para utilizar <i>log4j</i>	59
5.1. Validaciones del formulario “Registro”	61
5.2. Validaciones del formulario “Modificar mis datos”	62
5.3. Expresión regular <i>Framework</i> validación	62
5.4. Caracteres especiales para el <i>Framework</i> de validación	63
5.5. Archivo de configuración MySQL	63
5.6. Codificación ISO-8859-1	64
5.7. Codificación en MySQL	64
5.8. Usuarios registrados	64
5.9. Activación de cuenta por usuarios	65
5.10. Usuarios que contestaron la encuesta valores	66
5.11. Gráficas obtenidas de la encuesta valores	66
5.12. Usuarios que no terminaron la encuesta valores	66

B.1. Registro página principal	77
B.2. Datos requeridos del nuevo usuario	78
B.3. Confirmación del password	78
B.4. Finalizar el registro del nuevo usuario	79
B.5. Cuenta ya registrada	79
B.6. Página de confirmación del registro	80
B.7. Página activación de la cuenta	80
B.8. Cuenta no activada	81
B.9. Iniciar Sesión	81
B.10. Error inicio de sesión	82
B.11. Página principal al autenticarse en el sistema	82
B.12. Cierre de sesión	83
B.13. Modificar mis datos personales	84
B.14. Contestar Encuesta	85
C.1. Buscar usuario	86
C.2. Resultados de búsqueda en una tabla	87
C.3. Modificar los datos de un usuario	87
C.4. Modificar los datos de otros usuarios	88
C.5. Activar usuarios	89
C.6. Tabla ver usuarios	89
C.7. Tabla eliminar usuarios	90
C.8. Confirmar eliminar usuario	90
C.9. Tabla actualizada al eliminar un usuario	91
C.10. Tabla modificar permisos a los usuarios	91
C.11. Permisos del usuario	92
C.12. Enviar correo electrónico	92
C.13. Buscar usuario para enviar correo electrónico	93
C.14. Asunto y mensaje del correo electrónico individual	94
C.15. Asunto y mensaje del correo electrónico dirigido a todos	95
C.16. Resultados encuesta	95
C.17. Selección de una opción Resultados encuesta	96
C.18. Archivo PDF con los resultados de la encuesta según usuario	96
C.19. Encuesta no contestada	96
C.20. Generar reportes	97
C.21. Reporte en formato PDF	98
C.22. Gráficas de los reportes en PDF	98
C.23. Respaldo datos	99
C.24. Respaldo creado correctamente	99
C.25. Ver respaldos	100
C.26. Tabla respaldos	100
C.27. Cargar un respaldo al sistema	101
C.28. Archivo subido correctamente	101
C.29. Tabla respaldos	102
C.30. Descargar respaldo	102
C.31. Archivo no disponible	103
C.32. Confirmación de restaurar un respaldo	103
C.33. Restauración del respaldo correctamente	104

C.34.Confirmar eliminar respaldo	104
C.35.Eliminación del respaldo correctamente	105
C.36.Eliminar datos	105
C.37.Confirmación de eliminación de los datos	106
C.38.Pestaña licenciaturas	106
C.39.Agregar licenciatura	107
C.40.Atributo ya registrado en el sistema	108
C.41.Modificar el nombre la licenciatura	109
C.42.Atributo ya registrado en el sistema	109
C.43.Eliminar licenciatura	110
C.44.Eliminar todas las licenciaturas	110
C.45.Configurar el servidor	111
C.46.Cambios guardados correctamente en el servidor	112

Capítulo 1

Introducción

1.1. Planteamiento del problema

En el programa educativo de la Licenciatura en contabilidad del Centro Universitario UAEM Zumpango, se tiene la necesidad de contar con un sistema capaz de almacenar y facilitar el análisis de los datos acerca de alumnos de primer año de Contabilidad con la finalidad mantener una mejora continua en dicho programa educativo. Para ello se aplican cuestionarios o encuestas a los alumnos y posteriormente se analiza la información. En general el proceso de recolección de datos por medio de encuestas impresas lleva implícito el riesgo de que la información sea proporcionada de manera incorrecta o incompleta. Por otra parte, el análisis de los datos obtenidos por la encuestas resulta una tarea laboriosa en la mayor parte de los casos, sobre todo si existen errores de capura.

Este trabajo se enfoca disminuir el riesgo de que haya omisión de datos o errores de llenado en las encuestas contestadas por los alumnos de primer año del programa educativo mencionado anteriormente.

1.2. Justificación

El uso de tecnologías informáticas para la aplicación de encuestas o cuestionarios es común en estos días. La razón de su éxito radica en varios aspectos como su bajo costo de aplicación, facilidad para procesar la información recopilada, detección y corrección automática de errores y capacidad para llegar a regiones geográficamente distantes de manera simultánea. Otros aspectos operativos que deben ser tomados en cuenta para decidir usar tecnologías informáticas en la aplicación de cuestionarios son:

1. Tamaño de la muestra. Si el número de cuestionarios a aplicar es grande o si se requiere aplicar varios cuestionarios o entrevistas a un mismo grupo de personas, entonces el usar sistemas de información basados en *Web* es una opción viable.

2. Respaldo de información. Los sistemas de información modernos permiten almacenar la información de manera segura y su recuperación puede hacerse de manera remota.

3. Disponibilidad. La información generada por la aplicación de los cuestionarios o entre-

vistas por lo general es utilizada de manera diferente dependiendo de los usuarios, el acceso a la información puede ser simultáneo o en tiempo real.

La aplicación de cuestionarios o entrevistas utilizando material impreso no proporciona estas ventajas. Con base en lo expuesto, las ventajas de automatizar los instrumentos aplicados a los alumnos son evidentes, por lo que el desarrollo de un sistema informático que automatice la aplicación de tales instrumentos apoyará de manera ingente al proyecto "Análisis de los indicadores de ingreso de los alumnos aceptados al Programa Educativo de Contaduría en CU UAEM Zumpango".

1.3. Objetivo general

Diseñar e implementar un sistema de información automatizado basado en *Web* para la captura, almacenamiento, procesamiento y generación de reportes de los instrumentos aplicados a los alumnos aceptados del Programa educativo de Contaduría del Centro Universitario UAEM Zumpango.

1.3.1. Objetivos específicos

- Diseñar la arquitectura del sistema de información *Web*.
- Diseñar la base de datos relacional para el almacenamiento de los datos de los alumnos.
- Diseñar las interfaces de usuario *Web* para la captura de los datos por parte de los alumnos.
- Diseñar las interfaces de usuario *Web* para la generación de reportes.
- Implementar el sistema de información basado en *Web* con los puntos descritos anteriormente, para ser utilizando en cualquier computadora conectada a la red local del Centro Universitario UAEM Zumpango.
- Realizar las pruebas correspondientes terminando la fase de construcción.

1.4. Alcance del trabajo

SIWAE es un sistema *Web* para aplicar encuestas a los estudiantes de nuevo ingreso de Contaduría del Centro Universitario UAEM Zumpango. El SIWAE clasifica a los usuarios en tres tipos: Estudiante, Profesor y Administrador. Cada usuario tiene restricciones de acuerdo al tipo asignado, por ejemplo los usuario tipo Estudiante no pueden generar reportes. Los Administradores pueden cambiar los datos del perfil de todos los usuarios y generar reportes de la encuestas valores. Otras actividades que pueden desarrollar los administradores es activar usuarios, restablecer contraseñas, y configurar el servidor. No es permitido al administrador ver las contraseñas de los usuarios, solo reemplazarlas por una nueva.

El SIWAE es capaz de generar reportes sobre la cantidad de usuarios que contestaron una encuesta y el número de usuarios que eligieron cada respuesta para el caso de preguntas con opción múltiple.

1.5. Estructura de la tesina

En este documento se compone de 6 capítulos incluyendo el presente. En el capítulo 2 se exponen los conceptos básicos referentes a la arquitectura cliente servidor, en la que se basan la mayoría de las aplicaciones *Web*. Se presentan las tecnologías relacionadas, entre las que se pueden mencionar *Frameworks Web*, Patrones de Diseño, Bases de datos, Servidores *Web* y Servidores de Aplicaciones, *UML*, tecnologías que se ejecutan en el lado del servidor como *Javascript* y *CSS*.

El Análisis y Diseño del SIWAE se muestran en el capítulo 3. En primer lugar se mencionan algunas técnicas que se ocupan para la identificación de los requerimientos, lo cual es útil para poder elaborar el modelo de datos que sirve para tener una visión general del proyecto. En ese capítulo se presentan diagramas de casos de uso que permiten definir procesos con más detalle.

En el capítulo 4 se presenta el desarrollo del SIWAE, se describe su arquitectura, las tecnologías y *Frameworks* ocupados. La arquitectura se divide en 3 capas: el modelo, la vista y el controlador. Donde se describe la implementación de cada una de las capas mencionadas.

Las pruebas que fueron realizadas al SIWAE se presentan en el capítulo 5, así como los errores identificados y las soluciones que se aplicaron. Contiene recomendaciones que son importantes a tomar en cuenta al momento de desarrollar una aplicación *Web*, tales como la validación de formularios, el registro de *log*, el manejo de sesión, subir archivos al servidor por mencionar algunos.

El Capítulo 6 contiene las conclusiones a las que se llegaron una vez construida y probada la aplicación. También se mencionan algunas recomendaciones referidas a las tecnologías utilizadas. Y el trabajo a futuro con lo cual se pretende mejorar y extender el SIWAE.

Para finalizar, el documento incluye un anexo en donde se agregaron los manuales de usuario y de administrador, un glosario con los términos más relevantes y las referencias.

Capítulo 2

Fundamentos Tecnológicos

2.1. World Wide Web

Este capítulo presenta algunas definiciones las cuales son necesarias para comprender el funcionamiento técnico del sistema. También se describen algunas herramientas que pueden ser utilizadas para el desarrollo de aplicaciones *Web* dando una referencia breve de cada una de ellas.

World Wide Web (*WWW*) es una parte de *Internet* la cual es una red de computadoras que permite a las personas comunicarse entre sí desde cualquier parte del mundo, se pueden destacar tecnologías de comunicación como el correo electrónico, clientes de mensajería instantánea, páginas y sitios *Web* (conjunto de páginas *Web*). En un principio las páginas *Web* eran muy sencillas y estaban basadas en *HTML* (Lenguaje de Marcas de Hipertexto). Las personas podían compartir cualquier tipo de información, pero los visitantes no podían interactuar con la página. *Web* evolucionó rápidamente permitiendo insertar imágenes, tablas y formularios, consiguiendo una comunicación con los sitios *Web*, aunque esto no generó contenido verdaderamente dinámico. A continuación se describe el funcionamiento de la *Web*.

2.1.1. Funcionamiento de World Wide Web

WWW se basa en el modelo cliente-servidor, en el cual dos equipos de cómputo trabajan en conjunto para intercambiar diferente tipo de información y realizar una tarea. El ejemplo más sencillo es la comunicación entre un cliente y un servidor, donde el cliente solicita la información y el servidor posee la información. El proceso es sencillo, el cliente envía una petición al servidor el cual responde al cliente con la información solicitada. Los elementos más relevantes que conforman la *WWW* son:

El servidor es un equipo (Computadora) que contiene datos de sitios *Web*, como puede ser

texto, imágenes, sonido y video almacenados en discos duros por ser los más convencionales, estos datos son transmitidos por Internet en la mayoría de los casos puesto que la información puede ser solicitada localmente.

El cliente es el visitante de la página *Web* y es quien requiere de la información almacenada en el servidor *Web*.

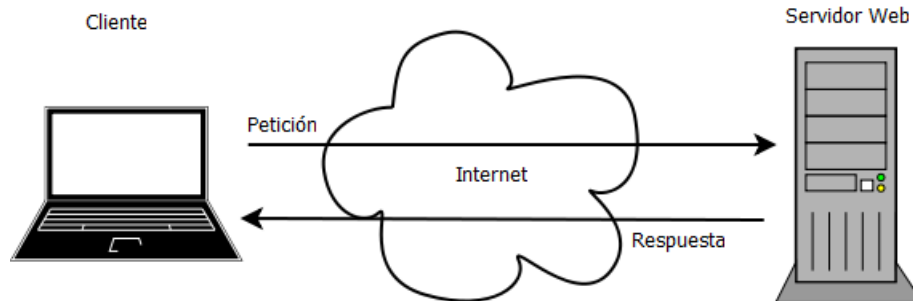


Figura 2.1: Modelo Cliente-Servidor

La página *Web* es la interfaz entre el cliente y el servidor *Web*, la cual se visualiza en un navegador. Algunos ejemplos de navegadores *Web* son *Internet Explorer*, *Mozilla FireFox*, *Opera*, *Google Chrome*, y muchos otros más. En los navegadores se visualiza texto, imágenes, video, etc. Existen dos tipos de páginas *Web* de acuerdo a la forma que se presenta el contenido.

- Página *Web* estática: Es aquella cuyo contenido no cambia con frecuencia, y donde los cambios se realizan de forma manual modificando el archivo *HTML*.
- Página *Web* Dinámica: Es aquella que genera contenido a partir de lo que el usuario introduzca en un formulario e incluso algún otro elemento dentro de la página, interactúa con una base de datos y mejora la interfaz entre el usuario y la página *Web*.

2.1.2. Aplicación Web

Una aplicación *Web* es un sistema donde la interfaz se provee desde una página *Web* y que se accede desde Internet o de forma local por medio de una red, por esta característica presenta un sistema con mayor portabilidad y alcance, pero con procedimientos de programación más complejos desde una perspectiva detrás de la Interfaz (*back-end*), con una lógica de negocios, donde la persistencia de la información se integra a una base de datos. Una aplicación *Web* a diferencia de una página *Web* estática, sirve para recabar información y/o permitir llevar a cabo diversos procesos.

2.2. Framework Web

Un *Framework* es un componente de *software* con una estructura bien definida. Un *Framework* tiene por objetivo hacer dos funciones primordialmente y con esto permitir construir una aplicación de forma más rápida. La primera función es automatizar las tareas más tediosas y repetitivas. La segunda es otorgar una solución de trabajo formal, esto quiere decir, que otorga una metodología que el desarrollador debe aprender y entender para poder utilizar el mismo. La mayoría de los *Frameworks Web* establecen mecanismos y soluciones sencillas y cómodas para realizar algunas tareas comunes, que permitan a los desarrolladores ahorrar esfuerzo y tiempo. Las tareas más comunes que puede presentar son las siguientes: conversión de datos, validación de datos, separación de la capa de negocio de los datos mostrados en la *Web*, internacionalización proceso a través del cual se van a diseñar sitios *Web* adaptables a diferentes idiomas y regiones sin necesidad de realizar cambios en el código.

El utilizar un *Framework* puede no ser necesario, pero entre más grande y complejo sea el proyecto, ayuda a simplificar procesos y otorgar un análisis y diseño más formal y comprensible para el programador.

A continuación se hace mención y una breve descripción de algunos *Frameworks Web* utilizados por empresas y comunidades en Internet y de los cuales existe soporte y documentación oficial.

2.2.1. Struts

Struts[23] es un *Framework* de código abierto (*open source*) gratuito creado por Craig R. McClanahan y donado a *Apache Software Foundation (ASF)* en el 2000. *Struts* ayuda a facilitar a los desarrolladores construir aplicaciones *Web* basadas en *Java Servlets*, *JavaServer Pages (JSP)*, *JavaBeans*, *ResourcesBundles* y *XML*. Permite el desarrollo de páginas *Web* dinámicas y la integración de base de datos. *Struts* utiliza el patrón de Diseño MVC (Modelo-Vista-Controlador), que es explicado más adelante.

El proyecto cuenta en la actualidad con varios *committers* (personas que tiene permiso para modificar un código fuente de un software en particular) de todo el mundo, y muchos desarrolladores están contribuyendo al bien general del *Framework*.

Struts es uno de muchos conocidos y exitosos proyectos de *Apache Jakarta* cuya misión general es proporcionar soluciones de calidad comercial, basada en la plataforma *Java*, de manera abierta y cooperativa.

Struts cuenta con tres componentes:

- La **solicitud** (*request*) controlador proporcionado por el desarrollador de la aplicación que se le asigna un *URL (Uniform Resource Locator)*.
- La **respuesta** (*response*) controlador que transfiere el control a otro recurso que completa la respuesta.
- Una **biblioteca de etiquetas** (*tags*) que ayuda a los desarrolladores a crear aplicaciones

interactivas basadas en formularios. Struts utiliza estas etiquetas para crear componentes dentro de las páginas *Web*, las cuales sirven como interfaz entre el usuario y la aplicación *Web* para el envío y solicitud de información.

2.2.2. Tapestry

Tapestry[25] es un *Framework* de código abierto (*open source*) para crear aplicaciones *Web* dinámicas, robustas y escalables. *Tapestry* funciona con cualquier servidor de aplicaciones (contenedor de *servlets*). Divide una aplicación *Web* en un conjunto de páginas, cada una compuesta por componentes, esto proporciona una estructura fácil de entender, lo que permite asumir la responsabilidad de la construcción del *URL* y el envío, almacenamiento de estado persistente en el cliente o en el servidor, la validación de entrada del usuario, internacionalización, y reportes de excepción. Permite la creación de plantillas *HTML* usando *HTML* plano y código *Java* en un modelo orientado a objetos. Está diseñado para crear nuevos componentes fácilmente y para crear pequeñas aplicaciones de una sola página así como aplicaciones de cientos de páginas. Se integra fácilmente con otros *Framework* como son *JEE*, *HiveMind*, *Spring* e *Hibernate*. Posee etiquetas muy parecidas al *HTML*.

Trabaja en tres capas, las cuales se mencionan a continuación:

- **Capa de Presentación:** Responsable de las peticiones y respuestas *HTML*.
- **Capa de Aplicación:** Responsable de la lógica de negocios puede ser implementada por *Enterprise JavaBeans (EJB)*.
- **Capa de la Base de Datos:** Es la responsable de la persistencia de los datos.

Con *Tapestry* se pueden diseñar sitios como *Amazon.com*, *Slashdot.org* e *Ebay.com* por mencionar algunos.

2.2.3. Java Server Faces

JSF o *Java Server Faces*[18] es un *Framework* que permite hacer aplicaciones *Web* fácilmente gracias a componentes de interfaz de usuario (campos de texto, combobox, botones, tablas, etc.). *JSF* es una parte de *J2EE (Java 2 Enterprise Edition)*. Una ventaja de este *Framework* es el poder utilizar *Rapid Application Development (RAD)*. El principal objetivo de *RAD* es permitir crear aplicaciones con un conjunto de componentes reutilizables visualmente, por mencionar algunos ejemplos de *IDE* con esta característica se encuentran: *Visual Basic*, *Power Builder*, *Delphi* y *Netbeans*. La tecnología *JSF* establece un estándar para la creación de interfaces de usuario. *JSF* incluye un conjunto de *API* para la gestión de estado, la gestión de eventos y validación de entrada, la definición de navegación de la página, y el apoyo a la internacionalización y accesibilidad. Además cuenta con una biblioteca de etiquetas (muy parecidas a las etiquetas *HTML*) para la creación de la interfaz de usuario dentro de una página *JSP*.

La arquitectura de *JSF* define de forma clara la separación entre la lógica y la presentación

de la aplicación al mismo tiempo que facilita la conexión de la capa de presentación con el código de aplicación por medio de beans (clases *Java*). *JSF* resuelve muchas tareas como son: Obtener y almacenar datos introducidos por el usuario en formularios, validación y conversión de datos, controlar eventos de periféricos (teclado, mouse, etc.), integración de otros lenguajes como *Javascript*, entre otras características más.

2.2.4. ASP.NET

ASP.Net[22] MVC es un *Framework* desarrollado por *Microsoft* que combina la eficiencia y elegancia del patrón MVC (Modelo-Vista-Controlador), además de poseer las técnicas más actualizadas para un desarrollo ágil de aplicaciones y proyectos *Web*. Es posible escribir código en *Visual Basic* o *C#* e incluso con algunos otros lenguajes y ser utilizado en la mayoría de los navegadores. Construir sitios *Web* en *ASP.NET* con *Visual Studio* (*IDE* de *Microsoft*) y correrlo en *ISS* (es un servidor *Web* y un conjunto de servicios para el sistema operativo *Microsoft Windows*). *Visual Studio* ofrece poderosas herramientas de desarrollo. *ASP.NET* es parte de *.Net*, que es una plataforma de programación. La página oficial de *ASP.NET* (<http://www.asp.net>) ofrece buen soporte e información adicional sobre este *Framework*.

2.2.5. Ruby on Rails

Ruby on Rails o *RoR*[21] es un *Framework* para el desarrollo de aplicaciones *Web* creado por David Heinemeier Hansson en el año 2003, escrito en lenguaje *Ruby* (Lenguaje Orientado a Objetos), el cual es un lenguaje moderno de programación que permite expresar ideas más claras en el código. *RoR* está diseñado para hacer más fácil la programación de aplicaciones *Web*. La filosofía de *Rails* sugiere que escribir el mismo código una y otra vez es algo malo. Trabaja con la arquitectura del patrón MVC.

2.3. Patrón de Diseño MVC

MVC (Modelo-Vista-Controlador) es un patrón de diseño de *software* en el cual todo el proceso está dividido en 3 capas: el Modelo, la Vista y el Controlador. [23, 3, 1]

- El Modelo incorpora la capa del dominio y persistencia, es la encargada de guardar los datos en un medio persistente (ya sea una base de datos, un archivo de texto, un archivo *XML*, un registro, etc.).
- La Vista se encarga de presentar la interfaz de usuario, mediante ventanas o páginas *Web* las cuales poseen componentes como botones, campos de texto, tablas, combobox, checkbox, etc.
- El Controlador es el que se encarga de comunicar la vista con el modelo, interpreta las solicitudes en la vista y regresa la petición del modelo. De una forma más sencilla, los

accesos a la base de datos se hacen en el modelo y el controlador es el que se encarga de la información que se envía a la vista.

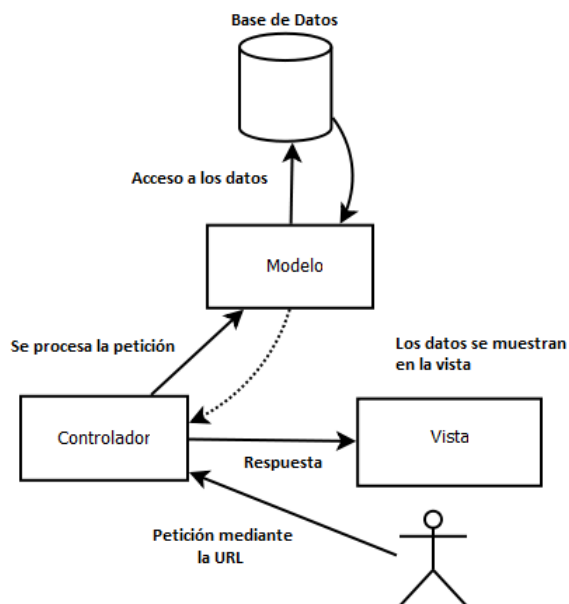


Figura 2.2: MVC

2.4. Base de Datos

Una base de datos es una colección estructurada de datos. En la cual se puede guardar cualquier tipo de información categorizada y ordenada en tablas para su mejor administración. Para añadir, acceder, y procesar los datos almacenados en ella, se necesita un sistema de gestión de base de datos. Una computadora es capaz de procesar grandes cantidades de información por lo que el gestor de base de datos juega un papel importante para el procesamiento dentro y fuera de las aplicaciones que se desarrollen.

2.4.1. Gestor de Base de Datos

Es un *software* que sirve de interfaz entre las bases de datos y las aplicaciones que requieren de los datos. Estas aplicaciones se conectan a una base de datos por medio del Gestor.

A continuación se mencionan algunos ejemplos de gestores de base de datos.

2.4.2. PostgreSQL

PostgreSQL[20], originalmente llamado *Postgres*, fue creado en la UCB (*University of California, Berkeley*) por un profesor de ciencias de la computación llamado Michael Stonebraker,

que pasó a convertirse en el director de tecnología de *Informix Corporation*. Desarrollado entre 1986-1994, fue un proyecto destinado a abrir nuevos caminos en los conceptos de bases de datos tales como la exploración de tecnologías objeto-relacional.

PostgreSQL es un potente sistema de base de datos relacional de código abierto, cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una reputación de fiabilidad, integridad de datos y corrección. Es compatible con *Linux*, *UNIX* (*AIX*, *BSD*, *HP-UX*, *SGI IRIX*, *Mac OS X*, *Solaris*, *Tru64*) y *Windows*. Tiene soporte completo para claves foráneas, uniones, vistas, triggers (disparadores) y procedimientos almacenados (en varios idiomas). Soporta la mayoría de datos *SQL* (*INTEGER*, *BOOLEAN*, *CHAR*, *VARCHAR*, *DATE*, *INTERVAL*, y *TIMESTAMP*). También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con soporte para *C* / *C++*, *Java*, *.Net*, *Perl*, *Python*, *Ruby*, *ODBC*.

2.4.3. FireBird

Firebird[6] es una base de datos relacional que ofrece muchas características del *SQL* estándar. Es compatible con *Linux*, *Windows* y una variedad de plataformas *Unix*. Ofrece una excelente concurrencia, alto rendimiento, y el apoyo de un poderoso lenguaje para procedimientos almacenados y disparadores, el cual podemos describir como un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (*INSERT*), actualización (*UPDATE*) o borrado (*DELETE*). El Proyecto *Firebird* es un proyecto comercial independiente basado en el código fuente liberado por *Inprise Corp* (ahora conocida como *Borland Software Corp*) el 25 de julio del 2000. Cualquiera puede construir una versión personalizada de *Firebird*, siempre y cuando las modificaciones se pongan a disposición, en las mismas condiciones de concesión de licencias *IDPL* (*Initial Developer's Public License*), para que otros puedan utilizar y aprovechar, sin costo alguno para su descarga, registro o licencia.

2.4.4. SQLite

SQLite[24] es gestor de base de datos transaccional y embebido, sin servidor, que no requiere configuración, creado por D. Richard Hipp. Es de código abierto y libre para cualquier propósito. Se encuentra actualmente en muchas aplicaciones, entre ellos varios proyectos de alto nivel. Lee y escribe directamente en archivos almacenados en el disco duro.

El formato de archivo de base de datos de *SQLite* es multiplataforma que libremente puede copiar una base de datos entre los 32 bits y 64 bits. Tiene una biblioteca compacta y el rendimiento es aceptable. Con todas las características habilitadas, el tamaño de la biblioteca puede ser inferior a 300 KB, dependiendo de la configuración del compilador. Si las características opcionales se omiten, el tamaño de la biblioteca puede reducirse por debajo 180 KB en entornos con poca memoria. Cada versión es cuidadosamente probada antes de

cada lanzamiento y tiene una reputación de ser muy fiable. La mayoría del código fuente de *SQLite* se dedica exclusivamente a las pruebas y verificación. En conclusión es rápido, fiable y fácil de usar.

2.4.5. Oracle

Oracle[17] es un gestor de bases de datos comercial vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes o multinacionales. En el desarrollo de páginas *Web*, pasa lo mismo, al ser un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, *Access*, *MySQL*, *SQL Server*, etc. Aunque actualmente existen diferentes versiones de acuerdo a las necesidades como son:

- *Enterprise Edition*
- *Standard Edition*
- *Standard Edition One*
- *Express Edition*

La versión *Oracle Database 10g Express Edition (Oracle Database XE)* es una base de datos de pequeñas dimensiones basado en la base de *Oracle Database 10g Release 2*. *Oracle Database XE* es libre para desarrollar, implementar y distribuir, fácil de descargar y administrar. Es una base de datos para los desarrolladores que trabajan en *PHP*, *Java*, *.NET*, *XML(Xtensible Markup Language)* y aplicaciones de código abierto. También para *DBAs* (Administradores de Base de Datos) que necesitan una base de datos inicial y sin costo alguno para la capacitación y el desarrollo, Proveedores de *software* Independiente y proveedores de *hardware* que quieren una base de datos para distribuir de forma gratuita, así mismo para las instituciones educativas y los estudiantes que necesitan una base de datos gratuita para su plan de estudios.

2.4.6. MySQL (Community Server)

MySQL[15] es un sistema gestor de bases de datos relacionales. Una base de datos relacional almacena datos en tablas, en lugar de poner todos los datos en un solo lugar, esto añade velocidad y flexibilidad. Esta versión de *MySQL* es de código abierto por lo que se puede modificar el *software* y adaptarlo a diversas necesidades bajo la licencia *GNU GPL*. El servidor de base de datos *MySQL* es muy rápido, fiable y fácil de usar. *MySQL Server* se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que otras soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. Ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad lo hacen altamente apropiado para acceder a bases de datos desde Internet. *MySQL* es un sistema cliente-servidor que consiste en un servidor *SQL* multihilo que trabaja con diferentes programas, bibliotecas, herramientas administrativas y amplias *APIs*. Por las características ya mencionadas *MySQL* fue elegido como gestor de base de datos para

el SIWAE.

2.5. Servidores *Web* y Servidores de Aplicaciones

Un servidor *Web* sirve para que las páginas puedan ser vistas a través de un navegador *Web*, mientras que un servidor de aplicaciones proporciona métodos que las aplicaciones cliente pueden llamar. Un servidor *Web* maneja exclusivamente peticiones *HTTP*, mientras que un servidor de Aplicaciones suministra la lógica de negocio a las aplicaciones cliente a través de uno o varios protocolos. En seguida se muestran los servidores *Web* más populares.

2.5.1. Apache Tomcat

Apache Tomcat[27] es un servidor *Web* de código abierto desarrollado en lenguaje *Java* por lo que soporta tecnologías *Java Servlet* y *Java Server Pages*, es desarrollado en un entorno abierto y participativo. *Apache Tomcat* tiene la intención de ser un proyecto con colaboración de los mejores desarrolladores de todo el mundo destinados a crear una implementación de código fuente robusta, de grado comercial, con muchas características a libre disposición de un servidor *Web*. Este proyecto forma parte de la *Apache Software Foundation*. Cientos de usuarios han contribuido con ideas, código y documentación para el proyecto. *Apache Software* existe para proporcionar implementaciones sólidas y de calidad comercial. Es una plataforma sobre la cual los desarrolladores pueden crear sistemas confiables, tanto para fines concretos como para fines experimentales.

2.5.2. IIS (Internet Information Services)

Internet Information Services o *IIS*[9] es un servidor *Web* y servidor de aplicaciones flexible, seguro y fácil de administrar que se ejecuta en sistemas operativos *Windows* y compatible con las arquitecturas x86 y 64-bit. Dentro de los servicios que soporta se encuentran *FTP*, *SMTP*, *NNTP* y *HTTP/HTTPS*. *IIS* tiene soporte para *ASP*, *ASP.NET*, *PHP* y *Perl* entre otros.

