

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO E IMPLANTACIÓN DEL SITIO WEB DE LA ADEPON

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS
INFORMÁTICOS Y DE COMPUTACIÓN

BRAVO LITUMA MARTÍN ERNESTO
martin.bravo@fisepn.ec

TORRES ZAMBRANO JOSÉ XAVIER
jose.torres@fisepn.ec

DIRECTOR: ING. MARCOS RAÚL CÓRDOVA BAYAS
raul.cordova@epn.edu.ec

Quito, Mayo 2013

DECLARACIÓN

Nosotros, Bravo Lituma Martín Ernesto y Torres Zambrano José Xavier declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

BRAVO LITUMA MARTÍN ERNESTO

TORRES ZAMBRANO JOSÉ XAVIER

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Martín Ernesto Bravo Lituma y José Xavier Torres Zambrano, bajo mi supervisión.

Ing. Raúl Córdova
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a mis amigos y compañeros que han compartido conmigo los éxitos y los fracasos, las alegrías y las penas, los buenos y malos ratos en esta etapa de mi vida, en especial a José Torres por su apoyo y paciencia.

A todos mis Maestros por haber sido parte de mi formación y aprendizaje en mi carrera, en especial al ingeniero Raúl Córdova quien hizo posible esta tesis.

A la ADEPON por habernos abierto sus puertas y permitido desarrollar esta tesis.

En general agradezco a todos aquellos que de una forma u otra han dejado huella en mi vida aunque ya no formen parte de ella.

Martín Bravo L.

AGRADECIMIENTO

A mis padres por brindarme la oportunidad de estudiar, haberme dado siempre la libertad de escoger mis metas y apoyarme para que las cumpla.

A mi hermano David.

Al ingeniero Raúl Córdova.

A la ADEPON y al ingeniero Fabián Barba.

A la Poli y a los amigos que hicieron de ella un lugar agradable y divertido.

José Torres Z.

DEDICATORIA

La presente tesis se la dedico ante todo a mi familia, pues su apoyo y confianza es lo que me permite seguir adelante buscando siempre ser un mejor hombre.

A mis padres que me dieron todo lo necesario para que pueda culminar mis estudios.

Al mejor de los hermanos.

Al ingeniero Raúl Córdova.

Martín Bravo L.

DEDICATORIA

A mis padres, a mi hermano.

Y a todos los amigos que me acompañaron durante estos 6 años.

José Torres Z.

CONTENIDO

CAPITULO I.....	1
1 Definición del problema	1
1.1 Descripción de la asociación de profesores de la Escuela Politécnica Nacional.....	1
1.2 Justificación de la metodología de desarrollo.....	3
1.2.1 Características de la metodología xp	3
1.2.2 La metodología XP en el desarrollo web.....	4
1.2.3 Comparación entre las características de la metodología XP y las del presente proyecto.....	5
1.3 Justificación de las herramientas a utilizar.....	7
1.3.1 JAVA.....	7
1.3.2 Plataforma de programación JEE6	8
1.3.3 Marco de trabajo ICEFACES	9
1.3.4 Gestor de base de datos relacional MYSQL	10
1.3.5 Entorno de desarrollo integrado NetBeans	10
CAPITULO II.....	12
2 Planificación y diseño	12
2.1 Historias de usuario	12
2.1.1 Valoración de historias de usuario	21
2.2 Plan de entrega	22
2.2.1 Calendario del plan de entrega	23
2.3 Metáfora del sistema	24
2.4 Tarjetas CRC	27
2.4.1 Tarjetas CRC del sistema.....	28
2.4.2 Estructura del sitio web	35
2.4.3 Diseño de interfaces gráficas.....	38

2.4.4	Especificación de la arquitectura del sistema	38
2.4.5	Diseño de la base de datos.....	39
CAPITULO III.....		41
3	Implementación, pruebas, implantación y análisis de resultados.	41
3.1	Implementación	41
3.1.1	Implementación de la capa de persistencia (Modelo en el patrón MVC)	42
3.1.1.1	Clases Entidad (Entities)	43
3.1.1.2	Componentes Session Bean	46
3.1.2	Implementación de la capa de lógica del negocio (Controlador en el patrón MVC).....	49
3.1.3	Implementación de la capa de Presentación (Vista en el patrón MVC)	54
3.1.4	Implementación de seguridades	58
3.2	Pruebas	63
3.2.1	Pruebas unitarias	63
3.2.2	Pruebas de aceptación	65
3.3	Implantación	84
3.3.1	Afinamiento del servidor GlassFish en el entorno de producción	86
3.3.1.1	Tamaño de la Pila de Java.....	86
3.3.1.2	Hilos de procesamiento de peticiones HTTP.....	88
3.3.1.3	Pool de Conexiones	89
3.3.1.4	Contenedor de EJBs	91
3.3.2	Pruebas de rendimiento	92
3.3.3	Presentación Oficial del Sitio web	94
3.4	Análisis de resultados.....	95
3.4.1	Descripción de la encuesta	95
3.4.2	Resultado de la encuesta.....	96

CAPITULO IV	99
4 Conclusiones y recomendaciones	99
4.1 Conclusiones	99
4.2 Recomendaciones	100
GLOSARIO	101
BIBLIOGRAFÍA.....	103
ANEXOS.....	105

ÍNDICE DE FIGURAS

Figura 2.1: Plantilla Historias de Usuario.....	12
Figura 2.2: Tarjeta CRC.....	28
Figura 2.3: Mapa del Sitio para el usuario Administrador.....	36
Figura 2.4: Mapa del Sitio para Usuarios Registrados.....	37
Figura 2.5: Mapa del Sitio para Usuarios No Registrados.....	38
Figura 2.6: Arquitectura del Sistema.....	39
Figura 2.7: Modelo Entidad Relación.....	40
Figura 3.1: Creación de Proyecto JEE.....	41
Figura 3.2: Estructura del Proyecto	42
Figura 3.3: Creación de una Clase Entidad.....	44
Figura 3.4: Código de una Clase Entidad.....	45
Figura 3.5: Creación de un componente Session Bean.....	46
Figura 3.6: Código de un componente Session Bean.....	47
Figura 3.7: Código de la clase Abstract Facade.....	48
Figura 3.8: Creación de un componente Managed Bean.....	50
Figura 3.9 Códido de un Managed Bean.....	50
Figura 3.10: Bloque de imports de un Managed Bean.....	51
Figura 3.11: Bloque de declaración de variables de un ManagedBean.....	52
Figura 3.12: Lógica de un Managed Bean.....	53
Figura 3.13: Métodos getters y setters de un Managed Bean.....	54
Figura 3.14: Creación de una página xhtml.....	55
Figura 3.15: Codigo de una página xhtml nueva.....	56
Figura 3.16: Código de la plantilla implementada por las páginas xhtml.....	57
Figura 3.17: Código de una de las páginas xhtml del proyecto.....	58
Figura 3.18 Configuración de un “realm” de seguridad en Glassfish.....	59
Figura 3.19 Autenticación de Usuarios.....	60
Figura 3.20. Tipo de Autenticación y Roles.....	60
Figura 3.21 Restricción de Seguridad.....	61
Figura 3.22 Archivo <i>glassfish-web.xml</i>	62
Figura 3.23 Método de encriptación SHA-256.....	62

Figura 3.24: Librerías para pruebas Unitarias en ambientes JEE.....	63
Figura 3.25: Prueba Unitaria, métodos: @BeforeClass y @AfterClass	64
Figura 3.26: Método prueba.....	65
Figura 3.27: Prueba de Aceptación.....	66
Figura 3.28: Panel de control del dominio	85
Figura 3.29: Proyecto en estado de Producción	86
Figura 3.30: Afinamiento Tamaño de la Pila de JAVA	87
Figura 3.31: Afinamiento Hilos de procesamiento de peticiones HTTP	89
Figura 3.32: Afinamiento Pool de conexiones.....	90
Figura 3.33: Afinamiento Contenedor EJBs	92
Figura 3.34 Tiempo de respuesta del servidor y conexiones activas.....	93
Figura 3.35 Tiempo de carga y conexiones activas.	94
Figura 43. Usabilidad del sitio.....	96
Figura 44. Diseño del Sitio.....	96
Figura 45. Beneficio a la ADEPON.	97
Figura 46. Tiempo de respuesta del sitio.	97
Figura 47. Satisfacción con el sitio.	98

ÍNDICE DE TABLAS

Tabla 1.1 Ganancias en productividad con XP.....	5
Tabla 1.2 Características del proyecto y de la metodología XP.	6
Tabla 1.3 Características del proyecto y de JAVA.....	8
Tabla 1.4 Características del proyecto y de JSF.	9
Tabla 2.1 : Historia de Usuario Publicación de noticias.	13
Tabla 2.2 : Historia de Usuario Presentación de noticias.....	14
Tabla 2.3: Historia de Usuario Publicación de Eventos.....	14
Tabla 2.4: Historia de Usuario Presentación Eventos.	15
Tabla 2.5: Historia de Usuario Publicación de Reglamentos y Resoluciones.	15
Tabla 2.6: Historia de Usuario Presentación de Reglamentos y Resoluciones.....	16
Tabla 2.7: Historia de Usuario Publicación Periódico.....	16
Tabla 2.8: Historia de Usuario Presentación Periódico.	17
Tabla 2.9: Historia de Usuario Publicación de encuestas.	17
Tabla 2.10: Historia de Usuario Responder encuestas.	18
Tabla 2.11: Historia de Usuario Resultados de Encuestas.....	18
Tabla 2.12: Historia de Usuario Visualizar video en vivo.....	19
Tabla 2.13: Historia de Usuario Registro de Usuarios.	19
Tabla 2.14: Historia de Usuario Edición Cuentas de usuarios.	20
Tabla 2.15 : Historia de Usuario Administración de Información sobre la ADEPON.	20
Tabla 2.16: Historia de Usuario Presentación de Información sobre la ADEPON.....	21
Tabla 2.17 : Estimación de Historias de Usuario.....	22
Tabla 2.18: Plan de Entregas.....	23
Tabla 2.19: Calendario del Plan de Entregas.....	24
Tabla 2.20: Tarjeta CRC ARCHIVO_PDF.....	28
Tabla 2.21: Tarjeta CRC COMENTARIO.....	29
Tabla 2.22: Tarjeta CRC DIRECTIVA_ADEPON.....	29
Tabla 2.23: Tarjeta CRC ENCUESTA.....	30
Tabla 2.24: Tarjeta CRC EVENTO.....	30
Tabla 2.25: Tarjeta CRC IMAGEN.....	31
Tabla 2.26: Tarjeta CRC INFORMACION.....	31

Tabla 2.27: Tarjeta CRC NOTICIA	32
Tabla 2.28: Tarjeta CRC OPCION.....	32
Tabla 2.29: Tarjeta CRC PERIODICO.....	33
Tabla 2.30: Tarjeta CRC PREGUNTA	33
Tabla 2.31: Tarjeta CRC REGLAMENTO	34
Tabla 2.32: Tarjeta CRC USUARIO	34
Tabla 2.33: Tarjeta CRC VIDEO.....	35
Tabla 3.1: Prueba de Aceptación Registro de Noticia	67
Tabla 3.2: Prueba de Aceptación Presentación de Noticias.....	68
Tabla 3.3: Prueba de Aceptación Registro de Evento.....	69
Tabla 3.4: Prueba de Aceptación Presentación de Eventos	70
Tabla 3.5: Prueba de Aceptación Registro de Documento.....	71
Tabla 3.6: Prueba de Aceptación Presentación de Documento	72
Tabla 3.7: Prueba de Aceptación Registro de Periódico	73
Tabla 3.8: Prueba de Aceptación Presentación de Periódicos.....	73
Tabla 3.9: Prueba de Aceptación Registro de Encuesta	74
Tabla 3.10: Prueba de Aceptación Responder Encuesta.....	75
Tabla 3.11: Prueba de Aceptación Presentación de Encuestas	76
Tabla 3.12: Prueba de Aceptación Comentar Evento	77
Tabla 3.13: Prueba de Aceptación Presentación de video en vivo.....	77
Tabla 3.14: Prueba de Aceptación Registro de usuarios	78
Tabla 3.15: Prueba de Aceptación Edición de la cuenta de usuario propia	79
Tabla 3.16: Prueba de Aceptación Registro de la directiva de la ADEPON	80
Tabla 3.17: Prueba de Aceptación Presentación de la información de la directiva de la ADEPON.....	81
Tabla 3.18: Prueba de Aceptación Edición de la misión y visión de la ADEPON.....	82
Tabla 3.19: Prueba de Aceptación Presentación de la misión y visión de la ADEPON.....	83

INTRODUCCIÓN

Es muy común afirmar que estamos transitando hacia nueva época histórica, en la cual la comunicación y la información juegan un rol preponderante, en ésta, la información es una necesidad y al mismo tiempo una obligación de cada individuo e institución. Por ello, en una organización gremial con más de 300 miembros, como es el caso de la Asociación de Docentes de la Escuela Politécnica Nacional (ADEPON), el mantener una comunicación fluida, oportuna, clara y relevante con cada uno de sus miembros y con la comunidad universitaria en general, no sólo es una obligación sino también una necesidad impostergable.

Ante esta situación y a solicitud de la ADEPON, se desarrolló una plataforma de comunicación o sitio Web oficial para la ADEPON utilizando la metodología XP, tecnología Java para su programación y una base de datos relacional MYSQL para la gestión de la información.

Se utilizó la metodología XP debido a que posibilita el desarrollo de software basado en valores como de comunicación y retroalimentación. También provee un conjunto de prácticas probadas muy útiles como la planificación, entregas pequeñas, metáfora, diseño sencillo, pruebas y refactorización, así como programación en parejas que posibilita mejorar la productividad en la programación. Se utilizó la tecnología Java debido a que no implica costos por licencia asociados al desarrollo del sitio, posibilita un fácil mantenimiento del código, entendimiento del sistema para programar su crecimiento y ofrece facilidades para construir aplicaciones colaborativas multiusuario en internet. Se utilizó la base de datos MYSQL por su potencia y seguridad en el manejo de datos y un entorno de desarrollo NETBEANS que integra todas las herramientas necesarias para el desarrollo de la aplicación.

El sitio fue diseñado de acuerdo a los requerimientos de la ADEPON, para lo cual se trabajó con historias de usuario y básicamente maneja publicación y presentación de noticias, publicación y presentación de eventos, publicación y presentación de documentos, publicación y presentación de un periódico de la ADEPON, elaboración, aplicación y

procesamiento de encuestas, publicación de videos en vivo, registro de usuarios y administración de la información del sitio.

Debido a que la ADEPON no cuenta con la infraestructura necesaria para el funcionamiento del sitio, se contrató los servicios de un servidor virtual privado (VPS) sobre el cual se construyó el entorno de operación del proyecto. Sobre esta base, el sitio fue probado y entregado a la ADEPON en sesión ordinaria del viernes 16 de abril del presente año. Para determinar el grado de satisfacción de los usuarios se realizó una encuesta, en la que se aprecia la conformidad con los resultados.

La tesis está organizada en 4 capítulos. En el **Capítulo Uno** se presenta el planteamiento del problema, la descripción de la Asociación, la justificación de la Metodología de Desarrollo y la justificación de las herramientas a utilizar en el proceso de desarrollo.

En el **Capítulo Dos** se describe el diseño del sitio, las historias de usuarios que representan los requerimientos del sistema, la planificación de iteraciones y el plan de entrega, así como la metáfora del sistema. Se presenta además las tarjetas CRC, la estructura del sitio Web, y la arquitectura del sistema.

En el **Capítulo Tres** se explica la implementación del sitio web y se realiza una descripción más detallada de la implementación de cada una de las capas del modelo MVC, de las pruebas unitarias, de aceptación y de rendimiento que se realizaron al sistema, así como del proceso de implantación del sistema, para terminar explicando los resultados que se obtuvieron después de la implantación.

Por último, en el **Capítulo Cuatro** se presenta las conclusiones y recomendaciones para realizar un trabajo más eficiente en la construcción de herramientas para promover procesos de comunicación e información más eficaces en instituciones como ADEPON.

CAPITULO I

1 DEFINICIÓN DEL PROBLEMA

Cuando una asociación llega a contar con una amplia cantidad de miembros, siente la necesidad de disponer de un medio de comunicación de amplia difusión y razonable costo, que facilite la gestión de la información, y que sea amigable, de fácil navegación y fácil manejo, además de que garantice la seguridad y privacidad de los datos que se manejan.

Este capítulo contiene la descripción de la situación actual de la ADEPON, la justificación de la metodología y de las herramientas a utilizar en el proyecto.

1.1 DESCRIPCIÓN DE LA ASOCIACIÓN DE PROFESORES DE LA ESCUELA POLITÉCNICA NACIONAL

La Asociación de Profesores de la Escuela Politécnica Nacional (ADEPON), se encuentra conformada por: “los profesores titulares de la Escuela Politécnica Nacional, con nombramiento definitivo o accidental, que libre y voluntariamente se asocien a la ADEPON; los profesionales que desarrollen actividades académicas vinculadas a la docencia, investigación y extensión en la Escuela Politécnica Nacional, con nombramiento definitivo o accidental que libre y voluntariamente se asocien a la ADEPON; los profesores y profesionales que luego de haberse acogido al derecho de jubilación en la Escuela Politécnica Nacional, manifiesten su deseo de continuar asociados a la ADEPON; los profesores honorarios de la Escuela Politécnica Nacional que deseen pertenecer a la Asociación. Los docentes que no deseen pertenecer a la ADEPON pueden manifestar su voluntad por escrito para desafiliarse”. [1]

La ADEPON se rige por un estatuto que norma su funcionamiento, en el cual se detalla que: “La ADEPON está dirigida por un presidente o presidenta, un vicepresidente o vicepresidenta, un tesorero o tesorera y un secretario o secretaria, quienes conforman la Directiva de la ADEPON. La Directiva junto a ocho vocales conforman el Directorio de la asociación; cuya sede se encuentra en el cuarto piso del Edificio de la Facultad de Ciencias de la EPN. El Directorio se reunirá ordinariamente una vez al mes y extraordinariamente por

disposición del Presidente o por solicitud escrita de por lo menos la tercera parte de sus miembros”. [1]

El Directorio tiene entre sus deberes y atribuciones:

- Aprobar y reformar los Reglamentos de la ADEPON.
- Considerar y resolver los asuntos planteados a la ADEPON.
- Aprobar el plan anual de actividades de la ADEPON y vigilar su cumplimiento.
- Designar las comisiones que fueren necesarias para cumplir con las finalidades de la ADEPON.
- Aprobar el presupuesto anual de la ADEPON y sus reformas.
- Designar delegados ante los diferentes organismos con los que la ADEPON mantenga relaciones.
- Atender y resolver sobre las solicitudes de afiliación de nuevos miembros.

“Las elecciones de la Directiva y de los Vocales del Directorio de la ADEPON se realizarán cada dos años, previa convocatoria por escrito y con por lo menos treinta días de anticipación, a la fecha de terminación del respectivo periodo”. [1]

“La ADEPON tiene como obligaciones, defender los derechos y los intereses profesionales de sus asociados, defender la dignidad profesional de los profesores en el desarrollo de sus actividades docentes e investigativas, defender la esencia académica del profesor y el desarrollo de la actividad académica universitaria basados en la planificación, evaluación y autonomía, y velar por el bienestar social de sus agremiados activos y jubilados”. [1]

En la actualidad, la ADEPON para difundir las actividades que lleva a cabo, cuenta desde enero de 2001 con una lista de correo electrónico en Yahoo groups, que permite la distribución de mensajes entre múltiples usuarios de Internet de forma simultánea. Hace aproximadamente 4 años se tiene una lista de correo electrónico alojada en un servidor del Centro de Cómputo de la Escuela Politécnica Nacional; la dirección de correo de dicha lista es adepon-oficial@epn.edu.ec. Todo socio puede dirigirse a ella para comunicarse con los demás socios y a su vez el directorio y la Sra. secretaria de la ADEPON se comunican con

los socios por este mismo medio para informar de las actividades realizadas.

A pesar de que el sitio web oficial de la EPN determina el URL: <http://www.adepon.net/>, como el Blog oficial de la ADEPON, no existe un Blog oficial que sea utilizado en la actualidad por la directiva de la ADEPON. También cabe resaltar que se pueden encontrar alojados en Internet Blogs de listas que fueron candidatas a dirigir la asociación, incluyendo un Blog de la lista que ganó las últimas elecciones.

El actual directorio, encabezada por su presidente Ing. Jaime Calderón, ha visto la necesidad de contar con un sitio web oficial para la ADEPON, un sitio moderno y sencillo, el cual sea una plataforma de comunicación y colaboración entre los integrantes de la asociación. Para la tarea, la ADEPON decide confiar en estudiantes Politécnicos, segura de alcanzar los objetivos planteados. A partir de esta visión se procede a realizar el desarrollo del sitio web oficial de la ADEPON.

1.2 JUSTIFICACIÓN DE LA METODOLOGÍA DE DESARROLLO.

1.2.1 CARACTERÍSTICAS DE LA METODOLOGÍA XP

XP es un estilo de desarrollo de software centrado en la excelente aplicación de técnicas de programación, comunicación clara y trabajo en equipo. XP incluye:

- Una filosofía de desarrollo de software basado en los valores de la comunicación, la retroalimentación, la sencillez, la valentía y el respeto.
- Un conjunto de prácticas probadas útiles para mejorar el desarrollo de software. Las cuales son: Planificación, Entregas pequeñas, Metáfora, Diseño sencillo, Pruebas, Refactorización, Programación en parejas, Propiedad colectiva, Integración continua, Máximo 40 horas semanales, Trabajar con el cliente, Estándares de codificación.
- Una comunidad que comparte estos valores y muchas de las mismas prácticas.

Se distingue de otras metodologías a través de:

- Sus cortos ciclos de desarrollo, dando lugar a retroalimentación temprana, concreta y

permanente.

- Su enfoque de planificación incremental, que provee un plan general al inicio del proyecto, el cual se espera que evolucione a través del ciclo de vida del proyecto.
- Su capacidad de programar en forma flexible las funcionalidades requeridas, respondiendo a las necesidades cambiantes del negocio.
- Su dependencia en la comunicación oral, pruebas, y código fuente para comunicar la estructura del sistema y su intención.
- Su dependencia en un proceso de diseño evolutivo que dura lo que dura el sistema.
- Su dependencia en la estrecha colaboración entre los interesados que participan activamente.

1.2.2 LA METODOLOGÍA XP EN EL DESARROLLO WEB

Aunque XP inicialmente estuvo pensado para cualquier tipo de desarrollo (sobre todo Java), realmente encaja a la perfección con un proceso de desarrollo Web.

