

Universidad de las Ciencias Informáticas

Facultad 3



Panel de administración para el marco de trabajo Sauxe.

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: Adrián Ávila Atencio

Tutor(es):

Ing. Juan José Rosales Rodríguez

Ing. Yanet Pérez Valcárcel

Ing. Ana I Ferrer Hernández

DECLARACIÓN DE AUTORÍA

Declaramos que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

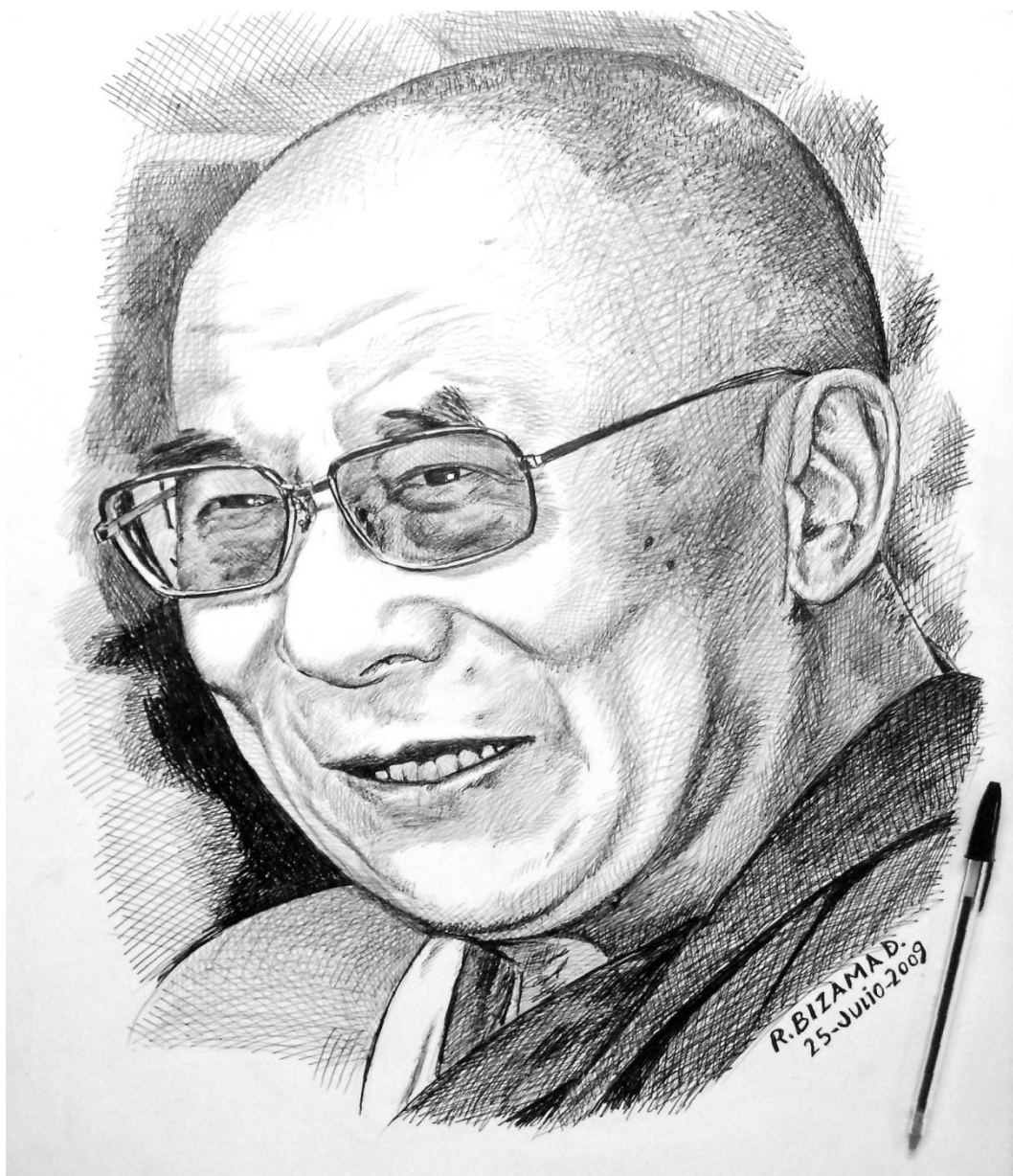
Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Autor: Adrián Ávila Atencio

Tutor: Ing. Juan José Rosales Rodríguez

Tutor: Ing. Ana I Ferrer Hernández

Tutor: Ing. Yanet Pérez Valcárcel



"Solo existen dos días en el año en que no se puede hacer nada. Uno se llama ayer y otro mañana..."