2.5.3. GlashFish

GlassFish[7] es un servidor de aplicaciones de código abierto desarrollado por *Sun Microsystems*, compatible con aplicaciones *Java EE*. *Sun Microsystems* fue comprada por *Oracle*, por lo que existen 2 tipos de licencia, una *GNU GPL* (*GNU General Public License*) y la versión comercial *Oracle GlashFish Enterprise Server*. Es compatible para plataformas *Windows*, *Linux*, *Unix* y *Mac*.

Es un servidor flexible basado en el estándar *OSGi* (estándar que cuenta con el apoyo de muchos vendedores del mundo *Java*: *IBM*, *RedHat*, *Software AG*.), permite a las organizaciones crear y desplegar aplicaciones *Web* de forma ligera sobre *Java EE 6* y así aprovechar el poder

de la plataforma *Java* para las aplicaciones empresariales. Los desarrolladores también se benefician de las mejoras simplificado el modelo de programación y productividad que ofrece *Java EE 6*. Utiliza una arquitectura microkernel basado en *OSGi*.

Se integra en la distribución de *NetBeans 6.9*, también ofrece soporte para *Eclipse*, y compatibilidad con lenguajes como *Ruby* y *Groovy*. Además una consola de administración fácil de usar y conectividad con un centro de actualizaciones.

2.5.4. WAS (Websphere)

IBM WebSphere Application Server o *WAS*[28] es un servidor de aplicaciones comercial de *IBM*. *WAS* está construido bajo estándares abiertos tales como *J2EE*, *XML*, y *Servicios Web*. *WebSphere Application Server* proporciona disponibilidad y seguridad. Además de generar, implementar y administrar ágilmente aplicaciones *SOA* robustas.

2.6. UML

El modelado de cualquier aplicación es un proceso complejo, para ello se han desarrollado técnicas, metodologías y estándares que ayuden a este proceso.

Lenguaje de Modelado Unificado o *UML*[14] es un lenguaje o notación en su mayor parte gráfica para analizar, diseñar y documentar un sistema. Fue creado en 1994 por Grafy Booch y Jim Rumbaugh basado en sus dos famosos métodos, el de Booch y el *OMT*(*Object Modeling Technique*) respectivamente, a quien más tarde se les unió Ivar Jacobson creador del método *OOSE* (*Object-oriented software engineering*). Existen diferentes diagramas de *UML* clasificados en 2 categorías: Diagramas de estructura y de comportamiento.

2.6.1. Diagramas de comportamiento

Son diagramas que permiten exhibir comportamientos de un sistema o de los procesos de las organizaciones. Incluyen los diagramas de secuencia, estado, actividades de colaboración y los casos de uso. A continuación se hace una breve descripción de cada uno de ellos.

Casos de uso

Es un documento narrativo que describe la secuencia de eventos de un actor (una persona, una computadora, etc.) que utiliza un sistema para llevar a cabo un proceso. Este diagrama y sirve para analizar los procesos que se llevan a cabo en el negocio mediante una descripción de los mismos.

Diagrama que muestra la forma en que interactúan varios objetos, especificando las relaciones entre ellos y el orden en el que ocurre el intercambio de mensajes. Un mensaje se traduce como un método de una clase que es invocado desde otra clase si se está utilizando el lenguaje de programación *Java*.

Diagrama de estado

Los diagramas de estado representan el comportamiento de las entidades capaces de comportamiento dinámico especificando su respuesta al recibir casos instancias de un evento, son útiles sólo para los objetos con un comportamiento significativo. El estado es el valor que adquiere determinado objeto a lo largo de un proceso o flujo.

Diagrama de actividades

Un diagrama de la actividad es un caso especial de un diagrama de estado en el cual todos los estados son acciones o estados de subactividad en el cual todas las transiciones son disparadas por acciones o subactividades en los estados. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad. En términos más sencillos es un diagrama de flujo que representa la transición de los componentes de un negocio.

Diagrama de colaboración

Es un diagrama que contiene un sistema de papeles o roles que se jugarán por objetos, así como sus relaciones requeridas dadas en un contexto particular, o presenta un CollaborationInstanceSet con una colección de objetos y sus relaciones.

2.6.2. Diagramas de Estructura

Son diagramas que describen elementos de una especificación que sean independientes del tiempo. Incluyen diagramas de clase, objetos, componentes y despliegue. A continuación se hace una breve descripción de cada uno de ellos.

Diagrama de clase

Es un documento en el cual se representan las clases, atributos de la clase y la relación entre las ya mencionadas de un sistema orientado a objetos.

Diagrama de objetos

Un Diagrama de objetos representa una situación concreta del dominio.

Diagrama de componentes

Un diagrama componente demuestra las dependencias entre componentes de software y los artefactos que los implementan; por ejemplo, los archivos de código fuente, archivos de código binario, y scripts ejecutables.

Diagrama de despliegue

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

2.7. Javascript

Javascript es un lenguaje orientado a objetos creado por *Netscape*. Este lenguaje no requiere de compilador ya que se ejecuta en el lado del cliente y es interpretado por los navegadores. Actualmente tiene soporte para la mayoría de navegadores. Permite tener una mejor interacción en las páginas *Web* permitiendo desarrollar otros *Frameworks* que reutilizan implementaciones y permiten crear mejor contenido de forma más sencilla y rápida.

2.8. JQuery

JQuery[11] es un *Framework* implementado con *Javascript* de código abierto que permite manipular componentes y estructura de los documentos *Web*, así como manejar eventos, desarrollar animaciones y agregar interacción en las páginas. Con *JQuery* es más fácil acceder a las propiedades de los componentes (etiquetas) por medio de sus correspondientes selectores (id o clases).

2.9. JQueryUI

JQueryUI[12] es la biblioteca de interfaz de usuario oficial de *JQuery*. Proporciona las interacciones, aplicaciones (*widgets*), efectos y temas (colores y diseño) de la creación de aplicaciones dinámicas. La ventaja es que ofrece variedad de componentes visuales e interactivos que hacen mas funcionales y atractivas las páginas.

2.10. CSS

CSS u Hojas de Estilo en Cascada es un lenguaje orientado para controlar las propiedades visuales de los componentes de la vista (color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página), aunque también soporta eventos del ratón. *CSS* trabaja con objetos bien definidos como puede ser párrafos, tablas, botones, etc. prácticamente todo componente de la vista. Otra característica es el uso de selectores a los cuales se pueden hacer referencias con un *Id* para un sólo elemento o con una clase para todos los elementos que posean el nombre de la clase.

2.11. IDE (Entorno de Desarrollo Integrado)

Las *IDE* son programas que ofrecen diferentes herramientas útiles para el desarrollo de aplicaciones, soportan diferentes lenguajes de programación, poseen un depurador que permite observar línea por línea el código de dicha aplicación, muestran mensajes de error, permite un mejor ordenamiento de los archivos asociados, genera código automáticamente.

2.11.1. Netbeans

Netbeans[16] es una *IDE* para desarrolladores de *software* libre, de código abierto. El cual cuenta con herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, *Web* y aplicaciones móviles con la plataforma *Java*, así como *C*, *C++*, *PHP*, *JavaScript*, *Groovy* y *Ruby*.

Puede aumentar su productividad cuando se trabaja con *Java SE*, *Java EE*, o la tecnología *Java ME*, así como *PHP*, *Ruby on Rails*, *Groovy*, *JavaScript*, *JavaFX*, *C* y *C++*. Tiene disponible herramientas visuales que generan código, lo que le permite crear aplicaciones simples sin tener que escribir una sola línea de código.

Las principales razones para utilizar *NetBeans* son:

1. Fácil descarga e instalación, todas las herramientas y características están integradas.
2. Libre y de Código Abierto.
3. Orientado para desarrolladores.
4. Generador de interfaces gráficas de usuario.
5. Compatibilidad con los estándares *Java* y plataformas. Proporciona soluciones de extremo a extremo para todas las plataformas de desarrollo *Java*, incluyendo los últimos estándares *Java*.
6. Herramientas de depuración.
7. Asistencia dinámica de idiomas.
8. Plataforma extensible. Ampliación de la plataforma y su fundamento basado en Swing, ahorra tiempo de desarrollo.
9. Personalización de Proyectos. Puede construir, ejecutar e implementar proyectos en servidores fuera de la *IDE*.
10. Soporte para código que no sea *Java*. No está limitado al lenguaje de programación *Java*, puede incluir muchos otros lenguajes de programación como *C*, *C++*, *Ruby* y *PHP*.

2.11.2. Eclipse

Eclipse[4] fue creado originalmente por *IBM* en noviembre de 2001 y apoyado por un consorcio de proveedores de *software*. La *Eclipse Foundation* se creó en enero de 2004 como en el independiente sin fines de lucro, para actuar como administrador de la comunidad. Fue creada para permitir que un proveedor neutral y comunidad abierta, transparente, que

se establecerá alrededor de esta *IDE*. No necesita ser instalada puesto que, con solo descargar y descomprimir la carpeta se puede ejecutar. Posee diferentes versiones, que se integran perfectamente a diferentes gestores de base de datos y servidores de aplicaciones, cuenta con buena documentación y muchos libros incluyen ejemplos.

2.12. Java

Java[8] es un lenguaje de programación que se ejecuta en una máquina virtual por lo que no es dependiente de un sistema operativo en específico. Es un lenguaje orientado a objetos por lo que cuenta con características como el polimorfismo y la herencia. Puede ser utilizado para diversos tipos de aplicaciones como son de escritorio, móviles y *Web* como es el caso en este sistema.

2.12.1. JPA

JPA[13] o *API* de Persistencia de *Java* es un *Framework* desarrollado para el lenguaje *Java*. Este *ORM* ofrece un mapeo por medio de anotaciones. Una anotación es una forma de añadir metadatos (datos de los datos) al código de *Java* y se usa como alternativa al lenguaje *XML* y las cuales pueden ser ocupadas en tiempo de compilación y ejecución.

2.12.2. Tiles

Es un *Framework* de plantillas diseñado para simplificar el desarrollo de interfaces de usuario de una aplicación *Web*. Este *Framework* se integra con *Struts 2* con una instalación e implementación sencilla.

Capítulo 3

Análisis y Diseño del Sistema

En esta sección se describen las fases de análisis y diseño del sistema. La primera fase se enfoca en determinar los requerimientos en el dominio del negocio, así mismo se establece el alcance del sistema. En la segunda fase -diseño del sistema- se establece una propuesta formal de solución y se dan detalles de implementación. Se utiliza *UML* como herramienta para explicar ambas fases.

3.1. Análisis del Sistema

En esta etapa es necesario recabar todos los requerimientos del sistema para determinar el alcance del mismo. Existen muchas técnicas para recabar información. A continuación se muestran algunas técnicas para obtener requerimientos:

- **Arqueología de documentos:** Consiste en buscar en los documentos, reportes y archivos existentes los requerimientos. Es más útil cuando ya se cuenta con un sistema funcionando y se planea modificarlo o renovarlo.
- **Entrevista:** Es una técnica muy efectiva para tener un mejor entendimiento del problema y para descubrir los requerimientos generales.
- **Taller de requerimientos:** Junta a todos los involucrados con el proyecto por un periodo corto pero intenso y se obtiene toda la información posible de los asistentes.
- **Lluvia de ideas (*Brainstorming*):** Tiene dos fases, la de generación de ideas y la de reducción de ideas. La primera se realiza mediante una pregunta que es lanzada por un moderador y los demás participantes escriben sus ideas, opiniones o sugerencias en hojas al termino de la fase las hojas son pegadas en la sala donde se está llevando la actividad. La segunda fase es leer cada una de las hojas, las cuales serán eliminadas por votación de los integrantes o complementadas entre ideas muy similares.
- **Presentación de ideas (*Storyboard*):** Es un prototipo del sistema, que se realiza con bosquejos, pueden ser hechos en rotafolios, pizarrones, o algún programa que permi-

ta rápidamente crear un diseño de baja fidelidad, la idea es presentar o exponer el entendimiento que se tiene de los requerimientos del cliente.

Para el análisis de esta aplicación (SIWAE) se utilizó la técnica de arqueología. Se obtuvieron las encuestas impresas previamente creadas por la coordinación de la licenciatura de Contaduría del CU UAEM Zumpango y se tomaron como punto de partida para generar interfaces de usuario y diseñar la base de datos. En la figura 3.1 se muestra el diagrama de dominio que se utilizó para tener una perspectiva general de las entidades (personas, objetos e incluso otro sistema que tiene relación con el expuesto en este documento) y las relaciones que tienen entre las mismas.

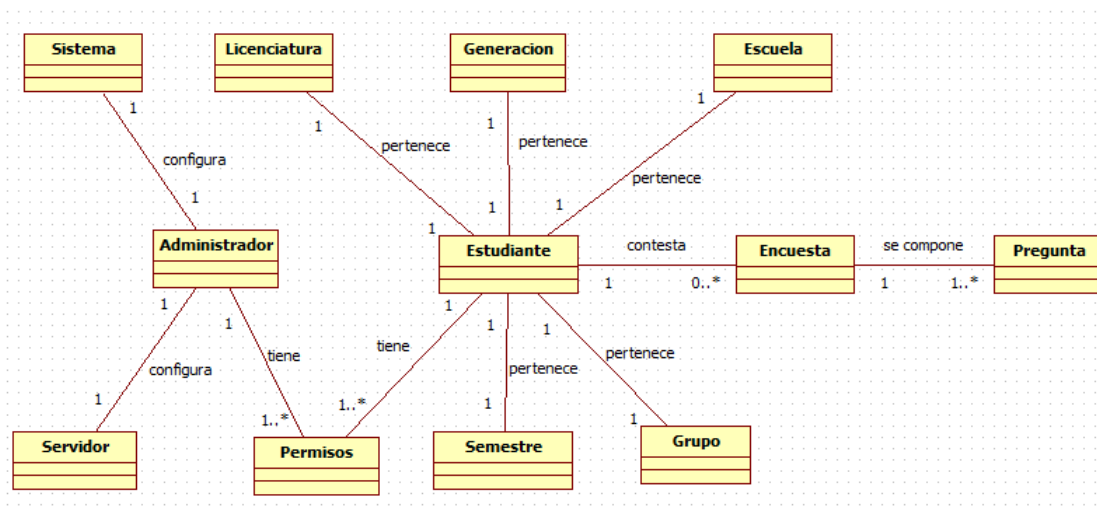


Figura 3.1: Diagrama de dominio para el SIWAE

El diagrama de dominio muestra las entidades que interactúan en el sistema y permite tener una perspectiva general del mismo. En el diagrama se muestran las relaciones que existen entre las entidades, las relaciones van de uno a uno y de uno a muchos, un estudiante contesta cero a muchas encuestas, una encuesta se compone de una a muchas preguntas, un estudiante tiene de uno a muchos permisos al igual que el Administrador. A continuación se muestran los casos de uso que se detectaron en la fase de análisis.

3.1.1. Casos de uso del Sistema

En la figura 3.2 se muestran los actores y casos de uso identificados en el sistema, también se muestra la inclusión de casos de uso con la palabra `<<include>>`, esto quiere decir, que para que se pueda ejecutar determinado caso de uso se tiene que ejecutar el caso de uso incluido.

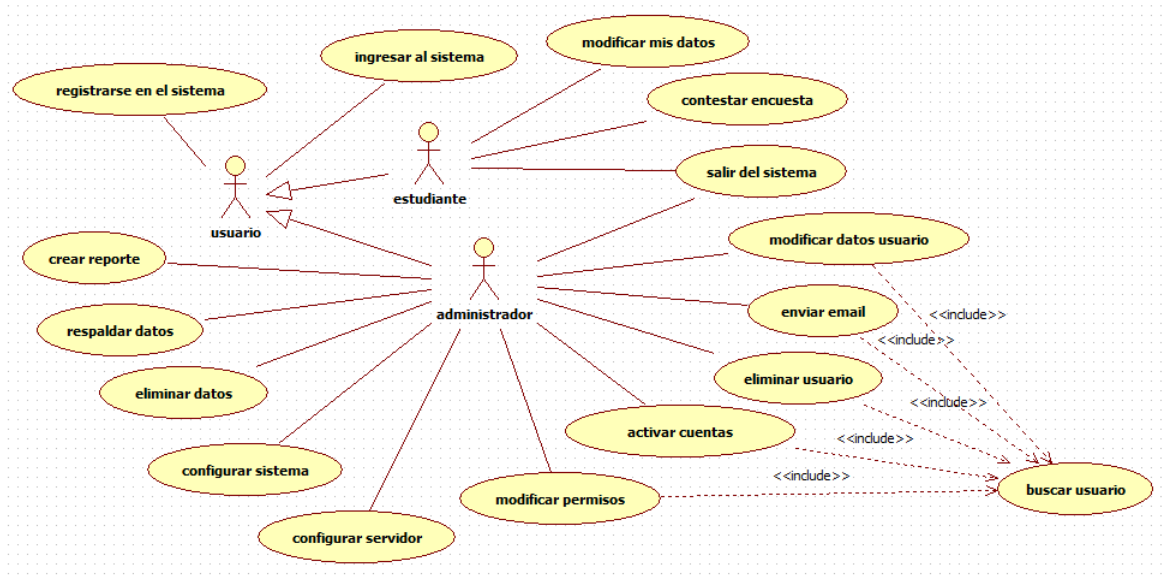


Figura 3.2: Casos de uso del SIWAE

En la figura 3.3 se muestra el caso de uso **Registrarse en el sistema**. Este caso de uso es de los más relevantes puesto que es el primer proceso para poder acceder al sistema, dicho diagrama explica la interacción entre el usuario y el sistema, las precondiciones que debe de tener el usuario son no estar registrado en el sistema y tener una cuenta de correo electrónico activa a la que tenga acceso, esto es necesario ya que para activar la cuenta el usuario debe ingresar a su correo y presionar el hipervínculo enviado por el SIWAE. Los pasos 6,7 y 8 del caso de uso de la figura 3.3 se refieren a la validación que se lleva dentro del formulario.

Caso de Uso	<i>Registrarse en el sistema</i>		
Autor	Christian Abraham Flores Lara		
Precondiciones	Tener una cuenta de correo electrónico activa No estar registrado en el sistema		
Postcondiciones	Ninguna		
Flujo de eventos	Actor	Sistema	
	1	Abrir la página principal del sistema	
	2		Envía la página principal con una sección de autenticación y registro para un nuevo usuario.
	3	Solicita al sistema registrarse al mismo, dando clic en el botón de Nuevo Usuario .	
	4		Envía una página con un formulario con la información necesaria para realizar el registro del usuario.
	5	Ingresar la información en los campos del formulario y acepta darse de alta dando clic en el botón Aceptar .	
	6		Valida la información de los campos.
	7		Si algún campo que es requerido no contiene información envía un mensaje de error.
	8		Si el usuario ya existe envía un mensaje de que el usuario ya existe.
	9		Si todos los datos se encuentran en forma correcta, el sistema envía un mensaje al correo del usuario en el cual debe activar su cuenta.
10	El usuario entra a su cuenta de correo electrónico y abre el mensaje enviado por el sistema, da clic en el hipervínculo para activar su cuenta.		
		El sistema envía una página donde notifica al usuario que su cuenta se encuentra activada.	

Figura 3.3: Caso de Uso - Registrarse en el sistema

En la figura 3.4 se muestra el caso de uso **Ingresar al sistema**. En este caso de uso se describe la forma en que el usuario ingresa al sistema y las validaciones que se llevan a cabo en el proceso. Una vez que el usuario ingresa al sistema se muestra una página con las opciones de acuerdo al tipo de usuario y los permisos que éste posee.

Caso de Uso	<i>Ingresar en el sistema</i>		
Autor	<i>Christian Abraham Flores Lara</i>		
Precondiciones	<i>Registrarse en el sistema Activar la cuenta</i>		
Postcondiciones	<i>Ninguna</i>		
Flujo de eventos		Actor	Sistema
	1	Abrir la página principal del sistema	
	2		Envía la página principal con una sección de autenticación y registro para un nuevo usuario.
	3	Introduce el usuario y contraseña en la sección de autenticación.	
	4		Valida los datos que introduce el usuario.
	5		Si el usuario o contraseña son incorrectos envía un mensaje de error.
	6		Si la cuenta no está activa envía un mensaje de error.
	7		Si los datos son correctos envía una página con las opciones correspondientes al usuario.

Figura 3.4: Caso de uso - Ingresar al sistema

En la figura 3.5 se muestra el caso de uso **Contestar encuesta**. Se describe de forma general el proceso que se debe seguir para que un usuario con permisos conteste una encuesta. Las precondiciones son ingresar al sistema y tener los permisos necesarios para ingresar al menú de encuestas.

Caso de Uso	<i>Contestar encuesta</i>		
Autor	<i>Christian Abraham Flores Lara</i>		
Precondiciones	<i>Ingresar en el sistema Tener permisos para contestar encuestas</i>		
Postcondiciones	<i>Ninguna</i>		
Flujo de eventos		Actor	Sistema
	1	Selecciona una encuesta del menú de encuestas	
	2		Envía una página con las preguntas de la respectiva encuesta
	3	El usuario contesta las preguntas de la encuesta y da clic en continuar.	
	4		Valida que se hayan contestado las preguntas requeridas obligatoriamente
	5		Si alguna pregunta no es respondida o el tipo de dato no es válido envía un mensaje de error.
	6		Si todo se encuentra válido envía la siguiente parte.
	7		Si la encuesta esta completa envía a la página principal de autenticación del usuario donde se muestra el menú con las opciones que el usuario tiene permitido hacer.

Figura 3.5: Caso de uso - Contestar encuesta

3.2. Diseño del sistema

En esta etapa se crearon diagramas de *UML* los cuales describen los procesos llevados a cabo por los usuarios dentro del sistema, esto sirve para describir su comportamiento y encontrar posibles mejoras a los mismos. Hay que recordar que no se tienen que ocupar todos los diagramas de *UML* sino únicamente los que sean necesarios. A continuación se presentan los diagramas de interacción de los procesos encontrados en el sistema con una descripción más detallada que en los casos de uso. Cuando un usuario no esté registrado en el sistema se

le donominará **usuario**, y una vez que acceda al sistema será llamado **Estudiante**, **Profesor** y **Administrador** según el caso.

3.2.1. Diagramas de interacción

En la figura 3.6 se muestra el diagrama de interacción “**Registrar Usuario**”, donde un usuario que no posee una cuenta, ingresa a la página principal y da clic en el botón “Nuevo Usuario”, SIWAE envía un formulario con los datos requeridos para que el usuario se registre.

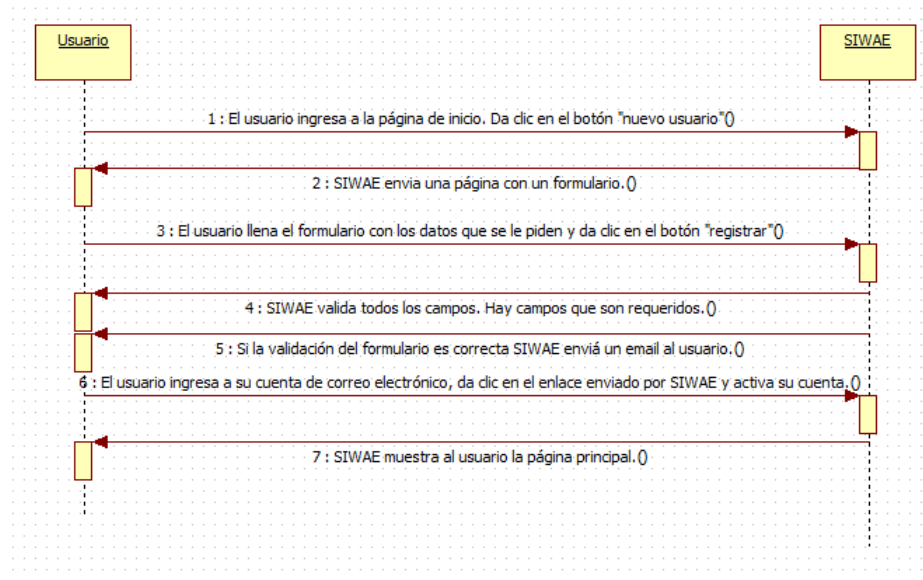


Figura 3.6: Diagrama de interacción “Registrar usuario”

En la figura 3.7 se muestra el diagrama de interacción “**Ingresar al sistema**”, como pre-condición el usuario tiene que estar registrado en el SIWAE y su cuenta debe estar activa. Si por alguna razón el usuario introduce su cuenta o contraseña incorrectos el sistema envía un mensaje de error.

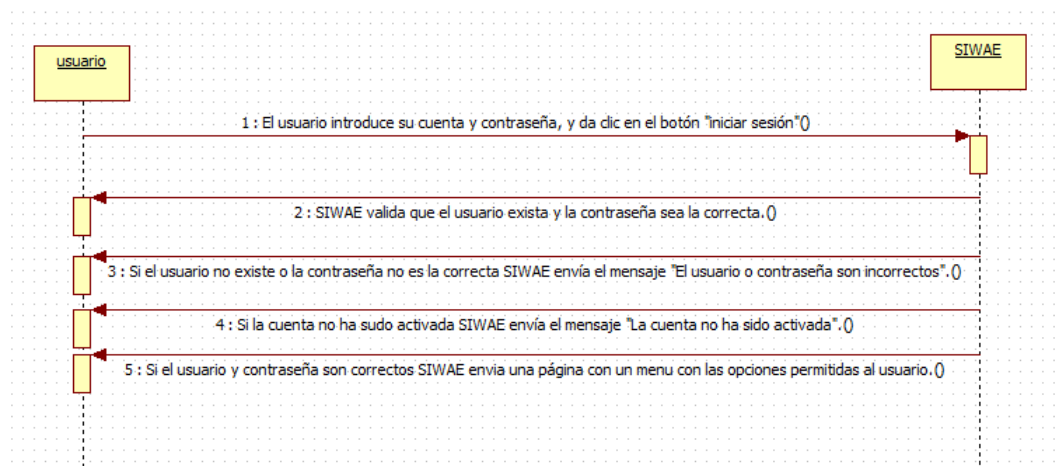


Figura 3.7: Diagrama de interacción “Ingresar al sistema”

En la figura 3.8 se muestra el diagrama de interacción “**Modificar mis Datos**”, el respectivo formulario pasa por una validación y al terminar el proceso se envía un mensaje, informando los datos modificados a la cuenta de correo electrónico del usuario “de” o “a” quien modificó sus datos.

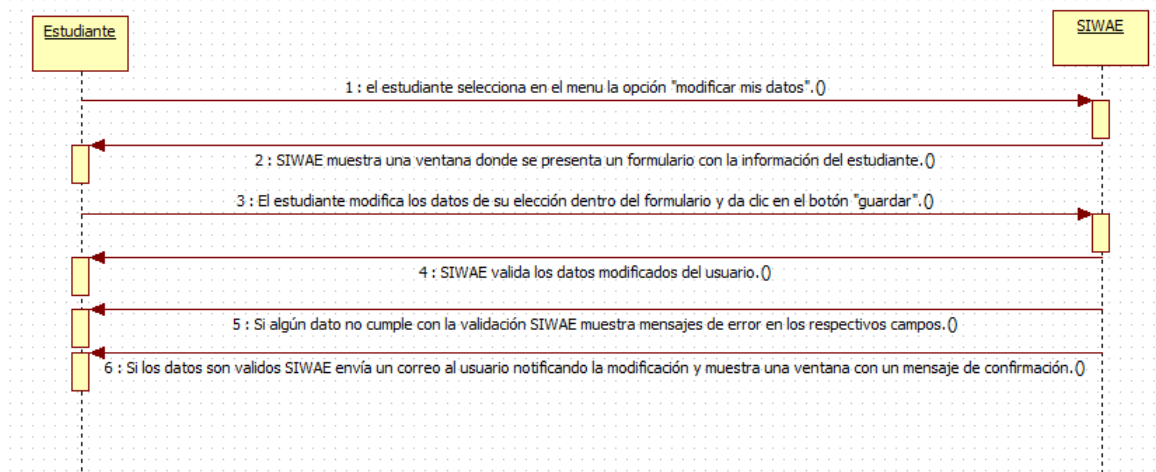


Figura 3.8: Diagrama de interacción “Modificar mis datos”

La figura 3.9 muestra el diagrama de interacción “**Buscar usuarios**”, este diagrama se presenta en diferentes procesos del SIWAE y es de los más utilizados en el mismo, ya sea para modificar los datos, para eliminar e incluso para los reportes de las encuestas. Otra característica importante es que si se presentan muchos resultados de búsqueda serán presentados en pantalla por medio de un paginador, lo que vuelve más sencilla la navegación, esto se describe con más detalle en el capítulo 4.

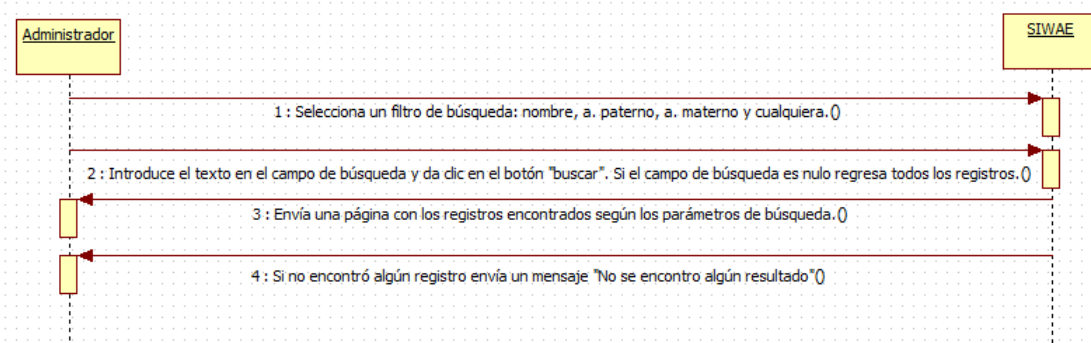


Figura 3.9: Diagrama de interacción “Buscar usuarios”

La figura 3.10 muestra en diagrama de actividades “**Activar usuario**”, este proceso se presenta cuando alguien que se registró no activó su cuenta, por lo que debe de solicitar al Administrador la activación manual. Para realizar esto se debe de buscar al Estudiante, Profesor o Administrador, en el diagrama 3.10 se observa cómo hace el llamado al diagrama de interacción “**Buscar usuarios**” con la finalidad de que no se vuelva a repetir la información expresada en otro diagrama.