La metodología XP es una de las muchas metodologías ágiles que existen en la actualidad, Frank Maurer y Sebastien Martel en su artículo “Extreme Programming Rapid Development for Web-Based Applications” escriben respecto al uso de la metodología XP en desarrollo Web “XP es una buena opción para muchos proyectos de software basados en Internet, que a menudo retrasan los esfuerzos de documentación, debido a las limitaciones de tiempo de salida al mercado. En XP, todos los desarrolladores trabajan en estrecha colaboración para que puedan comunicarse de manera informal en lugar de gastar tiempo en la documentación y en la toma de decisiones. Mientras los equipos siguen siendo pequeños, este enfoque vale la pena: Es más rápido hablar directamente que realizar anotaciones del desarrollo. Además, la comunicación directa se limita típicamente a los problemas existentes. Para producir la documentación, los escritores a menudo tienen que hacer suposiciones acerca de qué información será de utilidad para los lectores. Si estas suposiciones son incorrectas o si los cambios en el diseño cambian de manera drástica, el esfuerzo de documentación se desperdicia.” [2]

El artículo además presenta un estudio realizado por 16 meses en una empresa pequeña con 9 programadores a tiempo completo, desarrollando un sistema Web, en el cual se

compara la productividad de los trabajadores antes de utilizar XP con la productividad alcanzada utilizando XP. Esta información se muestra en la Tabla 1.1.

	NLC/esfuerzo	#métodos/esfuerzo	#clases/esfuerzo
Promedio Pre-XP	10.2	0.36	0.05
Promedio XP	17.0	1.45	0.21
Porcentaje de Ganancia	66.3	302.1	281.6

Tabla 1.1 Ganancias en productividad con XP.

Los indicadores medidos fueron:

- NLC: número de líneas nuevas de código (código fuente de Java y código HTML)
- #métodos: número de nuevos métodos
- # clases: número de nuevas clases.
- Se midió el esfuerzo como el número de horas facturado al cliente.

Los resultados muestran ganancias de productividad en los programadores de entre 66,3 por ciento y 302,1 por ciento sobre las métricas definidas.

1.2.3 COMPARACIÓN ENTRE LAS CARACTERÍSTICAS DE LA METODOLOGÍA XP Y LAS DEL PRESENTE PROYECTO

Para el presente proyecto es fundamental obtener resultados rápidos, por lo cual la productividad del equipo de trabajo es fundamental, la cual, como se aprecia en el estudio antes descrito, se ve favorecida con el uso de XP en proyectos Web; esto, junto con el número reducido de miembros del equipo de trabajo y otras características que se muestran en la Tabla 1.2 motivaron la selección de esta metodología de desarrollo.

Características Requeridas	Características de XP
Los requerimientos tienen alta probabilidad de variar en el tiempo.	XP maneja fácilmente los cambios de requerimientos gracias al “refactoring” y al uso de pruebas unitarias de manera constante, lo que permite que los cambios en el código no introduzcan errores.
Los miembros del directorio de la ADEPON, requieren realizar reuniones continuas para apreciar los avances del proyecto.	XP promueve la comunicación entre las partes, al estar el cliente integrado en el proyecto, aprecia los cambios después de cada iteración y su opinión se conoce en tiempo real.
Para que un sitio web tenga el éxito deseado, se debe lograr un diseño simple, fundamentalmente de interfaces de usuario.	La simplicidad es la base de la programación extrema. XP propone realizar un diseño simple que agilite el desarrollo y facilite el mantenimiento
Experiencia del Equipo de Trabajo	El equipo de desarrollo tiene experiencia, académica y empresarial, con el uso de la metodología XP, lo cual reduce los tiempos de desarrollo.

Tabla 1.2 Características del proyecto y de la metodología XP.

1.3 JUSTIFICACIÓN DE LAS HERRAMIENTAS A UTILIZAR.

1.3.1 JAVA

La tecnología Java se refiere tanto al lenguaje de programación como a la plataforma o entorno de computación originarios de Sun Microsystems y posteriormente adquirido por Oracle Corporation.

La tabla 1.3 muestra las características requeridas para la implementación del sitio web de la ADEPON y las características propias de la tecnología JAVA que motivaron su selección.

Características Requeridas	Características de JAVA
La ADEPON requiere los más bajos costos de producción posibles, sin que se afecte la calidad del sitio.	Java posee una Licencia Pública General de GNU, lo que implica que no existen costos asociados a licencias por el desarrollo en Java.
El código debe ser fácilmente mantenible para que concluido el desarrollo, el mantenimiento pueda ser realizado por cualquier otro equipo de trabajo.	La tecnología Java usada en cualquier portal web es fácilmente mantenible al poseer una estructura claramente diferenciada y entendible, pudiéndose identificar con facilidad el modelo MVC.
La ADEPON busca ofrecer, en un futuro, varios servicios a través de su portal.	Java es fácilmente mantenible y modular, lo que aporta a la escalabilidad. El hecho de que JAVA sea una técnica estructurada facilita el entendimiento del flujo del sistema y así también el crecimiento del mismo.

Experiencia del Equipo de Trabajo	El equipo de desarrollo tiene experiencia, académica y empresarial, con el uso de JAVA, lo cual reduce los tiempos de desarrollo.

Tabla 1.3 Características del proyecto y de JAVA.

Se puede concluir que la tecnología Java es una buena solución para el desarrollo del sitio web para la ADEPON, ya que además de tener Licencia Pública General de GNU (por lo que su uso comercial es libre y gratuito) es una de las tecnologías más maduras en el mercado, brindando alta seguridad, estabilidad, mantenibilidad, escalabilidad y modularidad.

1.3.2 PLATAFORMA DE PROGRAMACIÓN JEE6

JEE6 (Java Enterprise Edition 6) es el estándar actual para el desarrollo de aplicaciones empresariales, mantenido y desarrollado por ORACLE. Dentro de JEE la tecnología JavaServer Faces (JSF) es el marco de trabajo estándar para aplicaciones web. JSF está basado en el patrón MVC, ofrece una clara separación entre el comportamiento y la presentación de las aplicaciones web y presta varias bibliotecas de etiquetas complementarias a las etiquetas HTML.

La tabla 1.4 muestra las características de JEE y JSF que motivaron su uso.

Características Requeridas	Características de JSF
Se requiere una herramienta confiable, estable y que brinde soporte al momento del desarrollo.	JEE6 se desarrolla a través del Java Community Process (JCP) [3], es el estándar informal para el desarrollo de aplicaciones web de ORACLE, una de

	las mayores compañías de software del mundo según la firma PricewaterhouseCoopers (PwC) [4].
Se requiere una herramienta que agilite el proceso de desarrollo, de tal forma que permita entregar resultados rápidos.	JSF tiene como fin, simplificar el desarrollo de interfaces web en las aplicaciones JEE, lo que facilita el desarrollo.
Multiplataforma, el sitio debe ser soportado por diferentes dispositivos como teléfonos celulares, agendas, etc.,	JSF no es una tecnología que soporta solamente clientes web browser, como lo es Struts[5].
Experiencia del Equipo de Trabajo	El equipo de desarrollo tiene experiencia, académica y empresarial, con el uso de JEE y JSF, lo cual reduce los tiempos de desarrollo.

Tabla 1.4 Características del proyecto y de JSF.

1.3.3 MARCO DE TRABAJO ICEFACES

ICEfaces es un marco de trabajo de código abierto basado en JSF, para el desarrollo de Aplicaciones de Internet Enriquecidas (RIA - Rich Internet Application) utilizando el lenguaje JAVA.

El uso de ICEfaces aporta al proyecto tres factores claves con respecto a JSF:

- Componentes enriquecidos propios de ICEfaces los cuales poseen mayor funcionalidad que los propios de JSF y son mucho más agradable para el usuario.
- Automatic Ajax el cual es un mecanismo que calcula el número mínimo y necesario de actualizaciones de la página en un ciclo de vida JSF al siguiente, eliminando la necesidad que presenta JSF de etiquetas <f:ajax>.

- Ajax Push el cual permite que la aplicación actualice cualquier parte de la página en cualquier momento y por cualquier grupo de clientes. Este es un cambio pequeño en términos de las capacidades técnicas de las aplicaciones Ajax, pero es un cambio revolucionario en términos de lo que realmente puede ofrecer para desarrollar aplicaciones para los usuarios. Ajax Push es la clave para construir aplicaciones multi-usuario colaborativas en la web. [6]

1.3.4 GESTOR DE BASE DE DATOS RELACIONAL MYSQL

MySQL ofrece al sitio web una alta escalabilidad. MySQL ha sido probado con bases de datos que contienen 50 millones de registros. También se conoce de usuarios que usan MySQL con 200.000 tablas y cerca de 5.000.000.000 de filas [7].

Además MySQL brinda seguridad al sitio web al hacer uso de un sistema de privilegios y contraseñas flexible y seguro, que permite verificación basada en el host. Todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor [8].

Además MySQL fue seleccionado como gestor de base de datos por poseer una Licencia Pública General de GNU, lo que implica que no existen costos asociados a licencias por el desarrollo con MySQL.

Como herramienta visual para el diseño de la base de datos, la administración de la base de datos, y el mantenimiento de la base de datos se utilizara MySQL Workbench.

Esta herramienta visual para arquitectos de bases de datos, desarrolladores y Administradores de bases de datos, ofrece modelado de datos, desarrollo de SQL, herramientas de administración para la configuración de servidores y administración de usuarios, entre otros servicios. Está disponible en Windows, Linux y Mac OS.

1.3.5 ENTORNO DE DESARROLLO INTEGRADO NETBEANS

NetBeans es un entorno de desarrollo integrado o IDE (Integrated Development Environment) libre y de código abierto, escrito en Java, para desarrolladores de Software,

que integra todas las herramientas necesarias para crear aplicaciones web en, entre otras plataformas, Java.

Por las razones expuestas a lo largo de este capítulo se justifica el uso del marco de trabajo estándar para aplicaciones empresariales de Java (JEE) en su versión más reciente junto con las herramientas mencionadas.

CAPITULO II

2 PLANIFICACIÓN Y DISEÑO

2.1 HISTORIAS DE USUARIO

Cada historia de usuario es la representación de un requerimiento del usuario. Las historias de usuario son escritas por el usuario utilizando el lenguaje natural.

En la figura 2.1 se muestra el formato elegido para las historias de usuario, basado en el trabajo realizado por García Monge Ana Mercedes en su proyecto de titulación “Desarrollo e Implantación del Sistema de Administración Estudiantil vía web para la unidad educativa particular trilingüe HELMUT WALHMÜLLER” [8]

HISTORIA DE USUARIO	
Número:	Nombre:
Usuario:	
Número de Modificación de Historia:	Asignación Asignada:
Prioridad en Negocio:	Riesgo en desarrollo:
Descripción:	
Observaciones:	

Figura 2.1: Plantilla Historias de Usuario.

Donde:

Número: Representa el número secuencial de la historia de usuario.

Nombre: Es el proceso que realiza la historia de usuario.

Usuario: Es el responsable de realizar el proceso de la historia de usuario.

Número de Modificación de Historia: Nos indica la secuencia de cuantas veces se ha modificado la historia.

Descripción: Es el detalle de tareas a realizarse en la historia de Usuario.

El Ing. Jaime Calderón, Presidente de la ADEPON, fue el responsable de escribir la mayor parte de las historias de usuario; las tablas 2.2 a 2.16 describen a cada una de ellas.

HISTORIA DE USUARIO	
Número: 1	Nombre: Publicación de noticias
Usuario: Administrador	
Número de Modificación de Historia: 3	Asignación Asignada:
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Descripción: Se podrán agregar, modificar y eliminar noticia, cuyos registros incluirán: la información de la noticia: fecha, contenido, imagen, etc. Las noticias deben ser publicadas tanto en Twitter como en Facebook	
Observaciones:	

Tabla 2.1 : Historia de Usuario Publicación de noticias.

HISTORIA DE USUARIO	
Número: 2	Nombre: Presentación de noticias
Usuario: Usuario	
Número de Modificación de Historia: 3	Asignación Asignada:
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Descripción: Se muestra un resumen de la noticia en la sección correspondiente dentro del sitio web, ordenadas por fecha de publicación desde el registro más reciente al más antiguo. Desde la sección noticias se podrá ver el detalle de una noticia seleccionada y dejar comentarios si se trata de un usuario registrado.	
Observaciones:	

Tabla 2.2 : Historia de Usuario Presentación de noticias.

HISTORIA DE USUARIO	
Número: 3	Nombre: Publicación de Eventos
Usuario: Administrador	
Número de Modificación de Historia: 4	Asignación Asignada:
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Descripción: Se podrán agregar, modificar y eliminar eventos, cuyos registros constarán de: la información del evento fecha, lugar, imagen, descripción, fecha de publicación, etc. Los eventos deben ser publicados tanto en Twitter como en Facebook	
Observaciones:	

Tabla 2.3: Historia de Usuario Publicación de Eventos.

HISTORIA DE USUARIO	
Número: 4	Nombre: Presentación Eventos
Usuario: Usuario	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alta
Descripción: Se muestra los eventos ingresados por el administrador dentro del sitio web, ordenados por fecha de realización del evento desde el más próximo al más lejano. Desde la sección eventos se podrá ver el detalle de un evento seleccionado y dejar comentarios si se trata de un usuario registrado.	
Observaciones:	

Tabla 2.4: Historia de Usuario Presentación Eventos.

HISTORIA DE USUARIO	
Número: 5	Nombre: Publicación de Reglamentos y Resoluciones
Usuario: Administrador	
Número de Modificación de Historia: 3	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Descripción: Se podrán agregar, modificar, eliminar registros de los documentos correspondientes a las resoluciones y reglamentos de la ADEPON que contendrán la descripción del mismo, detalle, fecha de resolución, fecha de publicación. El administrador debe ser capaz de subir archivos con el contenido del reglamento.	
Observaciones:	

Tabla 2.5: Historia de Usuario Publicación de Reglamentos y Resoluciones.

HISTORIA DE USUARIO	
Número: 6	Nombre: Presentación de Reglamentos y Resoluciones
Usuario: Usuario	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Descripción: Se muestran a los usuarios los reglamentos y las resoluciones ordenadas por fecha de publicación desde el registro más reciente al más antiguo. Los usuarios deben ser capaces de descargar los archivos correspondientes a los reglamentos	
Observaciones:	

Tabla 2.6: Historia de Usuario Presentación de Reglamentos y Resoluciones.

HISTORIA DE USUARIO	
Número: 7	Nombre: Publicación Periódico
Usuario: Administrador	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Descripción: El sitio debe permitir al administrador agregar, eliminar y modificar registros del periódico. El administrador debe ser capaz de subir archivos con el contenido de los periódicos.	
Observaciones:	

Tabla 2.7: Historia de Usuario Publicación Periódico.

HISTORIA DE USUARIO	
Número: 8	Nombre: Presentación Periódico
Usuario: Usuario	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Descripción: El sitio contará con un archivo de los periódicos publicados por la ADEPON que será accesible para cualquier usuario. Se organizarán los registros por antigüedad. Los usuarios deben ser capaces de descargar los archivos correspondientes a los periódicos.	
Observaciones:	

Tabla 2.8: Historia de Usuario Presentación Periódico.

HISTORIA DE USUARIO	
Número: 9	Nombre: Publicación de encuestas
Usuario: Administrador	
Número de Modificación de Historia: 4	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Alta
Descripción: Se podrán agregar encuestas en el sitio, las preguntas serán definidas por el Administrador.	
Observaciones:	

Tabla 2.9: Historia de Usuario Publicación de encuestas.

HISTORIA DE USUARIO	
Número: 10	Nombre: Responder encuestas
Usuario: Usuario	
Número de Modificación de Historia: 3	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Alta
Descripción: Los usuarios registrados en el sitio podrán responde la encuestas, cada uno una única vez.	
Observaciones:	

Tabla 2.10: Historia de Usuario Responder encuestas.

HISTORIA DE USUARIO	
Número: 11	Nombre: Resultados de Encuestas
Usuario: Usuario	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Alta
Descripción: Todos los usuarios del sitio deben ser capaces de observar los resultados de las encuestas actuales y pasadas, sin necesidad de estar registrados	
Observaciones:	

Tabla 2.11: Historia de Usuario Resultados de Encuestas.

HISTORIA DE USUARIO	
Número: 12	Nombre: video en vivo
Usuario: Usuario	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Baja	Riesgo en desarrollo: Media
Descripción: El sitio permitirá a los usuarios visualizar vídeo en vivo.	
Observaciones:	

Tabla 2.12: Historia de Usuario Visualizar video en vivo.

HISTORIA DE USUARIO	
Número: 13	Nombre: Registro de Usuarios
Usuario: Administrador	
Número de Modificación de Historia: 5	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Descripción: El administrador del sitio podrá registrar nuevos usuarios miembros de la ADEPON, a lo cuales se les deberá enviar sus respectivos passwords y nombre de usuario a sus correos de la EPN	
Observaciones:	

Tabla 2.13: Historia de Usuario Registro de Usuarios.

HISTORIA DE USUARIO	
Número: 14	Nombre: Edición Cuentas de usuarios
Usuario: Usuario	
Número de Modificación de Historia: 6	Asignación Asignada:
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Descripción: Los usuarios registrados podrán cambiar sus contraseñas y nombre de usuario y agregar un correo, diferente al de la EPN, para recibir notificaciones, el mismo que podrá ser cambiado en el futuro	
Observaciones:	

Tabla 2.14: Historia de Usuario Edición Cuentas de usuarios.

HISTORIA DE USUARIO	
Número: 15	Nombre: Administración de Información sobre la ADEPON
Usuario: Administrador	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Baja	Riesgo en desarrollo: Baja
Descripción: El administrador del sistema será capaz de ingresar al sitio la siguiente información referente a la ADEPON: Misión, Visión, Autoridades	
Observaciones:	

Tabla 2.15 : Historia de Usuario Administración de Información sobre la ADEPON.

HISTORIA DE USUARIO	
Número: 16	Nombre: Presentación de Información sobre la ADEPON
Usuario: Usuario	
Número de Modificación de Historia: 1	Asignación Asignada:
Prioridad en Negocio: Baja	Riesgo en desarrollo: Baja
Descripción: Los usuarios del sitio podrán ver la siguiente información referente a la ADEPON: Misión, Visión, Autoridades	
Observaciones:	

Tabla 2.16: Historia de Usuario Presentación de Información sobre la ADEPON.

Obtenidas las historias de usuario, se presentan los módulos del sistema:

1. **Noticias**
2. **Eventos**
3. **Documentos (Reglamentos, Resoluciones, Estatuto)**
4. **Periódicos**
5. **Encuestas**
6. **Video en Vivo**
7. **Cuentas de Usuario**
8. **Información sobre la ADEPON(Misión, Visión, Directiva)**

2.1.1 VALORACIÓN DE HISTORIAS DE USUARIO

Dentro de la planificación de la entrega se tiene en cuenta la valoración de las historias de usuario, donde se especifica un tiempo estimado para la elaboración de cada historia. Debido a que ambos miembros del equipo trabajan a horario completo en las empresas Tandicorp y Montran respectivamente, se establece el día de trabajo en 3 horas y la semana

de trabajo en 5 días.

La tabla 2.17 presenta una estimación de las historias de usuario

Módulo	Historias de Usuario	N	Tiempo Estimado		
			Semanas	Días	Horas
Noticias	Publicación de noticias	1	2	10	30
	Presentación Noticias	2	1	5	15
Eventos	Publicación de Eventos	3	2	10	30
	Presentación Eventos	4	1	5	15
Documentos	Publicación de Documentos	5	1	5	15
	Presentación de Documentos	6	1	5	15
Periódicos	Publicación Periódico	7	1	5	15
	Presentación Periódico	8	1	5	15
Encuestas	Responder Encuesta	9	2	10	30
	Publicación de Encuesta	10	1	5	15
	Resultados de Encuestas	11	2	10	30
Video en Vivo	Publicación video en vivo	12	6	30	90
Cuentas de Usuario	Registro de Usuarios	13	1	5	15
	Edición Cuentas de usuarios	14	1	5	15
ADEPON	Administración de Información sobre la ADEPON	15	1	5	15
	Presentación de Información sobre la ADEPON	16	1	5	15

Tabla 2.17 : Estimación de Historias de Usuario.

2.2 PLAN DE ENTREGA

La tabla 2.18 muestra un estimado del tiempo requerido para el desarrollo de cada una de las historias de usuario, así como el número de iteración en la que se llevara a cabo el desarrollo y número de entrega en la que se la presentara al usuario, según las prioridades establecidas por la ADEPON en las historias de usuario.

Módulo	N	Historias de Usuario	CALENDARIO ESTIMADO			ITERACIÓN ASIGNADA				ENTREGA ASIGNADA		
			SEMANAS	DÍAS	HORAS	1	2	3	4	1	2	3
Noticias	1	Publicación de noticias	2	10	30	x				x		
	2	Presentación Noticias	1	5	15	x				x		
Eventos	3	Publicación de Eventos	2	10	30	x				x		
	4	Presentación Eventos	1	5	15	x				x		
Documentos	5	Publicación de Documentos	1	5	15		x			x		
	6	Presentación de Documentos	1	5	15		x			x		
Periódicos	7	Publicación Periódico	1	5	15		x			x		
	8	Presentación Periódico	1	5	15		x			x		
Encuestas	9	Responder Encuesta	2	10	30			x			x	
	10	Publicación de Encuesta	1	5	15			x			x	
	11	Resultados de Encuestas	2	10	30			x			x	
Video en Vivo	12	Publicación video en vivo	6	30	90				x			x
Cuentas de Usuario	13	Registro de Usuarios	1	5	15			x			x	
	14	Edición Cuentas de usuarios	1	5	15			x			x	
ADEPON	15	Administración de Información sobre la ADEPON	1	5	15				x			x
	16	Presentación de Información sobre la ADEPON	1	5	15				x			x

Tabla 2.18: Plan de Entregas.

2.2.1 CALENDARIO DEL PLAN DE ENTREGA

La tabla 2.19 presenta el plan de entrega definido para el proyecto.

Historias de Usuario	Fecha de Inicio	Fecha de Finalización
Publicación de noticias	20/08/2012	31/08/2012
Presentación Noticias	3/09/2012	7/09/2012
Publicación de Eventos	10/09/2012	21/09/2012
Presentación Eventos	24/09/2012	28/09/2012
Publicación de Documentos	01/10/2012	05/10/2012
Presentación de Documentos	08/10/2012	12/10/2012
Publicación Periódico	15/10/2012	19/10/2012
Presentación Periódico	22/10/2012	26/10/2012
Responder Encuesta	29/10/2012	9/11/2012
Publicación de Encuesta	12/11/2012	16/11/2012
Resultados de Encuestas	19/11/2012	30/11/2012
Publicación video en vivo	03/12/2012	11/01/2013
Registro de Usuarios	14/01/2013	18/01/2013
Edición Cuentas de usuarios	21/01/2013	25/01/2013
Administración de Información sobre la ADEPON	28/01/2013	01/02/2013
Presentación de Información sobre la ADEPON	04/02/2013	08/02/2013

Tabla 2.19: Calendario del Plan de Entregas.