Tenzin Gyatso Dalai Lama

Agradecimientos

Muchas personas me han apoyado tanto en la realización de este trabajo como en los 5 años de la carrera. Muchas gracias a todos.

A mis padres por tener toda confianza en mí y ayudarme en todo lo que estuviera a su alcance, sin ellos hubiera sido muy difícil terminar la carrera.

A mi novia que estuvo desde el comienzo de este trabajo brindándome su apoyo.

A mis tutores les agradezco pues sin su ayuda no hubiese sido posible la terminación de este trabajo, especialmente a Pepe, que más que un tutor es un amigo con el cual siempre se puede contar, gracias chama.

A las profesoras Denia, Yalice, Daileny por siempre ofrecerme su ayuda, Neysi, Chakal, a Eiler, Juan y Carlos por coordinarme a cada rato con algunas líneas de código.

Al team del aula con los cuales he pasado varios años de estudio y a todos mis compañeros de clases.

DEDICATORIA

Dedico este trabajo a mis padres, a mis abuelos, a mi hermano y todas aquellas personas que me apoyan.

RESUMEN

En el Centro de Informatización para la Gestión de Entidades de la Universidad de las Ciencias Informáticas se desarrolló el marco de trabajo Sauxe. La presente investigación constituye una propuesta para el desarrollo de una herramienta que permita al Sauxe realizar de forma visual la administración de las configuraciones del sistema en general. Actualmente en el marco de trabajo Sauxe no se está gestionando correctamente los elementos de configuración relacionados a la conexión a la base de datos, correo, mensajería y de caché, provocando que los desarrolladores tengan que realizar trabajos engorrosos y estén propensos a cometer errores durante el proceso de configuración. El desarrollo de la propuesta está previamente respaldado por un estudio sobre los principales conceptos tratados en la investigación y de los sistemas existentes relacionados con la administración.

En este documento se presenta el análisis y diseño de la herramienta desarrollada así como sus principales funcionalidades. Además se presenta el sistema, como una aplicación web guiada por las políticas de desarrollo del proyecto: de código abierto y multiplataforma. Se describen las pruebas de caja blanca y caja negra.

PALABRAS CLAVES

Marco de trabajo Sauxe, Administración, Configuración, Multiplataforma.

Índice de contenidos

Agradecimientos	IV
Dedicatoria.....	V
Resumen	VI
Índice de contenidos	VII
Introducción	1
Capítulo 1: FUNDAMENTACIÓN TEÓRICA.....	6
Introducción	6
1.1 Conceptos asociados al dominio del problema	6
1.1.1 Aplicación Web	6
1.1.2 Aplicación Web de gestión.....	6
1.1.3 Administración	7
1.1.4 Configuración informática	8
1.1.5 Seguridad informática	8
1.1.6 Usabilidad.....	8
1.1.7 Marcos de trabajo (Framework)	9
1.1.8 Marco de trabajo Sauxe	10
1.1.8.1 Componente de configuración actualmente independiente.	11
1.2 Administración y configuración en aplicaciones para el desarrollo Web.....	11
1.2.1 Administración en Drupal.....	11
1.2.2 Administración en Joomla.....	13
1.3 Modelo de desarrollo orientado a componentes.....	15
1.3.1. Modelo de ciclo de vida de los proyectos del CEIGE.....	16
1.3.2. Descripción de las fases del ciclo de vida de los proyectos.....	17
1.3.3. Ciclo de vida de proyectos del CEIGE.....	18
1.4 Tecnologías utilizadas para el desarrollo	18
1.4.1. Lenguajes de programación.....	19
1.4.2. Librerías y marcos de trabajo.....	21
1.4.3. Herramientas de desarrollo.....	24
1.5 Patrones de Arquitectura	29
1.6 Pruebas de software	30
1.6.1. Objetivos de la prueba	31