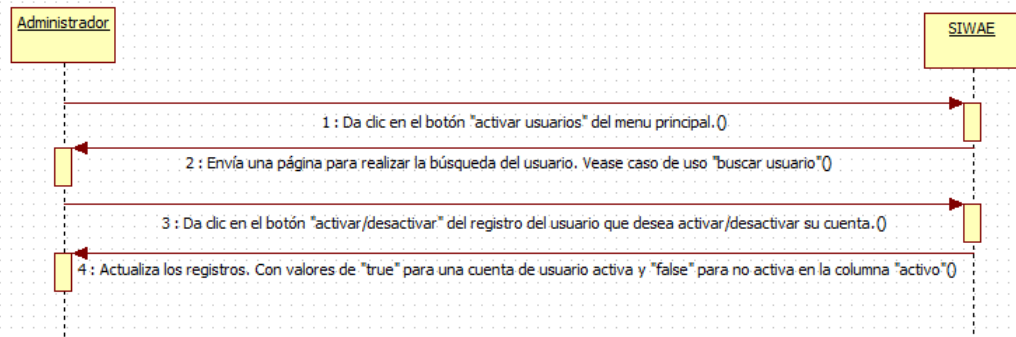


Figura 3.10: Diagrama de interacción “Activar usuario”

En la figura 3.11 se muestra el diagrama de interacción “**Configurar servidor**” el cual es un proceso que sólo los administradores pueden realizar. Este proceso también incluye validación de los datos y el envío de un mensaje de prueba por medio de la cuenta de correo electrónico con la que está configurado el SIWAE.

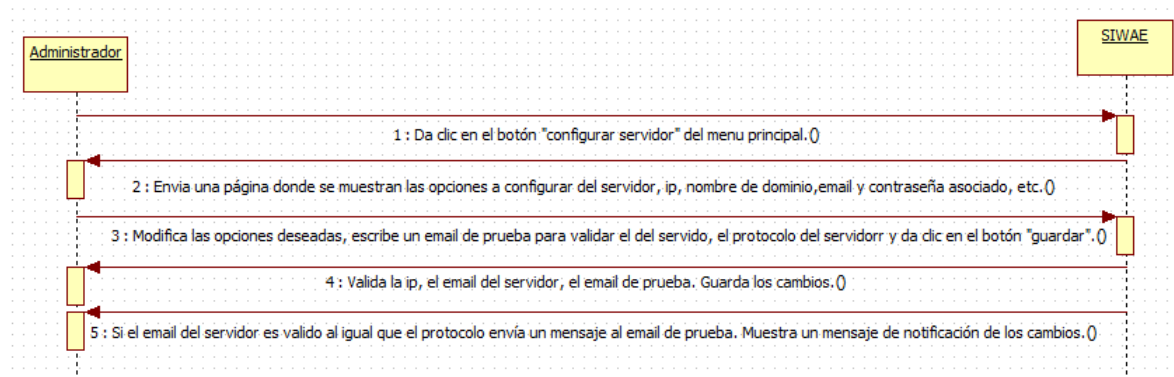


Figura 3.11: Diagrama de interacción “Configurar servidor”

En la figura 3.12 se muestra el diagrama de interacción “**Eliminar Usuario**” en donde se muestra el proceso donde un Administrador elimina un determinado Alumno, Profesor o Administrador del SIWAE. También se muestra la inclusión del diagrama de interacción “**Buscar usuario**” descrito anteriormente.

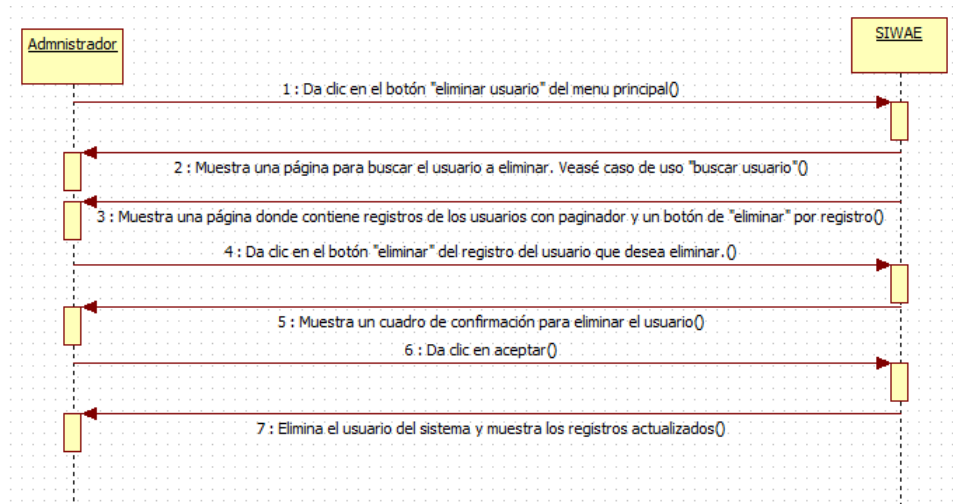


Figura 3.12: Diagrama de interacción “Eliminar usuario”

En la figura 3.13 se muestra el proceso “**Enviar Email**”, el mensaje puede ser enviado a cualquiera registrado en el SIWAE, ya sea de manera individual o para todos.

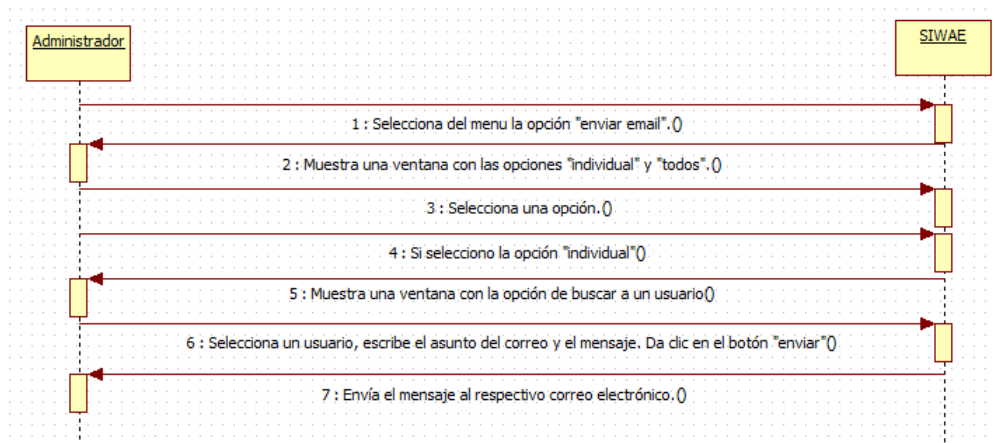


Figura 3.13: Diagrama de interacción “Enviar email”

En la figura 3.14 se muestra el diagrama “**Modificar permisos**” en donde se describe como un Administrador puede modificar permisos dentro del SIWAE a determinados Estudiantes, Profesores o incluso otros Administradores.

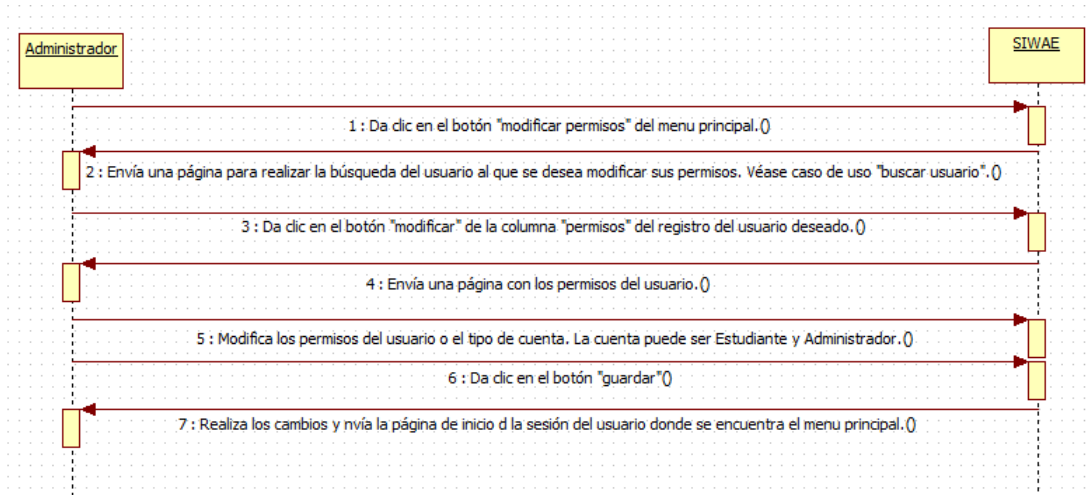


Figura 3.14: Diagrama de interacción “Modificar permisos”

En la figura 3.15 se muestra el diagrama “**Cerrar sesión**” el cual es importante por cuestiones de seguridad, ya que la información de los registrados en el sistema puede ser usada de manera incorrecta.

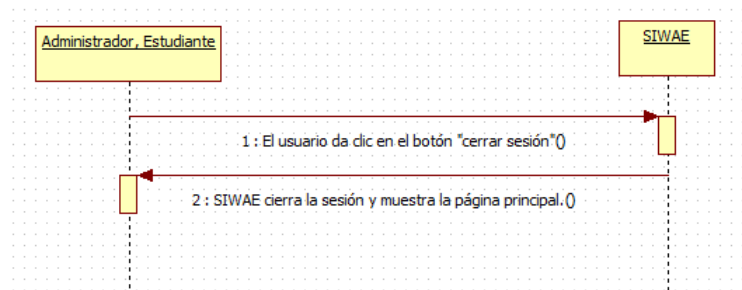


Figura 3.15: Diagrama de interacción “Cerrar sesión”

3.2.2. Diagrama de actividades

En la figura 3.16 se muestra el diagrama de actividad “**Registro de usuario**” el cual describe el proceso que se lleva a cabo cuando un usuario se registra en el sistema, se da el supuesto que se da clic en el botón “Registrarse”, enseguida se llena el formulario con los campos requeridos necesarios. En caso de no llenar todos los campos obligatorios el sistema indica los campos faltantes. Una vez validado por completo el formulario, el usuario presiona el botón “Registrar” y desde el servidor se realiza una nueva validación sobre la existencia del usuario, si éste ya existe se notifica con un mensaje, en caso de que no exista, el servidor envía un correo electrónico con un hipervínculo para la activación de la cuenta.

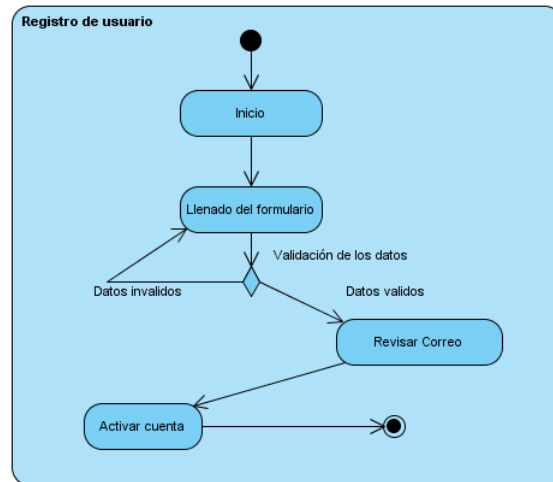


Figura 3.16: Diagrama de actividad “Registro de usuario”

En la figura 3.17 se muestra el diagrama de actividad “**Autenticación en el sistema**” donde se da el supuesto que el usuario ya se ha registrado, su cuenta ha sido activada y se encuentra en la página principal del SIWAE. En el diagrama se observa que el usuario introduce la cuenta y contraseña, luego hace clic en el botón “Iniciar Sesión”. El servidor verifica si la cuenta existe y/o si la contraseña es correcta, en caso contrario se envía un mensaje de error notificando que la cuenta o contraseña no son correctas. La actividad concluye cuando se introduce la cuenta y contraseña correcta y se accede al sistema.

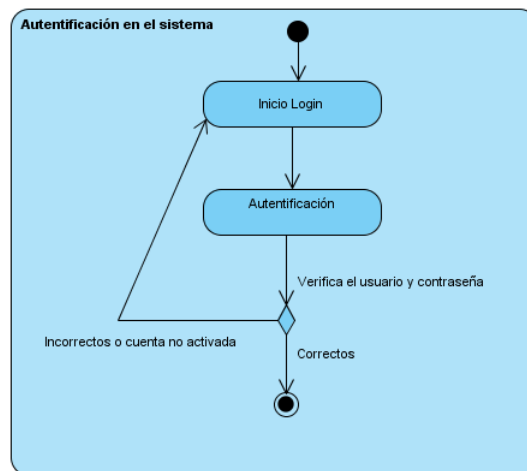


Figura 3.17: Diagrama de actividad “Autenticación del usuario”

La figura 3.18 muestra el diagrama de actividad “**Modificar datos**”, para esto se tiene el supuesto de que un Estudiante, Profesor o Administrador ya ingresó al sistema y presionó el botón “Modificar datos”. Los respectivos datos se actualizan de manera manual y se da clic en el botón “Guardar”. El formulario pasa por una validación que de ser aprobada concluye con una notificación al correo electrónico.

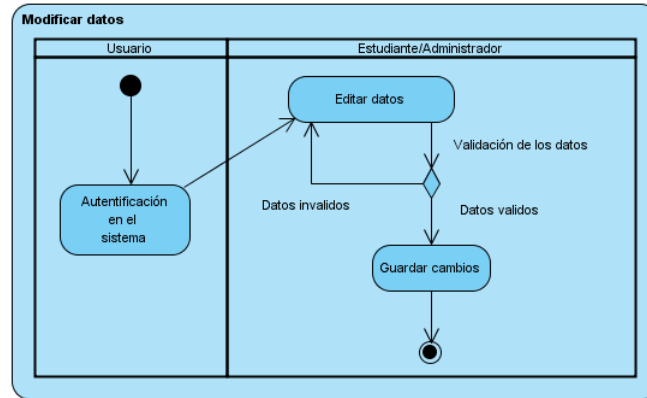


Figura 3.18: Diagrama de actividad “Modificar datos”

En la figura 3.19 se muestra el diagrama de actividad “**Eliminar Usuarios**” donde existe el supuesto de que un Administrador ingresó al SIWAE. Como primer paso dentro de esta actividad se busca al usuario que se desea eliminar, la búsqueda puede lanzar uno o más resultados por lo que se selecciona a quien eliminar, una vez seleccionado se presiona el botón “Eliminar” lo que muestra un mensaje de notificación “¿Desea realmente eliminar este Usuario?”, la actividad concluye con la respuesta a la notificación de “Si”, en caso de tener la seguridad de eliminarlo, o con un “No” para cancelar la eliminación.

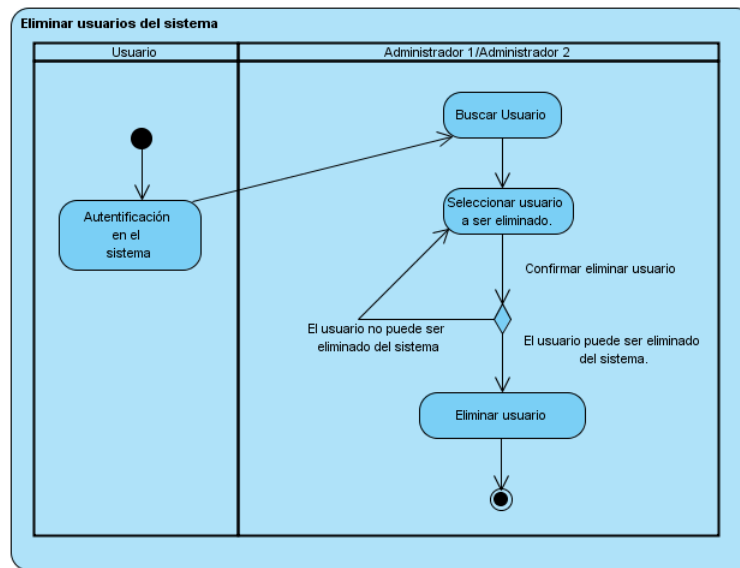


Figura 3.19: Diagrama de actividad “Eliminar usuarios”

La figura 3.20 muestra el diagrama de actividad “**Modificar Permisos**” donde existe el supuesto de que el Administrador ingresó al sistema. El Administrador busca al usuario, lo selecciona del resultado de la búsqueda, en la vista se muestran los permisos los cuales pueden ser modificados, se modifican los permisos deseados y da clic en el botón “Guardar” para concluir esta actividad.

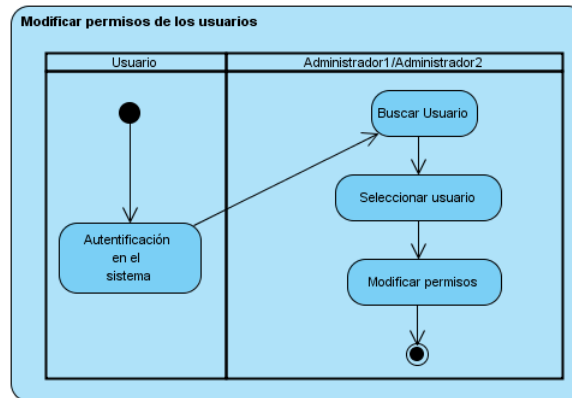


Figura 3.20: Diagrama de actividad "Modificar permisos"

La figura 3.21 muestra el diagrama de actividad "**Enviar Email**" el cual tiene como tarea la autenticación del usuario al sistema, el tipo de cuenta es Administrador y presiona el botón "Enviar Email".

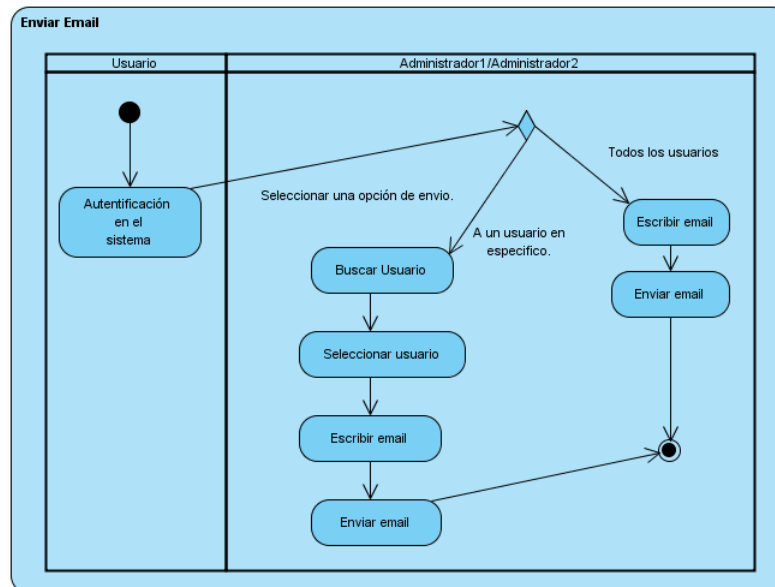


Figura 3.21: Diagrama de actividad "Enviar email"

La figura 3.22 muestra el diagrama de actividad "**Contestar Encuesta**" donde existe el supuesto que un usuario de cualquier tipo se autentica en el SIWAE y da clic en alguna encuesta activa. Una vez que ingresa a la encuesta el usuario debe contestar cada una de las preguntas y/o opciones. Una vez contestada puede imprimir esa sección de la encuesta y posteriormente dar clic en continuar para pasar a la siguiente sección, este paso se repite hasta llegar a la última sección donde tiene la opción de imprimir toda la encuesta dando como concluida esta actividad.

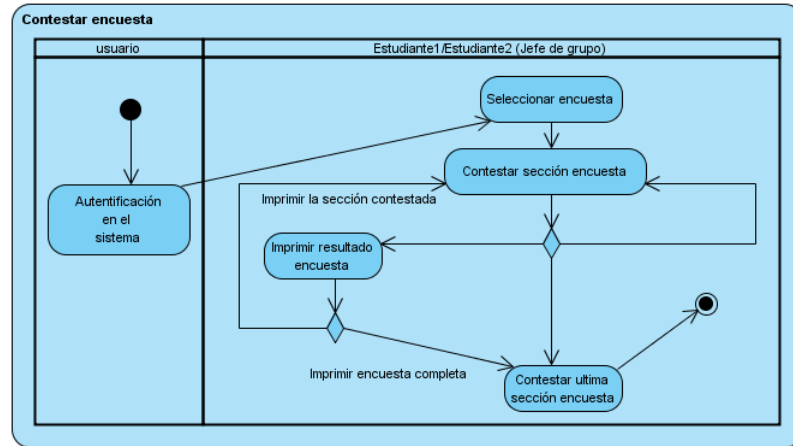


Figura 3.22: Diagrama de actividad "Contestar encuesta"

Capítulo 4

Desarrollo del Sistema

En este capítulo se describen las diferentes tecnologías ocupadas para el desarrollo de la aplicación descrita. Se mencionan algunas situaciones que se presentaron en las diferentes etapas de desarrollo y la mejor solución que se encontró para resolverlas.

4.1. Arquitectura Modelo-Vista-Controlador

Los sistemas *Web* están generalmente organizados en una arquitectura de tres capas, la capa cliente la cual presenta y/o obtiene los datos, la capa de datos la cual hace referencia a la base de datos y la capa intermedia cuyo objetivo es encargarse del procesamiento de los datos entre la capa cliente y la capa de datos.

El patrón de diseño MVC (Modelo Vista - Controlador) hace referencia precisa a la arquitectura antes mencionada. El modelo encapsula la lógica de negocio y el acceso a los datos. Es independiente de la vista y del controlador. La vista está compuesta por todas las páginas *Web*, y con las cuales interactuarán los usuarios y el sistema. Estas páginas *Web* contienen formularios que permiten mostrar y obtener datos, además de presentar efectos y estilos que pueden ser muy amigables para los usuarios finales. La capa controlador es la encargada de enlazar todos los procesos que se llevan a cabo en la aplicación *Web* y sirve de intermediario entre el modelo y la vista.

La ventaja de esta organización es la separación del análisis y reglas de negocio, por lo que hacer una modificación resulta más sencillo y rápido de realizar, otra ventaja es la forma en que los datos pasan de la vista al modelo y viceversa por medio del controlador, asociado a una clase de *Java* y un formulario definido en la *JSP*, la reutilización de código. En conclusión es una buena técnica para el desarrollo de aplicaciones. En la figura 4.1 se muestra la arquitectura del patrón MVC.

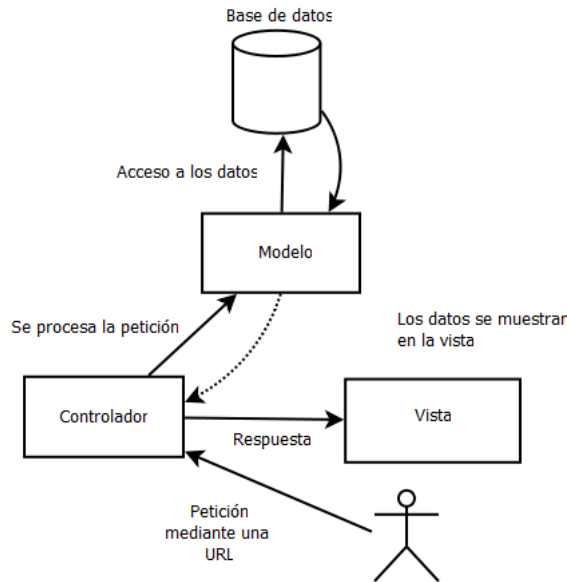


Figura 4.1: Arquitectura Modelo-Vista-Controlador

4.2. Descripción del Sistema

El sistema desarrollado tiene una arquitectura de tres capas como el de la figura 4.1, para la implementación del SIWAE se utilizó el Framework *Struts 2* [3].

4.2.1. Modelo

En *Struts 2* la lógica de negocio se encuentra a cargo de un componente llamado *Action*. Un *Action* se extiende de la interfaz *ActionSupport* y que posee un método *execute()* el cual regresa una cadena (*String*) como respuesta a una petición de la vista. En el *Action* se implementa la lógica de los datos desde una clase externa para tener una mejor organización de las clases, como se observa en la figura 4.2

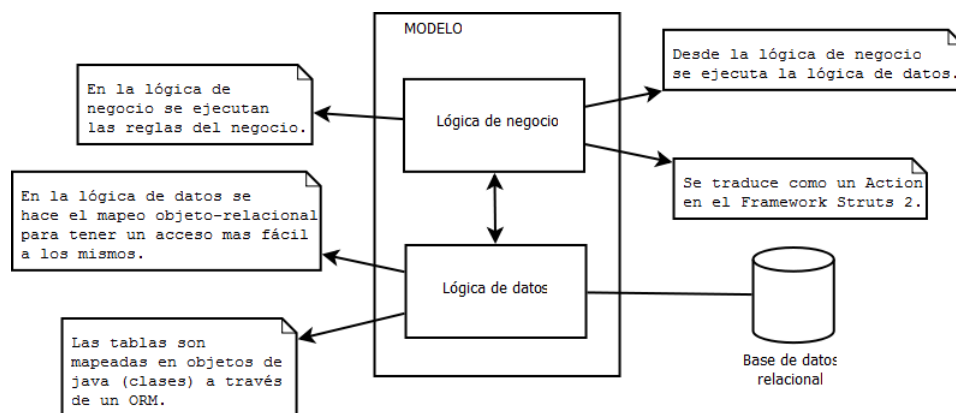


Figura 4.2: Representación de la capa modelo

Lógica del negocio

La figura 4.3 muestra el diagrama de clases de la lógica de negocio implementada por la clase *Action* asociada al *Framework* utilizado para desarrollar esta aplicación.

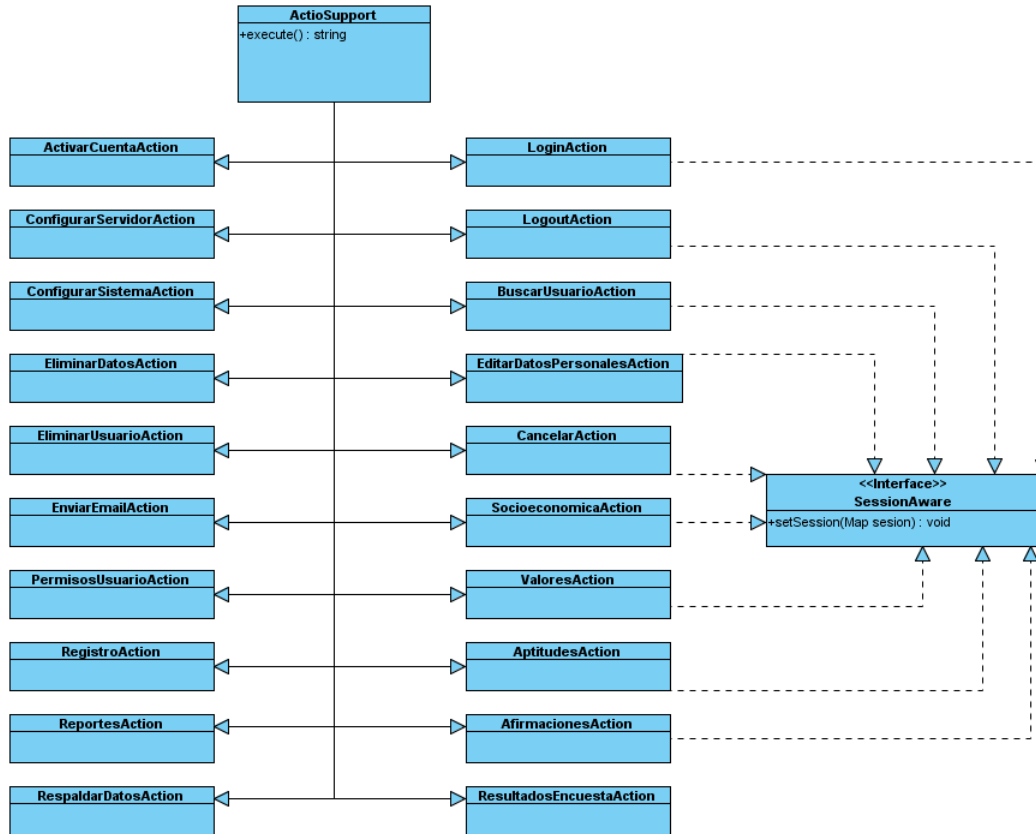


Figura 4.3: Diagrama de clases lógica de negocio

La figura 4.4 muestra el diagrama de clases de los *Actions* RegistroAction, ActivarCuentaAction, LoginAction, LogoutAction, CancelarAction, BuscarUsuarioAction y PermisosAction.

La clase RegistroAction realiza el proceso de registro del usuario y la consulta de la existencia del mismo; la clase ActivarCuentaAction se ejecuta cada vez que el usuario se ha registrado exitosamente y esté ingresa a su correo electrónico para activar su cuenta; La clase LoginAction se encarga de validar que el usuario exista y su contraseña sea la correcta, de ser así el usuario puede acceder al sistema y se crea una sesión donde se guardan datos que son utilizados hasta que salga del sistema o caduque su sesión, en caso de que alguna de estas dos condiciones no se cumplan el usuario no puede acceder al sistema; la clase LogoutAction limpia la sesión activa actual, se ejecuta una vez que el usuario se autentifica en el sistema y presiona la opción “Cerrar Sesión”, la clase CancelarAction es una clase muy general que se ejecuta en la mayoría de las ventanas que contengan un botón con la etiqueta “Cancelar” y su función es regresar a la pantalla principal, la clase BuscarUsuarioAction realiza una búsqueda en la base de datos con

algunos parámetros definidos por el usuario, esta clase se reutiliza en varias pantallas dentro de la aplicación, la clase `PermisosAction` se encarga de leer y modificar los permisos de los usuarios, esta función está disponible solo para los administradores.

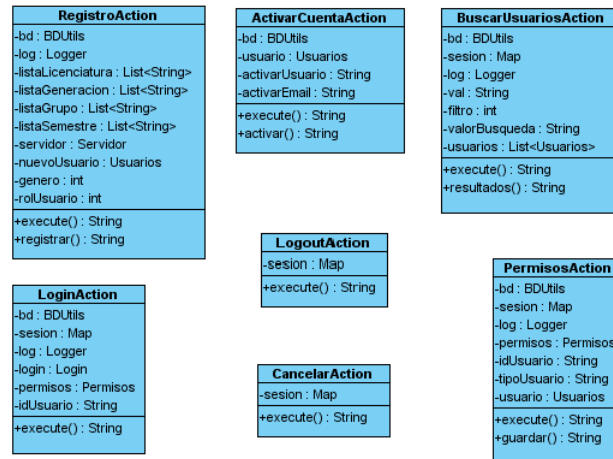


Figura 4.4: Clases `Action` para las actividades de registro, activación de cuenta, acceso, salir, cancelar, buscar usuario y permisos

La figura 4.5 muestra el diagrama de clases de los *Actions* `SocioeconomicaAction`, `ValoresAction`, `AptitudesAction` y `AfirmacionesAction`.