2.3 METÁFORA DEL SISTEMA

La metáfora del sistema tiene como fin explicar el funcionamiento del sistema de una manera clara, de tal forma que cualquier persona interesada la comprenda. Para escribirla se usa el lenguaje natural, tanto como sea posible.

Se ha dividido la metáfora del proyecto en dos partes: la administración del sitio web y la navegación en el sitio web las cuales son descritas a continuación:

2.3.1 Administración del Sitio Web

Para acceder a la administración del sitio web se ingresa a la dirección: www.adepon.org/WEBADEPON-war/admin.jsf donde el usuario deberá registrarse como usuario administrador tras lo cual será capaz de añadir, editar, consultar y eliminar los datos de: eventos, noticias, periódicos, reglamentos, resoluciones, estatuto, encuestas, videos, usuarios registrados en el sitio web, comentarios realizados por los usuarios sobre las noticias o los eventos, datos de la directiva de la ADEPON y la misión y la visión de la ADEPON.

- Manejo de los datos de los eventos.

Aquí se realiza el ingreso, edición, consulta o eliminación de los datos de un evento. Para ingresar un nuevo evento el administrador ingresa la siguiente información: lugar, fecha y hora, contenido del evento, imagen y/o video, los usuarios deben escoger si desean publicar el evento en Twitter y Facebook y si desean enviar la información del evento a los correos electrónicos de los usuarios registrados.

El administrador puede consultar eventos por: lugar, fecha y hora, y contenido del evento.

- Manejo de los datos de las noticias.

Aquí se realiza el ingreso, edición, consulta o eliminación de los datos de una noticia. Para ingresar una nueva noticia el administrador ingresa la siguiente información: título, resumen, contenido de la noticia, imagen y/o video. Los usuarios deben escoger si desean publicar la noticia en Twitter y Facebook y si desean enviar la información de la noticia a los correos electrónicos de los usuarios registrados.

El administrador puede consultar noticias por: título, resumen y contenido de la noticia.

- Manejo de los datos de los periódicos.

Aquí se realiza el ingreso, consulta o eliminación de los datos de un periódico. Se ingresa la siguiente información del periódico: título, descripción, fecha de publicación, archivo pdf del periódico.

El administrador puede consultar periódicos por: título, descripción y fecha de publicación.

- Manejo de los datos de los reglamentos, resoluciones y estatuto.
Aquí se realiza el ingreso, consulta o eliminación de los datos de los reglamentos, resoluciones y estatuto. Se ingresa la siguiente información del documento: título, descripción, fecha de publicación, archivo pdf del reglamento, resolución o estatuto. Se debe especificar si el documento es un reglamento, resolución o estatuto.
El administrador puede consultar reglamentos, resoluciones y estatuto por: título, descripción y fecha de publicación.
- Manejo de los datos de las encuestas.
Aquí se realiza el ingreso, edición, consulta o eliminación de los datos de una encuesta. Se ingresa la siguiente información de la encuesta: Título de la encuesta, preguntas y opciones de respuesta. Los usuarios son capaces de deshabilitar una encuesta para que los usuarios registrados no puedan seguir respondiéndola.
El administrador puede consultar reglamentos, resoluciones o el estatuto por: título, descripción y fecha de publicación.
- Manejo de los datos de los videos.
Aquí se realiza el ingreso, edición, consulta o eliminación de videos, los videos pueden ser subidos directamente desde el sitio a Youtube o se puede especificar un video ya subido previamente a Youtube por otro usuario. Se ingresa la siguiente información del video: título, categoría, descripción y dos palabras claves (para ubicarlo en Youtube) El usuario debe especificar su clave de GMAIL para acceder a Youtube.
El administrador puede consultar videos por: título, descripción o fecha.
- Manejo de los datos de los usuarios registrados en el sitio web.
Aquí se realiza el ingreso, consulta o eliminación de los datos de un usuario registrado en el sitio web. Se ingresa la siguiente información de los usuarios: nombre, apellido, nombre de usuario, rol (Administrador, Usuario), correo de la EPN.
El administrador puede consultar usuarios por: nombre, nombre de usuario, rol o correo.
- Manejo de los datos de los comentarios realizados por los usuarios sobre las noticias o los eventos.
Aquí se realiza la consulta o eliminación de los comentarios realizados por los usuarios sobre las noticias o los eventos.

El administrador puede consultar usuarios por: contenido del comentario, apellido o nombre del usuario, nombre de usuario, fecha, evento o noticia.

- Manejo de los datos de los de la directiva de la ADEPON.

Aquí se realiza el ingreso de los datos de los de la directiva de la ADEPON. Se ingresa la siguiente información de los miembros de la directiva: cargo, título, nombre, apellido y fotografía.

- Manejo de la misión y la visión de la ADEPON.

Aquí se realiza la edición de la misión y la visión de la ADEPON.

2.3.2 Navegación en el Sitio Web

Para acceder al sitio web se ingresa a la dirección: www.adepon.org aquí el usuario puede observar información sobre: noticias, eventos, reglamentos, resoluciones, estatuto, periódicos, encuestas, la directiva de la ADEPON y la misión y la visión de la ADEPON. Los usuarios también pueden ver video en vivo. Además los usuarios registrados pueden comentar eventos y noticias; y responder a las encuestas activas.

2.4 TARJETAS CRC

Las tarjetas CRC contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores, permitiendo al programador observar el desarrollo orientado a objetos.

En la figura 2.2 se muestra el formato elegido para las tarjetas CRC, basado en el trabajo realizado por García Monge Ana Mercedes en su proyecto de titulación “Desarrollo e Implantación del Sistema de Administración Estudiantil vía web para la unidad educativa particular trilingüe HELMUT WALHMÜLLER” [8]

Nombre de la Clase	
Responsabilidades	Colaboradores
Atributos	
Operaciones	

Figura 2.2: Tarjeta CRC.

Donde:

Nombre: Es el nombre de la clase que se describe en la tarjeta CRC

Responsabilidades: Describe la funcionalidad de la clase

Atributos: Características de la clase que contiene la tarjeta CRC

Operaciones: Son las operaciones que va a realizar la clase

Colaboradores: Clases que colaboran con la clase descrita

2.4.1 TARJETAS CRC DEL SISTEMA

Las tablas 2.20 a 2.33 describen cada una de las tarjetas CRC del Sistema

ARCHIVO_PDF	
Responsabilidades	Colaboradores
Atributos idARCHIVO ARCHIVO_BIN FECHA_UPLOAD TAMANO NOMBRE	PERIODICO REGLAMENTO
Operaciones Consultar Ingresar	

Tabla 2.20: Tarjeta CRC ARCHIVO_PDF

COMENTARIO	
Responsabilidades	Colaboradores
Atributos IdCOMENTARIO USUARIO_idUSUARIO EVENTO_idEVENTO NOTICIA_idNOTICIA FECHA COMENTARIO Operadores Ingresar Borrar	NOTICIA EVENTO USUARIO

Tabla 2.21: Tarjeta CRC COMENTARIO

DIRECTIVA_ADEPON	
Responsabilidades	Colaboradores
Atributos IdDIRECTIVA CARGO TITULO NOMBRE APELLIDO OFICINA EMAIL INFORMACION_ADICIONAL IMÁGEN_idIMAGEN Operadores Actualizar Consultar	IMAGEN

Tabla 2.22: Tarjeta CRC DIRECTIVA_ADEPON

ENCUESTA	
Responsabilidades	Colaboradores
Atributos IdENCUESTA TITULO FECHA_CREACION FECHA_PUB DURACION HABILITADA Operadores Ingresar Consultar Borrar	PREGUNTA USUARIO

Tabla 2.23: Tarjeta CRC ENCUESTA

EVENTO	
Responsabilidades	Colaboradores
Atributos idEVENTO FECHA_PUB FECHA_REALIZACION HORA LUGAR DESCRIPCION TITULO IMAGEN_idIMAGEN VIDEO_idVIDEO Operadores Ingresar Actualizar Consultar Borrar	IMAGEN VIDEO

Tabla 2.24: Tarjeta CRC EVENTO

IMAGEN	
Responsabilidades	Colaboradores
Atributos idIMAGEN IMAGEN FECHA_UPLOAD NOMBRE Operadores Consultar Ingresar	DIRECTIVA_ADEPON EVENTO NOTICIA

Tabla 2.25: Tarjeta CRC IMAGEN

INFORMACION	
Responsabilidades	Colaboradores
Atributos idINFORMACION NOMBRE_CAMPO DETALLE TITULO Operadores Ingresar Actualizar Consultar Borrar	

Tabla 2.26: Tarjeta CRC INFORMACION

NOTICIA	
Responsabilidades	Colaboradores
Atributos idNOTICIA TITULO RESUMEN CONTENIDO FECHA IMAGEN_idIMAGEN VIDEO_idVIDEO Operadores Ingresar Actualizar Consultar Borrar	IMAGEN VIDEO

Tabla 2.27: Tarjeta CRC NOTICIA

OPCION	
Responsabilidades	Colaboradores
Atributos idOPCIONES PREGUNTA_idPREGUNTA TEXTO VECES_SELEC Operadores Ingresar Actualizar Consultar Borrar	PREGUNTA

Tabla 2.28: Tarjeta CRC OPCION

PERIODICO	
Responsabilidades	Colaboradores
Atributos idPERIODICO FECHA FECHA_PUB DESCRIPCION ARCHIVO_PDF_idARCHIVO_PDF Operadores Ingresar Consultar Borrar	ARCHIVO_PDF

Tabla 2.29: Tarjeta CRC PERIODICO

PREGUNTA	
Responsabilidades	Colaboradores
Atributos idPREGUNTA ENCUESTA_idENCUESTA CUERPO NOTAS SELECCIÓN_MULTIPLE Operadores Ingresar Consultar Borrar	ENCUESTA OPCION

Tabla 2.30: Tarjeta CRC PREGUNTA

REGLAMENTO	
Responsabilidades	Colaboradores
Atributos IdREGLAMENTO FECHA FECHA_PUB DESCRIPCION ARCHIVO_PDF_idARCHIVO_PDF Operadores Ingresar Consultar Borrar	ARCHIVO_PDF

Tabla 2.31: Tarjeta CRC REGLAMENTO

USUARIO	
Responsabilidades	Colaboradores
Atributos idUSUARIO NOMBRE_USUARIO PASSWORD NOMBRE_GRUPO MAIL MAILPOLI NOMBRES APELLIDOS Operadores Ingresar Acutalizar Consultar Borrar	COMENTARIO ENCUESTA

Tabla 2.32: Tarjeta CRC USUARIO

VIDEO	
Responsabilidades	Colaboradores
Atributos idVIDEO NOMBRE_VIDEO URL DESCRIPCION_VIDEO FECHA_UPLOAD Operadores Ingresar Consultar Borrar	NOTICIA EVENTO

Tabla 2.33: Tarjeta CRC VIDEO

2.4.2 ESTRUCTURA DEL SITIO WEB

La figura 2.3 muestra un mapa del sitio para el usuario Administrador, en ella se describe las opciones de menú presentadas y las operaciones que el usuario es capaz de realizar dentro de cada una de ellas.

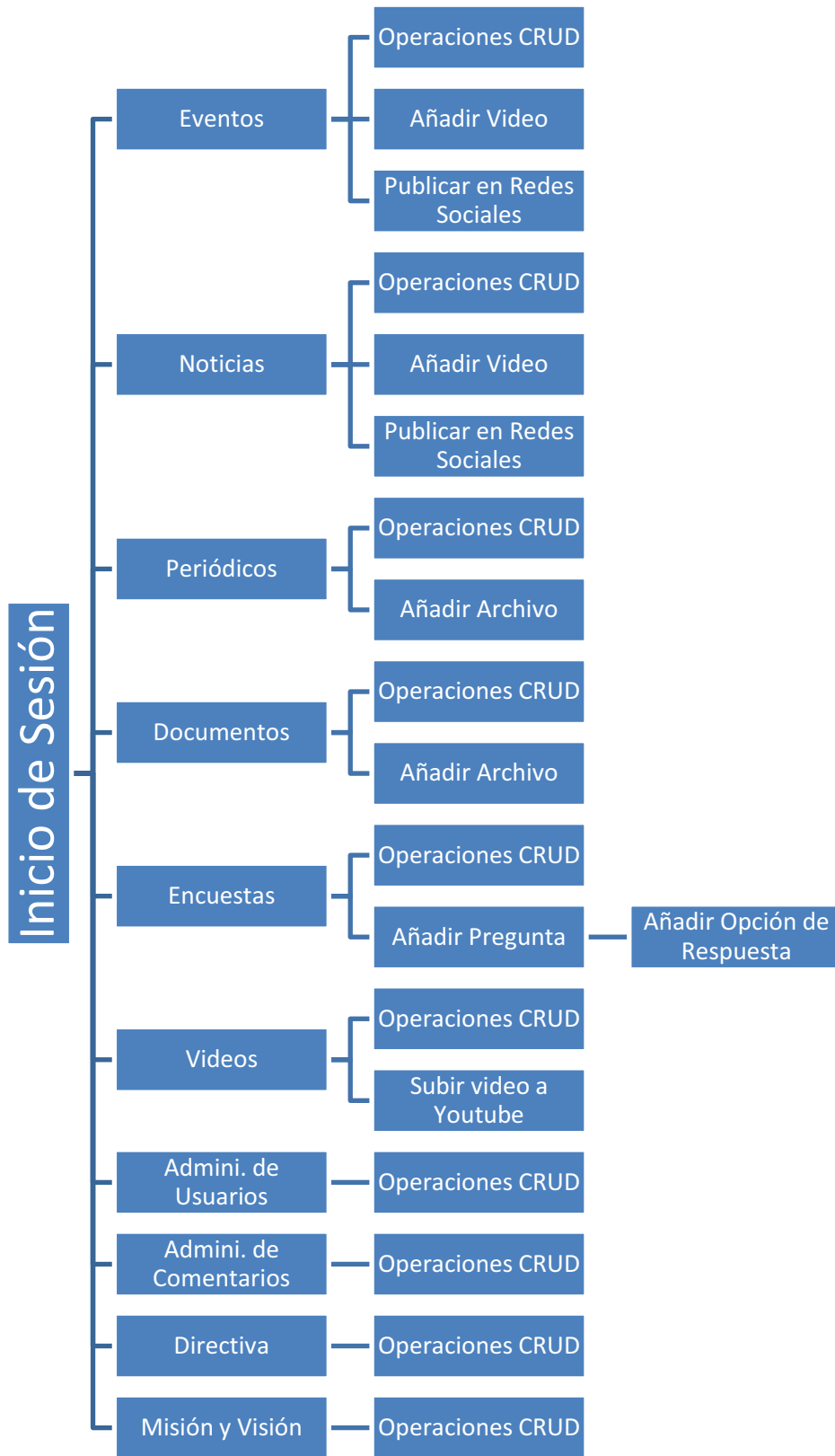


Figura 2.3: Mapa del Sitio para el usuario Administrador.

La figura 2.4 muestra un mapa del sitio para los usuarios registrados, en ella se describe las opciones de menú presentadas y las operaciones que el usuario es capaz de realizar dentro de cada una de ellas.

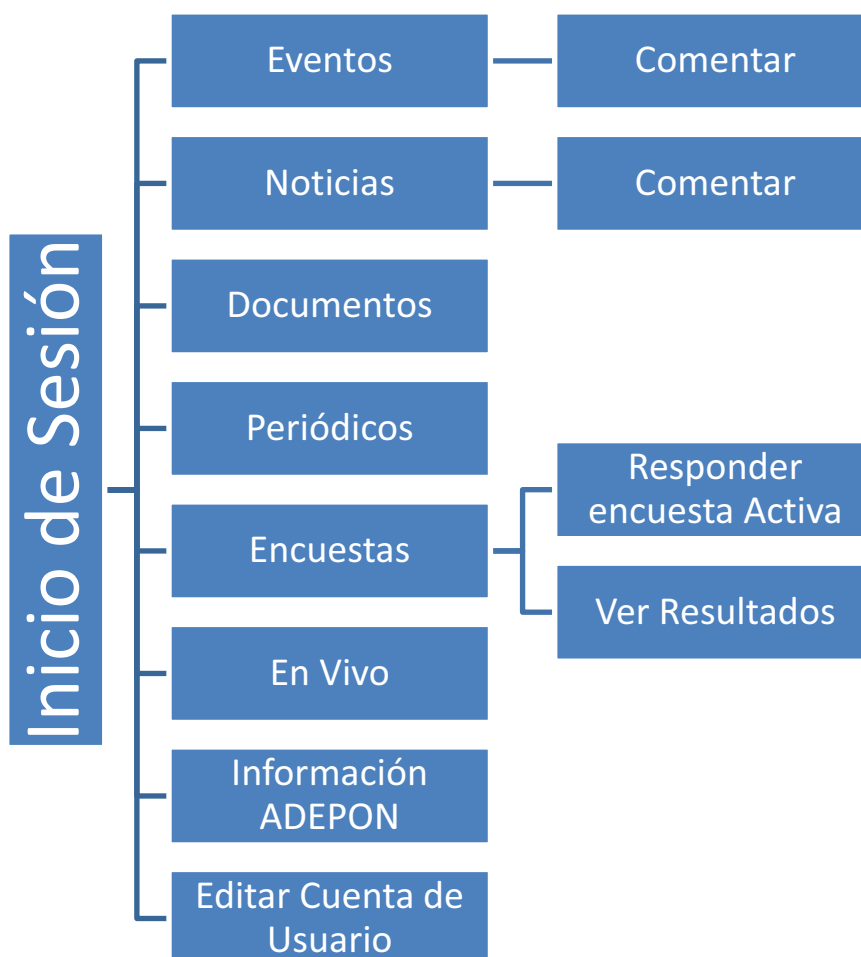


Figura 2.4: Mapa del Sitio para Usuarios Registrados.

La figura 2.5 muestra un mapa del sitio para los usuarios no registrados, en ella se describe las opciones de menú presentadas y las operaciones que el usuario es capaz de realizar dentro de cada una de ellas.



Figura 2.5: Mapa del Sitio para Usuarios No Registrados.

2.4.3 DISEÑO DE INTERFACES GRÁFICAS

Para las interfaces se utilizan archivos xhtml los cuales son el estándar para la tecnología JSF 2.0. Se utilizan 2 templates, uno para los administradores y otro para el sitio Web presentado a los usuarios. Cabe resaltar que la mayoría de los componentes utilizados son componentes enriquecidos propios de IceFaces o de JSF.

2.4.4 ESPECIFICACIÓN DE LA ARQUITECTURA DEL SISTEMA

Para el desarrollo del sistema se utiliza el patrón MVC. La figura 2.6 muestra la arquitectura del Sistema.

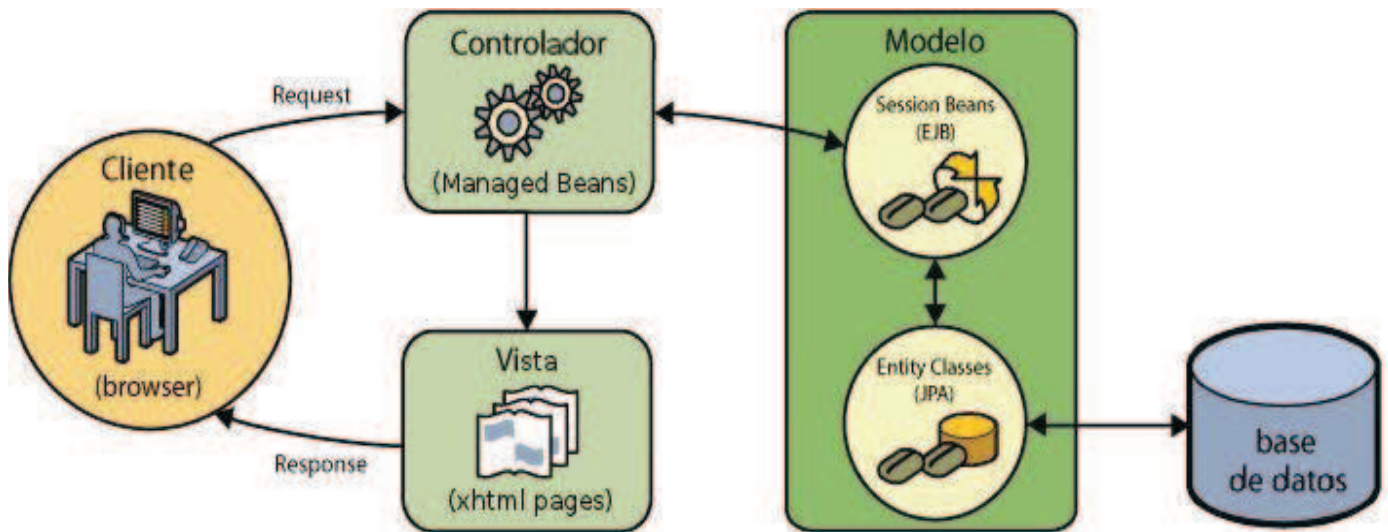


Figura 2.6: Arquitectura del Sistema.

Como se aprecia en la figura 2.6 el patrón MVC divide a la aplicación en 3 componentes interoperables, los cuales son:

Modelo: Representa a los datos del negocio. Permite al controlador acceder a sus datos y a su funcionalidad (operaciones CRUD), y notifica a la vista cuando surge un cambio.

Vista: La vista presenta la información al usuario. Obtiene los datos del modelo y especifica que datos deben ser presentados al usuario. En la vista se actualizan los datos cuando cambia el modelo. Además una vista permite al usuario hacer peticiones al controlador.

Controlador: El controlador define el comportamiento de la aplicación. Interpreta las entradas del usuario y establece las acciones que se deben realizar.

2.4.5 DISEÑO DE LA BASE DE DATOS

Una vez analizadas las Historias de Usuario y modelado el Sitio Web se procedió a realizar el diseño de la Base de Datos, proceso tras el cual se obtuvo el Modelo Entidad Relación presentado en la figura 2.7



Figura 2.7: Modelo Entidad Relación.