1.6.2.	Pruebas de Caja Negra y Caja Blanca.....	31
1.6.3.	Justificación de la selección de las pruebas.....	31
1.7.	Conclusiones	32
Capítulo 2:	ANÁLISIS Y DISEÑO.....	33
2.1.	Modelo conceptual.....	33
2.2.	Descripción de los procesos del negocio	34
2.2.1.	Descripción del proceso: Configuración de archivos del sistema.	34
2.3.	Especificación de requisitos.....	35
2.3.1.	Requisitos funcionales.....	35
	Especificación del requisito Modificar configuración de conexión a la base de datos.	35
2.3.2.	Requisitos no funcionales	36
2.4.	Patrones de diseño	38
2.4.1.	GRASP.....	38
2.5.	Modelo de diseño.....	39
2.6.	Conclusiones	42
Capítulo 3:	IMPLEMENTACIÓN, PRUEBAS Y VALIDACIÓN DE LA SOLUCIÓN.....	43
3.1.	Estructura del Marco de Trabajo	43
3.2.	Diagrama de despliegue	45
3.3.	Estándares de codificación	46
3.4.	Métricas de software.....	49
3.5.	Pruebas de caja blanca.....	56
3.6.	Pruebas de caja negra.....	58
3.7.	Validación de la solución.....	61
3.8.	Conclusiones	65
Conclusiones generales.....		66
Recomendaciones		67
Referencia Bibliográfica		68

Índice de figuras

Figura 1: Configuración de la base de datos en Drupal.	12
Figura 2: Configuración de las Actualizaciones en Drupal.	13
Figura 3: Configuración de la base de datos en Joomla.	14
Figura: 4 Fases del ciclo de vida de proyectos del CEIGE.....	17
Figura 5: Ciclo de vida proyectos CEIGE.....	18
Figura 6: Modelo conceptual.	33
Figura 7: Descripción del Proceso de configuración de archivos del sistema.	34
Figura 8: Prototipo de interfaz de configuración de conexión a la base de datos.	36
Figura 9: Diagrama de clase de la solución.	40
Figura 10: Diagrama de secuencia: Configurar mensajería.	42
Figura 11: Carpeta correspondiente al Subsistema Administrador.	43
Figura 12: Paquete administrador correspondiente a la carpeta de la aplicación.....	44
Figura 13: Paquete models correspondiente a la carpeta de la aplicación.....	44
Figura 14: Paquete views correspondiente a la carpeta de la aplicación.	44
Figura 15: Carpeta de diseño correspondiente al Subsistema Administrador.	45
Figura 16: Paquete views correspondiente a la carpeta de diseño.	45
Figura 17: Diagrama de despliegue.....	45
Figura 18: Descripción de las clases.	47
Figura 19: Ejemplo de comentario en una clase.	48
Figura 20: Descripción de las funciones.	48
Figura 21: Ejemplo de comentario en una clase para describir una función.	48
Figura 22: Ejemplo de comentario por líneas de código.	48
Figura 23: Representación del número de clases por procedimientos de la métrica TOC.	52
Figura 24: Resultados de la evaluación de la métrica TOC para el atributo responsabilidad.	53
Figura 25: Resultados de la evaluación de la métrica TOC para el atributo complejidad.	53
Figura 26: Resultados de la evaluación de la métrica TOC para el atributo reutilización.	54
Figura 27: Representación del número de dependencias de las clases de la métrica RC.	54
Figura 28: Resultados de la evaluación de la métrica RC para el atributo acoplamiento.	55
Figura 29: Resultados de la evaluación de la métrica RC para el atributo Reutilización.	55
Figura 30: Código fuente de la funcionalidad cargarStoreAction.....	56

Figura 31: Grafo del flujo asociado a la funcionalidad cargarStoreAction.	56
Figura 32: Evaluaciones por categorías.....	62
Figura 33: Estructura del marco de trabajo Sauxe.	¡Error! Marcador no definido.
Figura 34: Configuración de la caché en Joomla.	¡Error! Marcador no definido.
Figura 35: Configuración de la caché en Drupal.	¡Error! Marcador no definido.
Figura 36: Opciones avanzadas de la configuración de la base de datos en Drupal.....	¡Error! Marcador no definido.
Figura 37: Distribución de Chi Cuadrado de Pearson.	¡Error! Marcador no definido.