Estas clases realizan las operaciones de guardar y modificar las encuestas socioeconómica, valores, aptitudes y afirmaciones. Se ejecutan una vez que el usuario ha elegido una de ellas y ha contestado una sección de la misma, las encuestas están en secciones para una mejor vista para el usuario final.

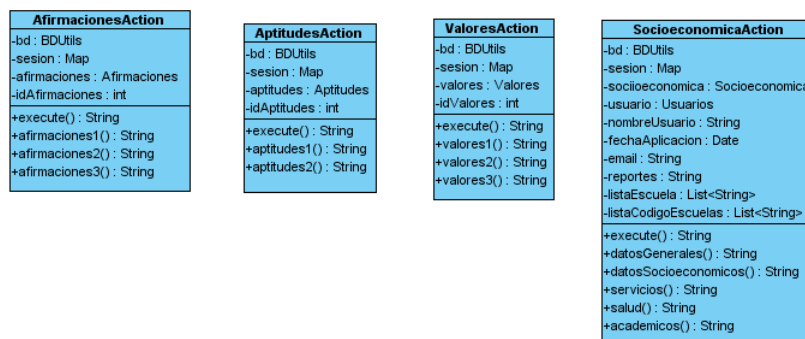


Figura 4.5: Clases `Action` para las encuestas socio económica, valores, aptitudes y afirmaciones

La figura 4.6 muestra el diagrama de clases de los *Actions* `RespaldarDatosAction`, `EditarDatosPersonalesAction` y `ReporteAction`. La clase `RespaldarDatosAction` es utilizada para respaldar la base de datos o tablas en particular seleccionadas por el usuario de tipo admi-

nistrador; la clase `EditarDatosPersonalesAction` permite a los usuarios modificar sus datos personales y la clase `ReporteAction` es empleada para crear todos los reportes y comprobantes realizados en la aplicación.

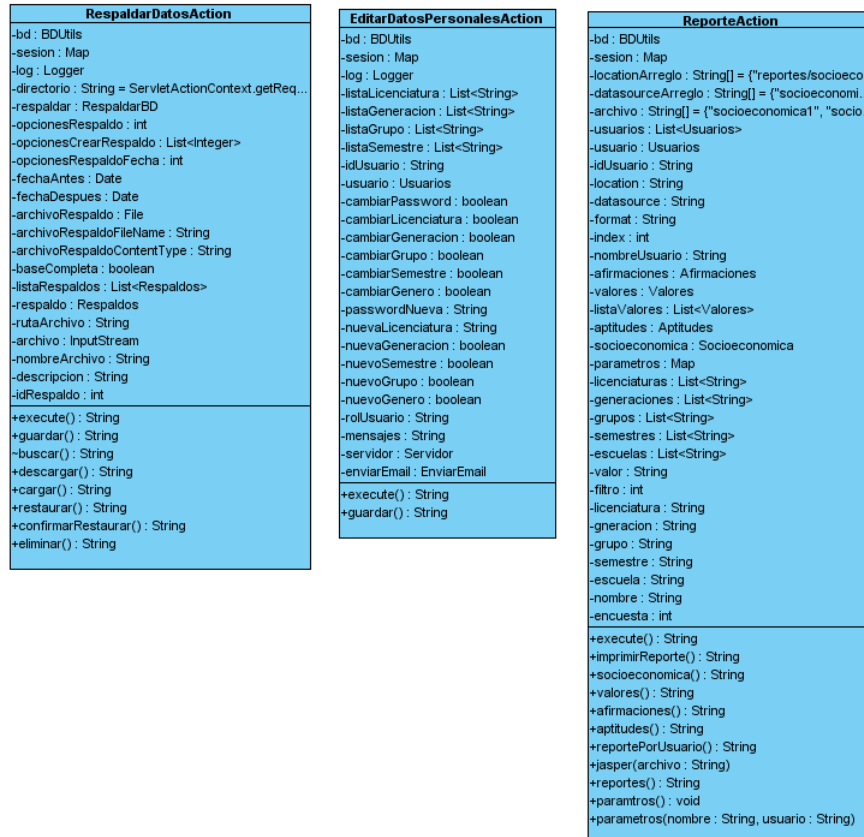


Figura 4.6: Clases Action para respaldo de datos, edición y reportes

La figura 4.7 muestra el diagrama de clases de los *Actions* `ConfigurarServidorAction`, `ConfigurarSistemaAction`, `EnviarEmailAction`, `ResultadosEncuestaAction` y `EliminarUsuarioAction`. La clase `ConfigurarServidorAction` es ejecutada cuando un administrador requiere cambiar algunas opciones del servidor como la dirección *IP*, nombre de dominio, el correo electrónico y su contraseña; la clase `ConfigurarSistemaAction` es necesaria para modificar algunas opciones dentro de la aplicación, la clase `EnviarEmailAction` trabaja con la *API JavaMail* y se utiliza para enviar algunas notificaciones a los usuarios de las aplicaciones, como el enlace para activar la cuenta y la notificación al cambiar algún dato personal del usuario; la clase `ResultadosEncuestaAction` se encarga de consultar las encuestas contestadas por los usuarios, mientras que la clase `EliminarUsuarioAction` es ejecutada al confirmar la eliminación de un usuario en el sistema por otro usuario de tipo administrador.

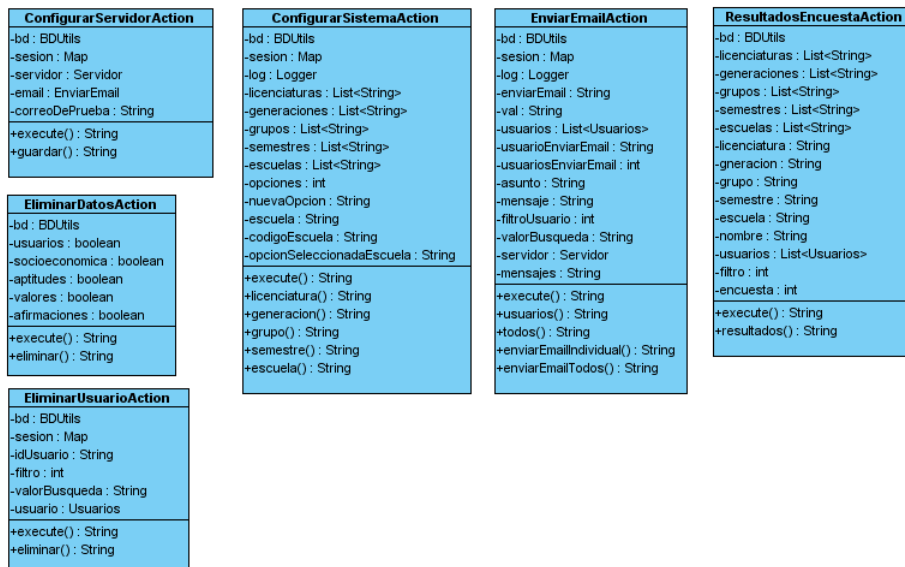


Figura 4.7: Clases Action para configurar al servidor y al sistema, enviar email y eliminar usuarios

Lógica de datos

A continuación se muestra el diagrama de clases de la lógica de datos, estas clases hacen un mapeo con la base de datos por medio de un *ORM* (Mapeo Objeto-Relacional). Como se puede observar las clases mapeadas implementan de la clase *Serializable* lo que permite que los objetos puedan viajar a través de una red y ser recuperados en su destino.

En la figura 3.8 se muestran algunas líneas de código referentes a las anotaciones, donde se realiza el mapeo de una tabla objeto-relacional de nombre usuarios, como se puede apreciar la anotación *@Table* se asocia a una tabla, *@id* a la llave primaria y *@Column* a un campo de la tabla. Otro aspecto importante es la implementación con la Interface *Serializable*, lo cual permite convertir los objetos en *bytes* para que éstos puedan viajar a través de la red y después ser recuperados sin ningún problema. En la figura 4.8 también se muestra el mapeo de una tabla con anotaciones y *JPA*.

```

@Entity
@Table(name = "usuarios")
public class Usuarios implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "id_usuario")
    private String idUsuario;
    @Column(name = "password")
    private String password;
    @Column(name = "nombre")
    private String nombre;
    @Column(name = "paterno")
    private String paterno;
}

```

Figura 4.8: Mapeo con *Anotaciones* de una tabla con *JPA*

Esta tecnología es un *ORM* que puede tener diferentes implementaciones: *hibernate*, *toplink*, *eclipselink* y otras más. Su configuración se realiza en un archivo *XML*. En este archivo se configura el gestor de la base de datos, así como el controlador, el usuario y contraseña, y la base de datos a la cual se conecta para realizar las transacciones dentro de la aplicación. En la figura 4.9 se muestra la configuración del archivo *persistence* para esta aplicación.

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="uaemPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <exclude-unlisted-classes>>false</exclude-unlisted-classes>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/BaseDeDatos"/>
      <property name="javax.persistence.jdbc.user" value="usuario"/>
      <property name="javax.persistence.jdbc.password" value="password"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
    </properties>
  </persistence-unit>
</persistence>

```

Figura 4.9: Archivo de configuración *JPA* con EclipseLink (*persistence.xml*)

El lenguaje de consulta para *JPA* se llama *JPQL* el cual es utilizado en lugar del *SQL* tradicional. En la figura 4.10 se muestra una consulta formada con este lenguaje para la obtención de los permisos de un determinado usuario asociado al parámetro *idUsuario*.

```

public String getIdUsuario() {
    return idUsuario;
}

public void setIdUsuario(String idUsuario) {
    this.idUsuario = idUsuario;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public String getNombre() {
    return nombre;
}

```

Figura 4.11: Métodos `get()` y `set()` para interfaz con JPA

```

public Permisos getPermisosUsuario(String idUsuario) {
    String consultaString = "SELECT p FROM Permisos p INNER JOIN p.usuarios u WHERE u.idUsuario=" + idUsuario + "'";
    try {
        consulta = getEm().createQuery(consultaString);
        permisos = (Permisos) consulta.getSingleResult();
        if (permisos != null) {
            return permisos;
        } else {
            return null;
        }
    } catch (NoResultException e1) {
        return null;
    } catch (NonUniqueResultException e2) {
        return null;
    }
}

```

Figura 4.10: Consulta con lenguaje *JPQL*

Esta técnica facilita la lógica de acceso y manipulación a los datos mediante métodos `get()` y `set()` de la tabla mapeada para cada uno de sus campos. Como se observa en la figura 4.11 cada atributo de la clase mapeada hay un método `get()` y `set()` a los cuales se puede acceder con una instancia a la clase.

Con las implementaciones de *JPA* están definidos métodos para el acceso a los datos (*INSERT*, *UPDATE* y *DELETE*). Dependiendo de la implementación estos métodos pueden diferenciar de nombre pero su función será la misma. En este caso se implementa con *EclipseLink* y la clase *EntityManager* donde sus métodos son: ***persist(Object)*** para guardar el objeto en la base de datos, ***merge(Object)*** que es utilizado para actualizar un objeto y ***remove(Object)*** para eliminarlo. En la figura 4.12 se muestra un método que realiza la persistencia de un objeto a la base de datos.


```

public void registrarUsuario(Usuarios nuevoUsuario) {
    usuario = new Usuarios();
    usuario.setIdUsuario(nuevoUsuario.getIdUsuario());
    usuario.setPassword(nuevoUsuario.getPassword());
    usuario.setNombre(nuevoUsuario.getNombre());
    usuario.setPaterno(nuevoUsuario.getPaterno());
    usuario.setMaterno(nuevoUsuario.getMaterno());
    usuario.setEmail1(nuevoUsuario.getEmail1());
    usuario.setEmail2(nuevoUsuario.getEmail2());
    usuario.setLicenciatura(nuevoUsuario.getLicenciatura());
    usuario.setGeneracion(nuevoUsuario.getGeneracion());
    usuario.setGrupo(nuevoUsuario.getGrupo());
    usuario.setGenero(nuevoUsuario.getGenero());
    usuario.setSemestre(nuevoUsuario.getSemestre());
    usuario.setTelefono1(nuevoUsuario.getTelefono1());
    usuario.setTelefono2(nuevoUsuario.getTelefono2());
    usuario.setCelular(nuevoUsuario.getCelular());
    usuario.setNotas(nuevoUsuario.getNotas());
    usuario.setTipoUsuario(nuevoUsuario.getTipoUsuario());
    usuario.setRolUsuario(nuevoUsuario.getRolUsuario());
    usuario.setEliminar(true);
    usuario.setActivo(false);
    getEm().persist(usuario);
}

```

Figura 4.12: Persistencia de un objeto Usuarios

En la figura 4.13 se muestra el diagrama de clases del modelo de datos que hace referencia una base de datos relacional. Mas adelante se describiran mas a detalle cada una de las clases expuestas en el modelo de datos.

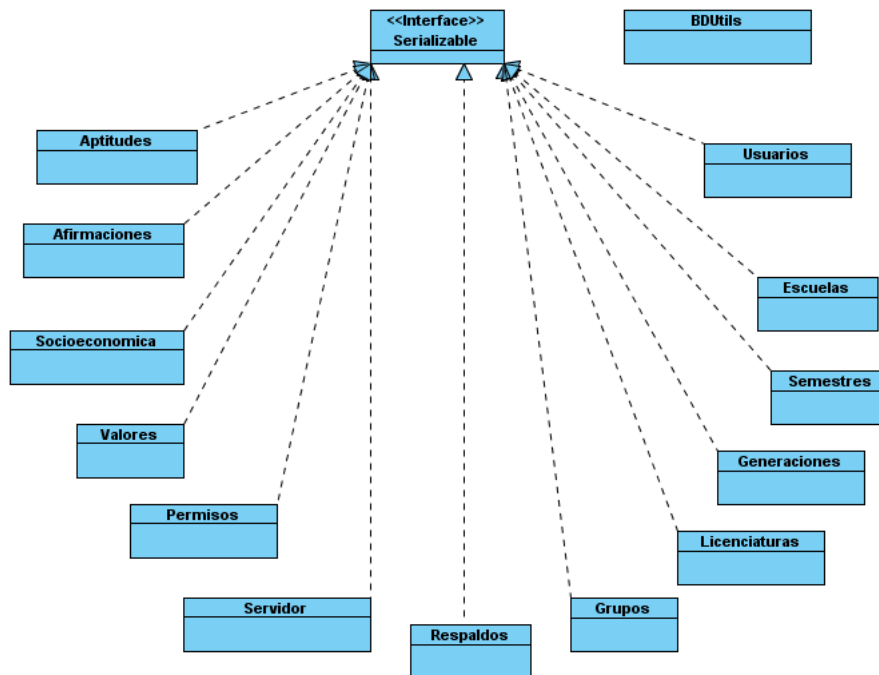


Figura 4.13: Diagrama de clases lógica de datos

A continuación se describen cada una de las clases, que por espacio no se puede mostrar en la figura 4.13.

En la figura 4.14 se muestran las clases Socioeconomica, Valores, Aptitudes y Afirmaciones con sus respectivos atributos, estas clases implementan *JPA* y hacen referencia a las tablas en

la base de datos asociadas a las encuestas.

Socioeconomica	Valores	Aptitudes	Afirmaciones
-idSocioeconomica : int	-idValores : Integer	-idAptitudes : int	-idAfirmaciones : int
-fechaAplicacion : Date	-tolernciaImportante : Integer	-operacionesMatematicas : int	-tomarApuntes : int
-fechaNacimiento : Date	-toleranciaUtil : Integer	-resolucionProblemas : int	-distribuirTiempo : int
-sexo : boolean	-toleranciaDuradera : Integer	-solucionEcuaciones : int	-dificultadEsforzarse
-edad : int	-toleranciaIncondicionada : Integer	-lenguajeExpresado : int	-ociosoNoEstudiar : int
-estadoCivil : String	-toleranciaDeseable : Integer	-operacionesSinCalculadora : int	-pierdoTiempoConAmigos : int
-notijos : int	-honestidadImportante : Integer	-resolverAcertijos : int	-estudiarAntesFiesta : int
-lugarProcedencia : String	-honestidadUtil : Integer	-lenguajeAlgebraico : int	-cansadoParaEstudiar : int
-domicilioCompleto : String	-honestidadDuradera : Integer	-localizarErroresHojaTabulada : int	-estudiarMateriasDisgustan : int
-telefono : String	-honestidadIncondicionada : Integer	-operacionesConCalculadora : int	-intentoEstudiar : int
-celular : String	-honestidadDeseable : Integer	-trabajarSistematicamente : int	-repararMentalmente : int
-email : String	-respetoImportante : Integer	-interpretarInformacionNumerica : int	-murmuroPalabras : int
-trabajos : boolean	-respetoUtil : Integer	-ordenCosasAprendidas : int	-relacionarUnirMaterial : int
-nombreEmpresa : String	-respetoDuradera : Integer	-problemasSinPelear : int	-sistematizarLoAprendido : int
-direccionEmpresa : String	-respetoIncondicionada : Integer	-desarrollarProyectos : int	-amplioLoDicho : int
-telefonoEmpresa : String	-respetoDeseable : Integer	-tomarDecisionesEficaces : int	-formarOpinion : int
-horaEntrada : String	-confianzaImportante : Integer	-ordenarLibros : int	-noRecordarLoAprendido : int
-horaSalida : String	-confianzaUtil : Integer	-distribuirTiempo : int	-recordarLoEstudiado : int
-ingresoMensual : Float	-confianzaDuradera : Integer	-organizarEventos : int	-estudiarFumando : int
-nombrePadre : String	-confianzaIncondicionada : Integer	-trabajarEnEquipo : int	-estudiarMaterial/VariasVeces : int
-vivePadre : Boolean	-confianzaDeseable : Integer	-ordenarActividades : int	-investigarSignificados : int
-nivelMaximoEstudiosPadre : String	-lealtadImportante : Integer	-distribuirTrabajo : int	-tendenciaAMemorizar : int
-trabajoPadre : String	-lealtadUtil : Integer	-decisionesEnEquipo : int	-repetirLoLeido : int
-direccionPadre : String	-lealtadDuradera : Integer	-leerLargoPeriodo : int	-encontrarPuntosImportantes : int
-telefonoPadre : String	-lealtadincondicionada : Integer	-comprenderDocumentos : int	-notasTomadasSinImportancia : int
-ingresoMensualPadre : Float	-lealtadDeseable : Integer	-comprenderArticuloInvestigacion : int	-dudasVuelvoALeer : int
-esperaPadre : String	-disciplinaImportante : Integer	-extraerIdeasPrincipales : int	-pierdoElHilo : int
-nombreMadre : String	-disciplinaUtil : Integer	-realizarEsquemas : int	-preguntoDudas : int
-viveMadre : Boolean	-disciplinaDuradera : Integer	-resumiDocumentos : int	-facilConcentracion : int
-nivelMaximoEstudiosMadre : String	-disciplinaIncondicionada : Integer	-indagarBuscarInvestigacion : int	-olvidoLoLeido : int
-trabajoMadre : String	-disciplinaDeseable : Integer	-hacerLecturaRapida : int	-piensoOtrasCosas : int
-direccionMadre : String	-amistadImportante : Integer	-habilExponer : int	-estudiarMeAburro : int
-telefonoMadre : String	-amistadUtil : Integer	-trasladarSituacionesReales : int	-ruidoMeDistrae : int
-ingresoMensualMadre : Float	-amistadDuradera : Integer	-imaginarSituaciones : int	-periodosEstudioCortos : int
-esperaMadre : String	-amistadincondicionada : Integer	-valorarMensajes : int	-estudioInmediato : int
-nombreHermano1 : String	-amistadDeseable : Integer	-expresarteEnClase : int	-interrupcionesAlEstudiar : int
-sexoHermano1 : Boolean	-iniciativaImportante : Integer	-expresarOpiniones : int	-planearRespuestasExamen : int
-edadHermano1 : Integer	-iniciativaUtil : Integer	-desarrollarExposiciones : int	-repararRespuestasExamen : int
-ocupacionHermano1 : String	-iniciativaDuradera : Integer	-expresarIdeasFormaEscrita : int	-hacerPreguntasExamen : int
-sueldoHermano1 : Float	-iniciativaIncondicionada : Integer	-expresarseFormaClara : int	-planearRespuestas : int
-nombreHermano2 : String	-iniciativaDeseable : Integer	-elegirPalabrasPrecisas : int	-resuelvoExamenesFacil : int
-sexoHermano2 : Boolean	-simpaticoImportante : Integer	-defenderOpinion : int	-calmoNerviosoExamen : int
-edadHermano2 : int	-simpaticoUtil : Integer	-expresarIdeasPorEscrito : int	-nerviosipidenExamen : int
-ocupacionHermano2 : String	-simpaticoDuradera : Integer	-expresarAprendisteLeyendo : int	-inquietudAlEstudiar : int
-sueldoHermano2 : Float	-simpaticoIncondicionada : Integer	-habilidadOrtografica : int	-timidezImpidePreguntar : int
-nombreHermano3 : String	-simpaticoDeseable : Integer	-usuarios : Usuarios	-aptitudMaestros : int
-sexoHermano3 : int	-liderImportante : Integer		-maestroPregunta : int
-edadHermano3 : int	-liderUtil : Integer		-prefieroAislarme : int
-ocupacionHermano3 : String	-liderDuradera : Integer		-malestarNoSerPrimero : int
-salarioHermano3 : Float	-liderIncondicionada : Integer		-usuarios : Usuarios
-nombreHermano4 : String	-liderDeseable : Integer		
-sexoHermano4 : Boolean	-seguroImportante : Integer		
-edadHermano4 : int	-seguroUtil : Integer		
	-seguroDuradera : Integer		
	-seguroIncondicionada : Integer		
	-seguroDeseable : Integer		

Figura 4.14: Base de datos mapeada con JPA parte 1

En la figura 4.15 se muestran las clases Usuarios, Servidor, Escuelas, Generaciones, Grupos, Semestres, Licenciaturas, Respaldos y Permisos con sus respectivos atributos, estas clases implementan *JPA* y hacen referencia a las tablas en la base de datos asociadas a las encuestas.

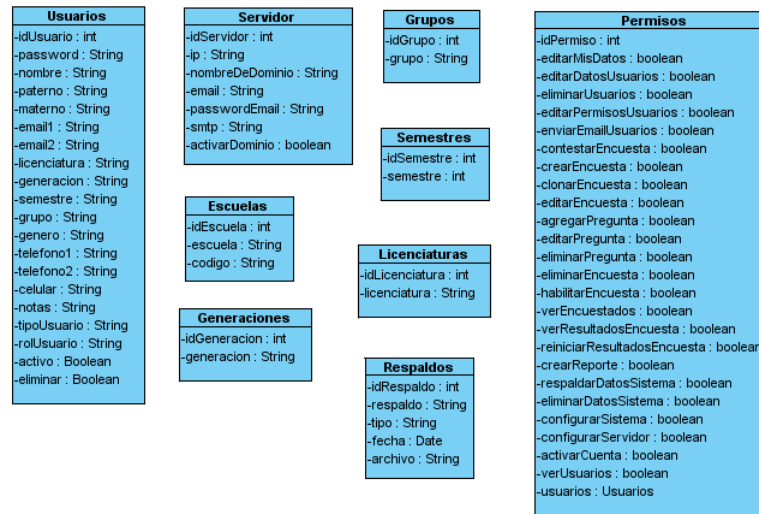


Figura 4.15: Base de datos mapeada con JPA parte 2

En la figura 4.16 se muestran la clase BDUtils la cual contiene todos los métodos para el acceso a la base de datos utilizados en la aplicación.



Figura 4.16: Clase para manipular la base de datos con JPA

4.3. Descripción de la Vista

En esta sección se describe de forma general lo que se refiere a la vista y *Frameworks* utilizados para darle presentación y funcionalidad a la misma. Otro aspecto importante es que la vista no sólo puede ser una página *Web* sino todo lo que se presente visualmente al usuario, por lo que en esta sección se pueden destacar los reportes que aunque no presenten

una comunicación síncrona entre la aplicación y el usuario, muestran datos importantes para los usuarios. Los reportes para la aplicación pueden ser exportados a formatos en *PDF* y a hojas de cálculo. Cabe también mencionar que algunos *Frameworks* aquí descritos también tienen cierta configuración con *Struts 2*, que es manejada por el controlador, sin embargo, interactúan más con la capa de la vista por lo que se decidió sean descritos en esta sección.

Ahora bien, de forma general la vista es la interfaz entre el usuario y el modelo de la aplicación, por lo general son *JSP* (*Java Server Pages*) para *Java*. Las *JSP* contienen formularios los cuales envían y muestran los datos asociados a las transacciones, una transacción se traduce como una operación en la base de datos. En la vista se pueden encontrar diferentes componentes los cuales interactúan con el usuario para realizar determinado proceso. Dentro de los componentes podemos encontrar los siguientes: Formularios, campos de texto, botones, checkbox, radio button, validadores y paginadores entre otros.

4.3.1. Vista - Result

En *Struts 2* la vista está asociada a un *Result*. Un *Result* es la respuesta al *Action* y que muestra como resultado una *JSP* (en otros casos un programa asociado a un tipo de archivo pero que es ejecutado desde el navegador, por ejemplo, un archivo *PDF*). Un *Action* debe tener como mínimo un *Result* o más dependiendo de la lógica de negocios o al evento ejecutado al presionar un botón o incluso algún otro componente de la vista.

4.3.2. Etiquetas Struts 2

Las etiquetas de *Struts 2* permiten crear la vista en una *JSP*, también se pueden utilizar etiquetas *HTML*, aunque hay algunas funcionalidades interesantes y algunos pros y contras en las etiquetas definidas en el *Framework* ya mencionado. Ahora bien, las etiquetas por omisión de *Struts 2* vienen definidas por plantillas hechas en Freemaker que a pesar de que se pueden llegar a modificar hay que analizar el lenguaje mencionado por lo que el tiempo de desarrollo se extendería al estudiar sobre el tema. El uso de las etiquetas de *Struts 2* ocupando las plantillas por omisión ocasionan que sea un tanto difícil poder ordenarlas en la *JSP*, puesto que están delimitadas por tablas no visibles para el programador lo que ocasiona un mal acoplamiento en la *JSP* a menos que se tenga bien en cuenta como está definida la plantilla y así se tendría que anidar muchas tablas para lograr un formato adecuado del formulario. Por otro lado, las etiquetas de *Struts 2* se asocian perfectamente a la internacionalización desde la propiedad *label* o más directamente la propiedad *key* de la etiqueta. La figura 4.17 muestra una etiqueta de *Struts 2* y su equivalencia en *HTML*.

```

TAG DE STRUTS 2
<s:select key="personBean.sport" list="sports" />

HTML DE UNA TAG DE STRUTS 2
<tr>
<td class="tdLabel">
<label for="save_personBean_sport" class="label">Favorite sport:</label>
</td>
<td>
<select name="personBean.sport" id="save_personBean_sport">
  <option value="football">football</option>
  <option value="baseball">baseball</option>
  <option value="basketball" selected="selected">basketball</option>
</select>
</td>
</tr>

```

Figura 4.17: etiquetas de *Struts 2* y equivalencia en *HTML*

4.3.3. Tiles

La instalación se realiza en el archivo *Web.xml* de la aplicación. En la figura 4.18 se muestra la integración del *Framework Tiles* en el archivo *Web.xml* de la aplicación.

```

<!--TILES-->
<context-param>
  <param-name>org.apache.tiles.impl.BasicTilesContainer.DEFINITIONS_CONFIG</param-name>
  <param-value>/WEB-INF/tiles.xml</param-value>
</context-param>

<listener>
  <listener-class>org.apache.struts2.tiles.StrutsTilesListener</listener-class>
</listener>

```

Figura 4.18: Instalación y configuración del *Framework Tiles* para *Struts 2*

Mientras que la implementación es a través de un archivo nombrado *tiles.xml* que también se hace referencia en el archivo *Web.xml*. Este archivo contiene todas las definiciones de las plantillas que usa la aplicación. En la figura 4.19 se muestra una plantilla con el nombre de *templatePrincipal*, y como se observa hay varios atributos los cuales están asociados a una página.

```

<tiles-definitions>
  <definition name="templatePrincipal" template="principal/templatePrincipal.jsp">
    <put-attribute name="title" value="Bienvenidos" />
    <put-attribute name="header" value="/principal/header.jsp" />
    <put-attribute name="body" value="/principal/login.jsp" />
    <put-attribute name="menu" value="/principal/inicio.jsp" />
    <put-attribute name="footer" value="/principal/footer.jsp" />
  </definition>

```

Figura 4.19: Plantilla de *Tiles*

Para definir una plantilla es a través de una *JSP* en la cual se insertan etiquetas que tiles interpreta desde el archivo *tiles.xml*. Cada etiqueta hace referencia al atributo. En la definición es donde se incrustan las correspondientes páginas.

```

<table width="100%" border="0" >
  <tr>
    <td colspan="2" ><tiles:insertAttribute name="header"/></td>
  </tr>
  <tr valign="top" >
    <td width="15%" >
      <table>
        <tr>
          <td ><tiles:insertAttribute name="menu"/></td>
        </tr>
      </table>
    <td width="85%">
      <table>
        <tr>
          <td><tiles:insertAttribute name="body"/></td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td colspan="2" ></td>
  </tr>
</table>

```

Figura 4.20: Plantilla de Tiles en una *JSP*

4.3.4. Validación

La validación es una parte muy importante en una aplicación *Web* ya que permite detectar errores y evitar inconsistencias, así mismo es útil para obligar al usuario introducir información en caso de que un dato sea requerido por reglas del negocio o por el diseño de la base de datos. Hay dos tipos de validación, del lado del servidor y del lado del cliente.

Validación del lado del servidor en Struts 2

Struts 2 cuenta con un interceptor que se encarga de las siguientes validaciones: correo electrónico, máximo y mínimo número de caracteres, un número en un rango, datos requeridos, datos idénticos, expresiones, etc., Sin embargo, por ser ejecutado del lado del servidor es una forma de validación más lenta y por lo tanto no muy recomendable salvo excepciones donde la validación requiera acceder a una base de datos para buscar datos.

Analicemos un ejemplo del por qué es más lento este tipo de validación. Tenemos un formulario donde se requieren validar 5 campos dentro de los cuales se debe validar un correo electrónico, un número entero, dos campos con los mismos datos y uno con una fecha. Al momento de enviar el formulario hay un tiempo de espera en que el servidor recibe los datos y realiza la correspondiente validación, en caso de algún error regresa a la misma página indicando que algo está incorrecto, esto lo ejecutará una y otra vez hasta que el formulario tenga una validación correcta de todos los campos y el servidor regrese la página correspondiente a recibir los datos de forma correcta.

Validación del lado del cliente

La validación del lado del cliente es más rápida ya que el formulario no se enviará hasta que la validación de todos los campos sea correcta y el servidor no procesará los datos una y otra vez, por lo que el usuario tiene una retroalimentación casi inmediata. Esta forma de validación se puede incorporar a cualquier página *Web* por medio de *Javascript* incluso hay diversas

implementaciones con *JQuery* y *CSS* para darle una mejor presentación. A continuación se hace mención de algunos *Frameworks* para validación.

Validation Engine

Validation Engine[19] es una solución para realizar validación del lado del cliente implementado con *JQuery* y *CSS* por lo que permite modificar su presentación. Puedes realizar las siguientes validaciones: dato requerido, correo electrónico, fechas, sólo letras, solo números, dirección de internet (*HTTP* y *FTP*), dirección *IP*, etc. además permite construir más validaciones por medio de expresiones regulares que el usuario puede generar desde su archivo .js de configuración.

4.3.5. JQuery

En la figura 4.21 se muestra algunas líneas de código donde se importan las bibliotecas necesarias para ejecutar *JQuery*.