CAPITULO III

3 IMPLEMENTACIÓN, PRUEBAS, IMPLANTACIÓN Y ANÁLISIS DE RESULTADOS.

3.1 IMPLEMENTACIÓN

La implementación del proyecto inicia con la creación de un nuevo proyecto JEE en el IDE NetBeans, para lo cual se realizan los siguientes pasos:

- Se inicia el IDE NetBeans 7.1
- Se escoge la opción “New Project” del menú: “File” lo que abre el wizard de creación de nuevos proyectos.
- En el wizard se selecciona la opción: “Enterprise Application” como se aprecia en la Figura 3.1

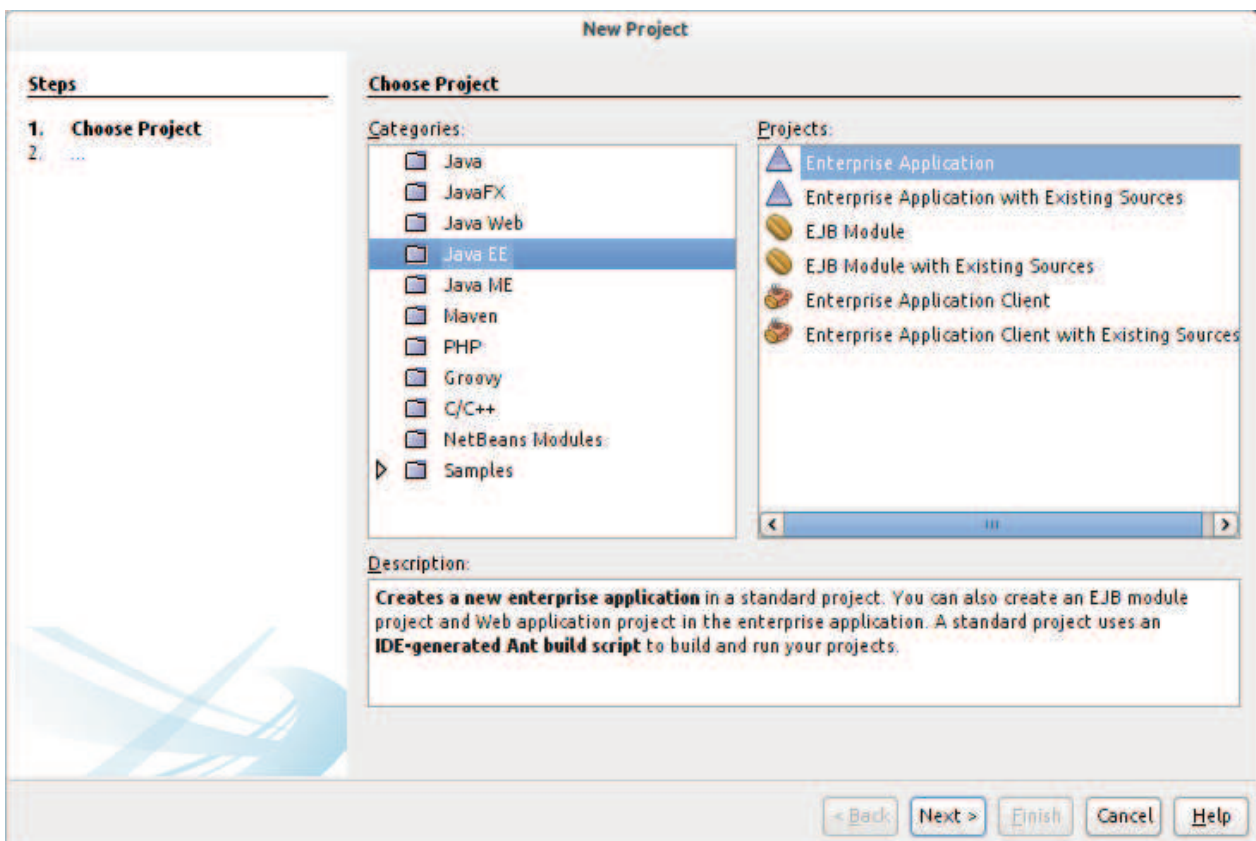


Figura 3.1: Creación de Proyecto JEE

- Después de dar clic en Next, se define el nombre del proyecto como: “WEBADEPON” y se finaliza el proceso, tras lo cual se obtiene el proyecto con dos módulos: WEBADEPON-ejb y WEBADEPON-war; y un contenedor de estos módulos: WEBADEPON, como se puede ver en la Figura 3.2

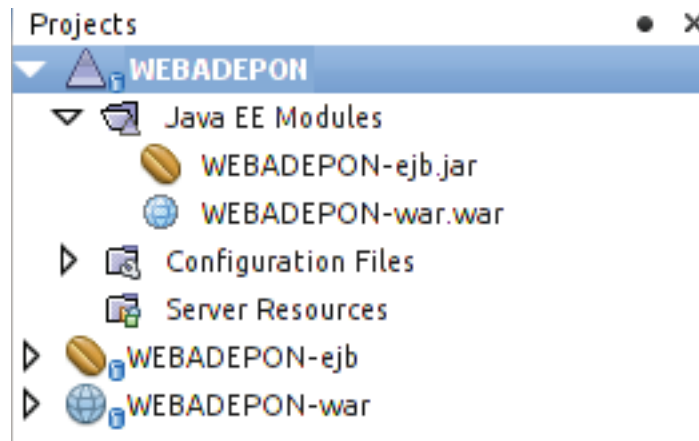


Figura 3.2: Estructura del Proyecto

A continuación se describe la implementación de los tres componentes del patrón MVC: Modelo, Vista y Controlador y la implementación de la seguridad.

3.1.1 IMPLEMENTACIÓN DE LA CAPA DE PERSISTENCIA (MODELO EN EL PATRÓN MVC)

Para la implementación de la capa de persistencia se siguieron los siguientes estándares:

- En la carpeta “entities” se encuentran las clases correspondientes a las tablas de la base de datos, es decir la capa de persistencia, para lo cual se utilizó el framework JPA que es el estándar de JEE.
- En la carpeta “sessions” se encuentran los componentes sessions beans para lo cual se utilizó el framework EJB que es el estándar de JEE.
- El nombre de las entidades es el mismo de las tablas de la base de datos, con la diferencia que únicamente la primera letra es con mayúscula.
- Los atributos de las clases y de los componentes se escriben con minúsculas, excepto la primera letra a partir de la segunda palabra.

- El nombre de los métodos de las clases y de los componentes se escriben con minúsculas, excepto la primera letra a partir de la segunda palabra.

Para la implementación de la capa de persistencia se hace uso de dos APIs (Application Programming Interface). Un API representa el conjunto de funciones y procedimientos que ofrece una biblioteca para ser utilizado por otro software. A continuación se describen los APIs utilizados dentro de la capa de persistencia del proyecto:

- JPA (Java Persistence API): Esta API es un sistema de mapeo objeto-relacional (ORM), lo que en la práctica simula una base de datos orientada a objetos, sobre la base de datos relacional.
- EJB. (Enterprise JavaBeans): Esta API permite el desarrollo rápido y sencillo de aplicaciones distribuidas, transaccionales, seguras y portátiles al ofrecer de forma transparente, varios servicios al programador, tales como: creación, consulta, actualización y eliminación (conocidas como operaciones CRUD, por sus siglas en inglés) de datos de la base de datos.

Utilizando estos frameworks se crean las clases entidad y los componentes sesión beans, los cuales forman la capa de persistencia y son descritos a continuación.

3.1.1.1 Clases Entidad (Entities)

Para cada tabla de la base de datos se creó una clase entidad con el uso del API JPA. Para crear las clases entidad del proyecto, en NetBeans 7.x, se realizan los siguientes pasos:

- Se crea el paquete “entities” en el módulo: WEBADEPON-ejb, para lo cual se da clic derecho sobre el módulo, se selecciona la opción New → Other. Esto abre el wizard de creación de nuevos archivos.
- En el wizard, se selecciona la opción: “Java Package” dentro de “Java”, tras lo cual se define el nombre del paquete como: “entities”, su ubicación y se da clic en finalizar.

- Después de crear el paquete “entities”, se da clic derecho sobre el paquete y se selecciona la opción New → Other. Una vez más se abre el wizard de creación de nuevos archivos
- En el wizard, se selecciona la opción: “Entity Classes from Database” dentro de “Persistence”, tras lo cual se selecciona el “Data Source” que apunte a la Base de Datos
- Se escogen las tablas de las cuales se desean crear las clases entidad, como se aprecia en la Figura 3.3.
- Se confirma el paquete en el que se almacenarán las clases entidad y se finaliza el proceso.

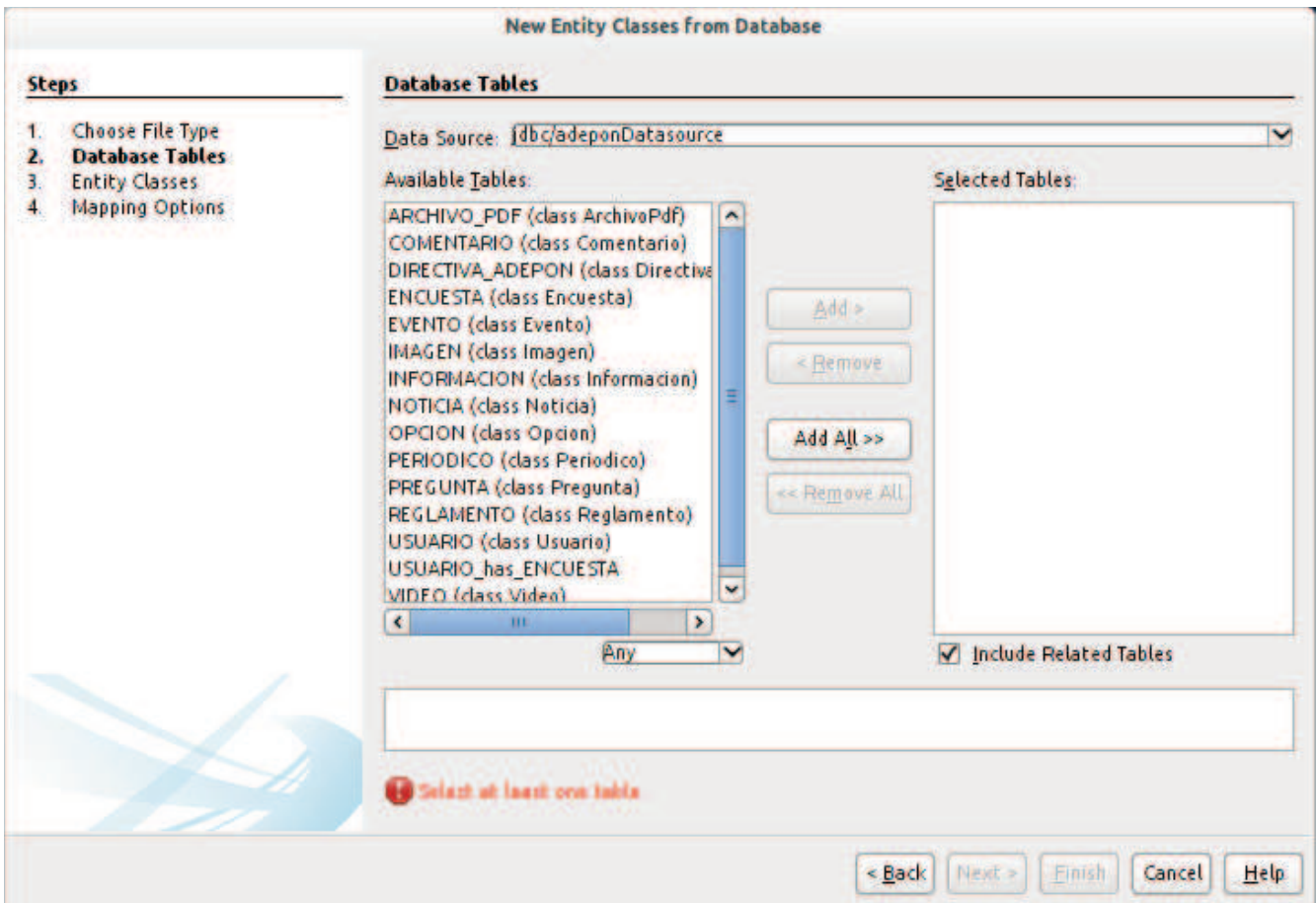


Figura 3.3: Creación de una Clase Entidad.

Realizado el proceso descrito se obtienen las clases entidad en el paquete “entities”. La Figura 3.4 muestra un ejemplo del código de una clase entidad.

```

public class Evento implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "idEVENTOS")
    private Integer idEVENTOS;
    @Column(name = "FECHA_PUB")
    @Temporal(TemporalType.DATE)
    private Date fechaPub;
    @Column(name = "FECHA_REALIZACION")
    @Temporal(TemporalType.DATE)
    private Date fechaRealizacion;
    @Size(max = 45)
    @Column(name = "HORA")
    private String hora;
    @Size(max = 200)
    @Column(name = "LUGAR")
    private String lugar;
    @Lob
    @Size(max = 16777215)
    @Column(name = "DESCRIPCION")
    private String descripcion;
    @Size(max = 200)
    @Column(name = "TITULO")
    private String titulo;
    @OneToMany(mappedBy = "eEVENTOSidEVENTOS")
    private Collection<Comentario> comentarioCollection;
    @JoinColumn(name = "VIDEO_idVIDEO", referencedColumnName = "idVIDEO")
    @ManyToOne
    private Video vIDEOidVIDEO;
    @JoinColumn(name = "IMAGENES_idIMAGENES", referencedColumnName = "idIMAGENES")
    @ManyToOne
    private Imagen iMAGENESidIMAGENES;

    public Evento() {
    }

    public Evento(Integer idEVENTOS) {
        this.idEVENTOS = idEVENTOS;
    }

    public Integer getIdEVENTOS() {
        return idEVENTOS;
    }

    public void setIdEVENTOS(Integer idEVENTOS) {
        this.idEVENTOS = idEVENTOS;
    }

    public Date getFechaPub() {
        return fechaPub;
    }
}

```

Figura 3.4: Código de una Clase Entidad.

3.1.1.2 Componentes Session Bean

Para cada clase entidad se creó un componente session bean con el uso del API EJB. Cada componente permitirá realizar operaciones CRUD sobre la base de datos de forma transparente. A continuación se describen los pasos para crear los componentes sessions beans del proyecto.

- Se crea el paquete “session” en el módulo: WEBADEPON-ejb, para lo cual se da clic derecho sobre el módulo, se selecciona la opción New → Other. Esto abre el wizard de creación de nuevos archivos.
- En el wizard, se selecciona la opción: “Java Package” dentro de “Java”, tras lo cual se define el nombre del paquete como: “session” y se da clic en finalizar.
- Después de crear el paquete “session”, se da clic derecho sobre el paquete y se selecciona la opción New → Other. Una vez más se abre el wizard de creación de nuevos archivos
- En el wizard, se selecciona la opción: “Session Beans from Entities Classes” dentro de “Enterprise JavaBeans”, tras lo cual se escogen las clases entidad de las cuales se desea crear los componentes sessions beans.
- Se confirma el paquete en el que se almacenarán los componentes sessions beans como se muestra en la Figura 3.5 y se finaliza el proceso.

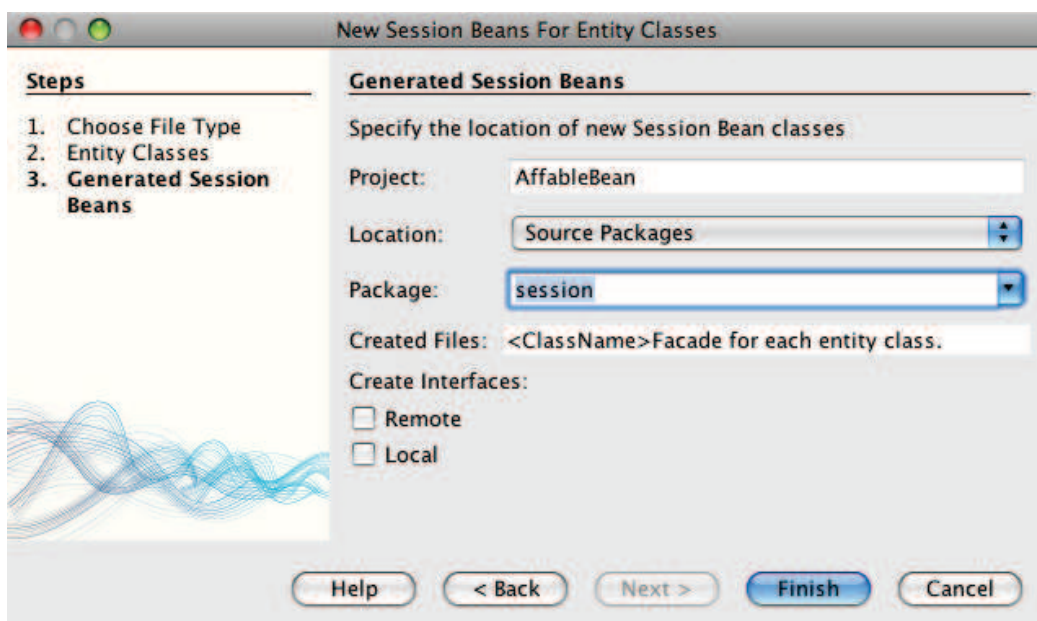


Figura 3.5: Creación de un componente Session Bean.

Realizado el proceso descrito se obtienen los componentes sessions beans en el paquete “session”. La Figura 3.6 muestra un ejemplo del código de un session bean.

```
package sessions;

import entities.Evento;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

/**
 *
 * @author joserock
 */
@Stateless
public class EventoFacade extends AbstractFacade<Evento> {
    @PersistenceContext(unitName = "WEBADEPON-ejbPU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public EventoFacade() {
        super(Evento.class);
    }
}
```

Figura 3.6: Código de un componente Session Bean.

Al crear los componentes sesión beans se crea también la clase abstracta “AbstractFacade” de la cual heredan todos los beans de sesión, y en la cual se pueden añadir nuevos métodos personalizados de búsqueda u otras interacciones con la base de datos. Para el presente proyecto se añadió el método: “findByParameter” el cual permite buscar en cualquiera de las clases entidad un parámetro, especificando el nombre de la columna.

La Figura 3.7 muestra el código de la Clase Abstracta: “AbstractFacade”

```

public abstract class AbstractFacade<T> {

    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public void create(T entity) {
        getEntityManager().persist(entity);
    }

    public void edit(T entity) {
        getEntityManager().merge(entity);
    }

    public void remove(T entity) {
        getEntityManager().remove(getEntityManager().merge(entity));
    }

    public T find(Object id) {
        return getEntityManager().find(entityClass, id);
    }

    public List<T> findAll() {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        return getEntityManager().createQuery(cq).getResultList();
    }

    public List<T> findByParameter(String column, Object value) {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        CriteriaBuilder cb = getEntityManager().getCriteriaBuilder();
        //CriteriaQuery cq = cb.createQuery();
        Root e = cq.from(entityClass);
        cq.where(cb.equal(e.get(column), value));
        Query query = getEntityManager().createQuery(cq);
        List<T> result = query.getResultList();
        return result;
    }

    public List<T> findRange(int[] range) {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        javax.persistence.Query q = getEntityManager().createQuery(cq);
        q.setMaxResults(range[1] - range[0]);
        q.setFirstResult(range[0]);
        return q.getResultList();
    }

    public int count() {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        javax.persistence.criteria.Root<T> rt = cq.from(entityClass);
        cq.select(getEntityManager().getCriteriaBuilder().count(rt));
        javax.persistence.Query q = getEntityManager().createQuery(cq);
        return ((Long) q.getSingleResult()).intValue();
    }
}

```

Figura 3.7: Código de la clase Abstract Facade.

3.1.2 IMPLEMENTACIÓN DE LA CAPA DE LÓGICA DEL NEGOCIO (CONTROLADOR EN EL PATRÓN MVC)

Para la implementación de la capa de lógica del negocio se siguieron los siguientes estándares:

- En la carpeta “beans” se encuentran los controladores del sitio, en donde están los métodos y por ende la lógica del negocio.
- Los atributos de las clases y de los componentes se escriben con minúsculas, excepto la primera letra a partir de la segunda palabra.
- El nombre de los métodos de las clases y de los componentes se escriben con minúsculas, excepto la primera letra a partir de la segunda palabra.

La capa de lógica del negocio hace uso de componentes *Managed Beans* (Java beans con inyección de dependencias) que están asociados con los componentes usados en las páginas que conforman la capa de presentación.

Los *Managed Beans* facilitan la comunicación con la capa de presentación, con respecto a otras herramientas utilizadas como controladores en la implementación del modelo MVC, como por ejemplo los Java Servlet.

Validar los datos de un componente, manejar un evento iniciado por un componente, o realizar procesamiento para determinar la siguiente página a la que la aplicación debe navegar, son algunas de las funciones que comúnmente realizan los métodos de un managed bean.

En NetBeans 7.1 para crear un managed bean se deben realizar los siguientes pasos:

- Click derecho en el nodo del proyecto seleccionar la opción New → Other. Esto abre el wizard de creación de nuevos archivos.
- En el wizard seleccionar JSF Managed Beans dentro de la categoría Java Server Faces (Figura 3.8).
- En el paso final se definen las propiedades del nuevo archivo (paquete, nombre,

ubicación, etc.), tras lo cual se crea un nuevo managed bean en la ubicación señalada.

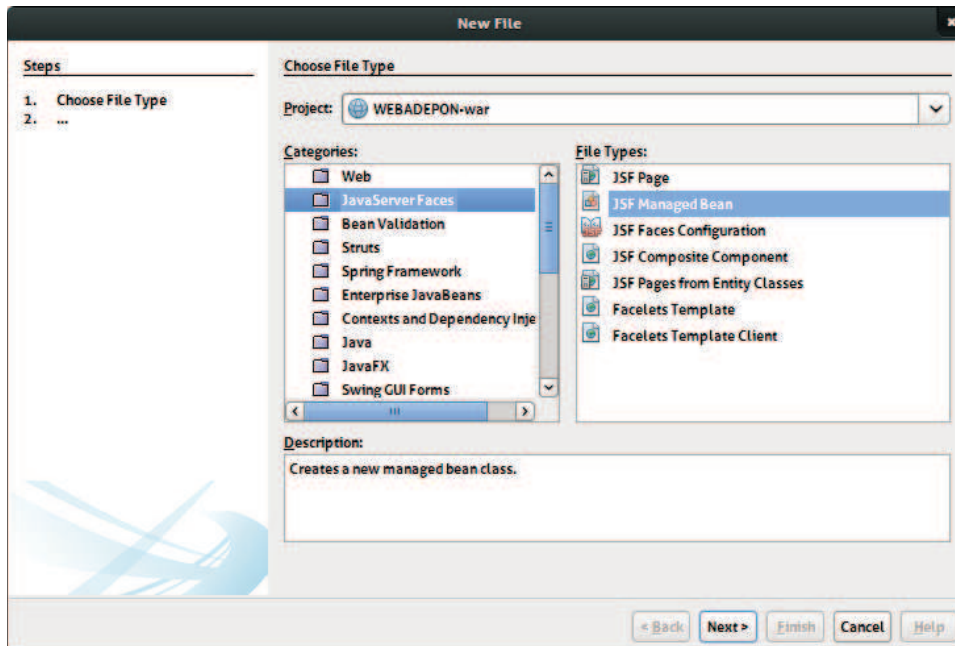


Figura 3.8: Creación de un componente Managed Bean.