Índice de tablas

Tabla 1: Descripción del requisito funcional Modificar configuración de conexión a la base de datos.	35
Tabla 2: Descripción de clases del diseño.	40
Tabla 3: Atributos de calidad evaluados por la métrica TOC.	50
Tabla 4: Criterios de evaluación para la métrica TOC.	50
Tabla 5: Atributos de calidad evaluados por la métrica RC.	51
Tabla 6: Criterios de evaluación para la métrica RC.	51
Tabla 7: Caso de prueba Modificar elementos de configuración de mensajería.	58
Tabla 8: Descripción de variables.	59
Tabla 9: Resultados de la prueba por requisitos funcionales.	60
Tabla 10: Resultados de la prueba de NC.	60
Tabla 11: Evaluación por preguntas de los expertos.	62
Tabla 12: Datos obtenidos de la encuesta.	64
Tabla 13: Cuestionario aplicado en la etapa de validación de la solución. ¡Error! Marcador no definido.	

INTRODUCCIÓN

Con el surgimiento de las Tecnologías de la Información y las Comunicaciones (TICs), ha evolucionado la forma de vida y el pensar del ser humano; dando un paso agigantado hacia el desarrollo en todas las esferas de la sociedad; prestando mayor atención a los desafíos que se impone el hombre para adaptarse a las necesidades tecnológicas del momento.

Con esta nueva era tecnológica, aparecen nuevas herramientas que ofrecen la posibilidad de encontrar soluciones novedosas a dichos desafíos. Las aplicaciones dirigidas a la plataforma web no están exentas a dicho impacto, pues son el reflejo directo de los cambios y contradicciones de la sociedad. Sus potencialidades como el gran poder de difusión debido al alcance que poseen, así como la variedad de usuarios que las utilizan, hacen que este tipo de software sea una herramienta de alta capacidad y de uso en diversos campos; entre los que se pueden señalar la planificación económica y la gestión de datos.

La utilización de estas aplicaciones en el sector empresarial, garantizan disminuir la complejidad y el tiempo de ejecución de muchos de sus procesos, además de las facilidades de adaptabilidad que posee la web ante el constante cambio al que está sometido.

Las aplicaciones web de gestión, debido a su enfoque directo a los negocios y las empresas, pretenden brindar a sus clientes un servicio de mayor calidad, seguridad y plena disponibilidad para su utilización. Muchas de estas aplicaciones son implementadas sobre marcos de trabajo, con el fin de optimizar tiempo y reutilizar soluciones. Los marcos de trabajo son estructuras tecnológicas que dan soporte para que otras aplicaciones puedan ser desarrolladas; los utilizados para desarrollar aplicaciones web, poseen gran demanda en la industria del software.

Nuestro país, inmerso en los nuevos avances tecnológicos y en función del desarrollo de nuestra sociedad, la producción y la optimización de los procesos empresariales, encontró necesario desarrollar sus propias aplicaciones informáticas libre de licencias; con el fin de lograr mejores resultados y hacer aportes a la economía nacional con la implantación de sistemas que gestionen sus procesos, tareas y personal de forma eficiente y segura. Estos productos informáticos deben ser adaptables a las particularidades económicas existentes, así como estar en constante actualización e intercambio con otros sistemas si es requerido.

Actualmente, existe el marco de trabajo para aplicaciones web: Sauxe, desarrollado por un equipo de desarrollo de software compuesto por estudiantes y profesores del Centro de Gestión de Entidades (CEIGE), de la Universidad de la Ciencias Informáticas (UCI). Sauxe, permite el desarrollo de aplicaciones web de gestión, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo.

Este marco de trabajo cuenta hoy día con una serie de herramientas para la realización del mantenimiento del sistema en general. A las cuales se accede de forma independiente. Aunque se han detectado problemáticas a la hora de modificar los elementos de configuración de Sauxe como la conexión a los servidores de BD, mensajería, correo, actualizaciones y de caché las cuales se hacen actualmente editando los ficheros de configuración de forma manual, provocando de esta forma un alto riesgo durante el proceso de configuración y mantenimiento, además de lo engorroso que se torna este proceso.

Igualmente se ha detectado la posible inclusión de nuevas funcionalidades que ayuden en el desempeño del sistema tales como: Elaboración de un componente que gestione las medias del sistema (imágenes, videos, documentos) que sea transparente al tipo de almacenamiento que se está utilizando.