```
<link rel="stylesheet" href="jquery/jqueryui 1.9/ui/themes/base/jquery.ui.all.css">
<link rel="stylesheet" href="jquery/jqueryui 1.9/ui/themes/base/jquery.ui.css">
<link rel="stylesheet" href="jquery/jqueryui 1.9/ui/themes/base/demos.css">
<link type="text/css" href="css/errorsTags.css" rel="stylesheet" />
<link type="text/css" href="css/general.css" rel="stylesheet" />
<script type="text/javascript" src="jquery/jqueryui 1.9/jquery-1.5.1.js"></script>
<script type="text/javascript" src="jquery/jqueryui 1.9/ui/jquery-ui.js"></script>
<script type="text/javascript" src="jquery/jqueryui 1.9/ui/jquery.ui.core.js"></script>
<script type="text/javascript" src="jquery/jqueryui 1.9/ui/jquery.ui.widget.js"></script>
<script type="text/javascript" src="jquery/jqueryui 1.9/ui/jquery.ui.button.js"></script>
<script type="text/javascript" src="jquery/jqueryui 1.9/ui/jquery.ui.position.js"></script>
```

Figura 4.21: Archivos necesarios para ejecutar *JQuery*

Este *Framework* está presente en casi todas las páginas de la aplicación para mejorar la intuición e interactividad a los usuarios. A continuación se presentan algunas herramientas desarrolladas con esta tecnología y utilizadas en el desarrollo del sistema descrito.

Tooltip

Un *tooltip* es una ayuda visual que aparece al presionar o pasar el puntero del mouse en cierto componente como un campo de texto, y el cual brinda información adicional al usuario respecto a dicho componente. En esta aplicación se utiliza *JQuery* para llevar a cabo esta función y darle un grado visual mejor. En la figura 4.22 se muestra un ejemplo de esta aplicación.

The image shows a registration form with four input fields: 'No. de Cuenta*', 'Password*', 'Confirma Password*', and 'Nombre(s)*'. The 'Password*' field contains ten black dots. A tooltip box is positioned to the right of the password field, containing the text 'Entre 6 y 15 caracteres'.

Figura 4.22: Tooltip con *JQuery*

Calendario (Datepicker)

En la aplicación desarrollada fue necesario introducir algunas fechas, por lo que el usuario puede de forma sencilla introducir en un campo de texto una determinada fecha en determinado formato, sin embargo, si el usuario por equivocación introduce una fecha incorrecta o en diferente formato esto puede ocasionar que el sistema no responda como debe ser o los datos no sean consistentes. Para solucionar esto se utilizó una aplicación (*widget*) para representar un calendario en el respectivo campo. En la figura 4.23 se muestra el calendario. De esta forma se previene lo ya mencionado por que el dato introducido no puede ser más que una fecha.

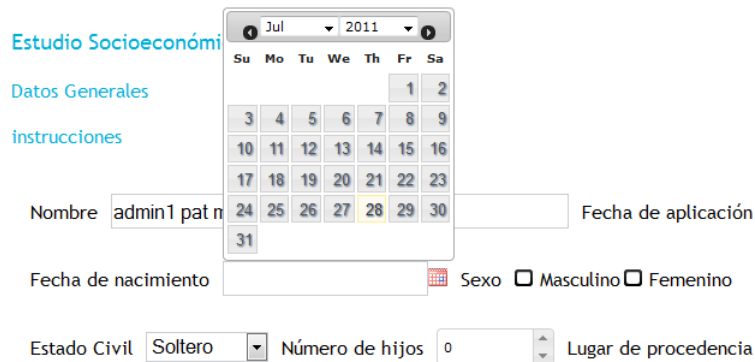


Figura 4.23: Calendario con *jQuery (Datepicker)* utilizado en la aplicación

Ezmark

Ezmark[5] es una aplicación desarrollada con *jQuery* y que permite cambiar la apariencia de determinados componentes de la *JSP*. En esta caso para modificar los checkbox y radiobutton utilizados en los formularios. En la figura 4.24 se muestran los componentes ya mencionados con *Ezmark*.

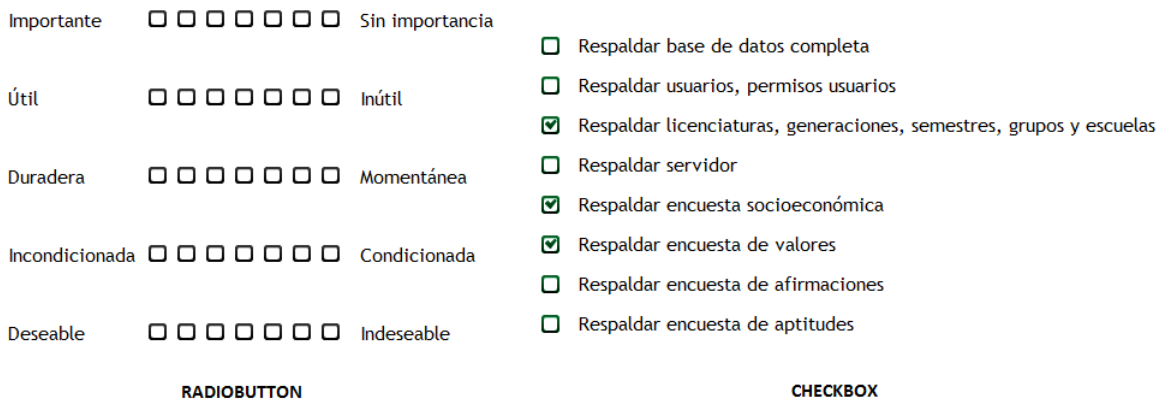


Figura 4.24: Checkbox y Radiobuttons con *Ezmark*

Mientras que en la figura 4.25 se muestran los componentes sin *Ezmark*.

Importante	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> Sin importancia	<input type="checkbox"/> Respalдар base de datos completa
Útil	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> Inútil	<input type="checkbox"/> Respalдар usuarios, permisos usuarios
Duradera	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Momentánea	<input checked="" type="checkbox"/> Respalдар licenciaturas, generaciones, semestres, grupos y escuelas
Incondicionada	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Condicionada	<input type="checkbox"/> Respalдар servidor
Deseable	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> Indeseable	<input checked="" type="checkbox"/> Respalдар encuesta socioeconómica
		<input checked="" type="checkbox"/> Respalдар encuesta de valores
		<input type="checkbox"/> Respalдар encuesta de afirmaciones
		<input type="checkbox"/> Respalдар encuesta de aptitudes
	RADIOBUTTON	CHECKBOX

Figura 4.25: Checkbox y Radiobuttons sin *Ezmark*

Editor texto enriquecido

Un editor de texto enriquecido es una aplicación la cual permite aplicar formato y que subsiste en una página de internet, hay diferentes implementaciones, pero para esta aplicación se utilizó *TinyMCE*[26] desarrollada con *Javascript* de licencia *GNU* que permite convertir una etiqueta de tipo *Textarea* e un editor con diferentes herramientas para aplicar formatos y estilos. En la figura 4.26 se muestra el editor.

Escriba el asunto

Escriba el mensaje

Este es un texto de prueba|

Aceptar

Figura 4.26: Editor de texto enriquecido

4.3.6. CSS

Para realizar el diseño de las páginas se utilizó *CSS* en la figura 4.27 se muestra una hoja de estilos para tres componentes.

```
.submit{
  color: #FFFFFF;
  background: #1ea54b ;
  border: 1px solid #000;
  height:25px;
  width:120px;
}

.area{
  border: 2px solid #CBCFD4 ;
  font-size: medium;
}

.field{
  height:25px;
  border: 1px solid #CBCFD4;
  font-size: medium;
}
```

Figura 4.27: Ejemplo de código en *CSS*

4.3.7. Paginación

La paginación se vuelve una tarea difícil de implementar puesto que dependiendo del número de resultados ya sea de una búsqueda o un catálogo se pueden generar numerosas páginas con contenido dinámico asociado a una base de datos. Por lo que se requirio el uso de algunas herramientas mas adelante descritas.

Display Tag

Display Tag[2] es una biblioteca de código abierto que provee etiquetas de alto contenido *Web* que trabajan con el patrón MVC. Para entender mejor su función hay que mencionar que sólo muestra tablas por medio de una lista que recibe como parámetro y soporta funcionalidades como ordenación de los registros, ordenación, agrupación, decoración e incluso exportación a varios formatos como son *PDF* u Hoja de cálculo, aunque la implementación básica no soporta esta última característica.

JMesa

JMesa[10] es una tabla dinámica en *HTML* que permite filtrar, ordenar, paginar, exportar y editar datos, estos dos últimos requieren de una configuración avanzada y *Ajax* para la edición de los datos. A diferencia de *Display Tag* la decoración incluye *JQuery* por lo que se presenta una implementación amistosa e interactiva. Aunque su instalación incluye más elementos que *Display Tag* es una buena opción para la paginación, en lo que respecta al uso es muy similar puesto que pasa una lista como parámetro. La figura 4.28 muestra una salida producida por *JMesa*.

Nombre ▲	Paterno	Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1	Eliminar
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com	Eliminar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com	Eliminar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com	Eliminar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com	Eliminar
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1		belinasan_dom@hotmail.com	Eliminar

Figura 4.28: Tabla con *JMesa*

Otra diferencia que posee este *Framework* sobre *Display Tag* es el uso de algunas funciones que se habilitan desde código *Javascript* las cuales sirven para efectuar algunos procedimientos adicionales como la exportación a algunos formatos. En la figura 4.29 se muestran funciones con *Javascript* implementadas por *JMesa*.

```
<script type="text/javascript">
  function onInvokeAction(id) {
    createHiddenInputFieldsForLimitAndSubmit(id);
  }
  function onInvokeExportAction(id) {
    var parameterString = createParameterStringForLimit(id);
    location.href = 'BuscarUsuarioAction?' + parameterString;
  }
</script>
```

Figura 4.29: Funciones *Javascript* de *JMesa*

En la figura se muestran las etiquetas con las que se permite crear una tabla con *JMesa*, estas etiquetas van incrustadas en una *JSP*.4.30

```
<form action="ResultadosBuscarUsuarioAction">
  <s:hidden name="filtro"></s:hidden>
  <s:hidden name="valorBusqueda"></s:hidden>
  <jmesa:tableModel exportTypes="pdf,jexcel,excel,csv" id="id_usuarios" items="{usuarios}" var="usuarios" >
    <jmesa:htmlTable>
      <jmesa:htmlRow >
        <jmesa:htmlColumn property="nombre" title="Nombre"></jmesa:htmlColumn>
        <jmesa:htmlColumn property="paterno" title="Paterno"/>
        <jmesa:htmlColumn property="materno" title="Materno"/>
        <jmesa:htmlColumn property="rolUsuario" title="Rol Usuario"/>
        <jmesa:htmlColumn property="licenciatura" title="Licenciatura"/>
        <jmesa:htmlColumn property="generacion" />
        <jmesa:htmlColumn property="semestre" />
        <jmesa:htmlColumn property="grupo" />
        <jmesa:htmlColumn property="telefono1" />
        <jmesa:htmlColumn property="email1" />
      </jmesa:htmlRow>
    </jmesa:htmlTable>
  </jmesa:tableModel>
</form>
```

Figura 4.30: Etiquetas de *JMesa*

4.3.8. Reportes (Ireport y Jasper Reports)

En la aplicación se utilizó para la generación de reportes, la herramienta de *Ireport* y *Jasper*

Reports, la cual consta de una plantilla con extensión *.jrxml* y un archivo binario con extensión *.jasper* por reporte. La plantilla es creada desde *Ireport* como se muestra en la figura 4.31. Este archivo tiene la extensión *.jrxml* el cual contiene código en *XML* figura 4.31, el archivo con extensión *.jasper* es creado al compilar el archivo *.jrxml* y es el mismo que interpretan las librerías de *Jasper Reports* para generar un archivo de salida y mostrar así el reporte en un archivo *PDF* o una hoja de cálculo.

The screenshot shows a report template for the Universidad Autónoma del Estado de México, Centro Universitario Zumpango. The header features the university's logo and name. Below the header, there are several data fields: 'REPORTE: SOCIOECONOMICA 1', 'FECHA APLICACIÓN: (new java.text)', 'ALUMNO: \${parametros.nombre}', and 'NO. CUENTA: \$P'. The main body of the form contains fields for 'Nombre: \${nombreUsuario}', 'Estado Civil: \${estadoCivil}', 'No. de Hijos: \$F', 'Lugar de Procedencia: \${lugarProcedencia}', 'Domicilio: \${domicilioCompleto}', 'Teléfono: \${telefono}', 'Celular: \${celular}', 'Email: \${email}', and '¿Trabajas?: \${trabajas}'. The form is displayed within a window titled 'Detail 1'.

Figura 4.31: Plantilla *Ireport*

```

<staticText>
  <reportElement x="0" y="120" width="75" height="20"/>
  <textElement verticalAlignment="Middle">
    <font fontName="SansSerif" size="12" isBold="true" pdfFontName="Helvetica-Bold"/>
  </textElement>
  <text><![CDATA[ALUMNO:   ]]></text>
</staticText>
<staticText>
  <reportElement x="386" y="120" width="82" height="20"/>
  <textElement textAlignment="Right" verticalAlignment="Middle">
    <font fontName="SansSerif" size="12" isBold="true" pdfFontName="Helvetica-BoldOblique"/>
  </textElement>
  <text><![CDATA[NO. CUENTA:   ]]></text>
</staticText>

```

Figura 4.32: Código *XML* de una plantilla *Ireport*

El proceso de los reportes es llevado a cabo desde un *Action* el cual es encargado de leer el archivo, compilarlo, procesarlo y exportarlo al formato deseado. En la figura 4.33 se muestran algunas líneas de código del *Action* ReporteAction el cual se encarga de compilar el archivo *.jrxml*.

```

public boolean.jasper(String archivo) {
    String jrxml,.jasper;
    String path = ServletActionContext.getRequest().getSession().getServletContext().getRealPath("/");
    jrxml = path + "reportes/" + archivo + ".jrxml";
   .jasper = path + "reportes/" + archivo + ".jasper";
    try {
        JasperCompileManager.compileReportToFile(jrxml, .jasper);
        return true;
    } catch (Exception e) {
        return false;
    }
}

```

Figura 4.33: Compilación de un archivo *.jrxml*

Esta herramienta permite la incorporación de gráficas de diferentes tipos, por ejemplo, barras de pastel como las mostradas en la figura 4.34.

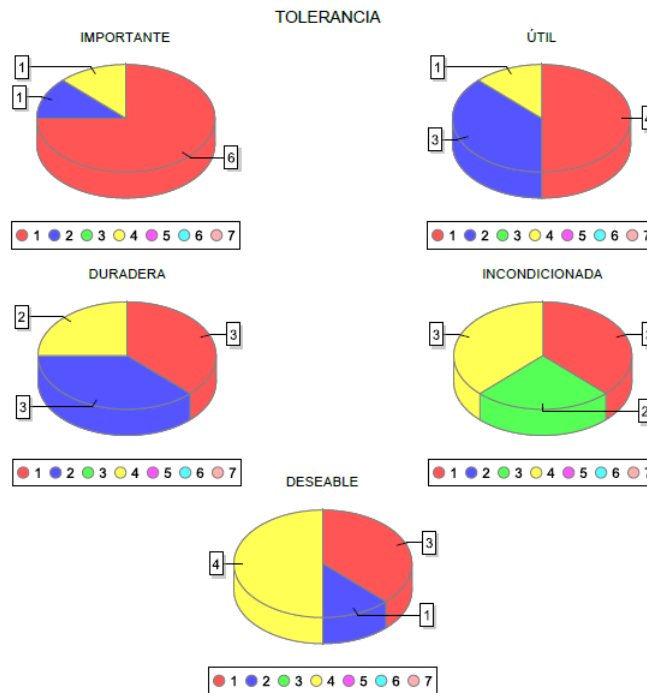


Figura 4.34: Gráficas de pastel en *Ireport*

4.4. Descripción del Controlador

4.4.1. Struts 2

A continuación se describen algunas herramientas que están asociadas al *Framework* como tal, y que se configuran y ejecutan desde el *Framework Struts 2* por lo que se decidió ponerlas en esta sección.

Struts 2 es un *Framework* basado en el patrón de diseño *MVC*. Y es en realidad el *Framework* el que realizará el trabajo del controlador, puesto es el encargado como intermediario de la vista y el modelo del procesamiento de los datos por medio de un filtro que

se configura en el archivo *Web.xml* de la aplicación. El trabajo de este filtro es recibir una petición, revisar que petición es y posteriormente de acuerdo al flujo de los datos (lógica de negocio) enviar una respuesta.

Filtro de Struts 2

Como se mencionó anteriormente el filtro se configura en el archivo *Web.xml* y cambia de acuerdo a la versión del *Framework*, por lo que hay que tener sumo cuidado al momento de cambiar las librerías puesto que el filtro puede cambiar también. En la figura 4.35 se muestra algunas líneas de código que configura el filtro del *Framework* descrito.

```
<filter>
  <filter-name>struts2</filter-name>
  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Figura 4.35: Filtro de *Struts 2*

4.4.2. Sesión

La sesión de usuario comienza cuando el usuario accede a la aplicación y termina cuando sale de esta. En una aplicación *Web* se vuelve una parte muy importante puesto que, en la sesión se almacenan datos que se usan entre las diferentes páginas que el usuario visite. Una sesión puede expirar de forma automática en determinado tiempo o de forma manual cuando el usuario lo decida. La expiración de sesión automática se presenta para brindar mayor seguridad al sistema puesto que al pasar por un tiempo de inactividad determinado el sistema elimina la sesión y por consiguiente este tiene que volver a iniciarla. Como se muestra en la figura 4.36 el tiempo de expiración se configura en el archivo *Web.xml*.

```
<session-config>
  <session-timeout>
    10
  </session-timeout>
</session-config>
```

Figura 4.36: Sesión en el archivo *Web.xml*

En esta aplicación hay diferentes datos que se utilizan al momento que el usuario se autentifica en el sistema entre ellos está el nombre de usuario el cual identifica al usuario que acaba de iniciar sesión. Otra información importante almacenada en la sesión son los permisos almacenados en una lista, los cuales permitirán o denegaran funciones al usuario.

En Struts la interface *SessionAware* nos permite manipular la sesión para almacenar, acceder y eliminar datos de la misma, por medio de un objeto de tipo *Map* y el método *setSession()* implementado en los *Actions* en los que se tendrá relación con la sesión como por ejemplo *LoginAction* donde se almacena el nombre, el tipo, el rol y los permisos de usuario. En la figura 4.37 se muestra un fragmento de código donde se manipula la sesión desde un *Action*.

```

if(nombreUsuario.getActivo()==true){
    permisos=bd.getPermisosUsuario(nombreUsuario.getIdUsuario());
    bd.cerrarConexion();

    sesion = ActionContext.getContext().getSession();
    sesion.put("usuario",nombreUsuario.getIdUsuario());
    sesion.put("nombre",nombreUsuario.getNombre().toUpperCase()+" "+
        +nombreUsuario.getPaterno().toUpperCase()+" "+nombreUsuario.getMaterno().toUpperCase()+" ");
    sesion.put("tipoUsuario",nombreUsuario.getTipoUsuario());
    sesion.put("rolUsuario",nombreUsuario.getRolUsuario());
    sesion.put("permisos",permisos);
    return nombreUsuario.getTipoUsuario();
}else{
    //La cuenta del usuario no esta activa
    bd.cerrarConexion();
    this.addActionError(getText("login.usuario.no.activo"));
    return INPUT;
}

```

Figura 4.37: Sesión y Action LoginAction

4.4.3. Interceptores

Los interceptores son componentes (métodos) que se ejecutan antes y después de un *Action*. Proporcionan un elemento arquitectónico en el que definen flujos de trabajo diferentes y tareas transversales para que puedan ser fácilmente reutilizados. Esta es una opción importante de *Struts 2*.

En *Struts 2* cada *Action* tiene una pila de interceptores por omisión que se puede configurar de acuerdo a las necesidades de la aplicación en el archivo *struts.xml*. También hay que tener en cuenta que los interceptores no necesariamente tienen que ejecutarse dos veces. Algunos interceptores sólo hacen su trabajo antes del *Action* que se ha ejecutado, y otros sólo lo hacen después. Para esta aplicación se trabaja con un *Interceptor* para la autenticación llamado *AutenticacionInterceptor* con este se logra que cuando se expira la sesión o un usuario trate de ingresar por medio de una *URL (Uniform Resource Locator)* a un *Action* el *Framework* comprueba si el usuario inició sesión, en caso contrario no puede ejecutar el *Action* y como *Result* se muestra la página de inicio. Esta es una buena solución para que solo los usuarios autenticados puedan acceder a la información en la aplicación. La figura 4.38 muestra parte del código del *Interceptor* descrito.

```

public class AutenticacionInterceptor extends AbstractInterceptor implements StrutsStatics {
    private static final String USER_HANDLE = "usuario";
    private static final String LOGIN_ATTEMPT = "loginAttempt";

    public void destroy() {
    }
    public void init() {
    }

    public String intercept(ActionInvocation invocation) throws Exception {

        final ActionContext context = invocation.getInvocationContext();
        HttpServletRequest request = (HttpServletRequest) context.get(HTTP_REQUEST);
        HttpSession session = request.getSession(true);

        Object user = session.getAttribute(USER_HANDLE);
        if (user == null) {
            String loginAttempt = request.getParameter(LOGIN_ATTEMPT);
            if (!StringUtils.isBlank(loginAttempt)) {
                return invocation.invoke();
            }
            return "login";
        } else {
            return invocation.invoke();
        }
    }
}

```

Figura 4.38: Interceptor *AutenticacionInterceptor*

4.4.4. Subir archivos al servidor

En la aplicación hay un apartado para poder subir los respaldos al servidor para ser restaurados posteriormente. *Struts 2* utiliza un *Interceptor* para subir archivos *FileUploadInterceptor* el cual posee las siguientes propiedades: *File* la ruta del archivo que está por subirse, *FileName* el nombre del archivo seleccionado y *ContentType* que hace referencia al tipo de archivo (imagen, audio, texto).

Hay una propiedad del formulario *enctype* que es utilizada sin excepción para poder subir archivos al servidor su valor es "*multipart/form-data*". En la figura 4.39 se muestra una imagen del formulario con dicha propiedad.

```

<s:form action="CargarRespaldarDatosAction" theme="simple" id="form3ID" enctype="multipart/form-data">
  <table>
    <tr>
      <td><s:text name="respaldo.cargar.archivo"></s:text></td>
      <td><s:file name="archivoRespaldo" key="" size="50" cssClass="field validate[required]"></s:file></td>
    </tr>
    <tr>
      <td><s:text name="respaldo.descripcion"></s:text></td>
      <td><s:textarea name="descripcion" theme="simple" cols="50" rows="5" cssClass="area" ></s:textarea> </td>
    </tr>
  </table>
  <br>
  <s:submit key="submit.aceptar" cssClass="submit" ></s:submit>
</s:form>

```

Figura 4.39: Propiedad *enctype*

Otro aspecto importante al momento de subir un archivo, es el tamaño del o los archivos

que serán importados al servidor. *Struts 2* posee un documento `struts.properties` el cual se sitúa en la carpeta `WEB-INF/classes/` de la aplicación, donde se pueden configurar otros aspectos del *Framework*. En este archivo se configura la propiedad `struts.multipart.maxSize` la cual posee un valor por omisión de 2048 bytes y la cual se puede modificar en el archivo mencionado por otro valor siempre en *bytes*. En la aplicación esta propiedad está configurada para soportar hasta 50 MB del o los archivos a subir. En caso de que el tamaño exceda el *Framework* enviará un mensaje de error a la respectiva vista. En la figura 4.40 se muestra el archivo `struts.properties` donde se encuentra la propiedad `struts.i18n.encoding` para poder ser editada según la aplicación lo requiera.

```
#Internacionalización
struts.custom.i18n.resources=ApplicationResources,EncuestaSocioeconomica,EncuestaPruebaValores,
#50 Mb
struts.multipart.maxSize=52428800
#Codificación
struts.i18n.encoding=ISO-8859-1
```

Figura 4.40: Tamaño máximo del archivo a subir en el servidor

4.4.5. Internacionalización y localización (i18n)

La internacionalización es un proceso con el cual se logra que una página *Web* ofrezca servicios en diferentes idiomas sin modificar el código de la misma. Esto se realiza a través de etiquetas referenciadas en el código y que se añaden a un archivo con extensión `.properties` donde se define la cadena de texto para cada etiqueta. Esto a su vez permite que en un sólo archivo se editen cada una de las etiquetas y de ser necesario sean modificadas sin ir a las respectivas líneas de código donde se encuentra la etiqueta. En la figura 4.41 podemos observar etiquetas definidas en el archivo `.properties`.

```
registro.password=Password
registro.passwordConfirm=Confirma Password
registro.password.diferente=El password debe ser el mismo
registro.nombre=Nombre(s)
registro.paterno=Apellido Paterno
registro.materno=Apellido Materno
registro.email1=Email 1
registro.email2=Email 2
registro.licenciatura=Licenciatura
registro.generacion=Generación
registro.semestre=Semestre
registro.grupo=Grupo
registro.genero=Genero
registro.telefono1=Telefono(casa)
registro.telefono2=Telefono(trabajo)
registro.celular=Celular
registro.notas=Notas
```

Figura 4.41: Archivo `.properties` de la aplicación

Hay que mencionar que se pueden generar más archivos de este tipo y que para que la aplicación los tome en cuenta hay que declararlos en el archivo `struts.properties` como se

muestra en la figura 4.42 y que estos a su vez serán separados por comas para diferenciar de uno a otro.

```
#Internacionalización
struts.custom.i18n.resources=ApplicationResources,EncuestaSocioeconomica,EncuestaPruebaValores, EncuestaAfirmaciones,
```

Figura 4.42: Archivos *.properties*

4.4.6. Log4j

La aplicación contiene configurado un registro (*Log4j*) el cual permite al programador realizar una bitácora en un archivo, de los eventos que van ocurriendo mientras la aplicación está siendo ejecutada, esto sirve tanto para saber quién, en qué tiempo y qué proceso fue realizado. También permite detectar errores que pudiesen ser ocurridos en determinado proceso y con esto se puede aplicar una solución mas rápida, puesto que se puede saber en qué parte del código ha ocurrido.

La configuración de *log4j* tiene varias vertientes, para esta aplicación se utilizó una configuración básica, donde los registros serán almacenados en un archivo de texto. El archivo para configurar *Log4j* es *log4j.properties* el cual está situado en la carpeta *WEB-INF/* de la aplicación. La configuración del archivo mencionado se puede observar en la figura 4.43 En este se puede determinar la cadena de impresión, es decir el formato, con el que se guarda cada mensaje en el archivo, la ruta del archivo y el nivel de prioridad del mensaje por mencionar algunos. Los niveles para *Log4j* son los siguientes: *all*, *trace*, *debug*, *info*, *warn*, *error*, *fatal* y *off*, en el orden de mayor a menor prioridad. Esta vez configurado con nivel *info*.

```
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n
log4j.rootLogger=info, stdout, LOGFILE
log4j.rootCategory=INFO, diario
log4j.appender.LOGFILE=org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOGFILE.file=${catalina.base}/logs/aplicacion.log
log4j.appender.LOGFILE.append=true
log4j.appender.LOGFILE.DatePattern='.'yyyy-MM-dd
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%-4r [%t] %-5p %c - %m%n
```

Figura 4.43: Configuración *Log4j*

Ya configurado el *log4j* es necesario implementarlo en las clases donde es necesario, esto depende de la lógica de negocio. En la figura 4.44 se muestran las líneas necesarias para llevar a cabo esta función, así como un ejemplo de los niveles de mensajes, aunque cabe mencionar que como esta aplicación solo tiene soporte para el nivel *info*, solo se registraran los mensajes con este nivel o niveles inferiores como *warn*, *error*, *fatal* y *off*.

```
public String execute(){
    URL url = Loader.getResource("log4j.properties");
    PropertyConfigurator.configure(url);
    log = Logger.getLogger(RegistroAction.class);
    log.info("mensaje nivel info");
    log.debug("mensaje nivel debug");
    log.error("mensaje nivel error");
    log.fatal("mensaje nivel fatal");
}
```

Figura 4.44: Código para utilizar *log4j*

Capítulo 5

Pruebas realizadas al SIWAE

Toda aplicación software debe pasar por un periodo de pruebas para detectar posibles errores. El tiempo y esfuerzo (costo) que se tiene que invertir para corregir los errores varía dependiendo de la etapa (Análisis, Diseño y/o Implementación) en donde se detecten. El costo de corregir los errores cometidos es más alto cuanto más cercanos se encuentren a las primeras etapas.

La primera prueba realizada fue el acceso al sistema SIWAE. En esta prueba se detectó un problema en la dirección *IP* de los recursos dentro del sistema, ya que el sistema estaba configurado con el servidor local (*IP* 127.0.0.1) en la página principal, esto provocaba un conflicto cuando se redireccionaba a la página donde el usuario realiza el acceso. La solución fue cambiar por la dirección *IP* y el puerto del servidor.

La segunda prueba realizada fue la del registro de un usuario. Se detectaron inconsistencias en la validación de datos de entrada durante la prueba, ya que en un principio la validación fue programada para realizarse en el lado del servidor la validación provocaba que se lanzaran excepciones y que el sistema mostrara pantallas de errores al usuario. Tomando ventaja de las características de Javascript, se decidió cambiar los validadores al lado del cliente y con esto se logró que los datos enviados al servidor no contengan errores (respecto al tipo e información faltante), asimismo se consiguió mejorar el tiempo de respuesta de las validaciones desde la perspectiva del usuario.