El código del managed bean tiene la forma que se presenta en la Figura 3.9.

```
package adepon.beans;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

/**
 *
 * @author martin
 */
@ManagedBean
@RequestScoped
public class NewJSFManagedBean {

    /**
     * Creates a new instance of NewJSFManagedBean
     */
    public NewJSFManagedBean() {
    }
}
```

Figura 3.9 Código de un Managed Bean.

Una vez añadida funcionalidad y código a cada managed bean, la estructura es la siguiente:

- Bloque de imports. En este bloque de código se hallan las clases adicionales que el managed bean usa en su funcionamiento; en la Figura 3.10 se observa un ejemplo de su contenido.

```
package adepon.beans;

import adepon.POJO.ReglamentosPdf;
import adepon.comparator.RegulationDateComparator;
import adepon.comparator.RegulationItemComparator;
import com.icesoft.faces.context.ByteArrayResource;
import entities.ArchivoPdf;
import entities.Reglamento;
import java.io.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;
import java.util.List;
import javax.ejb.EJB;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;
import org.icesoft.ace.component.fileentry.FileEntry;
import org.icesoft.ace.component.fileentry.FileEntryEvent;
import org.icesoft.ace.component.fileentry.FileEntryResults;
import sessions.ArchivoPdfFacade;
import sessions.ReglamentoFacade;
```

Figura 3.10: Bloque de imports de un Managed Bean.

- Bloque de declaración de variables. Por estándar los objetos y variables que se usan en cada managed bean se declaran al inicio de la clase, y pueden o no estar inicializados como se observa en la Figura 3.11.

```

@ManagedBean
@ViewScoped
public class RegulationsBean implements Serializable {

    private Reglamento regulations; // Objeto para administracion
    private List<Reglamento> regulationsList;
    private boolean panelCreation;
    private List<ReglamentosPdf> resourceList;
    private List<ReglamentosPdf> resourceListStatute;
    private RegulationItemComparator regulationItemComparator =
        new RegulationDateComparator();
    private Date dateReg;
    @EJB
    private ArchivoPdfFacade pdfFacade;
    @EJB
    private ReglamentoFacade regulationsFacade;
}

```

Figura 3.11: Bloque de declaración de variables de un ManagedBean.

- Lógica del managed bean. Este segmento del archivo es donde se encuentra la lógica propiamente dicha, incluye todos los métodos programados dentro de cada managed bean. En la Figura 3.12 se observa un fragmento de código con algunos de los métodos del Regulations Bean.

```

public RegulationsBean() {
    resourceList = new ArrayList<ReglamentosPdf>();
    resourceListStatute = new ArrayList<ReglamentosPdf>();
    regulations = new Reglamento();
    panelCreation = true;
}

public void pdfListener(FileEntryEvent e) throws FileNotFoundException, IOException {
    FileEntry fe = (FileEntry) e.getComponent();
    FileEntryResults results = fe.getResults();

    for (FileEntryResults.FileInfo i : results.GetFiles()) {
        if (i.isSaved()) {
            File file = i.getFile();
            if (file != null) {
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                Date date = new Date();
                String time = dateFormat.format(date);

                ArchivoPdf archivo = new ArchivoPdf(); // nuestra entidad
                File f = file; //asociamos el archivo fisico
                InputStream is = new FileInputStream(f); //lo abrimos. Lo importante es
                byte[] buffer = new byte[(int) f.length()]; //creamos el buffer
                is.read(buffer); //leemos el archivo al buffer
                archivo.setArchivoBin(buffer); //lo guardamos en la entidad
                archivo.setTamano(String.valueOf(i.getSize()));
                archivo.setFechaUpload(time);
                archivo.setNombre(i.getFileName());
                pdfFacade.create(archivo);
                regulations.setARCHIVOSPDFidARCHIVOSPDF(archivo);
            }
        }
    }
}

```

Figura 3.12: Lógica de un Managed Bean.

- Métodos Setters y Getters. Este bloque hace que la información dentro del managed bean sea accesible para la capa de presentación o para otros managed beans, normalmente son métodos autogenerados en base a los atributos de la clase (Figura 3.13).


```

public void setResourceList(List<ReglamentosPdf> resourceList) {
    this.resourceList = resourceList;
}

public Reglamento getRegulations() {
    return regulations;
}

public void setRegulations(Reglamento regulations) {
    panelCreation = false;
    this.regulations = regulations;
}

public boolean isPanelCreation() {
    return panelCreation;
}

public void setPanelCreation(boolean panelCreation) {
    this.panelCreation = panelCreation;
}

public List<Reglamento> getRegulationsList() {
    regulationsList = regulationsFacade.findAll();
    sortRegulationList();
    return regulationsList;
}

```

Figura 3.13: Métodos getters y setters de un Managed Bean.

3.1.3 IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN (VISTA EN EL PATRÓN MVC)

Para la implementación de la capa de presentación (Vista en el patrón MVC) se sigue el siguiente estándar:

- Los nombres de las vistas se escriben en letra minúscula; si tienen más de dos palabras, las primeras letras a partir de la segunda palabra se escriben en mayúsculas.

La capa de presentación del proyecto está compuesta por páginas.xhtml (versión extendida de.html) con tags.jsf comunes junto con tags de ICEFaces que proveen características avanzadas de una RIA a las vistas. Por ejemplo en los comentarios si un usuario hace un nuevo comentario todos los demás usuarios que estén en la misma página verán el nuevo comentario automáticamente sin necesidad de actualizar la página.

Para crear una página xhtml en NetBeans 7.x se debe:

- Hacer click derecho en el nodo del proyecto, seleccionar la opción New > Other. Esto abre el wizard de creación de nuevos archivos,
- En el wizard seleccionar JSF Page dentro de la categoría Java Server Faces, y añadir la extensión xhtml para el nuevo archivo (Figura 3.14).

El uso de xhtml garantiza que los documentos estén correctamente escritos y “bien formados” pues este lenguaje de marcado hace énfasis en el correcto uso y correcta sintaxis de los tags dentro de cada documento.

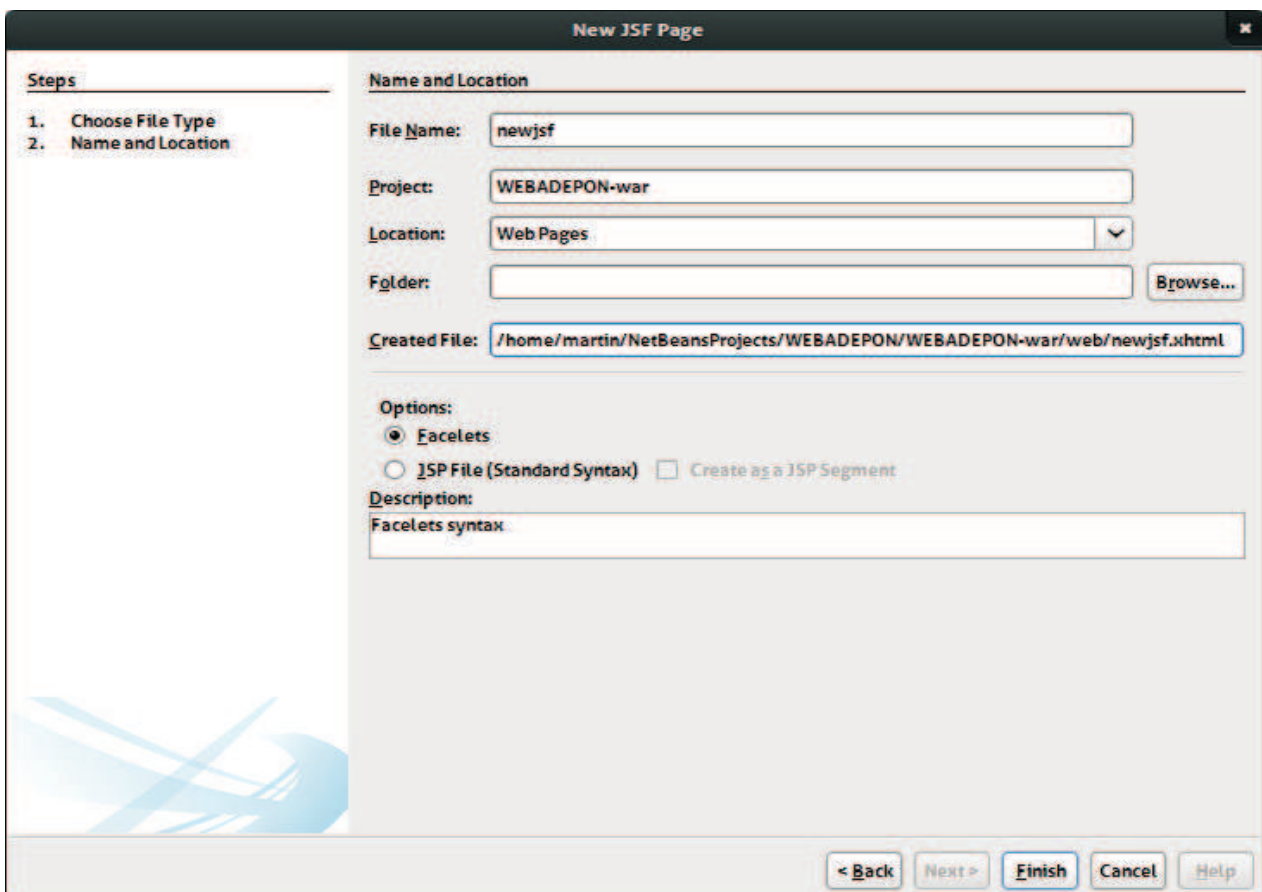


Figura 3.14: Creación de una página xhtml.

Una página xhtml nueva como la que se ve en la Figura 3.15 tiene el siguiente código:

- Declaración de XHTML DOCTYPE
- Los elementos <html>, <head>, <body> apropiadamente anidados.
- El atributo xmlns en <html> especificando el espacio de nombres XML para el documento.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    Hello from Facelets
  </h:body>
</html>
```

Figura 3.15: Código de una página xhtml nueva.

Las páginas de la capa de presentación tienen definido un cuerpo y una plantilla que evita repetir código innecesario en cada una de las páginas, esta plantilla (Figura 3.16) define las proporciones y ubicación del encabezado, menú, cuerpo, y pie de las páginas.


```

<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:icecore="http://www.icefaces.org/icefaces/core"
xmlns:ace="http://www.icefaces.org/icefaces/components"
xmlns:ice="http://www.icesoft.com/icefaces/component">
<h:head>
<title>ADEPON</title>
<ice:outputStyle href="/css/rime.css" />
<h:outputScript name="jsf.js" library="javax.faces"/>
<link href="http://fonts.googleapis.com/css?family=Yellowtail" rel="stylesheet" type="text/css"/>
<script type="text/javascript" src="js/jquery-1.4.2-min.js"/>
<script type="text/javascript" src="js/panelEncuesta.js"/>
<link href="favicon.ico" type="image/x-icon" rel="shortcut icon"/>
</h:head>
<h:body style="background-color: whitesmoke; padding-top: 0px;">
<div style="width: 83%; height:auto !important;
height:99%; min-height:99%; background-color: #f4f4f4;
margin-top: -8px; margin-left: auto;
margin-right: auto; border-bottom: 4px solid black;">
<div style="height: 7px;background-color: black;"></div>

<div id="header">
<ui:include src="/WEB-INF/components/header.xhtml">
<ui:param name="param_name" value="param_value"/>
</ui:include>
</div >
<center>
<br/><br/><br/>
<div id="content" >
<ui:insert name="page-content"/>
</div>
</center>
</div>
</h:body>
</html>

```

Figura 3.16: Código de la plantilla implementada por las páginas.xhtml.

El cuerpo de cada página está definido en un archivo diferente con el fin de facilitar el desarrollo. En la Figura 3.17 se puede observar el cuerpo de una de las páginas del proyecto.

```

<?xml version= 1.0 encoding= UTF-8 ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<div xmlns="http://www.w3.org/1999/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:icecore="http://www.icefaces.org/icefaces/core"
xmlns:ice="http://www.icesoft.com/icefaces/component"
xmlns:ace="http://www.icefaces.org/icefaces/components"
>
<br/><br/><br/>
<h:form>
<ace:dataTable value="#{regulationsBean.resourceList}"
var="pdf"
paginator="true"
paginatorPosition="bottom"
rows="5"
style="width: 710px">
<ace:column>
<f:facet name="header">
<ice:outputLabel value="Reglamentos" style="font-family: Yellowtail, cursive;
font-size:45px;
text-align:center; height: 210px"/>
</f:facet>
<ice:panelGrid columns="3">
<ice:panelGrid style="width: 180px; height: 90px; text-align: center;">
<ice:outputResource resource="#{pdf.archivoPDF}"
attachment="true"
label="#{pdf.name}"
image="/img/pdf.png"/>
</ice:panelGrid>
<ice:panelGrid style="width: 350px; text-align: left">
<ice:outputResource resource="#{pdf.archivoPDF}"
attachment="true"
label="#{pdf.name}"
style="font-size: 14px"/>
</ice:panelGrid>
<ice:outputText value="#{pdf.date}" style="font-style: oblique; font-weight : bold; font-size: 15px"/>
</ice:panelGrid>
</ace:column>
</ace:dataTable>
</h:form>
</div>

```

Figura 3.17: Código de una de las páginas xhtml del proyecto.

3.1.4 IMPLEMENTACIÓN DE SEGURIDADES

Al buscar ofrecer un producto de software que cumpla con los requerimientos de seguridad de cualquier sitio web actual, especialmente en su administración, se encontró la necesidad de añadir diferentes métodos para garantizar que tanto el acceso al sitio, manejo de los datos, y administración se realice de forma segura y privada. Las seguridades en el proyecto se implementaron usando características propias de JSF, orientadas a la seguridad, e integrando la aplicación con el servidor web Glassfish que provee varias herramientas para este fin; el proyecto incluye el uso de conexión segura a través del Protocolo seguro de transferencia de hipertexto (HTTPS por sus siglas en inglés), autenticación a través del servidor web Glassfish, encriptación de contraseñas y manejo de roles de usuarios para restringir el acceso a diferentes partes del sitio.

El servidor Glassfish es un producto muy robusto orientado para aplicaciones empresariales de mediana y gran escala, sus seguridades son sólidas y permite a través de configuraciones sencillas simplificar y hacer más seguro el proceso de autenticación. Como se observa en la Figura 3.18 basta con crear y configurar un “realm” de seguridad para que Glassfish se haga cargo de la mayor parte de seguridades de la aplicación. El servidor web se conecta directamente con la base de datos a través de los pool de conexión evitando pasos intermedios donde podría haber fallas de seguridad.

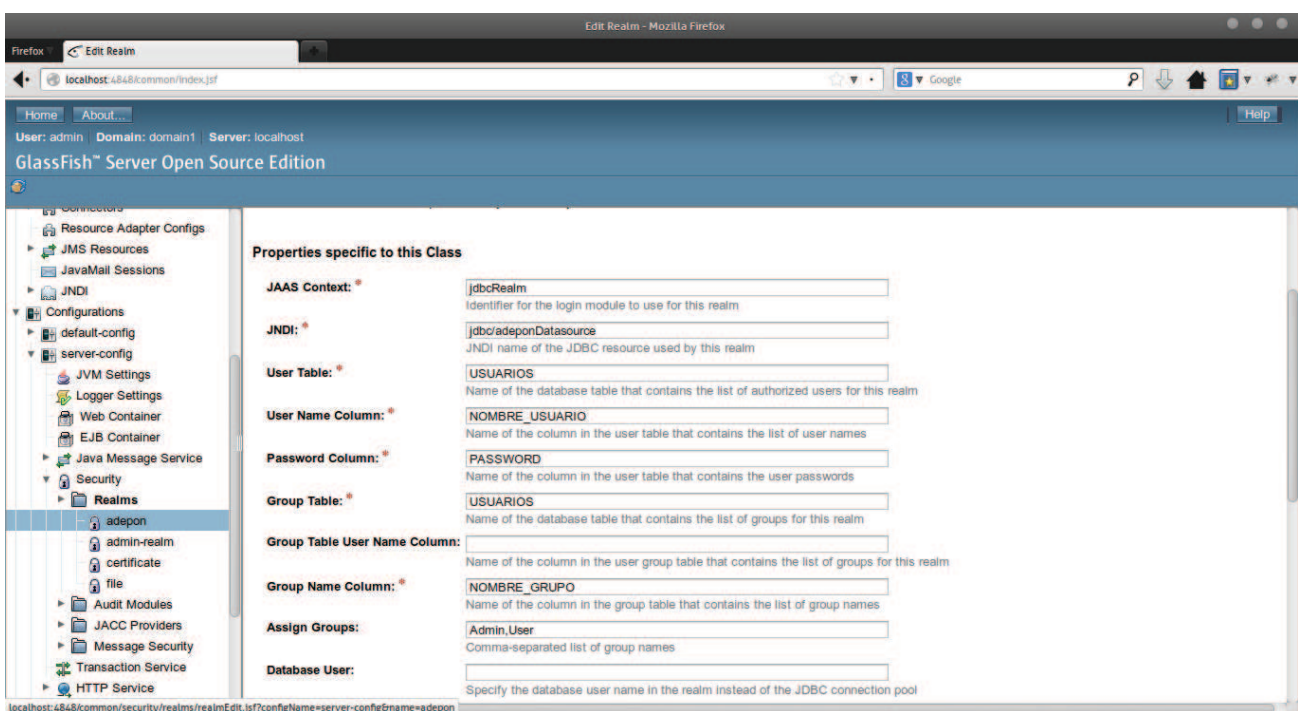


Figura 3.18 Configuración de un “realm” de seguridad en Glassfish

Entre otros aspectos se configuran la tabla y columnas que almacenan los datos de los usuarios (roles, nombres de usuario, contraseñas), el método de encriptación (que por defecto en Glassfish 3 es SHA-256), y el pool de conexiones que el realm va a usar. Este método de autenticación se puede invocar desde la aplicación de forma sencilla desde los métodos de *login* de las clases de seguridad como se muestra en la Figura 3.19, donde la línea resaltada es la que llama a la autenticación de Glassfish.

```

FacesContext context = FacesContext.getCurrentInstance();
HttpServletRequest request = (HttpServletRequest) context.getExternalContext().getRequest();
try {
    session.setAttribute("usernameAdmin", username);
    retrieveUserInfo();
    if (isAvlb && cUser != null && cUser.getNombreGrupo().equals("Admin") == true) {
        request.login(username, password);
        isLoggedIn = true;
        FacesContext.getCurrentInstance().getExternalContext().redirect(SystemVariable.getHost() + "WEBADEPON-
    } else {
        errorLogin = true;
        session.removeAttribute("usernameAdmin");
        // Handle unknown username/password in request.login().
        context.addMessage(null, new FacesMessage("Unknown login"));
    }
} catch (ServletException e) {
    errorLogin = true;
    // Handle unknown username/password in request.login().
    context.addMessage(null, new FacesMessage("Unknown login"));
}
}

```

Figura 3.19 Autenticación de Usuarios

Dentro de los archivos de configuración de la aplicación (*web.xml*, *glassfish-web.xml*) se especifica el tipo de autenticación y se declaran los roles de seguridad, esta vez a nivel de la aplicación (Figura 3.20).

```

<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>adepon</realm-name>
  <form-login-config>
    <form-login-page>/faces/adminLogin.xhtml</form-login-page>
    <form-error-page>/faces/adminLogin.xhtml</form-error-page>
  </form-login-config>
</login-config>

<security-role>
  <role-name>Admin</role-name>
</security-role>
<security-role>
  <role-name>User</role-name>
</security-role>

```

Figura 3.20. Tipo de Autenticación y Roles

Otra parte importante en la configuración de la seguridad dentro del archivo *web.xml*, es la definición de que páginas usan https y que rol de seguridad tiene acceso a dichas páginas. En la Figura 3.21 se observa el fragmento de código en xml con las configuraciones, el tag *<security-constraint>* indica el inicio de la definición de una restricción de seguridad, que contiene los patrones de url (localizador de recursos uniforme) que se incluyen dentro de la

restricción, al final de la restricción de seguridad se especifican el nombre del rol que tiene acceso a estos url y el uso de encriptación, es decir https, en dichas páginas.

```

<security-constraint>
  <web-resource-collection>
    <web-resource-name>admin-area</web-resource-name>
    <url-pattern>/faces/admin.xhtml</url-pattern>
    <url-pattern>/admin.jsf</url-pattern>
    <url-pattern>/faces/adminComment.xhtml</url-pattern>
    <url-pattern>/adminComment.jsf</url-pattern>
    <url-pattern>/faces/adminComment.xhtml</url-pattern>
    <url-pattern>/adminDirective.jsf</url-pattern>
    <url-pattern>/faces/adminDirective.xhtml</url-pattern>
    <url-pattern>/adminNews.jsf</url-pattern>
    <url-pattern>/faces/adminEvents.xhtml</url-pattern>
    <url-pattern>/adminEvents.jsf</url-pattern>
    <url-pattern>/faces/adminInquest.xhtml</url-pattern>
    <url-pattern>/adminInquest.jsf</url-pattern>
    <url-pattern>/faces/adminUser.xhtml</url-pattern>
    <url-pattern>/adminUser.jsf</url-pattern>
    <url-pattern>/faces/adminNewspaper.xhtml</url-pattern>
    <url-pattern>/adminNewspaper.jsf</url-pattern>
    <url-pattern>/faces/adminRegulations.xhtml</url-pattern>
    <url-pattern>/adminRegulations.jsf</url-pattern>
    <url-pattern>/faces/adminDirective.xhtml</url-pattern>
    <url-pattern>/adminDirective.jsf</url-pattern>
    <url-pattern>/faces/adminMision.xhtml</url-pattern>
    <url-pattern>/adminMision.jsf</url-pattern>
    <url-pattern>/faces/adminVideo.xhtml</url-pattern>
    <url-pattern>/adminVideo.jsf</url-pattern>
    <url-pattern>/faces/videoUpload.xhtml</url-pattern>
    <url-pattern>/videoUpload.jsf</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>Admin</role-name>
  </auth-constraint>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>

```

Figura 3.21 Restricción de Seguridad

Para el correcto funcionamiento de las configuraciones de seguridad del sistema se necesita mapear los roles de seguridad del sistema en el archivo *glassfish-web.xml* como se muestra en la Figura 3.22.