De lo anteriormente planteado se deriva el siguiente **problema a resolver**: ¿Cómo mejorar los procesos de administración del marco de trabajo Sauxe para facilitar la usabilidad y seguridad en la configuración del sistema?

Para la solución del problema planteado se define como **objeto de estudio** los marcos de trabajo para aplicaciones web de gestión.

A modo de dar solución al problema anterior se traza como **objetivo general**: implementar una herramienta que facilite la usabilidad y seguridad durante la configuración del sistema en los procesos de administración del marco de trabajo Sauxe.

Fueron trazados para dar cumplimiento al objetivo general los siguientes **objetivos específicos**:

- Realizar el marco teórico de la investigación.
- Realizar la especificación de requisitos.
- Realizar el análisis y diseño de solución.

- Implementar la solución
- Validar la solución aplicando pruebas, métricas de software y el criterio de expertos.

Como **campo de acción** se definió el marco de trabajo Sauxe y como **idea a defender**: con el desarrollo de una herramienta que facilite la usabilidad y seguridad durante la configuración del sistema se mejorarán los procesos de administración del marco de trabajo Sauxe.

Tareas a realizar:

1. Realización del estado del arte de la investigación.
2. Estudio y análisis del marco de trabajo de Sauxe.
3. Estudio y análisis de las tecnologías utilizadas para el desarrollo de sistemas de gestión web y de gestión de administración.
4. Estudio y análisis de la metodología o modelo de desarrollo a utilizar para la elaboración de la solución.
5. Implementación de las funcionalidades que permitan la configuración del entorno de Sauxe (permitir el cambio de los diferentes entornos de Sauxe tales como: producción, desarrollo, prueba).
6. Implementación de las funcionalidades de configuración del estado de Sauxe (poner Sauxe on/off line)
7. Implementación de las funcionalidades de configuración al servidor de Correo (permite configurar los aspectos de acceso al servidor de correo)
8. Implementación de las funcionalidades de configuración del servicio de mensajería XMPP.
9. Implementación de las funcionalidades de configuración de las actualizaciones de Cedrux (hacerlas automáticas con hora de inicio de las actualizaciones, frecuencia de las mismas).
10. Implementación de las funcionalidades de configuración de los parámetros de la conexión a la BD (modules config).
11. Implementación de las funcionalidades de configuración del servidor de caché.
12. Implementación de las funcionalidades que permitan la integración de todas las herramientas en el panel de administración.
13. Realización de los diagramas de clases del análisis.
14. Realización de los diagramas de clases del diseño.

15. Realización de las pruebas de calidad correspondientes para el panel de administración.

Entre los métodos empleados para la investigación científica están:

Métodos teóricos:

- **Analítico – Sintético:** posibilitará realizar un estudio teórico de la investigación permitiendo el análisis de documentos y la extracción de elementos importantes acerca del proceso de actualización de aplicaciones. Se utilizará para la precisión de las características del diseño propuesto, así como para el arribo de las conclusiones de la investigación.
- **Histórico – Lógico:** viabilizará el análisis de las características de las herramientas utilizadas y las tendencias evolutivas actuales de su uso en los proyectos productivos de la universidad, así como la definición de los antecedentes de las mismas y la vinculación con tecnologías confinadas para el desarrollo del componente.
- **Modelación:** proporcionará la realización de los diagramas necesarios en el proceso de desarrollo de software, al hacer una representación abstracta de la solución y facilitar así el desarrollo de la misma.

Métodos empíricos:

- **Observación:** permite un mejor entendimiento, comprensión y caracterización del proceso de actualización de aplicaciones.
- **Medición:** a través del uso de métricas de calidad y pruebas para comprobar la funcionalidad, usabilidad y seguridad, entre otros requerimientos, se podrá asegurar la obtención de un producto con calidad.

Con la investigación se pretende obtener una herramienta para administrar las configuraciones del sistema del marco de trabajo Sauxe que integre los componentes de configuración ya desarrollados y los nuevos a implementar. La investigación está estructurada en tres capítulos de la siguiente forma:

Capítulo 1: Fundamentación teórica.