Una vez que todas las validaciones de los datos de formularios se cambiaron hacia el lado del cliente, se realizó otra prueba similar a la segunda, en la que se encontraron nuevamente inconsistencias, pero esta vez en los tipos de caracteres que recibían los campos. Así que se procedió a crear tablas como la mostrada en la figura 5.1 para determinar sistemáticamente los tipos de datos que puede recibir cada campo. Los datos a evaluar son los siguientes:

1. **Sólo letras.** Campo que permite introducir letras.
2. **Sólo números.** Campo que permite introducir números.
3. **Número telefónico.** Campo que requiere de un número telefónico.

4. **correo electrónico.** Campo que requiere un correo electrónico (correo@dominio.com).
5. **Número de caracteres.** Campo que requiere un mínimo y/o máximo de caracteres.
6. **Datos requeridos.** Campos que son necesarios para enviar el formulario.
7. **Fechas.** Campo que requiere una fecha en formato dd/mm/aaaa.

Para la tercera prueba llenaron las correspondientes tablas para todos los formularios, estas ayudaron a dar seguimiento a los casos de uso en los formularios del SIWAE. La figura 5.1 que corresponde a la tabla del formulario para registro contiene dos columnas: “Acepta” y “Longitud”. La primera se subdivide en “Letras”, “Números” y “Caracteres especiales”, y los posibles valores para cada una de las celdas son “Si” y “No”, indicando si se permite introducir o no dicho carácter. La Columna de “longitud se subdivide en “Mínima” y “Máxima”, indicando cuales son los caracteres minimos y máximos que se permiten en el campo. La leyenda N/A indica que el campo no permite introducir caracteres, por lo que puede ser un control tipo caja de opciones, casilla de verificación o botones de selección con exclusión mutua (conocidos como botones de radio). La columna “Requerido” indica que si la información es obligatoria para poder enviar el formulario. A continuación se muestran algunas tablas con las que se puede validar dichos casos de prueba en las ventanas de la aplicación. A continuación se muestran algunas tablas relacionadas con los formularios más representativos del SIWAE.

Formulario: Registro						
Campo	Acepta			Longitud		Requerido
	Letras	Números	Caracteres Especiales	Mínima	Máxima	
No. De Cuenta	No	Si	No	7	7	Si
Contraseña	Si	Si	Si	6	15	Si
Confirmar Contraseña	Si	Si	Si	6	15	Si
Nombre	Si	No	Si	1	45	Si
Apellido Paterno	Si	No	Si	1	45	Si
Apellido Materno	Si	No	Si	1	45	Si
Email 1	Si	Si	No	1	45	Si
Email 2	Si	Si	No	1	45	Si
Perfil	N/A	N/A	N/A	N/A	N/A	Si
Genero	N/A	N/A	N/A	N/A	N/A	Si
Teléfono (Casa)	No	Si	No	N/A	N/A	No
Teléfono (Trabajo)	No	Si	No	N/A	N/A	No
Celular	No	Si	No	N/A	N/A	No
Notas	Si	Si	Si	0	1000	No

Figura 5.1: Validaciones del formulario “Registro”

En la figura 5.2 se presenta otro ejemplo que consiste en la tabla de las validaciones para el formulario “Modificar datos”.

Formulario: Modificar Datos						
Campo	Acepta			Longitud		Requerido
	Letras	Números	Caracteres Especiales	Mínima	Máxima	
Nombre	Si	No	No	1	45	Si
Apellido Paterno	Si	No	Si	1	45	Si
Apellido Materno	Si	No	Si	1	45	Si
Email 1	Si	Si	Si	1	45	Si
Email 2	Si	Si	Si	1	45	Si
Genero	N/A	N/A	N/A	N/A	N/A	Si
Teléfono (Casa)	No	Si	No	N/A	N/A	No
Teléfono (Trabajo)	No	Si	No	N/A	N/A	No
Celular	No	Si	No	N/A	N/A	No
Notas	Si	Si	Si	0	1000	No

Figura 5.2: Validaciones del formulario “Modificar mis datos”

En una cuarta prueba, el sistema SIWAE fue operado por un grupo de usuarios reales, se observó un problema al llenar los formularios: El *Framework JQuery* utilizado para la validación trabaja con expresiones regulares, por lo que hay ciertos caracteres y símbolos que no están incluidos y que al utilizarlos se detectan como errores. Estos caracteres y símbolos detectados son:

- Letras Ñ, ñ
- Las letras acentuadas Á, á, É, é, Í, í, Ó, ó, Ú, ú.

La solución a este problema consiste en agregar los caracteres al respectivo patrón en el archivo *jquery.validationEngine-es*, las expresiones regulares del *Framework JQuery* se muestran en la figura 5.3.

```

"onlyLetterSp": {
  "regex": /^[a-zA-Z\ \'\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D3\u00F3\u00DA\u00FA\u00d1\u00f1]+$/,
  "alertText": "** S#243lo letras con espacios"
},
"onlyLetter": {
  "regex": /^[a-zA-Z\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D3\u00F3\u00DA\u00FA\u00d1\u00f1]+$/,
  "alertText": "** S#243lo letras sin espacios"
},
"onlyLetterNumber": {
  "regex": /^[0-9a-zA-Z]+$/,
  "alertText": "** No se permiten caracteres especiales, s#243lo letras y n#250meros"
},

```

Figura 5.3: Expresión regular *Framework* validación

En la figura 5.47 se muestra una tabla con los caracteres necesarios en la expresión regular del *Framework* validation Engine con las respectivas codificaciones.

Tabla codificaciones.	
UTF-8	Unicode
Á	\u00c1
É	\u00c9
Í	\u00cd
Ó	\u00d3
Ú	\u00da
á	\u00e1
é	\u00e9
í	\u00ed
ó	\u00f3
ú	\u00fa
Ñ	\u00d1
ñ	\u00f1

Figura 5.4: Caracteres especiales para el *Framework* de validación

Una vez agregados los caracteres especiales al patrón, se volvió a realizar otra prueba similar a la cuarta, es decir, con usuarios reales. En esta prueba el formulario del registro se llenó correctamente, pero se detectó un problema al momento de que el formulario era enviado: No se almacenaba en la base de datos la información enviada. El problema fue que el sistema tenía configurada la dirección 127.0.0.1, pero esta vez con respecto al gestor de base de datos. Se cambió la dirección en el sistema en el archivo de configuración de *JPA* (*persistence.xml*), se realizaron pruebas y se observó que nuevamente que no se guardaba información en la base de datos. Revisando el código fuente y los archivos de configuración, se encontró que el gestor de base de datos (*MySQL*) no estaba configurado para un acceso remoto, esto se solucionó modificando el host del gestor de la base de datos. Para modificar el host o dirección *IP* del gestor, en este caso en *Ubuntu*, es necesario modificar la línea del archivo */etc/mysql/my.cnf* que se muestra en la figura 5.5. Como se observa, de manera predefinida se tiene una dirección *IP* la cual debe ser cambiada por la dirección *IP* del servidor.

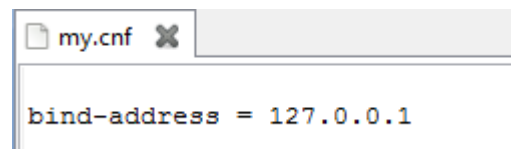


Figura 5.5: Archivo de configuración MySQL

Al realizar el cambio de *IP* en el archivo de configuración del gestor de la base de datos, ya fue posible almacenar la información, sin embargo, al realizar pruebas se presentó nuevamente un inconveniente al momento de guardar algunos caracteres, puesto que en la base de datos fueron guardados correctamente con excepción de símbolos especiales como los acentos, esto puede deberse a la codificación ya sea desde la vista, el controlador, o incluso la base de datos misma. Para solucionar esto se cambió la codificación en cada una de ellas, en primera instancia se cambió la codificación UTF-8 por la ISO-8859-1 en cada una de las páginas del

sistema, como se observa en la figura 5.6.

```
<? page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" ?>
<?taglib prefix="s" uri="/struts-tags" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<meta content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
```

Figura 5.6: Codificación ISO-8859-1

El segundo cambio se realizó en el archivo *struts.properties* el cual tiene predeterminado el valor UTF-8. El atributo *struts.i18n.encoding* se le asignó el valor ISO-8859-1. Por último, se cambió la codificación al gestor de base de datos en el archivo */etc/mysql/my.cnf* donde se modifica el valor a la variable *default-character-set*, ver figura 5.7.

```
#
[client]
port=3306
[mysql]
default-character-set=latin1

# SERVER SECTION
# -----
#
# The following options will be read by the MySQL Server. Make sure that
# you have installed the server correctly (see above) so it reads this
# file.
#
[mysqld]
```

Figura 5.7: Codificación en MySQL

Una vez realizados estos cambios, se realizaron pruebas con resultados satisfactorios. Todos los registros fueron guardados correctamente en la base de datos. Se registró un grupo de usuarios y se obtuvieron los datos que se muestran en la figura 5.8

#	/	id_usuario	/	nombre	paterno	materno
1		00920669		maria del carmen	escorcia	cajiga
2		00920675		beatriz	munoz	lozano
3		0620745		YARICENI	HERNANDEZ	AVILA
4		0822936		alejandra	garcia	garcia
5		090662		YAIR ULISES	CRUZ	MATURANO
6		0920658		ana karen	barrera	cordero
7		0920660		Mabel	Clemente	Ponce
8		0920661		celestelizabeth	decaro	barrera
9		0920662		YAIR ULISES	CRUZ	MATURANO
10		0920663		LUIS ERNESTO	BENITEZ	VALENCIA
11		0920664		Claudia	García	Galván
12		0920665		ALMA VICTORIA	GARAY	HERNANDEZ
13		0920666		jose felix	garcia	hernandez
14		0920667		mariadelsocorro	godoy	garcia
15		0920668		christian simon	juarez	oropeza
16		0920669		maria del carmen	escorcia	cajiga
17		0920672		ANA LAURA	LOPEZ	VARGAS
18		0920674		Victor hugo	Maturano	Gordiano
19		0920675		beatriz	munoz	lozano
20		0920677		ADRIANA	MIMBRERA	BUREOS

Figura 5.8: Usuarios registrados

En la parte del registro todos los usuarios que realizaron este proceso fueron dados de alta en el sistema, sin embargo solo 10 usuarios, es decir, solo el 50 % activaron su cuenta, ver figura 5.9.

#	id_usuario	tipo_usuario	rol_usuario	activo
1	00920669	E1	Alumno	<input type="checkbox"/>
2	00920675	E1	Alumno	<input type="checkbox"/>
3	0620745	E1	Alumno	<input checked="" type="checkbox"/>
4	0822936	E1	Alumno	<input checked="" type="checkbox"/>
5	090662	E1	Alumno	<input checked="" type="checkbox"/>
6	0920658	E1	Alumno	<input checked="" type="checkbox"/>
7	0920660	E1	Alumno	<input checked="" type="checkbox"/>
8	0920661	E1	Alumno	<input type="checkbox"/>
9	0920662	E1	Alumno	<input type="checkbox"/>
10	0920663	E1	Alumno	<input checked="" type="checkbox"/>
11	0920664	E1	Alumno	<input checked="" type="checkbox"/>
12	0920665	E1	Alumno	<input type="checkbox"/>
13	0920666	E1	Alumno	<input checked="" type="checkbox"/>
14	0920667	E1	Alumno	<input type="checkbox"/>
15	0920668	E1	Alumno	<input type="checkbox"/>
16	0920669	E1	Alumno	<input type="checkbox"/>
17	0920672	E1	Alumno	<input checked="" type="checkbox"/>
18	0920674	E1	Alumno	<input type="checkbox"/>
19	0920675	E1	Alumno	<input type="checkbox"/>
20	0920677	E1	Alumno	<input checked="" type="checkbox"/>

Figura 5.9: Activación de cuenta por usuarios

Los posibles problemas para este hecho son los siguientes:

1. Probablemente los usuarios introdujeron una dirección de correo electrónico no existente o inválida en el formulario del registro,
2. El correo es detectado como spam por el servidor de correo y por lo mismo no era almacenado en la bandeja de entrada de la cuenta de correo del usuario,
3. Otra posible causa es el tiempo de solicitud del envío, es decir, si el tiempo de respuesta es alto el sistema detecta que algo no procede bien, lanza una excepción y el correo no es enviado.

Un problema derivado de lo anterior, es que un usuario puede duplicar su registro. Para solucionar esto se habilitó una página donde hay un filtro de búsqueda para los usuarios, que permite al administrador activar y/o desactivar las cuentas de los usuarios.

La última prueba realizada al SIWAE fue el llenado de la encuesta valores (mostrada en la figura 5.10) durante la prueba se registraron más usuarios. En esta parte se estiman pocos usuarios que finalizaron la prueba, otros tantos la dejaron incompleta, y algunos que no obtuvieron acceso a la misma, esto se observa y concluye de acuerdo a los registros que se dieron en la base de datos.

id_usuario	tolerancia_importante	tolerancia_util	tolerancia_duradera
0620745	1	1	1
0920681	2	2	4
0920663	4	4	4
0920694	1	2	1
0920684	1	1	1
0822936	1	1	2
0920696	1	2	2
0920672	1	1	2
0421716	1	1	1

Figura 5.10: Usuarios que contestaron la encuesta valores

En la figura 5.11 se presentan las gráficas de los usuarios que terminaron la encuesta y los resultados de la misma. Esto es un ejemplo de las capacidades de generación de reportes del SIWAE.

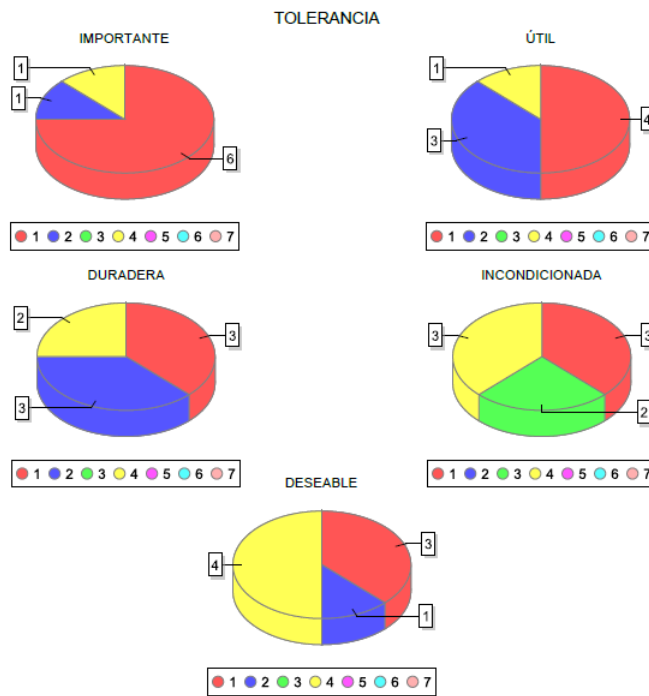


Figura 5.11: Gráficas obtenidas de la encuesta valores

Para consultar a los usuarios que dejaron incompleta el llenado de la encuesta valores, el SIWAE puede generar un reporte, similar al mostrado en l figura 5.12.

id_usuario	tolerancia_importante	tolerancia_util	tolerancia_duradera
0421716	1	1	1
0920695	1	1	1
0926907	1	1	1
0920666	1	1	3
0920658	1	4	4

Figura 5.12: Usuarios que no terminaron la encuesta valores

No se detectaron más errores en el SIWAE durante la última prueba. Se realizaron más

registros de usuarios de prueba para comprobar que todo funcionaba correctamente, se contestaron nuevamente encuestas con esos usuarios sin detectar problemas.

Debido al equipo donde se ejecuta el sistema (Procesador Pentium IV con 1 GB RAM), el tiempo de respuesta es alto y la cantidad de usuarios que pueden atenderse de manera simultánea es de 15 como máximo. No se considera que esto sea un problema del SIWAE, sino parte de las limitaciones del hardware.

Capítulo 6

Conclusiones, Recomendaciones y Trabajo Futuro

6.1. Conclusiones

En este trabajo se desarrolló un sistema *Web* denominado SIWAE (Sistema Informático *Web* para Aplicación de Encuestas), mismo que fue orientado hacia la aplicación de encuestas a alumnos aceptados del programa educativo de Contaduría del Centro Universitario UAEM Zumpango. El sistema desarrollado permite aplicar encuestas de valores, estudio socio económico y prueba de valores, también puede generar reportes y gráficas tipo pastel para facilitar el posterior análisis de las respuestas. El SIWAE fue completamente implementado utilizando software libre.

Durante el proceso de desarrollo del SIWAE se tuvieron dificultades técnicas y metodológicas, mismas que fueron exitosamente resueltas, lo que permitió adquirir experiencia en la realización de este tipo de sistemas. La obtención de requerimientos fue una de las etapas que más dificultad presentó al inicio del desarrollo del SIWAE, ya que aunque teóricamente se conocía la metodología y se tenía algo de experiencia en la realización de proyectos de menor complejidad, en este trabajo se verificó que un análisis poco profundo o incompleto hace que la implementación sufra de retrocesos importantes en etapas avanzadas del proyecto. Durante la etapa de obtención de requerimientos, se invirtió una cantidad considerable de tiempo y se utilizó como herramienta de modelado a *UML* para representar gráficamente al SIWAE desde diversas perspectivas. Esto permitió tener idea clara para su posterior implementación en software.

La elección del *Framework Web* previo a la escritura de código es de suma importancia, y para su selección es necesario considerar cuestiones como la vista, validación de datos y generación de vistas dinámicas. Se eligió *Struts 2* debido a que permite una arquitectura basada en el modelo vista controlador, además de soportar el trabajo junto con otros *Frameworks* para mejorar la presentación de la interfaz de usuario, como *jQuery*, también empleado en el SIWAE.

El *software* del lado del servidor fue escrito en lenguaje *Java*, y aunque el ejecutable

(comprimido en un archivo .jar) se supone multiplataforma, en la práctica se tuvieron algunos problemas, como por ejemplo, rutas absolutas no reconocidas en sistema operativo *Windows* y *Linux*, problemas de compatibilidad con algunas bibliotecas de *Struts* o con la base de datos *MySQL*.

Se muestra a continuación una lista de conclusiones sobre diversas partes del proyecto.

- No existe una compatibilidad completa entre navegadores *Web*, por lo que al desarrollar una aplicación *Web* es necesario decidir la versión del navegador a usar.
- Es sumamente importante conocer las características de la plataforma donde se ejecutará la aplicación final, ya que no es lo mismo desarrollar para sistema operativo *Windows*, que para las diferentes versiones de *Linux*. Por otra parte, las características del hardware de la plataforma determinan en gran medida el desempeño del sistema.
- La apariencia y funcionamiento de las páginas *Web* se puede mejorar considerablemente utilizando un *Framework Javascript*, además su uso permite un acceso más fácil a las propiedades y eventos de los componentes de la página. En este trabajo se usó *JQuery* por las características que brinda.
- Utilizar la validación del lado del cliente es una solución viable, puesto que no hay que recargar la toda la página *Web* cada vez que se valide un campo. Por otra parte, las validaciones de lado del cliente resultan más atractivas para el usuario debido a que los tiempos de respuesta resultan menores comparadas con validaciones realizadas del lado del servidor.
- Utilizar un *ORM* facilita la manipulación de los datos, hay muchas implementaciones por lo que elegir una depende de los requerimientos de la aplicación.
- Utilizar sesiones es en definitiva es necesario para una aplicación *Web*, puesto que permite realizar diversas comprobaciones y controlar el acceso a páginas *Web* del sistema.

6.2. Recomendaciones

Como recomendaciones generales, se indican las siguientes.

- Es necesario reunir todos los requerimientos de la aplicación desde el inicio. Se da un énfasis especial en este paso crucial, ya que entre más clara y completa esté la información, permitirá un mejor análisis y diseño del sistema final. Para reunir los requerimientos se recomienda realizar entrevistas, analizar procesos y revisar documentos previos si existe ya un sistema funcionando.

- Utilizar *UML* y en especial los casos de uso como diagramas iniciales en cada proyecto a desarrollar. Esto permite tener una idea más clara de lo que se desarrolla, definir los actores, limitar y definir procesos.

- Realizar análisis y diseño en la base de datos. Un diseño de la base de datos bien pensado desde el inicio evita modificaciones en la aplicación a lo largo de su proceso de desarrollo.

- Realizar pruebas sobre cada módulo. Esto es lo más adecuado, ya que así se verificará que realmente realice lo que se pretende.

- Documentar código fuente. La documentación es una buena práctica y debe de realizarse en el momento de escribir el código ya que el dejarlo para después es complicado puesto que hay que volver a leer y analizar las clases.

- Si se usa el *Web Framework Struts 2*, se recomienda establecer las etiquetas con la propiedad *theme* con valor “*simple*” y tablas *HTML* para su ordenamiento, puesto que resulta más sencillo y más preciso al momento de acomodar los campos en el formulario.

- Se recomienda usar la capacidad de internacionalización de *Struts 2*, para así poder editar las cadenas de texto que se presenten en los formularios, la edición se vuelve más sencilla puesto que se lleva a cabo en un archivo y no hay que realizar una búsqueda en cada página donde se requiera modificar la cadena de texto.

- Hacer pruebas en diferentes navegadores de Internet. No todos los navegadores son compatibles al 100%, y por lo tanto la presentación puede ser diferente en cada uno.

6.3. Trabajo futuro

El SIWAE, como cualquier otro sistema, puede ser sujeto a mejoras. A lo largo del desarrollo y una vez terminado el sistema, se han detectado algunas áreas de oportunidad, mismas que muestra en la siguiente lista.

- Cambiar el servidor *Web* por un servidor de aplicaciones con soporte completo para *J2EE* (por ejemplo, *GlassFish* o *JBoss*), para mejorar el desempeño del sistema y facilitar la administración del mismo.
- Implementar la persistencia a través de *EJB* o alguna otra tecnología que tenga soporte para la inyección de dependencias, para actualizar la lógica del negocio y agregar más funcionalidades al SIWAE de manera dinámica.
- Agregar módulos al SIWAE para realizar extracción de conocimiento a los datos almacenados en la base de datos.
- Desarrollar una versión para acceso a través de dispositivos móviles.

Bibliography

- [1] Apache. <http://struts.apache.org/2.x/>. Wep page, September 2011.
- [2] DISPLAYTAG. <http://www.displaytag.org/1.2/>, 2011.
- [3] Scott Stanlick Donald Brown, Chad Michael Davis. *Struts 2 in Action*. Manning, USA, 2008.
- [4] Eclipse. <http://www.eclipse.org/>, 2011.
- [5] EZMARK. <http://www.itsalif.info/content/ezmark-jquery-checkbox-radiobutton-plugin>, 2011.
- [6] Firebird. <http://www.firebirdsql.org/>, 2011.
- [7] GlassFish. <http://glassfish.java.net/es/>, 2011.
- [8] Paul J. Deitel Harvey M. Deitel. *Como programar en Java*. Prentice Hall, 2004.
- [9] ISS. <http://www.iis.net/>, 2011.
- [10] JMESA. <http://code.google.com/p/jmesa/>, 2011.
- [11] JQUERY. <http://www.jquery.com>. Wep page, 2011.
- [12] JQUERYUI. <http://jqueryui.com/>, 2011.
- [13] Mike Keith and Merrick Schnicariol. *JPA 2*. Apress, 2009.
- [14] Craig Larman. *UML y patrones*. Pearson, 1999.
- [15] MYSQL. <http://www.mysql.com/>. Wep page, 2011.
- [16] NETBEANS. <http://www.netbeans.org>. Wep page, 2011.
- [17] ORACLE. <http://www.oracle.com/index.html>, 2011.
- [18] ORACLE. <http://www.oracle.com/technetwork/java/javase/javaserverfaces-139869.html>, 2011.

-
- [19] Position-absolute. <http://www.position-absolute.com/articles/jquery-form-validator-because-form-validation-is-a-mess/>. Wep page, 2011.
 - [20] Postgresql. <http://www.postgresql.org/>, 2011.
 - [21] ROR. <http://rubyonrails.org/>, 2011.
 - [22] Steven Sanderson. *ASP.NET MVC 2 Framework*. APRESS, 2007.
 - [23] Antonio Martin Sierra. *Struts*. Ra-Ma, 2008.
 - [24] Sllite. <http://www.sqlite.org/>, 2011.
 - [25] Tapestry. <http://http://tapestry.apache.org/>, 2011.
 - [26] TINYMCE. <http://www.tinymce.com/>, 2011.
 - [27] Tomcat. <http://tomcat.apache.org/>, 2011.
 - [28] WAS. <http://www-01.ibm.com/software/webservers/appserv/was/>, 2011.

Appendix A

Glosario

- **API:** Interfaz de Programación de Aplicaciones, es un conjunto de funciones y procedimientos que son utilizados por otro software como una capa de abstracción. Representa la capacidad de comunicación entre componentes de software.
- **ASF** (Advanced Systems Format) Es un formato contenedor de audio y video digital, diseñado especialmente para el streaming.
- **Back-End:** Es la parte que procesa la entrada desde el front-end.
- **Bean:** Es un componente hecho en software que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.
- **Committers:** Personas que tiene permiso para modificar un código fuente de un software en particular
- **Contenedor de Servlets:** Es un programa capaz de recibir peticiones de páginas web y redireccionar estas peticiones a un objeto Servlet.
- **CPU:** Unidad Central de Procesamiento.
- **CSS** (Cascading Style Sheets): Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.
- **DBA:** Administrador de base de datos.
- **Etiqueta** (tag): Es una marca que delimita una región en los lenguajes basados en XML.
- **Framework:** Es una estructura tecnológica con un soporte definido con módulos de software concretos, para ayudar a desarrollar y unir los diferentes componentes de un proyecto.
- **Front-End:** Es la parte del software que interactúa directamente con los usuarios.

- **GNU GPL** (GNU General Public License): Posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia.
- **Hibernate**: Es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.
- **HTML** (HyperText Markup Language): Es el lenguaje de marcado que establece la estructura y el contenido para la elaboración de páginas Web.
- **IDE** (Entorno de desarrollo integrado): Es una aplicación de software que proporciona servicios integrales a los programadores para el desarrollo de software .
- **IDPL** Initial Developer's Public License Interfaz
- **Internacionalización**: Es el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería o en el código
- **JavaBean**: Es un componente de software reutilizable, que se puede manipular visualmente en una herramienta de construcción.
- **JavaEE** (Java Platform, Enterprise Edition): Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.
- **Javascript**: Es un lenguaje de programación que permite a los desarrolladores crear pequeños programas que son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos.
- **JPA** (Java Persistence API): Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).
- **jQuery**: Es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas Web.
- **JSF** (JavaServer Faces): Tecnología para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.
- **JSP** (JavaServer Pages): Tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

- **Microkernel:** Es un tipo de núcleo de un sistema operativo que suministra un conjunto de primitivas o llamadas al sistema mínimas, para implementar servicios básicos como espacios de direcciones, comunicación entre procesos y planificación básica.
- **Open Source** (codigo abierto): Término con el que se conoce al software distribuido y desarrollado libremente.
- **ORM**(Object-Relational mapping): Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.
- **OSGi** (estándar que cuenta con el apoyo de muchos vendedores del mundo Java: IBM, RedHat, Software AG).
- **Persistencia:** Propiedad de los datos para que estos sobrevivan de alguna manera.
- **RAD:** Rapid Application Development Resource-bundle.
- **Ruby on Rails o RoR:** Es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración.
- **Servicios Web:** Es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
- **Servlet:** Es un programa que se ejecuta en un servidor. Generan páginas Web de forma dinámica a partir de los parámetros de la petición que envía el navegador Web.
- **SOA** (Arquitectura orientada a servicios de cliente): Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.
- **Spring:** Es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java.
- **SQL**(lenguaje de consulta estructurado) Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones.
- **Tooltip:** Es una herramienta de ayuda visual, que funciona al situar el cursor sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra.
- **Transacción:** Es una interacción con una estructura de datos compleja, compuesta por varios procesos que se han de aplicar uno después del otro. La transacción debe realizarse de una sola vez y sin que la estructura a medio manipular pueda ser alcanzada por el resto del sistema hasta que se hayan finalizado todos sus procesos.

- **UML**(Lenguaje Unificado de Modelado): Es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para para visualizar, especificar, construir y documentar un sistema.
- **URL** (Uniform Resource Locator): Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.
- **WWW o World Wide Web**: Es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.
- **XML**(Extensible Markup Language) Es un conjunto de reglas para la codificación de documentos, haciendo hincapié en la simplicidad, la generalidad y la facilidad de uso en Internet .

Appendix B

Manual de Usuario

B.1. Registro de un usuario

Para registrarse en el sistema, hay que seguir los siguientes pasos:

1. Ingresar a la página principal del sistema.
2. Presionar el botón “Nuevo Usuario”.



The image shows a registration form for the Universidad Autónoma del Estado de México. The form is yellow and contains two input fields: 'Usuario' and 'Contraseña'. Below these fields are two buttons: 'Iniciar Sesión' and 'Nuevo Usuario'. A red error message, '*Este campo es requerido', is displayed above the 'Usuario' field, indicating that this field is required.

Figure B.1: Registro página principal

3. El sistema despliega una página con un formulario, se introducen los datos que son requeridos. En caso de que el usuario no llene un campo requerido este será notificado con un mensaje en un recuadro de color rojo como se muestra en la figura B.2.