```

<glassfish-web-app error-url="">
  <context-root>/WEBDEPON-war</context-root>
  <security-role-mapping>
    <role-name>Admin</role-name>
    <group-name>Admin</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>User</role-name>
    <group-name>User</group-name>
  </security-role-mapping>
  <class-loader delegate="true" />
  <jsp-config>
    <property name="keepgenerated" value="true">
      <description>Keep a copy of the generated servlet class' java code.</description>
    </property>
  </jsp-config>
</glassfish-web-app>

```

Figura 3.22 Archivo *glassfish-web.xml*

Finalmente para la encriptación de la contraseña de los usuarios hace uso del algoritmo SHA-256, el sistema graba en base la cadena resultante de aplicar el algoritmo a la contraseña provista por el usuario, para la autenticación el sistema aplica nuevamente el algoritmo a la contraseña que se ingresó y compara la cadena resultante a la almacenada en la base de datos. Por la lógica manejada en el sistema las contraseñas no son recuperables. En la Figura 3.23 se observa uno de los métodos que encriptan una cadena entrante haciendo uso del algoritmo SHA-256.

```

public String stringToSha(String password) throws NoSuchAlgorithmException {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    md.update(password.getBytes());

    byte byteData[] = md.digest();

    //convert the byte to hex format method 1
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < byteData.length; i++) {
        sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
    }
    return sb.toString();
}

```

Figura 3.23 Método de encriptación SHA-256

3.2 PRUEBAS

3.2.1 PRUEBAS UNITARIAS

XP propone realizar pruebas unitarias basadas en el enfoque "Test before Programming" el cual se basa en elaborar y ejecutar un completo conjunto de pruebas para un módulo del programa incluso antes de que el módulo esté acabado, con la finalidad de trabajar siempre en un entorno seguro y libre de errores [9].

Los programadores escriben pruebas unitarias de manera que puedan confiar en el funcionamiento del programa. El resultado es un programa que se vuelve más y más confiable con el tiempo, y se vuelve más capaz de aceptar el cambio. [10]

No es necesario escribir una prueba para cada método único que se escribe, sólo para los métodos de producción que se podrían romper [10]. Una prueba unitaria debe ser: automatizable, completa e independiente de otras pruebas.

Sin embargo, escribir pruebas unitarias es una tarea tediosa y pesada, y a menudo, difícil. Una forma de facilitar esto es utilizar un framework que automatice algunos de los detalles de la ejecución de pruebas, como JUnit. "JUnit es un framework de pruebas sencillo y práctico para clases Java, que alienta la estrecha integración entre las pruebas y el proceso de desarrollo. Sin embargo, incluso con herramientas como JUnit, escribir pruebas unitarias a menudo requiere un gran esfuerzo" [11].

Para utilizar JUnit en un entorno JEE, se hace necesario manejar un contenedor EJB, de tal manera que sea posible cargar los beans de sesión. En la figura 3.24 se observan las librerías necesarias para realizar las pruebas.

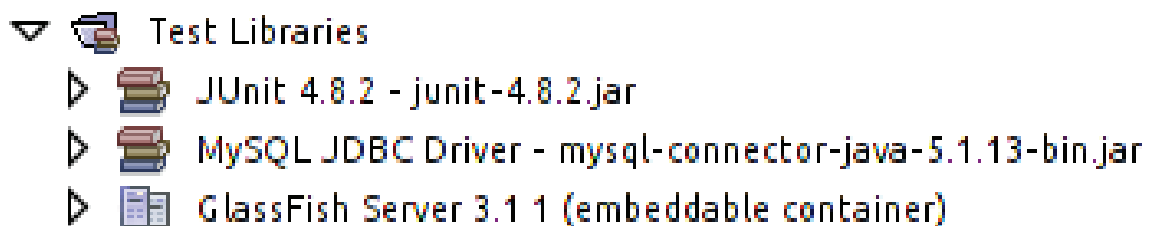


Figura 3.24: Librerías para pruebas Unitarias en ambientes JEE

En la Figura 3.25 se describe una clase de prueba creada con JUnit, a la cual se la adaptó después a las necesidades de la prueba. En la Figura se aprecian los 3 atributos indispensables para realizar las pruebas unitarias sobre un ManagedBean, los cuales son:

- Un objeto EJB: el cual será enviado como parámetro al objeto del Managed Bean a probar.
- Un objeto del ManagedBean a probar
- Un contenedor de EJBs

En la Figura 3.25 se aprecia también el uso de la anotación `@BeforeClass` y `@AfterClass`. La anotación `@BeforeClass` permite que el método a continuación sea ejecutado una única vez antes de realizarse la prueba unitaria, por lo que en este método se realizó la instanciación del objeto contenedor de los EJBs. Mientras la anotación `@AfterClass` define la ejecución del método que la sigue al final de la prueba unitaria, y en este método se definió el cierre del objeto contenedor de los EJBs.

```

@EJB
private NoticiaFacade noticiasFacade;
private NewsBean newsBean;
private static EJBContainer container;

@BeforeClass
public static void setUpClass() throws Exception {
    Map<String, Object> properties = new HashMap<String, Object>();
    properties.put(EJBContainer.MODULES, new File("build/jar"));
    container = EJBContainer.createEJBContainer(properties);
    System.out.println("Opening the container");
}

@AfterClass
public static void tearDownClass() throws Exception {
    container.close();
    System.out.println("Closing the container");
}

```

Figura 3.25: Prueba Unitaria, métodos: `@BeforeClass` y `@AfterClass`

En la Figura 3.26 se aprecia un método prueba, que se lo define con el uso de la anotación @Test. El método presentado en la Figura 3.26 se encarga de probar el método: “create-News” del ManagedBean: “NewsBean”, que registra la noticia en la base de datos y la publica en las redes sociales: Twitter y FaceBook.

```

@Test
public void testCreateNews() throws NamingException {
    noticiasFacade = (NoticiaFacade) container.getContext().lookup("java:global/classes/NoticiasFacade");
    newsBean = new NewsBean(noticiasFacade);
    Noticia news = new Noticia();
    news.setContenido("contenido");
    news.setFecha("01/01/2013");
    news.setResumen("RESUMEN");
    news.setTitulo("TITULO CREADA");
    newsBean.setNews(news);
    newsBean.createNews();

    if (noticiasFacade.findByParameter("titulo", news.getTitulo()).isEmpty()) {
        fail("not create the news");
    }
}
}

```

Figura 3.26: Método prueba

En el Anexo 1 se detallan todas las pruebas unitarias realizadas.

3.2.2 PRUEBAS DE ACEPTACIÓN

Son las pruebas finales basadas en el uso del programa al terminar su proceso de implementación. Para el registro de las pruebas de aceptación se utilizó el formato descrito en la Figura 3.27, basado en el trabajo realizado por García Monge Ana Mercedes en su proyecto de titulación “Desarrollo e Implantación del Sistema de Administración Estudiantil vía web para la unidad educativa particular trilingüe HELMUT WALHMÜLLER” [8].

PRUEBA DE ACEPTACIÓN	
Caso de Prueba:	Opción de Prueba:
Número de caso de prueba:	Número de Historia de Usuarios:
Nombre de Caso de prueba:	
Precondiciones:	
Datos:	
Pasos:	
Resultado Esperado:	
Evaluación:	

Figura 3.27: Prueba de Aceptación

Donde:

Caso de Prueba: Es el nombre del Módulo

Opción de Prueba: Es el nombre de opción que tiene en el menú

Número de Caso de Prueba: Es el número de la prueba

Número de Historia de Usuario: Es el número de la historia de usuario

Nombre de Caso de Prueba: Es el nombre de la historia de usuario

Precondiciones: Condiciones que debe cumplirse

Datos: Todo lo que el usuario ingresa

Pasos: Los pasos que tiene que hacer el usuario

Resultado Esperado: Es el resultado que se espera

Evaluación: Es el resultado que se obtiene

Las tablas 3.1 a 3.19 describen a cada una de las pruebas de aceptación realizadas.

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Noticias	Opción de Prueba: Noticias
Número de caso de prueba: 1	Número de Historia de Usuarios: 1
Nombre de Caso de prueba: Registro de Noticia	
Precondiciones:	
Datos:	
TITULO	VARCHAR(100)
RESUMEN	VARCHAR(200)
CONTENIDO	MEDIUMTEXT
FECHA	VARCHAR(45)
IMAGEN_idIMAGEN	LONGBLOB
VIDEO_idVIDEO	VARCHAR(100)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración del sitio. 2. Ingresar a Noticias. 3. Desplegar el panel de registro de Noticias. 4. Seleccionar y subir una imagen o asociar un video con la Noticia. 5. Llenar los campos del formulario. 6. Hacer click en el botón de registro. 	
Resultado Esperado: Todos los datos de la noticia se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos de la noticia se guardan en la base de datos sin ningún error.	

Tabla 3.1: Prueba de Aceptación Registro de Noticia

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Noticias	Opción de Prueba: Noticias
Número de caso de prueba: 1	Número de Historia de Usuarios: 2
Nombre de Caso de prueba: Presentación de Noticias	
Precondiciones:	
Datos:	
TITULO	VARCHAR(100)
RESUMEN	VARCHAR(200)
CONTENIDO	MEDIUMTEXT
FECHA	VARCHAR(45)
IMAGEN_idIMAGEN	LONGBLOB
VIDEO_idVIDEO	VARCHAR(100)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar al sitio. 2. Ingresar a Noticias. 	
Resultado Esperado: Se presentan al usuario las noticias almacenadas en la base de datos con sus respectivos datos	
Evaluación:	
Se presentan al usuario las noticias almacenadas en la base de datos con sus respectivos datos	

Tabla 3.2: Prueba de Aceptación Presentación de Noticias

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Eventos	Opción de Prueba: Eventos
Número de caso de prueba: 1	Número de Historia de Usuarios: 3
Nombre de Caso de prueba: Registro de Evento	
Precondiciones:	
Datos:	
FECHA_PUB	DATE
FECHA_REALIZACION	DATE
LUGAR	VARCHAR(200)
DESCRIPCION	MEDIUMTEXT
TITULO	VARCHAR(200)
IMAGEN_idIMAGEN	LONGBLOB
VIDEO_idVIDEO	VARCHAR(100)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración del sitio. 2. Ingresar a Eventos. 3. Desplegar el panel de registro de Eventos. 4. Seleccionar y subir una imagen o asociar un video con el Evento. 5. Llenar los campos del formulario. 6. Hacer click en el botón de registro. 	
Resultado Esperado: Todos los datos del evento se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos del evento se guarden en la base de datos sin ningún error.	

Tabla 3.3: Prueba de Aceptación Registro de Evento

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Eventos	Opción de Prueba: Eventos
Número de caso de prueba: 1	Número de Historia de Usuarios: 4
Nombre de Caso de prueba: Presentación de Eventos	
Precondiciones:	
Datos:	
FECHA_PUB	DATE
FECHA_REALIZACION	DATE
LUGAR	VARCHAR(200)
DESCRIPCION	MEDIUMTEXT
TITULO	VARCHAR(200)
IMAGEN_idIMAGEN	LONGBLOB
VIDEO_idVIDEO	VARCHAR(100)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar al sitio. 2. Ingresar a Noticias. 	
Resultado Esperado: Se presentan al usuario los eventos almacenados en la base de datos con sus respectivos datos	
Evaluación:	
Se presentan al usuario los eventos almacenados en la base de datos con sus respectivos datos	

Tabla 3.4: Prueba de Aceptación Presentación de Eventos

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Documentos	Opción de Prueba: Documentos
Número de caso de prueba: 1	Número de Historia de Usuarios: 5
Nombre de Caso de prueba: Registro de Documento	
Precondiciones:	
Datos:	
FECHA	VARCHAR(45)
FECHA_PUB	VARCHAR(45)
DESCRIPCION	VARCHAR(300)
ARCHIVO_PDF_idARCHIVO_PDF	LONGBLOB
TIPO	VARCHAR(45)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración del sitio. 2. Ingresar a Documentos. 3. Desplegar el panel de registro de Documentos. 4. Seleccionar y subir un archivo. 5. Llenar los campos del formulario. 6. Hacer click en el botón de registro. 	
Resultado Esperado: Todos los datos del documento se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos del documento se guardan en la base de datos sin ningún error.	

Tabla 3.5: Prueba de Aceptación Registro de Documento

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Documentos	Opción de Prueba: Documentos
Número de caso de prueba: 1	Número de Historia de Usuarios: 6
Nombre de Caso de prueba: Presentación de Documento	
Precondiciones:	
Datos: FECHA	VARCHAR(45)
FECHA_PUB	VARCHAR(45)
DESCRIPCION	VARCHAR(300)
ARCHIVO_PDF_idARCHIVO_PDF	LONGBLOB
TIPO	VARCHAR(45)
Pasos: 1. Ingresar al sitio. 2. Ingresar a Resoluciones, Reglamentos, Estatuto.	
Resultado Esperado: Se presentan al usuario los documentos almacenados en la base de datos con sus respectivos datos	
Evaluación: Se presentan al usuario los documentos almacenados en la base de datos con sus respectivos datos	

Tabla 3.6: Prueba de Aceptación Presentación de Documento

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Periódicos	Opción de Prueba: Periódicos
Número de caso de prueba: 1	Número de Historia de Usuarios: 7
Nombre de Caso de prueba: Registro de Periódico	
Precondiciones: Ingresar Archivo PDF	
Datos: FECHA	VARCHAR(45)
FECHA_PUB	VARCHAR(45)
DESCRIPCION	VARCHAR(300)
ARCHIVO_PDF_idARCHIVO_PDF	LONGBLOB

<p>Pasos: 1. Ingresar a administración del sitio.</p> <p>2. Ingresar a Periódico.</p> <p>3. Desplegar el panel de registro de Periódico.</p> <p>4. Seleccionar y subir un archivo.</p> <p>5. Llenar los campos del formulario.</p> <p>6. Hacer click en el botón de registro.</p>
<p>Resultado Esperado: Todos los datos del periódico se guardan en la base de datos sin ningún error.</p>
<p>Evaluación:</p> <p>Todos los datos del periódico se guardan en la base de datos sin ningún error.</p>

Tabla 3.7: Prueba de Aceptación Registro de Periódico

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Periódicos	Opción de Prueba: Periódicos
Número de caso de prueba: 1	Número de Historia de Usuarios: 8
Nombre de Caso de prueba: Presentación de Periódicos	
Precondiciones:	
Datos:	
FECHA	VARCHAR(45)
FECHA_PUB	VARCHAR(45)
DESCRIPCION	VARCHAR(300)
ARCHIVO_PDF_idARCHIVO_PDF	LONGBLOB
<p>1. Pasos: Ingresar al sitio.</p> <p>2. Ingresar a Resoluciones, Reglamentos, Estatuto.</p>	
Resultado Esperado: Se presentan al usuario los periódicos almacenados en la base de datos con sus respectivos datos	
Evaluación:	
Se presentan al usuario los periódicos almacenados en la base de datos con sus respectivos datos	

Tabla 3.8: Prueba de Aceptación Presentación de Periódicos

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Encuestas	Opción de Prueba: Encuestas
Número de caso de prueba: 1	Número de Historia de Usuarios: 9
Nombre de Caso de prueba: Registro de Encuesta	
Precondiciones:	
Datos:	
Pregunta	VARCHAR (45)
SELECCIÓN_MULTIPLE	TINYINT(1)
Opciones de Respuesta	VARCHAR (45)
HABILITADA	TINYINT(1)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración. 2. En el menú ir a Encuestas 3. Ingresar el título de la encuesta, y dar clic en Subir encuesta 4. En el panel administrar encuesta añadir una pregunta dando clic en el botón “+” ingresar el texto de la pregunta y dar clic en subir pregunta 5. Añadir las respuestas de cada pregunta dando clic en el botón “Rpts” 	
Resultado Esperado:	
Todos los datos de la encuesta se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos de la encuesta se guardan en la base de datos sin ningún error.	

Tabla 3.9: Prueba de Aceptación Registro de Encuesta

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Encuestas	Opción de Prueba: Encuestas
Número de caso de prueba: 1	Número de Historia de Usuarios: 10
Nombre de Caso de prueba: Responder Encuesta	
Precondiciones: Crear un usuario. Crear una encuesta	
Datos: Pregunta VARCHAR (45) SELECCIÓN_MULTIPLE TINYINT(1) Opciones de Respuesta VARCHAR (45) HABILITADA TINYINT(1)	
Pasos: 1. Ingresar a administración. 2. En el menú ir a Encuestas 3. Seleccionar una de las encuestas activas 4. Iniciar Sesión 5. Responder la Encuesta 6. Clic en el botón Responder	
Resultado Esperado: Todos las respuestas de la encuesta enviadas por el usuario se almacenan en la base de datos sin ningún error	
Evaluación: Todos las respuestas de la encuesta enviadas por el usuario se almacenan en la base de datos sin ningún error	

Tabla 3.10: Prueba de Aceptación Responder Encuesta

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Encuesta	Opción de Prueba: Encuestas
Número de caso de prueba: 1	Número de Historia de Usuarios: 11
Nombre de Caso de prueba: Presentación de Encuestas	
Precondiciones: Crear una encuesta	
Datos:	
Pregunta	VARCHAR (45)
SELECCIÓN_MULTIPLE	TINYINT(1)
Opciones de Respuesta	VARCHAR (45)
HABILITADA	TINYINT(1)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar al sitio 2. En el menú ir a Encuestas 3. Clic en la opción Ver resultados 	
Resultado Esperado:	
Se presentan al usuario los resultados de las encuestas almacenados en la base de datos.	
Evaluación:	
Se presentan al usuario los resultados de las encuestas almacenados en la base de datos.	

Tabla 3.11: Prueba de Aceptación Presentación de Encuestas

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Eventos	Opción de Prueba: Eventos
Número de caso de prueba: 1	Número de Historia de Usuarios: 3
Nombre de Caso de prueba: Comentarios de eventos	
Precondiciones: Crear un usuario, Crear un evento	
Datos:	
Comentario	MEDIUMTEXT

<p>Pasos:</p> <ol style="list-style-type: none"> 1. Ingresar al sitio 2. Dar clic sobre uno de los eventos 3. Iniciar Sesión 4. Ingresar el texto del comentario 5. Clic en el botón enviar
<p>Resultado Esperado:</p> <p>El texto del comentario se guarda en la base de datos asociado al evento y al usuario sin ningún error.</p>
<p>Evaluación:</p> <p>El texto del comentario se guarda en la base de datos asociado al evento y al usuario sin ningún error.</p>

Tabla 3.12: Prueba de Aceptación Comentar Evento

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Video	Opción de Prueba: Video
Número de caso de prueba: 1	Número de Historia de Usuarios: 13
Nombre de Caso de prueba: Presentación de video en vivo	
Precondiciones: Iniciar transmisión de video en vivo	
Datos: -----	
<p>Pasos:</p> <ol style="list-style-type: none"> 1. Ingresar al sitio 2. En el menú ir a Video en vivo 3. Se muestra video en vivo de haber una transmisión en ese momento 	
Resultado Esperado:	
Se observa video en vivo	
Evaluación:	
Se observa video en vivo	

Tabla 3.13: Prueba de Aceptación Presentación de video en vivo

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Cuentas de Usuario	Opción de Prueba: Usuarios
Número de caso de prueba: 1	Número de Historia de Usuarios: 14
Nombre de Caso de prueba: Registro de usuarios	
Precondiciones:	
Datos:	
NOMBRE_USUARIO	VARCHAR(45)
PASSWORD	VARCHAR(64)
NOMBRE_GRUPO	VARCHAR(45)
MAIL	VARCHAR(45)
MAILPOLI	VARCHAR(45)
NOMBRES	VARCHAR(45)
APELLIDOS	VARCHAR(45)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración. 2. En el menú ir a Usuarios 3. Llenar los datos del formulario de Usuarios 4. Clic en el botón subir usuario 	
Resultado Esperado:	
Todos los datos del usuario se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos del usuario se guardan en la base de datos sin ningún error.	

Tabla 3.14: Prueba de Aceptación Registro de usuarios

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: Cuentas de Usuario	Opción de Prueba: Administrar Cuenta
Número de caso de prueba: 1	Número de Historia de Usuarios: 15
Nombre de Caso de prueba: Edición de la cuenta de usuario propia	
Precondiciones:	
Datos:	
NOMBRE_USUARIO	VARCHAR(45)
PASSWORD	VARCHAR(64)
NOMBRE_GRUPO	VARCHAR(45)
MAIL	VARCHAR(45)
MAILPOLI	VARCHAR(45)
NOMBRES	VARCHAR(45)
APELLIDOS	VARCHAR(45)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar al sitio 2. Iniciar sesión, clic sobre la opción Ver perfil 3. Cambiar los datos del perfil de usuario (Contraseña, nombre de usuario o email) 	
Resultado Esperado:	
Todos los datos a actualizar del usuario se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos a actualizar del usuario se guardan en la base de datos sin ningún error.	

Tabla 3.15: Prueba de Aceptación Edición de la cuenta de usuario propia

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: ADEPON	Opción de Prueba: Directiva
Número de caso de prueba: 1	Número de Historia de Usuarios: 16
Nombre de Caso de prueba: Registro de la directiva de la ADEPON	
Precondiciones:	
Datos:	
CARGO	VARCHAR(45)
TITULO	VARCHAR(45)
NOMBRE	VARCHAR(45)
APELLIDO	VARCHAR(45)
OFICINA	VARCHAR(45)
EMAIL	VARCHAR(45)
INFORMACION_ADICIONAL	VARCHAR(300)
IMÁGEN_idIMAGEN	LONGBLOB
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración. 2. En el menú ir a Usuarios 3. Llenar los datos del formulario de Usuarios 4. Clic en el botón subir usuario 	
Resultado Esperado:	
Todos los datos de miembro de la directiva de la ADEPON se guardan en la base de datos sin ningún error.	
Evaluación:	
Todos los datos de miembro de la directiva de la ADEPON se guardan en la base de datos sin ningún error.	

Tabla 3.16: Prueba de Aceptación Registro de la directiva de la ADEPON

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: ADEPON	Opción de Prueba: Directiva
Número de caso de prueba: 1	Número de Historia de Usuarios: 16
Nombre de Caso de prueba: Presentación de la información de la directiva de la ADEPON	
Precondiciones:	
Datos:	
CARGO	VARCHAR(45)
TITULO	VARCHAR(45)
NOMBRE	VARCHAR(45)
APELLIDO	VARCHAR(45)
OFICINA	VARCHAR(45)
EMAIL	VARCHAR(45)
INFORMACION_ADICIONAL	VARCHAR(300)
IMÁGEN_idIMAGEN	LONGBLOB
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar al sitio 2. En el menú ir a Directiva 	
Resultado Esperado:	
Se presentan al usuario los datos de los miembros de la directiva de la ADEPON almacenados en la base de datos.	
Evaluación:	
Se presentan al usuario los datos de los miembros de la directiva de la ADEPON almacenados en la base de datos.	

Tabla 3.17: Prueba de Aceptación Presentación de la información de la directiva de la ADEPON

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: ADEPON	Opción de Prueba: Misión y Visión
Número de caso de prueba: 1	Número de Historia de Usuarios: 17
Nombre de Caso de prueba: Edición de la misión y visión de la ADEPON	
Precondiciones:	
Datos:	
NOMBRE_CAMPO	VARCHAR(45)
DETALLE	MEDIUMTEXT
TITULO	VARCHAR(45)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar a administración. 2. En el menú ir a Misión y Visión 3. Llenar el campos de la misión 4. Clic en el botón subir 5. Llenar el campos de la visión 6. Clic en el botón subir 	
Resultado Esperado:	
El texto de la misión y la visión se guarda en la base de datos sin ningún error.	
Evaluación:	
El texto de la misión y la visión se guarda en la base de datos sin ningún error.	

Tabla 3.18: Prueba de Aceptación Edición de la misión y visión de la ADEPON

PRUEBA DE ACEPTACIÓN	
Caso de Prueba: ADEPON	Opción de Prueba: Misión y Visión
Número de caso de prueba: 1	Número de Historia de Usuarios: 17
Nombre de Caso de prueba: Presentación de la misión y visión de la ADEPON	
Precondiciones:	
Datos:	
NOMBRE_CAMPO	VARCHAR(45)
DETALLE	MEDIUMTEXT
TITULO	VARCHAR(45)
Pasos:	
<ol style="list-style-type: none"> 1. Ingresar al sitio 2. En el menú ir a Misión y Visión 	
Resultado Esperado:	
Se presenta al usuario la misión y la visión de la ADEPON almacenados en la base de datos.	
Evaluación:	
Se presenta al usuario la misión y la visión de la ADEPON almacenados en la base de datos	

Tabla 3.19: Prueba de Aceptación Presentación de la misión y visión de la ADEPON

3.3 IMPLANTACIÓN

La ADEPON no cuenta con la infraestructura necesaria para implantar el presente proyecto en sus instalaciones; esto hizo necesario buscar las soluciones disponibles en el mercado que se adecuaran a las necesidades del proyecto. Debido a las características del proyecto desarrollado se llegó a la conclusión de que la solución más adecuada es el uso de un servidor virtual privado (VPS) sobre el cual se puede configurar y optimizar adecuadamente el entorno de operación del proyecto.

El siguiente paso fue la selección del proveedor del servicio de VPS y de registro del dominio www.adepon.org; debido a que se requiere de un servicio serio, confiable y con el soporte adecuado se escogió a GoDaddy, empresa estadounidense líder a nivel mundial en servicios de hosting, VPS, y registro de dominios, que garantiza una disponibilidad de 99.9% [12], rápida solución de problemas, fácil mantenimiento, y confiabilidad probada por sus millones de clientes.

Características del VPS:

Sistema Operativo:	Fedora 14
Ancho de Banda:	2,000 GB/mes
Disco duro:	60 GB
Memoria RAM:	2 GB
Dirección IPV4:	184.168.86.43
Precio:	\$35.99/mes

El siguiente paso, una vez contratado el VPS y registrado el dominio, es asociar la dirección del VPS con el dominio, en GoDaddy esta operación se realiza de forma sencilla desde el panel de control del dominio (Figura 3.28).

The screenshot displays the GoDaddy Domain Details control panel for the domain ADEPON.ORG. The interface includes a navigation menu with options like Domains, Hosting, and Email. The main content area is divided into three sections: Name Servers, DNS Manager, and TLD Specific. The Name Servers section shows NS63.DOMAINCONTROL.COM and NS64.DOMAINCONTROL.COM. The DNS Manager section is available and shows records for A, CNAME, and MX. The TLD Specific section is not applicable. At the bottom, there is a table for contact information.

Registrant	Administrative	Technical	Billing
Registration Private Domains By Proxy, LLC DomainsByProxy.com	Registration Private Domains By Proxy, LLC DomainsByProxy.com	Registration Private Domains By Proxy, LLC DomainsByProxy.com	Registration Private Domains By Proxy, LLC DomainsByProxy.com

Figura 3.28: Panel de control del dominio

La configuración del VPS se realiza conectándose remotamente al mismo, desde un cliente ssh como putty en Windows o desde una terminal cualquiera en Linux. Para liberar capacidad de procesamiento y memoria se tiene deshabilitado por default el entorno gráfico, por lo que toda interacción se realiza a través de línea de comandos.

El primer aspecto a configurar es la instalación de la base de datos, máquina virtual java y servidor de aplicaciones Glassfish, lo que se puede realizar totalmente a través del propio gestor de software del sistema operativo (yum en el caso de Fedora). Posteriormente se configura el dominio de la aplicación dentro de Glassfish, incluyendo opciones de seguridad, conexión a la base de datos y rendimiento, todo este proceso se realiza desde la consola de administración del servidor de aplicaciones.

Para la implantación del sitio web es importante establecer al proyecto como Proyecto en estado de Producción, para lo cual se establece en el archivo "web.xml" para la variable "javax.faces.PROJECT_STAGE" el valor "Production" como se aprecia en la figura 3.29 (Su valor por defecto es "Development").