Serán descritos los principales conceptos relacionados con el tema del trabajo. Se realizará un estudio del estado del arte que incluye la disertación de las aplicaciones que serán utilizadas, las tecnologías y metodologías, haciendo un análisis de sus características, ventajas y desventajas, con el propósito de obtener base y conocimientos para proponer la mejor solución al problema que se plantea.

Capítulo 2: Análisis y diseño.

El capítulo se centra en definir los temas de análisis y diseño de la solución, donde se plantean las características de la herramienta para su posterior implementación. Se examinarán cada uno de los requisitos funcionales y no funcionales, prototipos de interfaces, modelos, y diagramas que serán la guía para la siguiente etapa de desarrollo.

Capítulo 3: Implementación y prueba.

En este capítulo se expondrá una explicación de cómo se lleva a cabo el proceso para la implementación de la solución y se validará la solución desarrollada mediante la realización de métricas para el diseño, pruebas de caja blanca y negra para la implementación, las cuales garantizarán la calidad de la solución.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En este primer capítulo se abordan algunos conceptos de utilidad para comprender mejor el problema que se plantea, se realiza un análisis de las principales características de las aplicaciones web, haciendo énfasis en las aplicaciones web de gestión. Se realiza también, un estudio detallado sobre del marco de trabajo Sauxe y sus componentes de configuración actualmente independientes.

Un tema de suma importancia en este capítulo es el proceso de administración y configuración en aplicaciones web de gestión, tema medular para la realización del panel de administración del Sauxe. Así como la ingeniería de requisitos, la cual es imprescindible para identificar las nuevas funcionalidades a desarrollar. Además se describen las tecnologías que serán utilizadas durante la implementación.

1.1 Conceptos asociados al dominio del problema

1.1.1 Aplicación Web

Con el surgimiento de la www (world wide web) aparecen las aplicaciones web, estas se desarrollan como una extensión de los sistemas web, para agregarle funcionalidades del negocio al proceso. En otros términos, una aplicación web es un sistema web que permite a los usuarios ejecutar lógica de negocio usando un navegador, o de igual forma: modificar el estado del negocio. Su arquitectura general es la de Cliente – Servidor, donde el protocolo principal de comunicación es HTTP, el cual funciona normalmente desconectado, es decir, el cliente realiza la petición al servidor, este la procesa y envía el resultado de la petición (1).

1.1.2 Aplicación Web de gestión

Las aplicaciones web de gestión permiten trabajar en un entorno consolidado y centralizado, con acceso seguro e inmediato a los datos y a las tareas de interés para la empresa, usando una combinación de CMS, CRM y ERP.

CMS (Content Management System): Sistema gestor de contenidos web, o aplicación automatizada que permite gestionar los contenidos de la web.

CRM (Customer Relationship Management): Gerencia de relaciones con los clientes. Mediante el software CRM se logra conocer más a fondo las necesidades y preferencias de los clientes, y de esta forma poder ofrecerles un producto con mayor valor agregado dependiendo de cada usuario.

ERP (Enterprise Resource Planning): Sistema que permite gestionar los recursos empresariales (recursos humanos, proveedores, recursos de capital, materias primas, procesos internos, entre otros).

Resumiendo, una aplicación web de gestión es una aplicación web utilizada en la gestión de los procesos de negocio, donde se vinculan otros componentes como los sistemas de gestión de contenidos web (CMS), gestión de contenido de gerencia de relaciones con los clientes (CRM) y el sistema de gestión de recursos empresariales (ERP). Estas aplicaciones están muy relacionadas con los marcos de trabajo (2).

1.1.3 Administración

Según la definición de varios autores la administración es la actividad de planeación, organización, ejecución y control, desempeñada para determinar y alcanzar los objetivos con el uso de seres humanos y otros recursos (materiales y económicos) (3).

En el proceso de construcción de un software el concepto de administración es un elemento trascendental para su adecuado funcionamiento, pues facilita que el sistema funcione de una manera ordenada, segura y consistente.

Dentro de un sistema informático la administración se compone por un módulo o subsistema que proporciona diferentes funcionalidades administrativas que permiten:

- Controlar las operaciones que garanticen la seguridad.
- Definir las estructuras lógicas con el fin de tener una buena organización del sistema.
- Gestionar las configuraciones globales o específicas de los negocios que integra.
- Gestión de usuarios y estaciones de trabajo.