ESTADOS UNIDOS MEXICANOS
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO ZUMPANGO

Capture los datos que a continuación se le piden

No. de Cuenta* * Este campo es requerido
* Sólo números

Password* * Este campo es requerido
* Mínimo de 6 caracteres autorizados

Confirma Password*

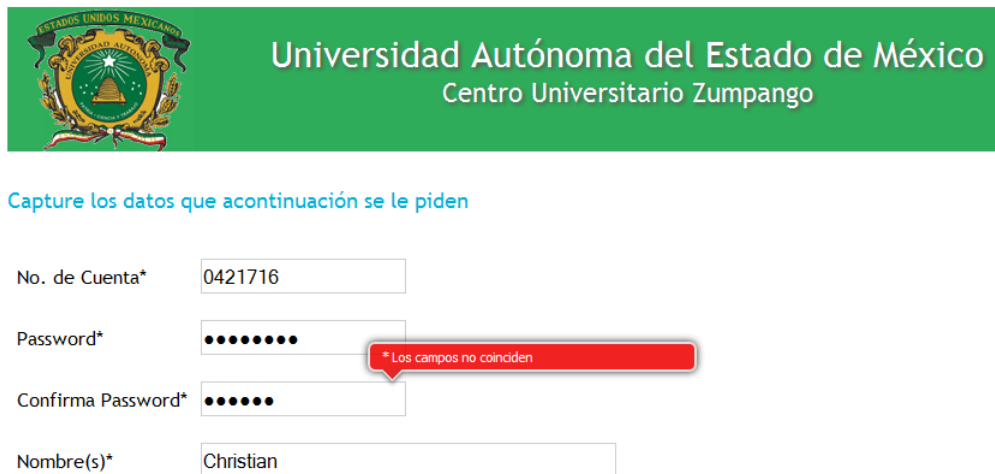
Nombre(s)*

Apellido Paterno*

Apellido Materno*

Figure B.2: Datos requeridos del nuevo usuario

- Para asegurar que el usuario introduzca la contraseña que este pretende, debe repetirlo en un segundo campo. Si estos no coinciden aparecerá un mensaje notificando que la contraseña no es el mismo.



ESTADOS UNIDOS MEXICANOS
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO ZUMPANGO

Capture los datos que a continuación se le piden

No. de Cuenta*

Password*

Confirma Password* * Los campos no coinciden

Nombre(s)*

Figure B.3: Confirmación del password

- Una vez llenado y validado de forma correcta hay que presionar el botón “Registrar” para finalizar el proceso de registro.

Perfil*	<input type="checkbox"/> Alumno <input checked="" type="checkbox"/> Profesor
Genero*	<input type="checkbox"/> Masculino <input type="checkbox"/> Femenino
Telefono(casa)	<input type="text"/>
Telefono(trabajo)	<input type="text"/>
Celular	<input type="text"/>
Notas	<input type="text"/>
	<input type="button" value="Registrar"/>
	<input type="button" value="Cancelar"/>

Figure B.4: Finalizar el registro del nuevo usuario

En caso de que el usuario haya introducido una cuenta ya registrada en el sistema se muestra el mensaje de la figura B.5.



Universidad Autónoma del Estado de México
Centro Universitario Zumpango

Capture los datos que a continuación se le piden

No. de Cuenta*	<input type="text" value="0421717"/>	El usuario ya esta registrado en el sistema
Password*	<input type="password"/>	
Confirma Password*	<input type="password"/>	
Nombre(s)*	<input type="text" value="Christian"/>	

Figure B.5: Cuenta ya registrada

- Una vez llenado el registro correctamente y haber presionado el botón “Registrar” aparecerá la página mostrada en la figura B.6. Esta página indica al usuario que se ha registrado satisfactoriamente y que solicita que active su cuenta desde el correo introducido en el formulario previamente llenado.



Figure B.6: Página de confirmación del registro

1. Al ingresar al correo electrónico abre el mensaje con el asunto “Activar cuenta sistema uaem” y presiona el botón “Finalizar” en el enlace enviado para activar tu cuenta y poder acceder al sistema. En la figura B.7 se muestra la página donde se realiza la activación de la cuenta.



Figure B.7: Página activación de la cuenta

En caso de que no haya activado la cuenta por alguno de los siguientes motivos: no recibió el correo, no escribió un correo existente o no puede acceder a su correo electrónico, el sistema mostrará un mensaje de error mostrado en la figura B.8. por lo que habrá que notificarlo con el administrador o profesor a cargo.



ESTADOS UNIDOS MEXICANOS
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO ZUMPANGO

• El usuario no ha activado su cuenta

Usuario 0421717

Contraseña

Iniciar Sesión

Nuevo Usuario

Figure B.8: Cuenta no activada

Nota: En caso de que el usuario haya introducido un correo electrónico no existente o que haya olvidado la contraseña del mismo y no pueda acceder a activar la cuenta, debe notificarlo al administrador del sistema para que éste active o elimine la cuenta. En caso de eliminar la cuenta debe de volver a realizar el proceso de registro y no introducir un correo electrónico que no exista o que no cuente con acceso.

B.2. Autenticación en el sistema

Una vez activada la cuenta, introduzca los datos correspondientes en la página mostrada en la figura B.9. Posteriormente presione el botón “Iniciar sesión”.



ESTADOS UNIDOS MEXICANOS
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO ZUMPANGO

Usuario 0421716

Contraseña

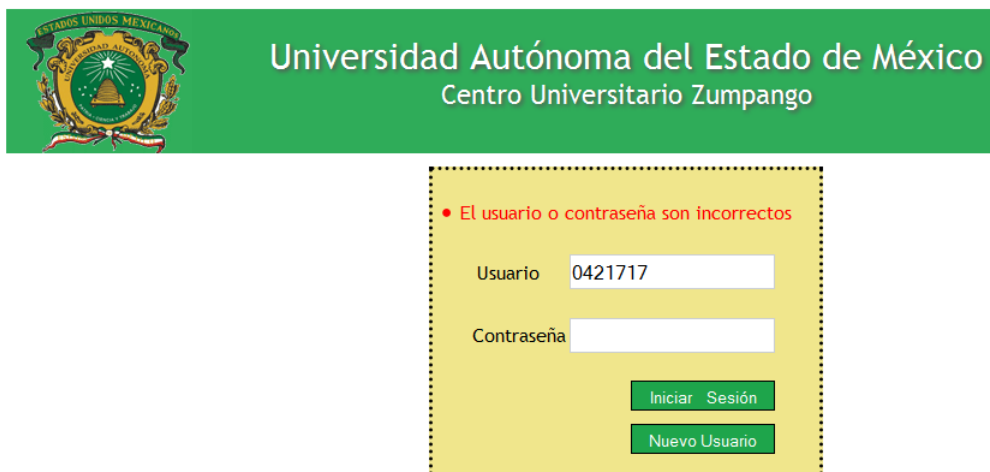
Iniciar Sesión

Nuevo Usuario

Figure B.9: Iniciar Sesión

Es posible que haya introducido un dato incorrecto (usuario o contraseña), de ser así el sistema muestra un mensaje de error como el mostrado en la figura B.10. Por lo que hay que

introducir nuevamente los datos.



The screenshot shows the login interface for the Universidad Autónoma del Estado de México, Centro Universitario Zumpango. The header is green with the university's logo on the left and the name in white text. Below the header, a yellow box with a dashed border contains an error message in red: "El usuario o contraseña son incorrectos". Underneath the message are two input fields: "Usuario" with the value "0421717" and "Contraseña" which is empty. At the bottom of the yellow box are two green buttons: "Iniciar Sesión" and "Nuevo Usuario".

Figure B.10: Error inicio de sesión

Al ingresar correctamente, el sistema envía la página de la figura B.11. En la parte izquierda se observa el menú con las opciones permitidas al usuario, éstas pueden cambiar de acuerdo a los permisos y tipo de usuario asignados por el administrador.



The screenshot shows the main page after successful authentication. The header is green with the university's logo on the left and the name in white text. On the right side of the header, it says "USUARIO DE PRUEBA" and "Cerrar Sesión". Below the header, there is a green menu on the left with the following options: "Usuarios", "Encuestas", "Encuesta Socioeconomica", "Encuesta de Valores", "Encuesta de Afirmaciones", and "Encuesta de Aptitudes".

Figure B.11: Página principal al autenticarse en el sistema

B.3. Cerrar Sesión

Cada que el usuario ingresa al sistema se crea una sesión, y cuando desee salir tendrá que cerrar la misma. Para esto una vez que ingresa puede observar su nombre de usuario y en la parte inferior un enlace con el cual sale del sistema de forma correcta terminando la sesión. Este enlace se muestra en todas las páginas que usted navegue una vez autenticándose en el sistema. Figura B.12.



Figure B.12: Cierre de sesión

B.4. Modificar mis datos personales

Para modificar datos del usuario, en caso de que éstos hayan cambiado por alguna razón o por un error en el registro, el usuario debe realizar los siguientes pasos:

1. Dar clic sobre el menu "Usuarios".
2. Dar clic en el submenu "Modificar mis datos".
3. Modificar y/o llenar los datos en el respectivo formulario.
4. Presionar el botón "Guardar".

Una vez guardado los cambios el sistema envía un correo con sus respectivas modificaciones al correo electrónico del usuario.



ESTADOS UNIDOS MEXICANOS
Universidad Autónoma del Estado de México
Centro Universitario Zumpango

USUARIO PATERNO
Cerrar Sesión

Usuarios
Modificar mis datos
Encuestas

Modifique los campos correspondientes y de clic en el botón aceptar de la parte inferior.

Nombre

Apellido Paterno

Apellido Materno

Cambiar Password

Genero Cambiar Genero

Telefono(casa)

Telefono(trabajo)

Celular

Notas

Guardar

Cancelar

Figure B.13: Modificar mis datos personales

B.5. Contestar Encuesta

Para contestar una encuesta hay que seguir los siguientes pasos:

1. Seleccionar el menú “Encuestas”.
2. Seleccionar la encuesta que desea contestar en el submenú.
3. Contestar pregunta por pregunta siguiendo las validaciones del sistema.
4. Presionar el botón “Imprimir” como comprobante de haber llenado la encuesta. (Opcional)

5. Presionar el botón “Aceptar”
6. Repetir del punto 3 al 5 hasta finalizar toda la encuesta, ya que esta se encuentra en varias partes para un mejor ordenamiento.
7. En la última parte se puede obtener un archivo PDF con los datos que introdujo en toda la encuesta, para esto presione el botón “Imprimir todo”. (Opcional)

Usuarios
Encuestas
Encuesta Socioeconomica
Encuesta de Valores
Encuesta de Afirmaciones
Encuesta de Aptitudes

Estudio Socioeconómico

Datos Generales

instrucciones

Nombre Fecha de aplicación

Fecha de nacimiento Sexo Masculino Femenino

Estado Civil Número de hijos Lugar de procedencia

Domicilio completo

Telefono Celular Email

¿Trabajas? Si No

Figure B.14: Contestar Encuesta

Appendix C

Manual del administrador

C.1. Buscar Usuario

Esta función es utilizada en primer instancia para otras funciones como es el caso de modificar los datos del usuario, activar usuarios, ver usuarios, eliminar usuarios, modificar permisos y enviar correo electrónico.

Para realizar la búsqueda de un usuario se realizan los siguientes pasos:

1. Seleccionar el tipo de búsqueda por nombres(s), apellido paterno, apellido materno o cualquiera. La última opción se refiere buscar las coincidencias del texto introducido.
2. Introducir la cadena de texto a buscar.
3. Presionar el botón “Buscar”

Nota: En caso de no introducir ningún texto la búsqueda realizada muestra como resultado todos los usuarios registrados en el sistema.

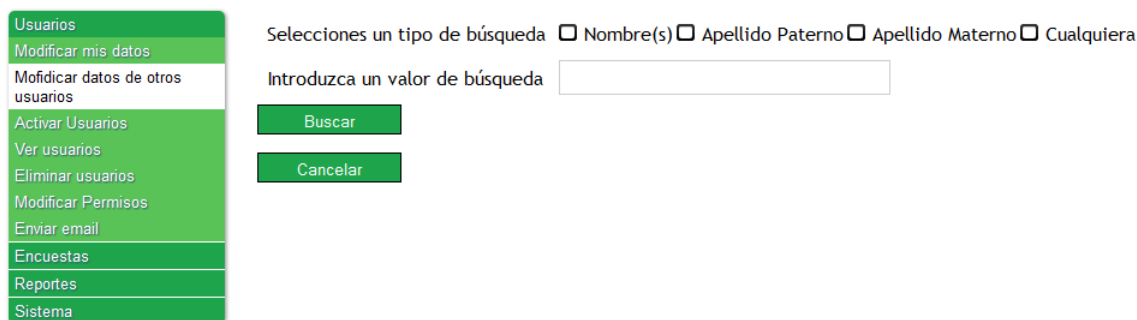
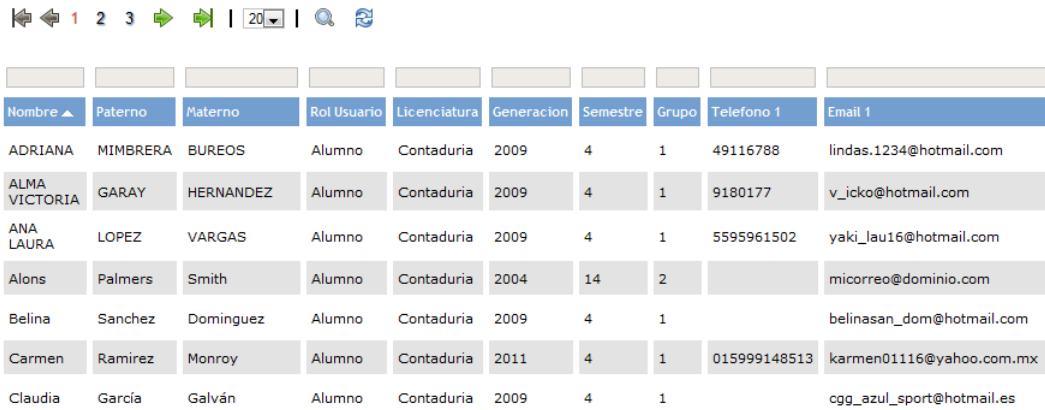


Figure C.1: Buscar usuario

Una vez que la búsqueda ha finalizado se muestra una tabla, la cual contiene información referente de los usuarios encontrados y opciones relacionadas al submenú que previamente seleccionó el usuario. En la figura C.2 se muestra la respectiva tabla. La tabla contiene algunas

funciones como el ordenamiento de cierto número de registros que van de los 20 por omisión, a los 40 y 60 por página mejor conocido como paginación.



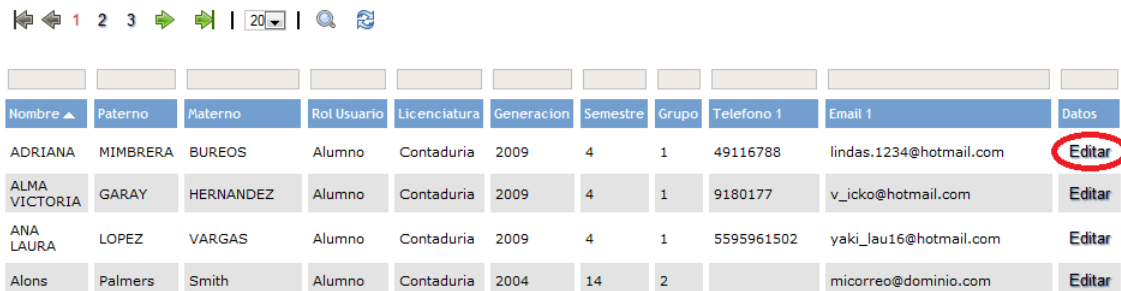
Nombre ▲	Paterno	Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1		belinasan_dom@hotmail.com
Carmen	Ramirez	Monroy	Alumno	Contaduria	2011	4	1	015999148513	karmen01116@yahoo.com.mx
Claudia	García	Galván	Alumno	Contaduria	2009	4	1		cgg_azul_sport@hotmail.es

Figure C.2: Resultados de búsqueda en una tabla

C.2. Modificar datos de los usuarios

Para modificar los datos de los usuarios se deben seguir los siguiente pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Usuarios”.
3. Seleccionar el submenu “Modificar Datos Usuarios”.
4. Seleccionar el tipo de búsqueda e introducir el valor de búsqueda.
5. Presionar el botón “Buscar”.
6. Presionar el enlace editar del usuario al que desea modificar sus datos.



Nombre ▲	Paterno	Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1	Datos
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com	Editar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com	Editar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com	Editar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com	Editar

Figure C.3: Modificar los datos de un usuario

7. Modificar los datos correspondientes del usuario seleccionado con sus respectivas validaciones y campos requeridos.

The screenshot shows a green header bar with the text "Universidad Autónoma del Estado de México" and "Centro Universitario Zumpango". On the right side of the header, there is a link "USUARIO PATERNO" and a button "Cerrar Sesión". Below the header, there is a blue instruction: "Modifique los campos correspondientes y de clic en el botón aceptar de la parte inferior." The form contains three input fields: "Nombre" with the value "Usuario", "Apellido Paterno" with the value "Paterno", and "Apellido Materno" with the value "Materno". At the bottom, there is a checkbox labeled "Cambiar Password" which is currently unchecked.

Figure C.4: Modificar los datos de otros usuarios

8. Presionar el botón “Guardar”.

Si todo se encuentra correctamente el sistema envía un mensaje de confirmación y un correo electrónico al usuario modificado.

C.3. Activar usuarios

Para activar/desactivar los usuarios se deben seguir los siguiente pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Usuarios”.
3. Seleccionar el submenu “Activar Usuarios”.
4. Seleccionar el tipo de búsqueda e introducir el valor de búsqueda.
5. Presionar el botón “Buscar”.
6. Presionar el enlace “Activar/Desactivar” del registro correspondiente al usuario. Como se observa en la figura C.5 la columna “Activo” cambia de true a false y viceversa. True indica que el usuario se encuentra activo, mientras que false indica que no lo está.

The image shows two screenshots of a user management interface. The top screenshot shows a list of users with the following data:

Nombre	Apellido Paterno	Apellido Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Email 1	Activo	Acciones
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	lindas.1234@hotmail.com	true	Activar/Desactivar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	v_icko@hotmail.com	false	Activar/Desactivar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	yaki_lau16@hotmail.com	true	Activar/Desactivar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2	micorreo@dominio.com	false	Activar/Desactivar
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1	belinasan_dom@hotmail.com	true	Activar/Desactivar

The bottom screenshot shows the same list of users, but the 'Activo' status has been updated to 'Desactivar' for all users:

Nombre	Apellido Paterno	Apellido Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Email 1	Activo	Acciones
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	lindas.1234@hotmail.com	true	Activar/Desactivar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	v_icko@hotmail.com	true	Activar/Desactivar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	yaki_lau16@hotmail.com	true	Activar/Desactivar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2	micorreo@dominio.com	false	Activar/Desactivar
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1	belinasan_dom@hotmail.com	true	Activar/Desactivar

Figure C.5: Activar usuarios

C.4. Ver usuarios

Para ver los datos de los usuarios se deben seguir los siguiente pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Usuarios”.
3. Seleccionar el submenu “Ver Usuarios”.
4. Seleccionar el tipo de búsqueda e introducir el valor de búsqueda.
5. Presionar el botón “Buscar”.

The image shows a screenshot of a user management interface with a table of users. The table has the following data:

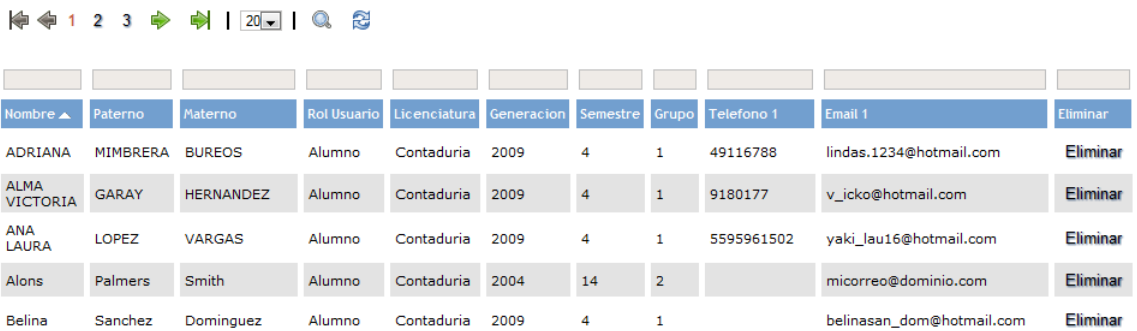
Nombre	Apellido Paterno	Apellido Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1	Acciones
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com	Desactivar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com	Desactivar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com	Desactivar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com	Desactivar
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1		belinasan_dom@hotmail.com	Desactivar

Figure C.6: Tabla ver usuarios

C.5. Eliminar usuarios

Para eliminar los datos de los usuarios se deben seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Usuarios”.
3. Seleccionar el submenu “Eliminar Usuarios”.
4. Seleccionar el tipo de búsqueda e introducir el valor.
5. Presionar el botón “Buscar”.
6. Presionar el enlace “Eliminar”.



Navigation: Home, 1, 2, 3, Search, Refresh

Nombre ▲	Paterno	Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1	Eliminar
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com	Eliminar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com	Eliminar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com	Eliminar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com	Eliminar
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1		belinasan_dom@hotmail.com	Eliminar

Figure C.7: Tabla eliminar usuarios

7. Confirmar la eliminación del usuario presionando en el botón “Aceptar” o dar clic fuera del cuadro de confirmación para cancelar.

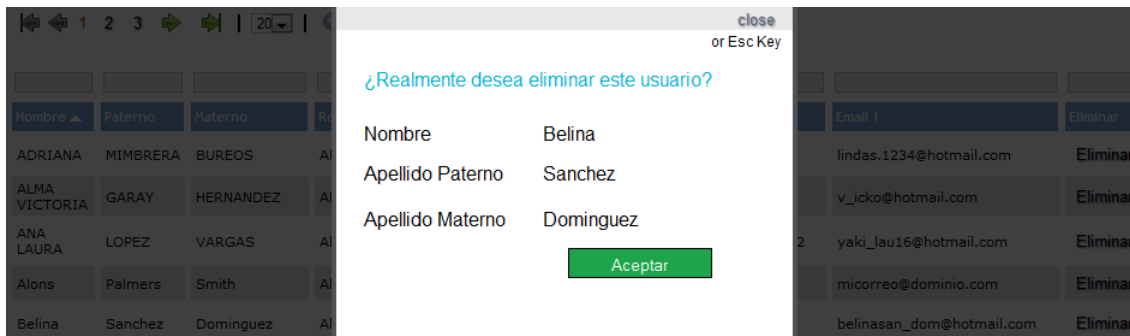



Figure C.8: Confirmar eliminar usuario

Una vez eliminado el usuario se actualiza la tabla como se observa en la figura C.9.




Nombre ▲	Paterno	Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1	Eliminar
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com	Eliminar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com	Eliminar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com	Eliminar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com	Eliminar

Figure C.9: Tabla actualizada al eliminar un usuario

C.6. Modificar permisos usuarios

Para modificar los permisos de los usuarios se deben seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Usuarios”.
3. Seleccionar el submenu “Modificar Permisos”.
4. Seleccionar el tipo de búsqueda e introducir el valor de búsqueda.
5. Presionar el botón “Buscar”.
6. Seleccionar de la tabla el usuario al que se modificaran los permisos dando un clic en el enlace modificar.



Nombre ▲	Paterno	Materno	Rol Usuario	Licenciatura	Generacion	Semestre	Grupo	Telefono 1	Email 1	Permisos
ADRIANA	MIMBRERA	BUREOS	Alumno	Contaduria	2009	4	1	49116788	lindas.1234@hotmail.com	Modificar
ALMA VICTORIA	GARAY	HERNANDEZ	Alumno	Contaduria	2009	4	1	9180177	v_icko@hotmail.com	Modificar
ANA LAURA	LOPEZ	VARGAS	Alumno	Contaduria	2009	4	1	5595961502	yaki_lau16@hotmail.com	Modificar
Alons	Palmers	Smith	Alumno	Contaduria	2004	14	2		micorreo@dominio.com	Modificar
Belina	Sanchez	Dominguez	Alumno	Contaduria	2009	4	1		belinasan_dom@hotmail.com	Modificar

Figure C.10: Tabla modificar permisos a los usuarios

7. Seleccionar el tipo de usuario de la página enviada por el sistema en el punto anterior.
8. Modificar los permisos de usuario si es posible de acuerdo al tipo de usuario.

Estudiante 1	Estudiante 2 (Jefe de grupo)	Administrador 1	Administrador 2 (Todos los permisos)
<input checked="" type="checkbox"/> Editar mis datos	<input checked="" type="checkbox"/> Editar mis datos	<input checked="" type="checkbox"/> Editar mis datos	<input checked="" type="checkbox"/> Editar mis datos
<input type="checkbox"/> Editar Datos Usuarios	<input type="checkbox"/> Editar Datos Usuarios	<input checked="" type="checkbox"/> Editar Datos Usuarios	<input checked="" type="checkbox"/> Editar Datos Usuarios
<input type="checkbox"/> Activar/Desactivar usuarios	<input type="checkbox"/> Activar/Desactivar usuarios	<input checked="" type="checkbox"/> Activar/Desactivar usuarios	<input checked="" type="checkbox"/> Activar/Desactivar usuarios
<input type="checkbox"/> Ver usuarios	<input type="checkbox"/> Ver usuarios	<input checked="" type="checkbox"/> Ver usuarios	<input checked="" type="checkbox"/> Ver usuarios
<input type="checkbox"/> Eliminar Usuarios	<input type="checkbox"/> Eliminar Usuarios	<input checked="" type="checkbox"/> Eliminar Usuarios	<input checked="" type="checkbox"/> Eliminar Usuarios
<input type="checkbox"/> Editar permisos usuarios	<input type="checkbox"/> Editar permisos usuarios	<input type="checkbox"/> Editar permisos usuarios	<input checked="" type="checkbox"/> Editar permisos usuarios
<input type="checkbox"/> Enviar Email Usuarios	<input type="checkbox"/> Enviar Email Usuarios	<input checked="" type="checkbox"/> Enviar Email Usuarios	<input checked="" type="checkbox"/> Enviar Email Usuarios
<input checked="" type="checkbox"/> Contestar Encuesta	<input checked="" type="checkbox"/> Contestar Encuesta	<input checked="" type="checkbox"/> Contestar Encuesta	<input checked="" type="checkbox"/> Contestar Encuesta
<input type="checkbox"/> Ver Resultados Encuesta	<input checked="" type="checkbox"/> Ver Resultados Encuesta	<input checked="" type="checkbox"/> Ver Resultados Encuesta	<input checked="" type="checkbox"/> Ver Resultados Encuesta
<input type="checkbox"/> Crear Encuesta	<input type="checkbox"/> Crear Encuesta	<input checked="" type="checkbox"/> Crear Encuesta	<input checked="" type="checkbox"/> Crear Encuesta
<input type="checkbox"/> Crear Reporte	<input type="checkbox"/> Crear Reporte	<input checked="" type="checkbox"/> Crear Reporte	<input checked="" type="checkbox"/> Crear Reporte
<input type="checkbox"/> Respaldar Datos del Sistema	<input type="checkbox"/> Respaldar Datos del Sistema	<input type="checkbox"/> Respaldar Datos del Sistema	<input checked="" type="checkbox"/> Respaldar Datos del Sistema
<input type="checkbox"/> Eliminar Datos del sistema	<input type="checkbox"/> Eliminar Datos del sistema	<input type="checkbox"/> Eliminar Datos del sistema	<input checked="" type="checkbox"/> Eliminar Datos del sistema
<input type="checkbox"/> Configurar Sistema	<input type="checkbox"/> Configurar Sistema	<input checked="" type="checkbox"/> Configurar Sistema	<input checked="" type="checkbox"/> Configurar Sistema
<input type="checkbox"/> Configurar Servidor	<input type="checkbox"/> Configurar Servidor	<input checked="" type="checkbox"/> Configurar Servidor	<input checked="" type="checkbox"/> Configurar Servidor
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>

Figure C.11: Permisos del usuario

9. Presionar el botón “Guardar”.

C.7. Enviar correo electrónico

Para enviar un correo electrónico desde el sistema, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menú “Usuarios”.
3. Seleccionar el submenú “Enviar Email”.

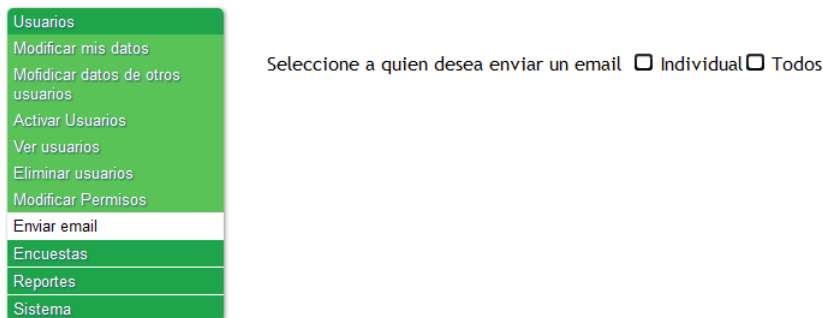


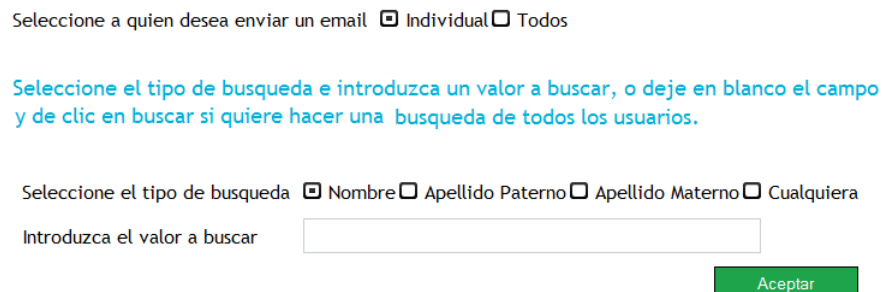
Figure C.12: Enviar correo electrónico

4. Seleccionar a quien desea enviar el correo electrónico.

C.8. Enviar correo electrónico individual

Para enviar un correo electrónico individual, hay que seguir los siguientes pasos:

1. Realizar los pasos para enviar un correo electrónico.
2. Seleccionar la opción “Individual”.
3. Seleccionar el tipo de búsqueda.
4. Introducir un valor de búsqueda.
5. Presionar el botón “Aceptar”



Seleccione a quien desea enviar un email Individual Todos

Seleccione el tipo de búsqueda e introduzca un valor a buscar, o deje en blanco el campo y de clic en buscar si quiere hacer una búsqueda de todos los usuarios.

Seleccione el tipo de búsqueda Nombre Apellido Paterno Apellido Materno Cualquiera

Introduzca el valor a buscar

Figure C.13: Buscar usuario para enviar correo electrónico

6. Seleccionar de la tabla el usuario al que desea enviar el correo electrónico.
7. Escribir el asunto del correo electrónico
8. Escribir el mensaje del correo electrónico.

maria del carmen	escorcia	cajiga	marikar_1628@hotmail.com	<input type="checkbox"/>
phillip	bautista	bautista	fsafa@sasag.com	<input type="checkbox"/>
Alons	Palmers	Smith	micorreio@dominio.com	<input checked="" type="checkbox"/>
admin1	pat	mat	christianflara@hotmail.com	<input type="checkbox"/>
admin1	pat	mat	christianflara@hotmail.com	<input type="checkbox"/>

Escriba el asunto

Escriba el mensaje

Font family Font size Paragraph

B *I* U | [List icons] | [Link/Unlink icons] | HTML | [Color icon] | ab

Mensaje del correo con formato

Figure C.14: Asunto y mensaje del correo electrónico individual

9. Presionar el botón “Aceptar”.

C.9. Enviar correo electrónico a todos

Para enviar un correo electrónico dirigido a todos, hay que seguir los siguientes pasos:

1. Realizar los pasos para enviar un correo electrónico.
2. Seleccionar la opción “Todos”.
3. Escribir el asunto del correo electrónico.
4. Escribir el mensaje del correo electrónico.