```
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Production</param-value>
</context-param>
```

Figura 3.29: Proyecto en estado de Producción

Al compilar el proyecto dentro de NetBeans se genera un archivo EAR que es el que se usa para el despliegue en el servidor de aplicaciones, desde la consola de administración de Glassfish se despliega la aplicación.

3.3.1 AFINAMIENTO DEL SERVIDOR GLASSFISH EN EL ENTORNO DE PRODUCCIÓN

Con el fin de obtener un rendimiento óptimo para el sitio web mientras se encuentre en producción, se deben ajustar varios de los parámetros del servidor desde la consola de administración. Para la afinación del servidor Glassfish se utilizó como referencia el documento “Optimize GlassFish Performance in a Production Environment” [13]. Aunque el documento se centra en GlassFish v2, las sugerencias son aplicables a GlassFish v3. Los parámetros que se deben ajustar son:

- Tamaño de la Pila de Java
- Hilos de procesamiento de peticiones HTTP
- Pool de Conexiones
- Contenedor EJB

3.3.1.1 Tamaño de la Pila de Java

El tamaño de la pila se determina por la opción `-Xmx` (maximum) y `-Xms` (minimum). Mientras más grande es la pila puede contener más objetos y reducir la frecuencia con la que se ejecuta el “garbage collection”.

Para determinar el tamaño de la pila de Java se siguen los pasos descritos a continuación:

- Acceder a la consola de Glassfish a través de un browser y registrarse en ella con un usuario administrador de Glassfish.
- En el árbol que aparece en la columna izquierda de la consola de administración, expandir el nodo “Configurations” y hacer clic en “JVM Settings”.
- En el panel principal, seleccionar la ficha “JVM Opciones”, donde se aprecian las variables `-Xmx` y `-Xms`.

Todo esto se aprecia en la Figura 3.30.

Para el proyecto se definieron los valores de las variables `-Xmx` y `-Xms` en 1024m y 800m respectivamente.

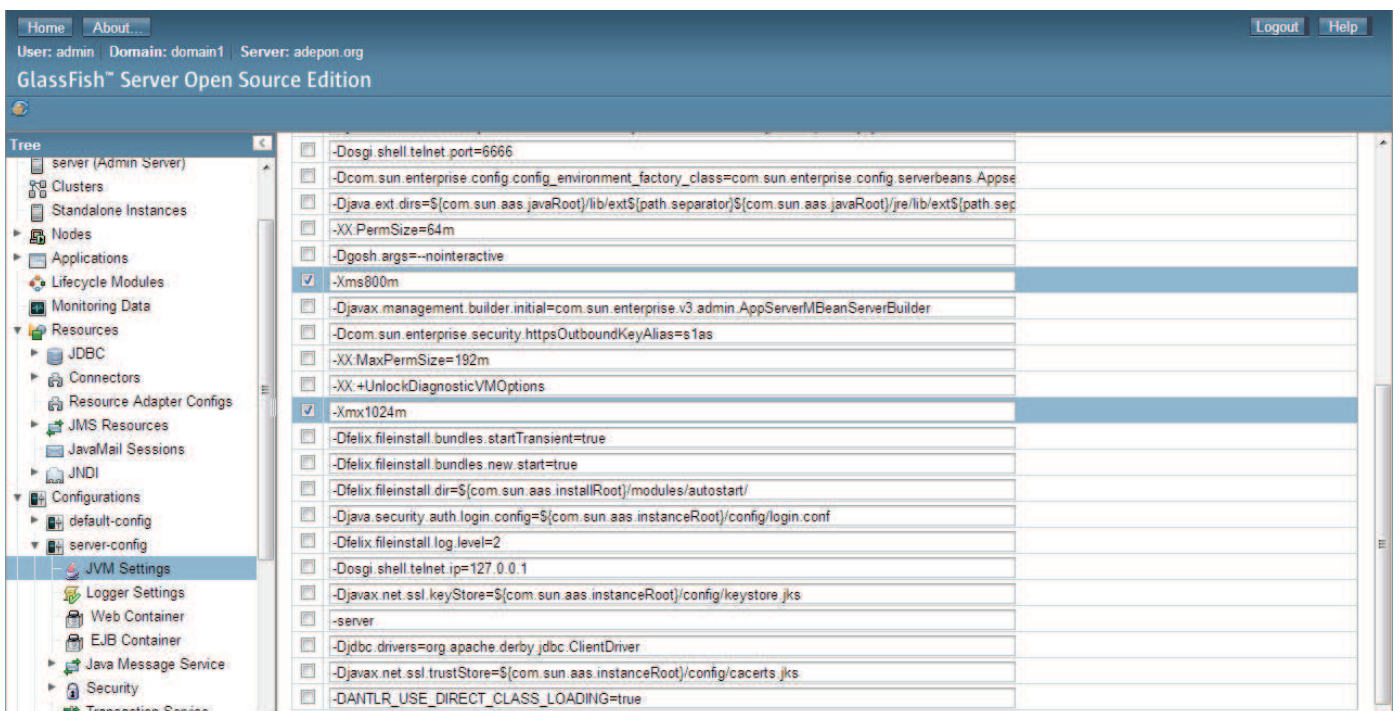


Figura 3.30: Afinamiento Tamaño de la Pila de JAVA

3.3.1.2 Hilos de procesamiento de peticiones HTTP

Para el afinamiento de Glassfish se debe también especificar el número máximo y mínimo de solicitudes simultáneas que el servidor puede manejar, lo cual es determinando por los parámetros “*Thread Max Pool Size*” y “*Min Thread Pool Size*”. Cuando el servidor ha alcanzado el límite de hilos o solicitudes, difiere el proceso de nuevas solicitudes hasta que el número de solicitudes activas desciende por debajo del valor máximo. El aumento de estos valores reducirá los tiempos de latencia de respuesta HTTP. También se debe establecer la cantidad máxima de tiempo (en segundos) que un hilo puede permanecer inactivo en el pool. Después de este tiempo de inactividad el hilo se retira del pool.

Para determinar el número de hilos de procesamiento de peticiones HTTP se siguen los pasos descritos a continuación:

- Acceder a la consola de Glassfish a través de un browser y registrarse en ella con un usuario administrador de Glassfish.
- En el árbol que aparece en la columna izquierda de la consola de administración, expandir el nodo Configurations dar clic en Thread Pool y seleccionar http-thread-pool
- Tras lo cual se aprecian los parámetros “*Thread Max Pool Size*”, “*Min Thread Pool Size*” y “*Idle Thread Timeout*”.

Todo esto se aprecia en la Figura 3.31.

Para el proyecto los parámetros fueron definidos de la siguiente manera:

- Thread Max Pool Size: 50 hilos
- Min Thread Pool Size: 10 hilos
- Idle Thread Timeout: 300 segundos

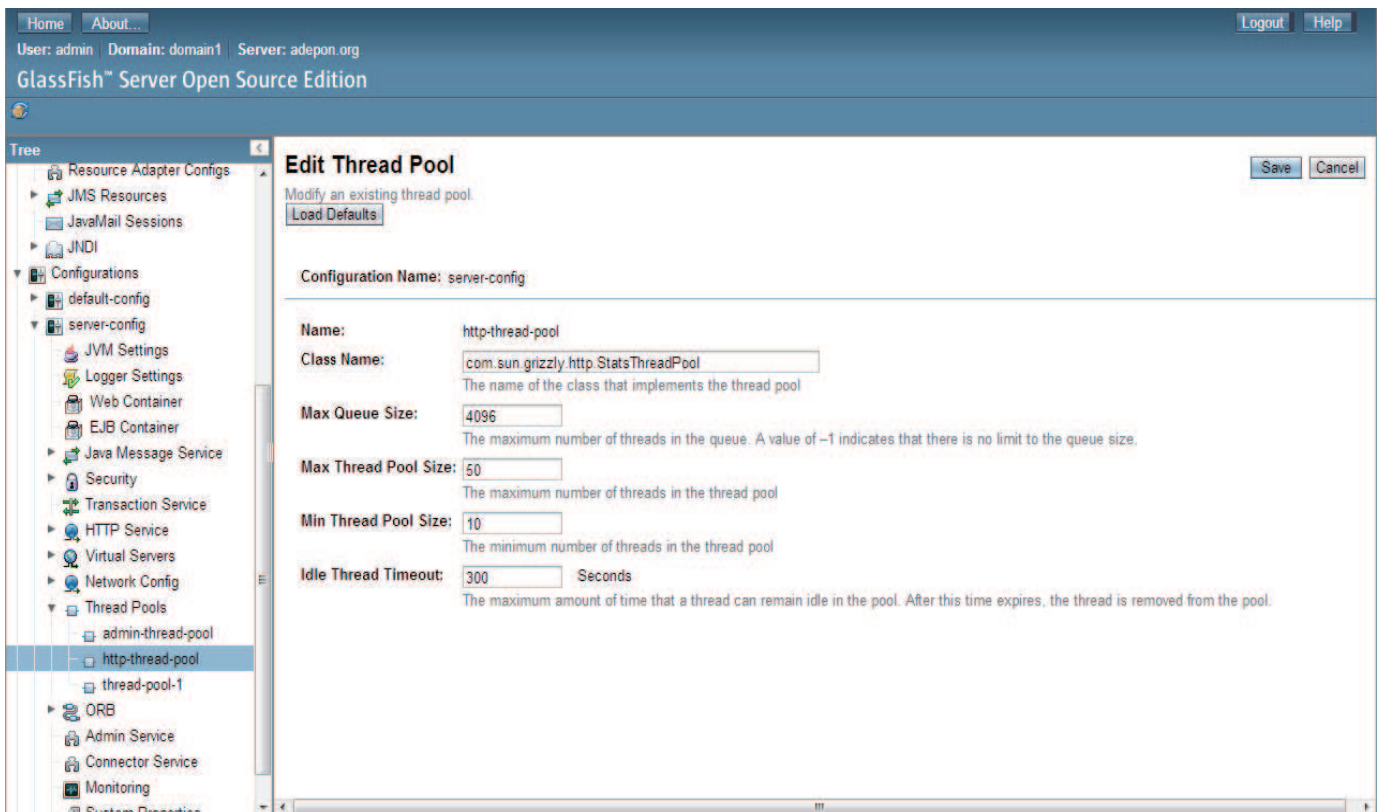


Figura 3.31: Afinamiento Hilos de procesamiento de peticiones HTTP

3.3.1.3 Pool de Conexiones

El Pool de Conexiones se refiere al manejo de una colección de conexiones abiertas a una base de datos, permitiendo reutilizar conexiones por nuevas consultas a la base de datos. Para afinar el pool de conexiones es importante ajustar el valor de los parámetros “max-pool-size” (número máximo de conexiones permitidas en el pool) e “initial and minimum pool size” (número mínimo e inicial de conexiones permitidas en el pool) con el mismo número de hilos de procesamiento de peticiones HTTP. Además se deben configurar los siguientes parámetros:

- Tiempo ocioso: (idle-timeout, en segundos). El tiempo máximo que la conexión puede permanecer inactiva en el pool.
- Cantidad quitada del pool: (pool-resize-quantity). El número de conexiones que se quitan cuando expira el tiempo ocioso establecido.

- Máximo tiempo de espera: (max-wait-time, en milisegundos). Cantidad de tiempo en que espera quien realiza la petición antes de que se envíe una conexión.

Para determinar el número de conexiones del pool se siguen los pasos descritos a continuación:

- Acceder a la consola de Glassfish a través de un browser y registrarse en ella con un usuario administrador de Glassfish.
- En el árbol que aparece en la columna izquierda de la consola de administración, expandir el nodo Resources → JDBC Connection → Pool de Conexiones, a continuación se hace clic en el pool de conexiones del proyecto.
- Donde se aprecian las variables –Xmx y –Xms.

Todo esto se aprecia en la Figura 3.31.

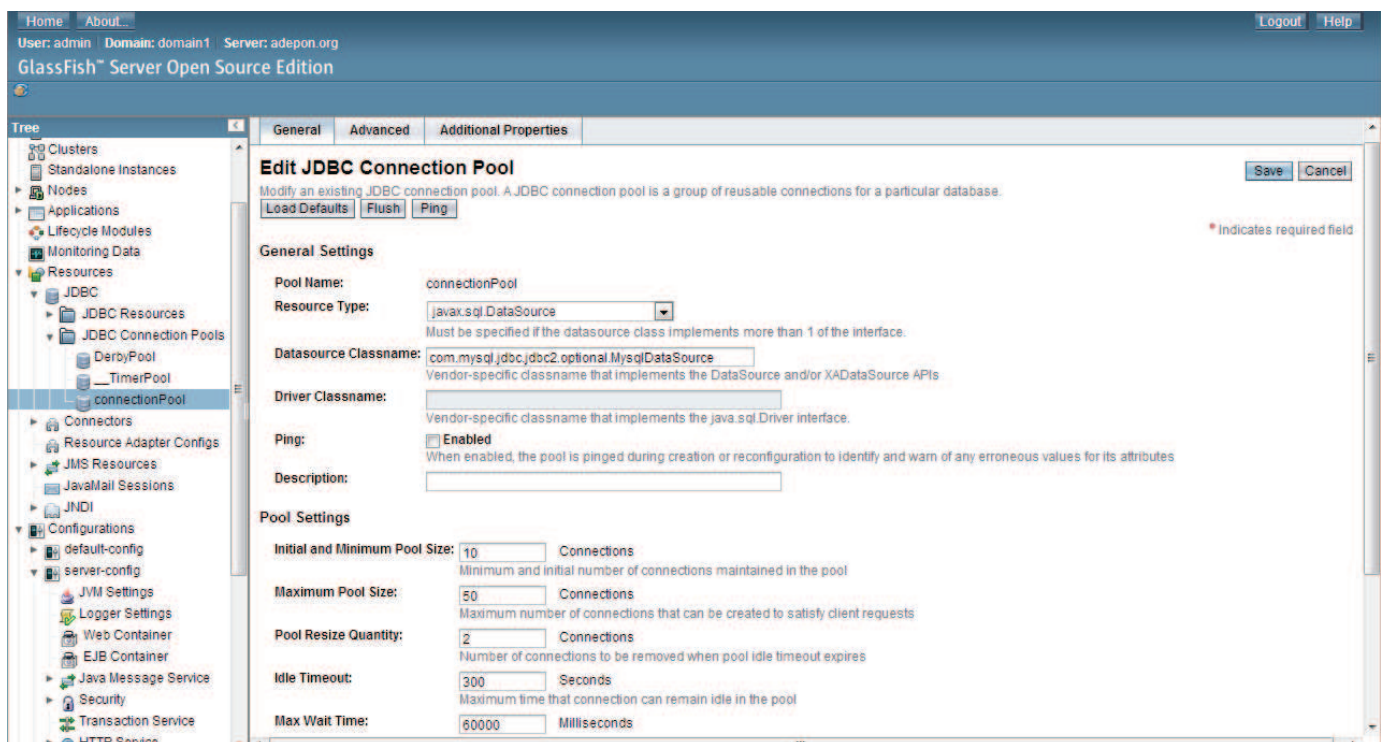


Figura 3.32: Afinamiento Pool de conexiones

Para el proyecto los parámetros fueron definidos de la siguiente manera:

- Initial and minimum pool size: 10 conexiones
- Max pool size: 50 conexiones
- Pool resize quantity: 2 conexiones
- Idle timeout: 300 segundos
- Max wait time: 60000 milisegundos

3.3.1.4 Contenedor de EJBs

El Contenedor de EJBs maneja los EJBs abiertos por un proyecto JEE. Para afinar el contenedor de EJBs es importante ajustar el valor de los siguientes parámetros: “max-pool-size”, “initial and minimum pool size”, “pool resize quantity” y “pool Idle timeout”

Para determinar el número de beans del contenedor de EJBs se siguen los pasos descritos a continuación:

- Acceder a la consola de Glassfish a través de un browser y registrarse en ella con un usuario administrador de Glassfish.
- En el árbol que aparece en la columna izquierda de la consola de administración, expandir el nodo “Configurations” → EJB “Container”.
- Donde se aprecian las variables “max-pool-size”, “initial and minimum pool size”, “pool resize quantity” y “pool Idle timeout”.

Esto se ve en la Figura 3.27.

Para el proyecto los parámetros fueron definidos de la siguiente manera:

- Initial and minimum pool size: 10 beans
- Max pool size: 50 beans
- Pool resize quantity: 8 beans

- Pool Idle timeout: 300 segundos

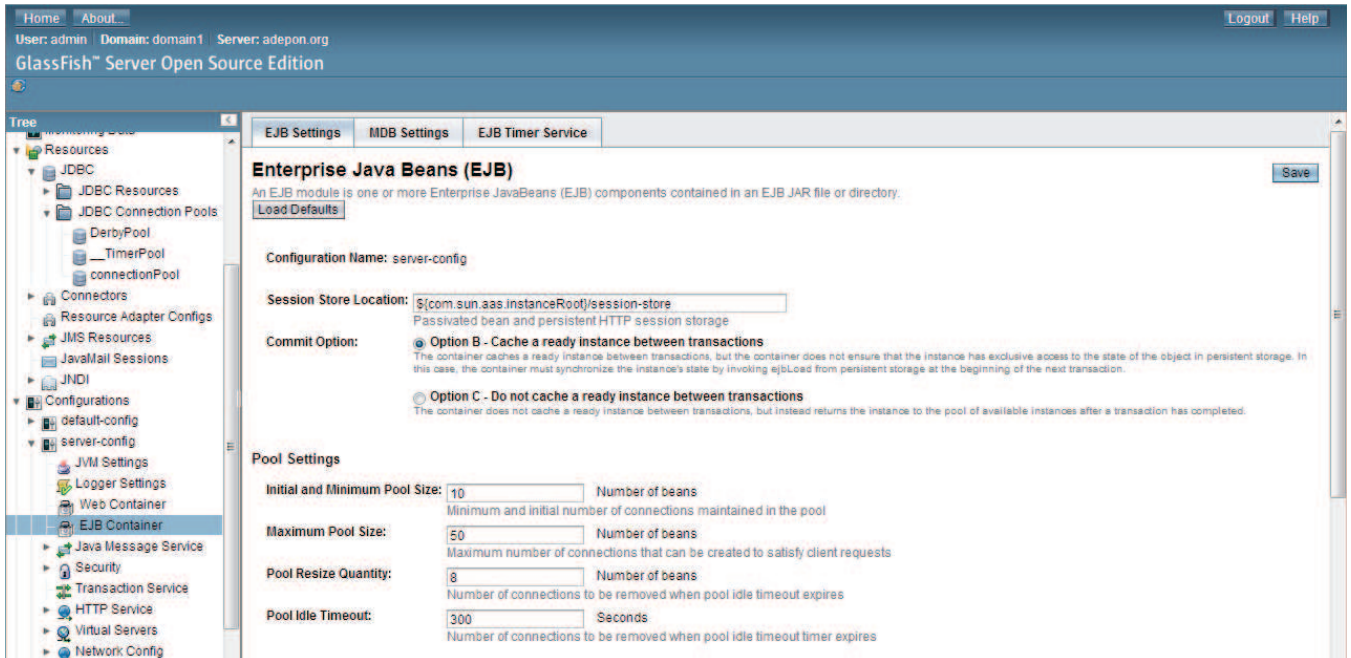


Figura 3.33: Afinamiento Contenedor EJBS

3.3.2 PRUEBAS DE RENDIMIENTO

Para comprobar el rendimiento del servidor, y por ende del sitio web implantado se realizaron varias comprobaciones de tiempo de respuesta y rendimiento ante altos niveles de carga (varias conexiones simultáneas y altos niveles de peticiones por segundo al servidor). Las soluciones disponibles para realizar estas pruebas son muy abundantes, para este proyecto se seleccionó una herramienta web gratuita (<http://loadimpact.com>), que ofrece resultados detallados y que permite monitorear la eficiencia del servidor bajo un nivel de carga relativamente alto.

La primera prueba realizada mide el tiempo de respuesta del servidor cuando se hace una solicitud a la página principal, en la Figura 3.34 se observa el número de sesiones activas (línea en verde) que llegan hasta 50 sesiones activas simultáneas, tiempo de respuesta del servidor (línea azul) que como se observa nunca sobrepasó los 250 ms y el número de

peticiones por segundo. La gráfica muestra que el servidor responde de forma efectiva ante un nivel de carga medio, y su capacidad es suficiente para el fin del mismo.

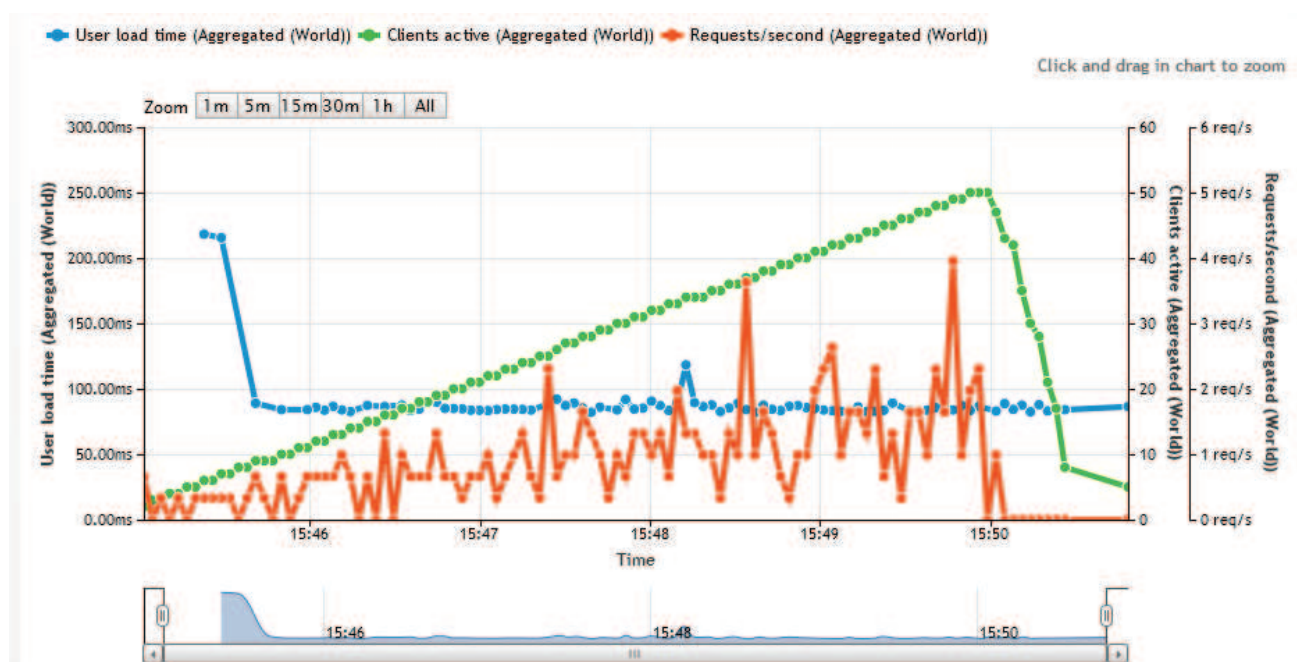


Figura 3.34 Tiempo de respuesta del servidor y conexiones activas.

Se puede concluir que el VPS contratado y el servidor web instalado en el mismo ofrecen un tiempo de respuesta óptimo, incluso contando con varias conexiones activas simultáneas.

La Figura 3.35 muestra el tiempo que tardó en cargarse una de las páginas más pesadas del sitio web (sección noticias) bajo un alto nivel de solicitudes; como se observa, la prueba duró 5 minutos, durante los cuales el servidor recibió un total de 5504 solicitudes con un máximo de 50 sesiones activas simultáneas, durante ese lapso de tiempo se recibieron 358 Mb de información y el tiempo promedio de carga de la página fue de 7.5 s. Hay que señalar que a pesar de parecer un tiempo elevado el servidor estuvo sometido a una carga muy fuerte lo que no significó una baja sensible en su rendimiento, por lo que se puede afirmar que cumple con las expectativas del proyecto.

Load test: Auto-generated (Wednesday, April 10, 2013 4:20:15 PM)

Target URL: <http://www.adepon.org/WEBADEPON-war/news.jsf>

Test result public URL: <http://loadimpact.com/load-test/www.adepon.org-63e8977fc7f>

Load zone data source Aggregated (World)	Status Test finished	User type: VU ? Started: Wed, 10 Apr 2013 22:19:01 Ended: Wed, 10 Apr 2013 22:24:44
---	--------------------------------	--

Choose from which load zone to display data.

VUs active 0 <small>Total number of simulated clients (VU or SBU) active.</small>	Connections active 0 <small>Number of open TCP connections to target system.</small>	Bandwidth 0 bit/s <small>Current throughput to target system.</small>	Data received 385.78 MiB <small>Total number of bytes received during test.</small>	Requests 5504 (0 req/s) <small>Current number of requests per second.</small>
--	---	--	--	--

+ Map and Console

- Charts

Switch view

Add graph

Remove graph

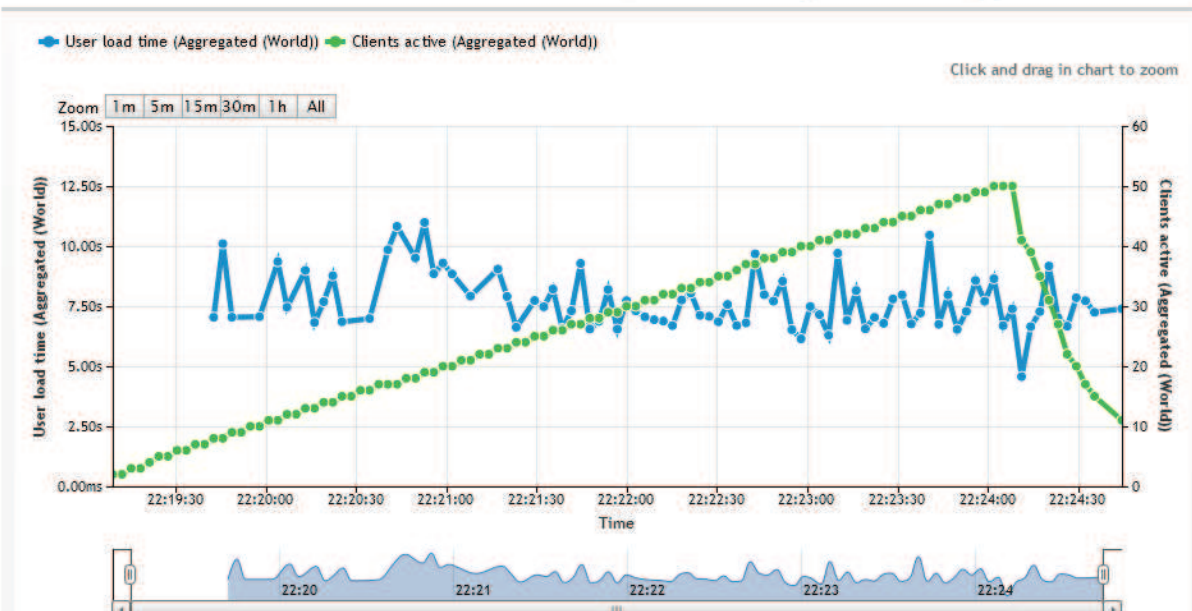


Figura 3.35 Tiempo de carga y conexiones activas.

Se puede concluir que el sitio web de la ADEPON está preparado para soportar un nivel aceptable de usuarios simultáneos sin que se observe una disminución apreciable de rendimiento.

3.3.3 PRESENTACIÓN OFICIAL DEL SITIO WEB