Seleccione a quien desea enviar un email Individual Todos

Introduzca el asunto y el mensaje a enviar

Escriba el asunto

Escriba el mensaje

Mensaje del email con formato

Figure C.15: Asunto y mensaje del correo electrónico dirigido a todos

5. Presionar el botón “Aceptar”.

C.10. Resultados encuestas

Para ver los resultados de la encuesta, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menú “Encuestas”.
3. Seleccionar el submenú “Resultados encuesta”.

- Usuarios
- Encuestas
- Resultados encuesta
- Encuesta Socioeconómica
- Encuesta de Valores
- Encuesta de Afirmaciones
- Encuesta de Aptitudes
- Reportes
- Sistema

Seleccione un tipo de búsqueda

Todos Por Licenciatura Por Generación Por Semestre Por Grupo Por Nombre

Figure C.16: Resultados encuesta

4. Seleccionar el tipo de búsqueda. (Todos, licenciatura, generación, semestre, grupo, nombre).
5. Seleccionar o introducir una opción de acuerdo al tipo de búsqueda.

Selecciones un tipo de búsqueda

Todos
 Por Licenciatura
 Por Generación
 Por Semestre
 Por Grupo
 Por Nombre

Contaduría ▾

Aceptar

Cancelar

Figure C.17: Selección de una opción Resultados encuesta

6. Presionar el botón “Aceptar”.
7. Se muestra una tabla con el resultado de la búsqueda. Seleccione el enlace del registro del usuario según la encuesta que desee (Socioeconómica, valores, aptitudes y afirmaciones).
8. Se muestra un archivo en PDF con los resultados de la encuesta.

75% Buscar


 Universidad Autónoma del Estado de México
 Centro Universitario Zumpango

REPORTE: SOCIOECONOMICA FINAL **FECHA APLICACIÓN:** 21/07/2011
ALUMNO: ADRIANA MIMBRERA BUREOS **NO. CUENTA:** 0920677

Nombre: ADRIANA MIMBRERA BUREOS
Estado Civil: Soltero **No. de Hijos:** 0
Lugar de Procedencia: SAN ANDRES JALTENCO ESTADO DE MEXICO
Domicilio: CALLE 7 DE MARZO SN, SAN ANDRES JALTENCO, ESTADO DE MEXICO, C.P.
Teléfono: 49116788 **Celular:** 5525586334 **Email:** lindas.1234@hotmail.com
¿Trabajas?: No
Nombre de la Empresa:
Dirección de la Empresa:

Figure C.18: Archivo PDF con los resultados de la encuesta según usuario

En caso de que el usuario no haya finalizado de contestar la encuesta se muestra una página como en la figura C.19.



No existen datos para crear el reporte correspondiente.

[Cerrar esta ventana](#)

Figure C.19: Encuesta no contestada

C.11. Generar reportes

Para generar reportes se deben seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Reportes”.
3. Seleccionar el submenu “Generar reportes”.

Figure C.20: Generar reportes

4. Seleccionar un tipo de búsqueda (todos, por licenciatura, por generación, por semestre, por grupo, por escuela, por nombre).
5. Seleccionar la opción valores de la categoría de reportes.
6. Seleccionar el subreporte de la encuesta valores (tolerancia, confianza, honestidad, lealtad, respeto, disciplina, amistad, iniciativa, simpatico, lider, seguro).
7. Seleccionar el formato de salida (PDF o Excel).
8. Presionar el botón “Aceptar” para crear el reporte.

En la figura C.21 se muestra la primer página que genera el archivo la cual contiene una tabla de los usuarios y respuestas de la encuesta.

No. Cuenta	TOLERANCIA				
	Importante	Útil	Duradera	Incondicionada	Deseable
092068	5	4	4	5	4
0920684	1	1	1	1	1
0822936	1	1	2	1	1
0920663	4	4	4	4	4
0620745	1	1	1	4	4
0920694	1	2	1	3	4
0920696	1	2	2	3	2
0920681	2	2	4	4	4
0920672	1	1	2	1	1

Figure C.21: Reporte en formato PDF

En la figura C.22 se muestra las gráficas de pastel correspondiente al número de usuarios que contestaron la encuesta y sus respuestas.

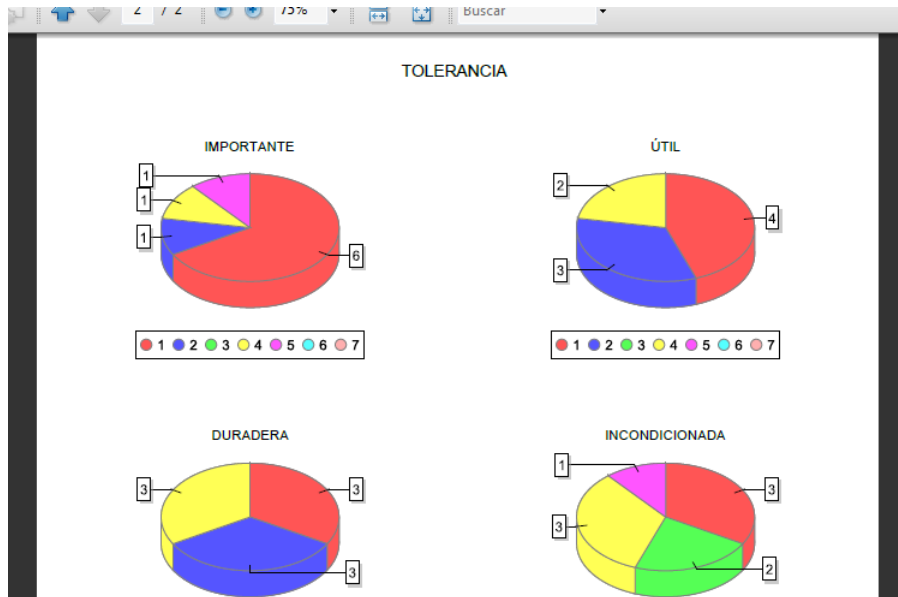


Figure C.22: Gráficas de los reportes en PDF

C.12. Crear respaldo

Para respaldar los datos del sistema como son usuarios, permisos, resultados de la encuesta, etc. hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Respaldo datos”.
4. Seleccionar la opción “Crear respaldo”.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

- Respaldo base de datos completa
- Respaldo usuarios, permisos usuarios
- Respaldo licenciaturas, generaciones, semestres, grupos y escuelas
- Respaldo servidor
- Respaldo encuesta socioeconómica
- Respaldo encuesta de valores
- Respaldo encuesta de afirmaciones
- Respaldo encuesta de aptitudes

Descripción

Figure C.23: Respaldo datos

5. Seleccionar respaldar la base de datos completa, o los datos que desee.
6. Llenar el area de texto descripción . (Opcional).
7. Presionar el botón “Aceptar”.

Si el respaldo se creó correctamente se muestra el mensaje de la figura C.24.

- El respaldo fue guardado exitosamente.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Figure C.24: Respaldo creado correctamente

C.13. Ver respaldos

Para respaldar los datos del sistema como son usuarios, permisos, resultados de la encuesta, etc. hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea este permiso.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Respaldar datos”.
4. Seleccionar la opción “Ver respaldos”.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Seleccione una opción Antes de la fecha Despues de la fecha Entre la fecha

De la fecha

Figure C.25: Ver respaldos

5. Seleccionar el criterio de búsqueda por fecha (antes de la fecha, después de la fecha, entre las dos fechas).
6. Seleccionar la fecha dando un clic al calendario.
7. Presionar el botón “Aceptar”.

Si existen respaldos con las fechas previamente seleccionadas se muestra una página con la una tabla de los respaldos.

Respaldo	Fecha	Archivo	Archivo	Restaurar	Eliminar
Base datos completa	18/07/2011	respaldos/respaldo2011-07-1816-5-29.sql	Descargar	Restaurar	Eliminar

Resultados 1 - 1 de 1.

Figure C.26: Tabla respaldos

C.14. Cargar o subir respaldo

Para cargar o subir un respaldo al sistema, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Respaldar datos”.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Subir respaldo

Descripción

*Este campo es requerido

Figure C.27: Cargar un respaldo al sistema

4. Seleccionar la opción “Cargar respaldo”.
5. Presionar el botón examinar para buscar el archivo a subir.
6. Escribir la descripción del archivo. (Opcional).
7. Presionar el botón “Aceptar”.

Si el archivo se subió correctamente se muestra un mensaje como el que se muestra en la figura C.28.

- El archivo ha sido subido exitosamente.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Figure C.28: Archivo subido correctamente

C.15. Descargar respaldo

Para descargar el respaldo del sistema, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.

2. Seleccionar el menú “Sistema”.
3. Seleccionar el submenú “Respalidar datos”.
4. Seleccionar la opción “Ver respaldos”.
5. Seleccionar el criterio de búsqueda por fecha (antes de la fecha, después de la fecha, entre la fecha).
6. Seleccionar la fecha dando un clic al icono del calendario.
7. Presionar el botón “Aceptar”.
8. Seleccionar el enlace descargar de la tabla de respaldos.

Respaldo	Fecha	Archivo	Archivo	Restaurar	Eliminar
Base datos completa	18/07/2011	respaldos/respaldo2011-07-1816-5-29.sql	Descargar	Restaurar	Eliminar

Figure C.29: Tabla respaldos

9. Aceptar la descarga del archivo.

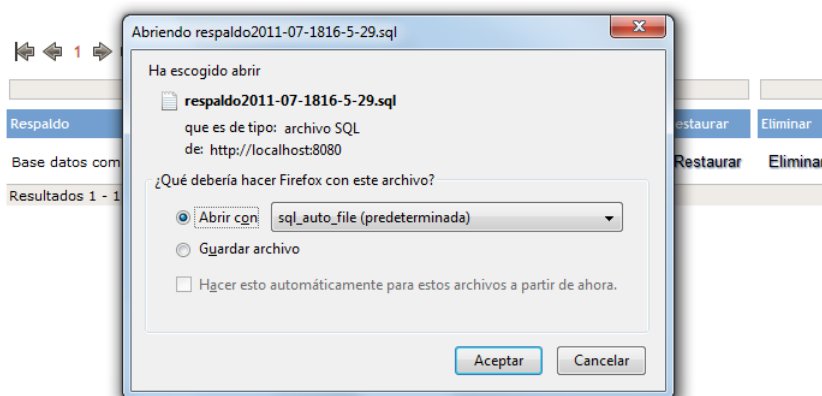


Figure C.30: Descargar respaldo

En caso de que el sistema muestre el mensaje de la figura C.31 quiere decir que el archivo no se encuentra disponible ya que este pudo ser eliminado manualmente por lo que ya no será accesible.

- Ha ocurrido un error es posible que el archivo no se encuentre disponible.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Figure C.31: Archivo no disponible

C.16. Restaurar respaldos

Para descargar el respaldo del sistema, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Respaldar datos”.
4. Seleccionar la opción “Ver respaldos”.
5. Seleccionar el criterio de búsqueda por fecha (antes de la fecha, despues de la fecha, entre la fecha).
6. Seleccionar la fecha dando un clic al icono del calendario.
7. Presionar el botón “Aceptar”.
8. Seleccionar el enlace restaurar del respectivo respaldo dentro de la tabla de respaldos.
9. Presionar el botón “Aceptar” si desea confirmar la restauración. En caso contrario de clic fuera del cuadro de confirmación.

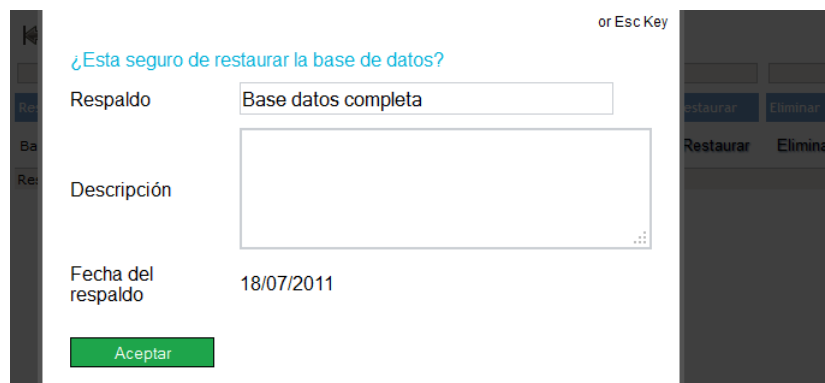


Figure C.32: Confirmación de restaurar un respaldo

Si la restauración se llevo a cabo correctamente se muestra el mensaje como en la figura C.33.

- El sistema fue restaurado correctamente.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Figure C.33: Restauración del respaldo correctamente

C.17. Eliminar respaldos

Para descargar el respaldo del sistema, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Respaldo datos”.
4. Seleccionar la opción “Ver respaldos”.
5. Seleccionar el criterio de búsqueda por fecha (antes de la fecha, despues de la fecha, entre la fecha).
6. Seleccionar la fecha dando un clic al icono del calendario.
7. Presionar el botón “Aceptar”.
8. Seleccionar el enlace “Eliminar” del respectivo respaldo dentro de la tabla de respaldos.
9. Confirmar la eliminación del respaldo dando presionando el botón “Aceptar” del cuadro de confirmación.

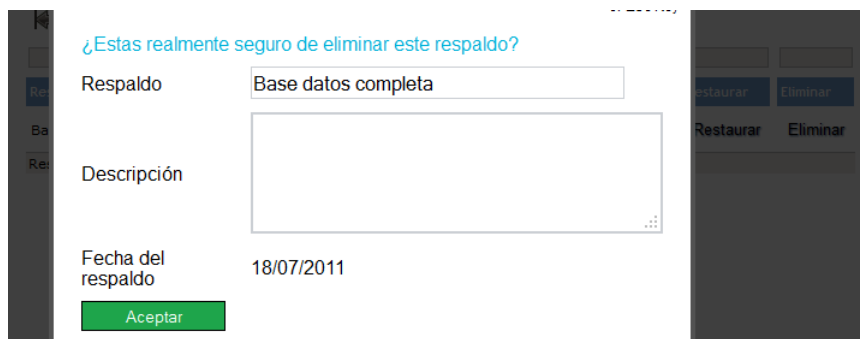


Figure C.34: Confirmar eliminar respaldo

Si el respaldo fue eliminado correctamente se muestra un mensaje como el de la figura C.35.

- El respaldo fue eliminado exitosamente.

Seleccione una opción Crear respaldo Ver respaldos Cargar respaldo

Figure C.35: Eliminación del respaldo correctamente

C.18. Eliminar Datos

Para eliminar datos del sistema, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Eliminar datos”.

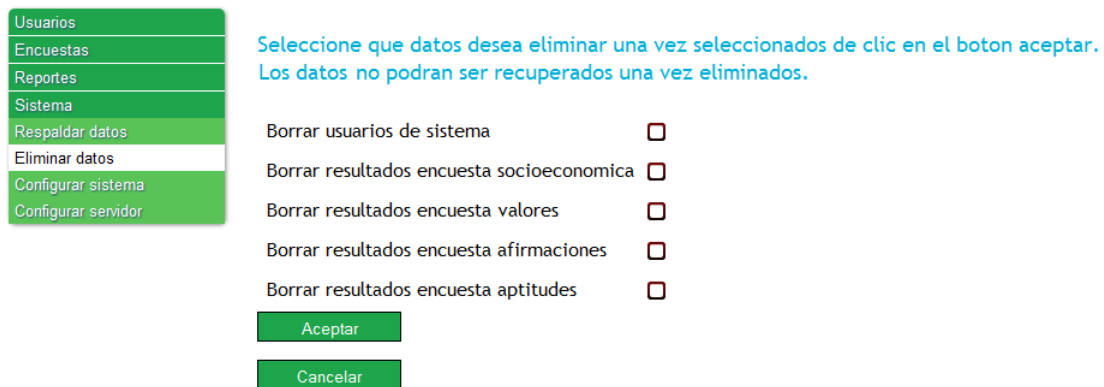


Figure C.36: Eliminar datos

4. Seleccionar los datos que desee eliminar del sistema (Usuarios, encuesta socioeconómica, valores, afirmaciones, aptitudes).
5. Presionar el botón “Aceptar”.
6. Se muestra una página para confirmar la eliminación. En esta parte no se pueden modificar las opciones previamente seleccionadas.

¿Realmente desea eliminar estos datos?

- Borrar usuarios de sistema
- Borrar resultados encuesta socioeconomica
- Borrar resultados encuesta valores
- Borrar resultados encuesta afirmaciones
- Borrar resultados encuesta aptitudes

Aceptar

Cancelar

Figure C.37: Confirmación de eliminación de los datos

7. Presionar el botón “Aceptar”.

C.19. Configurar sistema

Para configurar ciertos aspectos del sistema, como lo son las licenciaturas, semestres, generaciones, grupos y escuelas, hay que seguir los siguientes pasos:

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Configurar sistema”.
4. Seleccionar una pestaña correspondiente a los datos que quieras modificar. (Licenciaturas, semestres, generaciones, grupos y escuelas).

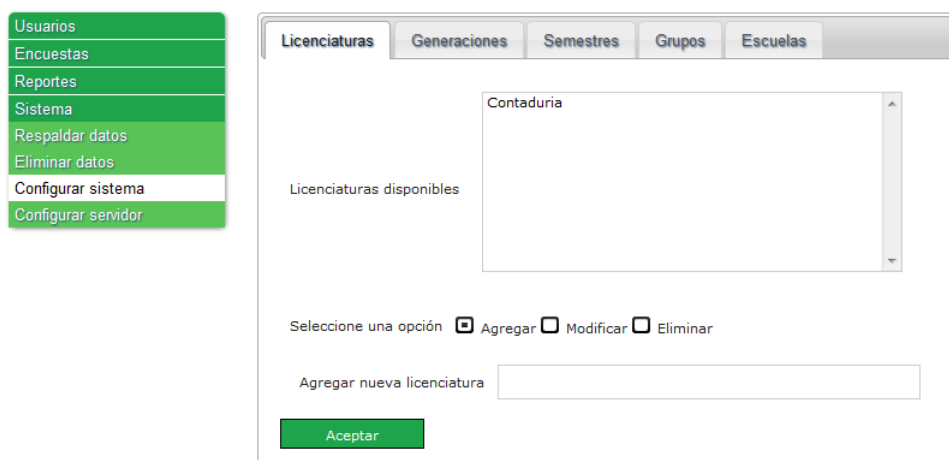
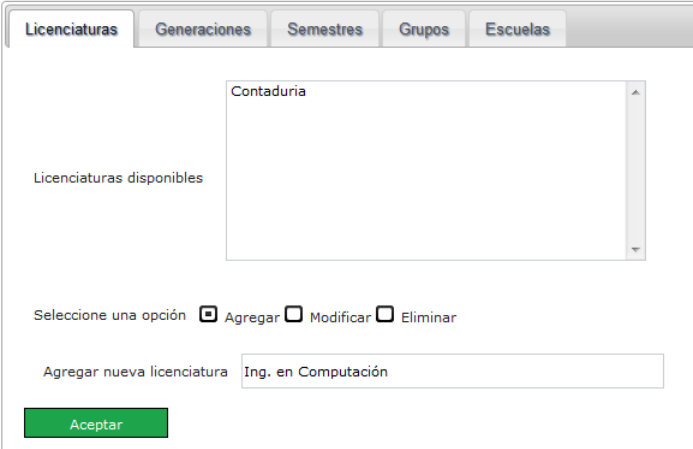


Figure C.38: Pestaña licenciaturas

C.20. Agregar licenciatura, generación, semestre, grupo y escuelas

Para agregar una nueva licenciatura, hay que seguir los siguientes pasos:

1. Previamente hacer los pasos para configurar sistema.
2. Seleccionar la pestaña “Licenciaturas”.
3. Seleccionar la opción “Agregar”.
4. Escribir el nombre de licenciatura en el respectivo campo.
5. Presionar el botón “Aceptar”.



The screenshot shows a web application interface for managing degrees. At the top, there are five tabs: 'Licenciaturas', 'Generaciones', 'Semestres', 'Grupos', and 'Escuelas'. The 'Licenciaturas' tab is active. Below the tabs, there is a list of available degrees. The first item is 'Contaduría'. Below the list, there is a section labeled 'Licenciaturas disponibles'. Underneath, there are three radio buttons: 'Agregar' (selected), 'Modificar', and 'Eliminar'. Below the radio buttons, there is a text input field labeled 'Agregar nueva licenciatura' with the text 'Ing. en Computación' entered. At the bottom, there is a green button labeled 'Aceptar'.

Figure C.39: Agregar licenciatura

En caso de que se repita la licenciatura se muestra un mensaje como el de la figura C.40.

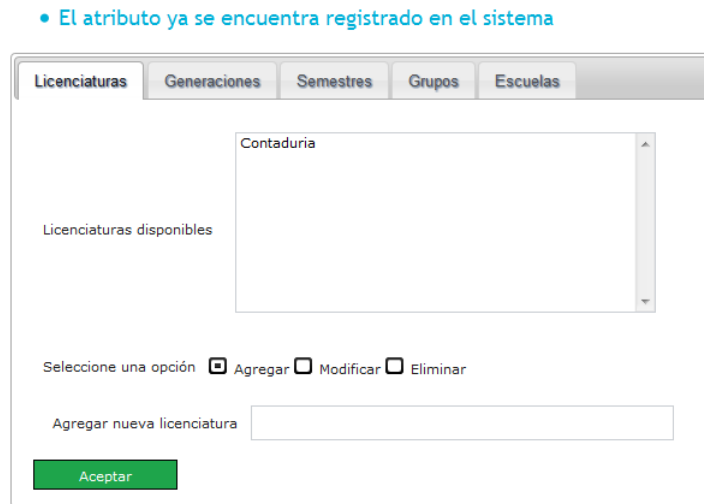


Figure C.40: Atributo ya registrado en el sistema

Nota: Este procedimiento es similar para las pestañas licenciaturas, generaciones, semestres y grupos en sus respectivas pestañas.

C.21. Modificar licenciatura, generación, semestre, grupo y escuelas

Para modificar una licenciatura, hay que seguir los siguientes pasos:

1. Previamente hacer los pasos para configurar sistema.
2. Seleccionar la pestaña “Licenciaturas”.
3. Seleccionar la opción “Modificar”.
4. Seleccionar dentro del recuadro la licenciatura que desea modificar.
5. Escribir el nombre de licenciatura en el respectivo campo por el cual quiere reemplazar.

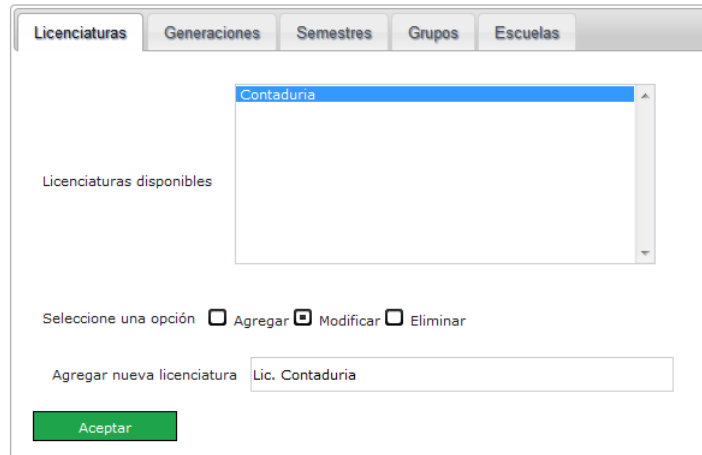


Figure C.41: Modificar el nombre la licenciatura

En caso de que se repita la licenciatura se muestra un mensaje como el de la figura C.42.

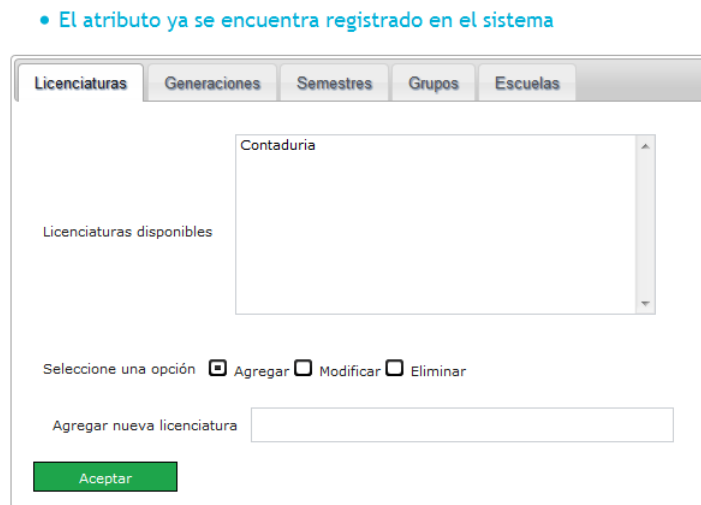


Figure C.42: Atributo ya registrado en el sistema

Nota: Este procedimiento es similar para las pestañas licenciaturas, generaciones, semestres y grupos en sus respectivas pestañas.

C.22. Eliminar licenciatura, generación, semestre, grupo y escuelas

Para eliminar una licenciatura, hay que seguir los siguientes pasos:

1. Previamente hacer los pasos para configurar sistema.

2. Seleccionar la pestaña “Licenciaturas”.
3. Seleccionar la opción “Eliminar”.
4. Seleccionar dentro del recuadro la licenciatura que desea eliminar.
5. Presionar el botón “Aceptar”.

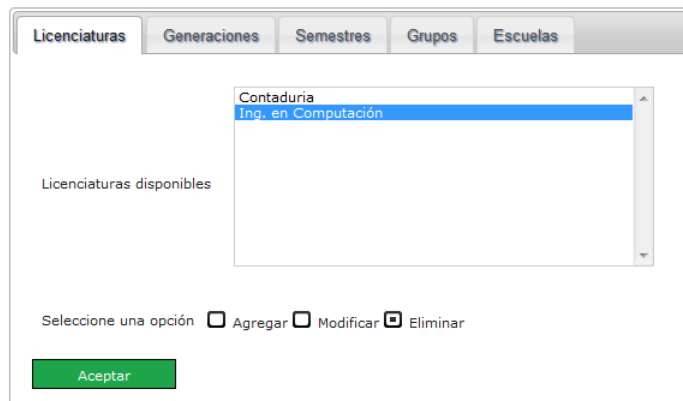


Figure C.43: Eliminar licenciatura

Por seguridad en el funcionamiento de la aplicación debe de tener por lo menos 1 licenciatura registrada, por lo cual no es posible eliminar todas las licenciaturas. En caso de tratar de eliminarlas todas se muestra el mensaje de la figura C.44.

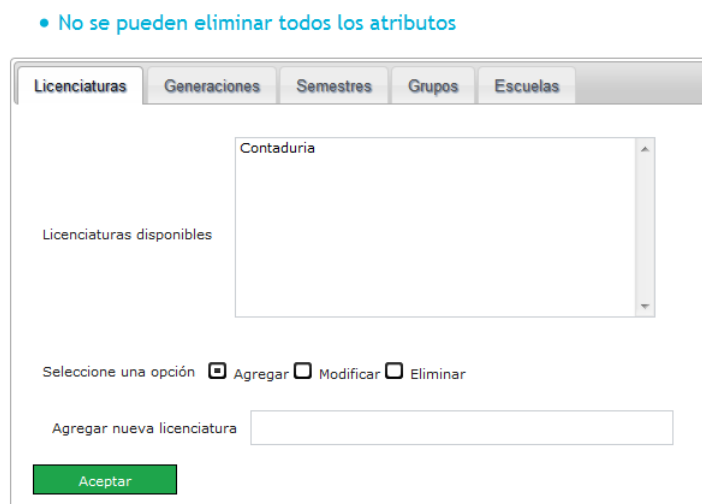


Figure C.44: Eliminar todas las licenciaturas

Nota: Este procedimiento es similar para las pestañas licenciaturas, generaciones, semestres y grupos en sus respectivas pestañas.

C.23. Configurar servidor

Para configurar algunas opciones del servidor como lo es la dirección IP de donde esta montada la aplicación, el nombre de dominio, la cuenta de correo electrónico y contraseña con la que trabajara.

1. Ingresar al sistema con una cuenta administrativa que posea el permiso de respaldar datos.
2. Seleccionar el menu “Sistema”.
3. Seleccionar el submenu “Configurar servidor”

Ip del servidor	<input type="text" value="148.215."/>
Activar nombre de dominio	<input type="checkbox"/>
Nombre del dominio	<input type="text" value="midominio.com"/>
Email	<input type="text" value="correo@gmail.com"/>
Password	<input type="password"/>
SMTP (Protocolo Simple de Transferencia de Correo)	<input type="text" value="smtp.gmail.com"/>
Ingrese un email de prueba al que pueda acceder	<input type="text"/>
<input type="button" value="Guardar"/>	
<input type="button" value="Cancelar"/>	

Figure C.45: Configurar el servidor

4. Realizar los posibles cambios en la IP, o si desea ocupar un nombre de dominio seleccionar la casilla. (Opcional)
5. Cambiar la cuenta de correo electrónico utiliza por el sistema. (Opcional pero debe poseer una cuenta válida).
6. Introducir la contraseña del correo electrónico utilizada por el sistema.
7. Ingresar un correo electrónico de prueba. Esto es para verificar que la cuenta de correo electrónico utilizada por el servidor sea completamente válida.
8. Presionar el botón “Guardar”.

Al guardar los respectivos cambios, se muestra un mensaje como en la figura C.46. Como sugerencia revise que efectivamente le haya llegado un correo electrónico a la cuenta introducida.

- Se han guardado los cambios revise el email que introdujo de prueba.

Ip del servidor	<input type="text" value="148.215."/>
Activar nombre de dominio	<input type="checkbox"/>
Nombre del dominio	<input type="text" value="midominio.com"/>
Email	<input type="text" value="correo@gmail.com"/>
Password	<input type="text"/>
SMTP (Protocolo Simple de Transferencia de Correo)	<input type="text" value="smtp.gmail.com"/>
Ingrese un email de prueba al que pueda acceder	<input type="text"/>

Figure C.46: Cambios guardados correctamente en el servidor