El día viernes 12 de Abril del 2013 a las 12:30 PM se realizó la presentación oficial del sitio web de la ADEPON, en la asamblea general de la ADEPON, realizada en el hemicycle politécnico, ubicado en el edificio de Administración de la Escuela Politécnica Nacional. En la

presentación se explicó a la audiencia el funcionamiento del sitio web, sus módulos y los beneficios que ofrece el sitio web a la ADEPON, luego de lo cual el sistema ha sido puesto en operación.

3.4 ANÁLISIS DE RESULTADOS

Después de implantar el sitio web de la ADEPON y presentárselo a la asamblea general de la asociación, la directiva de la ADEPON expresó su conformidad con el sitio web a través de una carta de aceptación dirigida al Decano de la Facultad de Sistemas y firmada por el Secretario de la ADEPON. La carta se la presenta en el Anexo 2.

Para comprobar el grado de satisfacción de los miembros de la ADEPON con el sitio web de su asociación, se realizó una encuesta en el mismo sitio web, en la que únicamente puedan votar los profesores miembros de la ADEPON (la cual se encuentra en el Anexo 3), sin embargo 4 semanas después de la publicación de la encuesta, únicamente votaron 6 de los 360 profesores registrados (Los resultados de esta encuesta se pueden apreciar en el Anexo 4), motivo por el cual se procedió a realizar la encuesta a usuarios no registrados en el sitio web.

3.4.1 DESCRIPCIÓN DE LA ENCUESTA

La encuesta realizada constó de 5 preguntas, detalladas a continuación:

- a) ¿Resulta sencillo el manejo del Sistema?
- b) ¿Es agradable el diseño del sitio?
- c) ¿Beneficia el Sitio web a la ADEPON?
- d) ¿Considera que la velocidad de respuesta del sitio es aceptable?
- e) ¿Se siente satisfecho con el Sitio?

La encuesta se realizó usando la funcionalidad de formulario dentro de Google Drive para que pudiera ser distribuirla a través de internet.

3.4.2 RESULTADO DE LA ENCUESTA

Se obtuvieron un total de 82 respuestas a la encuesta realizada, los resultados por pregunta son los siguientes:

a) ¿Resulta sencillo el manejo del Sistema?

La figura 43 muestra que 76 personas creen que el sitio es fácil de usar, teniendo un 93% de aceptación.

¿Resulta sencillo el manejo del Sistema?



Figura 36. Usabilidad del sitio.

b) ¿Es agradable el diseño del sitio?

La figura 44 muestra que 62 personas creen que el sitio es agradable, teniendo un 76% de aceptación.

¿Es agradable el diseño del sitio?

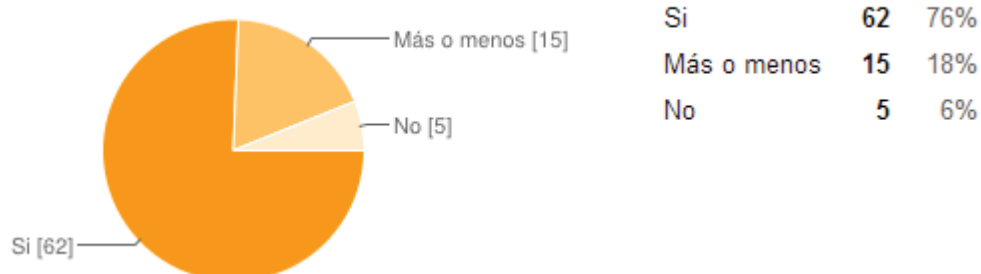


Figura 37. Diseño del Sitio.

c) ¿Beneficia el Sitio web a la ADEPON?

La figura 45 muestra que 74 personas creen que el sitio web beneficiará a la ADEPON, teniendo un 91% de aceptación.

¿Beneficia el Sitio web a la ADEPON?



Figura 38. Beneficio a la ADEPON.

d) ¿Considera que la velocidad de respuesta del sitio es aceptable?

La figura 46 muestra que 67 personas creen que el sitio web tiene un tiempo de respuesta aceptable, teniendo un 82% de aceptación.

¿Considera que la velocidad de respuesta del sitio es aceptable?

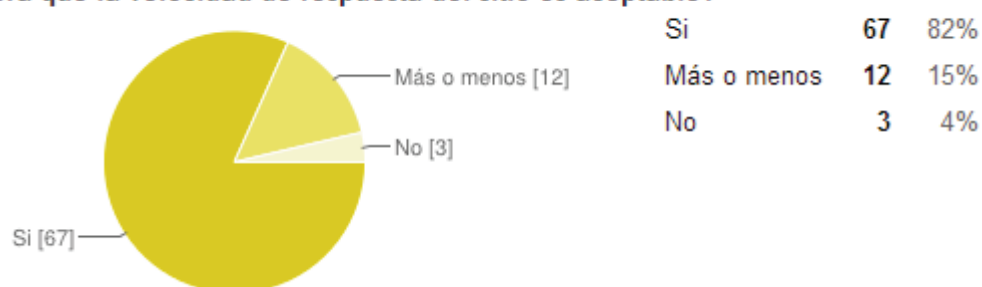


Figura 39. Tiempo de respuesta del sitio.

e) ¿Se siente satisfecho con el Sitio?

Finalmente, la figura 47 muestra que 66 personas se sienten satisfechas con el sitio web, teniendo un 80% de aceptación.

¿Se siente satisfecho con el Sitio?

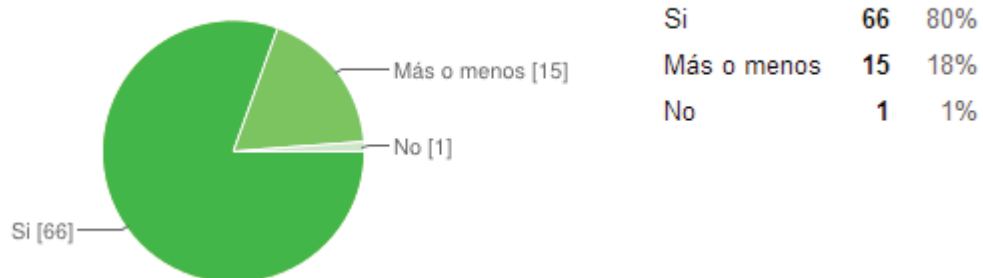


Figura 40. Satisfacción con el sitio.

De los resultados descritos, se puede apreciar un alto grado de aceptación del sitio web, tanto por parte de la directiva de la ADEPON, manifestada en su carta de aceptación, como por parte de los usuarios en general, expresada por medio de la encuesta de satisfacción.

CAPITULO IV

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El proyecto fue exitoso gracias a la apertura que tuvo la ADEPON y al tiempo que dedicaron varios de los miembros de su directiva al proyecto.
- La metodología XP se acopló adecuadamente al desarrollo del sitio Web de la ADEPON, disminuyendo de forma apreciable los tiempos de entrega y facilitando la implementación de los cambios a los requerimientos.
- El utilizar herramientas conocidas e incluso manejadas previamente por los desarrolladores del sitio web en proyectos empresariales, fue vital para el éxito del proyecto y el cumplimiento de los plazos establecidos
- El uso de JSF facilitó el desarrollo del sitio web, ya que permitió el acceso de forma directa desde las vistas del proyecto a las propiedades y métodos de los Managed Beans, encargados del control de la lógica del sistema.
- El uso del framework IceFaces permitió añadir fácilmente características de Aplicaciones de Internet Enriquecidas (RIA) al sitio web.
- El implantar el sitio web en un VPS (Servidor Virtual Privado) hizo posible realizar pruebas en un entorno de producción desde una fase temprana del desarrollo y abarató los costos de implantación del proyecto.

4.2 RECOMENDACIONES

- Se recomienda realizar constantemente entrevistas con los usuarios, cuando se desarrolle un sitio web, para conocer su opinión y realizar a tiempo cualquier cambio a los requerimientos.
- Se recomienda a la ADEPON que designe un administrador del sitio web, con conocimientos medios de Linux y de base de datos MySQL, para garantizar el correcto uso y mantenimiento del sitio web.
- Se recomienda a la ADEPON mantener constantemente actualizada la información del sitio web para promover su uso.
- Se recomienda a la ADEPON incrementar la capacidad del VPS contratado a medida que el sitio web tenga un mayor nivel de uso, y un aumento en el tráfico de datos en el mismo.

GLOSARIO

ADEPON.- Asociación de profesores de la Escuela Politécnica Nacional

EPN.- Escuela Politécnica Nacional

XP.- Programación Extrema

WEB.- sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.

JAVA.- Lenguaje de programación originalmente desarrollado por Sun Microsystems, adquirida por Oracle

JEE.- Java EE plataforma de programación, parte de Java, para desarrollar y ejecutar aplicaciones empresariales de n capas

JSF.- Marco de Trabajo para aplicaciones web en Java EE

RIA.- Aplicación de Internet Enriquecidas

ICEFACES.- Marco de trabajo de código abierto basado en JSF, para el desarrollo de Aplicaciones de Internet Enriquecidas

MYSQL.- Sistema de gestión de bases de datos relacional.

MySQL Workbench.- Herramienta visual de diseño de bases de datos para MySQL

IDE.- Entorno de desarrollo integrado, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica

NetBeans.- Entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java

MVC.- Modelo Vista Controlador, es un patrón de desarrollo de software que separa los datos de una aplicación en tres componentes distintos (modelo, vista y controlador).

XHTML.- eXtensible HyperText Markup Language, HTML con las especificaciones, más estrictas, de XML.

ORM.- mapeo objeto-relacional, es una técnica de programación para convertir datos del tipo utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional

JPA.- Java Persistence API, es la API de persistencia desarrollada para la plataforma Java EE

EJB.- Enterprise JavaBeans, API que forma parte del estándar de construcción de aplicaciones empresariales JEE

API.- Interfaz de programación de aplicaciones, conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software

Framework.- Marco de Trabajo conjunto estandarizado de conceptos, prácticas y criterios para enfrentar un problema

Wizard.- Interfaz de usuario que presenta a un usuario una secuencia de cuadros de diálogo que conducen al usuario a través de una serie de pasos bien definidos.

Operaciones CRUD.- Acrónimo de Crear, Obtener, Actualizar y Borrar, funciones básicas de Bases de Datos

Glassfish.- Servidor de aplicaciones de software libre desarrollado por Sun Microsystems, adquirido después por Oracle.

HTTPS.- Protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto

SHA 256.- Secure Hash Algorithm, Algoritmo de Hash Seguro, es un algoritmo de encriptación.

VPS.- Servidor Virtual Privado, un VPS puede alojar numerosos sitios web sin los gastos derivados de tener que adquirir su propio servidor físico independiente.

Hosting.- Alojamiento web, es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

DNS.- Sistema de nombres de dominio, es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada

BIBLIOGRAFÍA

- [1] ADEPON, ESTATUTO DE LA ASOCIACIÓN DE PROFESORES DE LA ESQUELA POLITÉCNICA NACIONAL, 2010.
- [2] MAURER Frank, MARTEL Sebastien, Extreme Programming Rapid Development for Web-Based Applications, 2002.
- [3] ORACLE, The Java EE 6 Tutorial, <http://docs.oracle.com/javaee/6/tutorial/doc/bnaaw.html>, enero 2013.
- [4] PRICEWATERHOUSECOOPERS, Global 100 Software Leaders http://www.pwc.com/en_GX/gx/technology/publications/global-software-100-leaders/assets/global-software-100.pdf, 2010.
- [5] DÍAZ Francisco J., QUEIRUGA Claudia A., FAVA Laura A., Struts y JavaServer Faces, cara a cara, http://sedici.unlp.edu.ar/bitstream/handle/10915/20451/Documento_completo.pdf?sequence=1, enero 2013.
- [6] ICEFACES, ICEfaces Overview <http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>, abril 2013.
- [7] MYSQL, <http://dev.mysql.com/doc/refman/5.0/en/features.html>, Febrero del 2013.
- [8] GARCÍA Mercedes, Desarrollo e implantación del sistema de administración estudiantil vía web para la unidad educativa particular trilingüe “helmut walhmüller”, 2012.
- [9] CORTIZO José Carlos, EXPÓSITO Diego, RUIZ Miguel, eXtreme Programming, 2011.
- [10] BECK Kent, Extreme Programming Explained. Segunda Edición, 2004

[11] CHEON Yoonsik, LEAVENS Gary T. A Simple and Practical Approach to Unit Testing: The JML and JUnit Way, 2001.

[12] GODADDY, Virtual Private Servers, <http://www.godaddy.com/hosting/vps-hosting.aspx?isc=gtngflam30&ci=9013> Enero 2013.

[13] SUN MICROSYSTEMS, Optimize GlassFish Performance in a Production Environment, 2009.

ANEXOS

ANEXO 1: Pruebas Unitarias (Anexo Digital)

ANEXO 2: Certificado de la Institución (Anexo Digital)

ANEXO 3: Encuesta (Anexo Digital)

ANEXO 4: Resultados de la Encuesta a los profesores miembros de la ADEPON (Anexo Digital)

ANEXO 5: Manual de Usuario (Anexo Digital)