1.1.4 Configuración informática

La configuración informática es el grupo de datos e información que caracterizan diferentes elementos de una computadora, como son programas, aplicaciones o elementos de hardware y software. Se conforma por el conjunto de artefactos y programas que constituyen un sistema informático.

Partiendo del concepto anterior es preciso considerar entonces que la configuración informática es la manera de adaptar una aplicación de software al resto de los componentes del entorno de desarrollo y a las necesidades del usuario. Por tal motivo, es una tarea esencial antes de trabajar con un nuevo sistema informatizado.

La tendencia que existe actualmente para reducir las necesidades de configuración es hacer uso de sistemas que permitan detectar en que entorno se va a instalar el nuevo elemento, configurándose automáticamente sin la necesidad de requerir la participación del usuario y en caso de ser necesaria se intenta facilitar al máximo el proceso de configuración (4).

1.1.5 Seguridad informática

La seguridad de software es un tema que, junto a muchos otros, se trata como algo engorroso y su definición se maneja con cierto grado de incertidumbre. Según el diccionario de la Real Academia Española de la Lengua (RAE), el término seguridad se refiere a la certeza o garantía de que algo va a cumplirse; mecanismo que previene algún riesgo o asegura el buen funcionamiento de alguna cosa, precaviendo que falle. Si se adapta el concepto al campo de la informática se puede definir que la Seguridad Informática generalmente consiste en asegurar que los recursos tecnológicos y activos tangibles e intangibles de un sistema de información de una organización sean utilizados de la manera que se estableció.

Las tendencias actuales en el mundo de la seguridad de software establecen que “El objetivo de la Seguridad Informática será mantener la Integridad, Disponibilidad, y Confidencialidad de la información manejada por la computadora” (5).

1.1.6 Usabilidad

Numerosos autores han propuesto diversas definiciones de usabilidad en dependencia de los diferentes atributos o factores mediante los que puede ser evaluada. Se seleccionó para este

trabajo la definición más extendida, que es la ofrecida por la ISO, y que define usabilidad como el " grado de eficacia, eficiencia y satisfacción con la que usuarios específicos pueden lograr objetivos específicos, en contextos de uso específicos".

La usabilidad se compone de dos tipos de atributos:

- Atributos cuantificables de forma objetiva: como son la eficacia o número de errores cometidos por el usuario durante la realización de una tarea, y eficiencia o tiempo empleado por el usuario para la consecución de una tarea.
- Atributos cuantificables de forma subjetiva: como es la satisfacción de uso, medible a través de la interrogación al usuario, y que tiene una estrecha relación con el concepto de Usabilidad Percibida (6).

1.1.7 Marcos de trabajo (Framework)

Un framework simplifica el desarrollo de una aplicación mediante la informatización de algunos de los patrones utilizados para resolver las tareas comunes. Además proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y fácil de mantener. Por último, facilita la programación de aplicaciones, permitiendo que se encapsulen operaciones complejas en instrucciones sencillas (7).

Características principales de los marcos de trabajo

A continuación se enuncian una serie de características que poseen la mayoría de los marcos de trabajo existentes (8):

- **Abstracción de URLs y sesiones:** No es necesario manipular directamente las URLs ni las sesiones, el marco de trabajo ya se encarga de hacerlo.
- **Acceso a datos:** Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en bases de datos, XML, etc.
- **Controladores:** La mayoría de los marcos de trabajo implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.

- **Autenticación y control de acceso:** Incluyen mecanismos para la identificación de usuarios mediante *nombre de usuario* y *contraseña* lo que permite restringir el acceso a determinadas páginas de determinados usuarios.
- **Separación entre diseño y contenido:** La mayoría de los marcos de trabajo implementan el patrón MVC (modelo - vista - controlador). Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

1.1.8 Marco de trabajo Sauxe

A partir de la extensión de algunos componentes de Zend Framework surge ZendExt, desarrollado por el Departamento de Tecnología y la UCID (Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa), con el objetivo de crear un marco de trabajo extensible y configurable, centrando el desarrollo de las aplicaciones en la lógica del negocio, en las interfaces de usuario, alejando a los programadores de los detalles arquitectónicos, con soporte para entornos multientidad y una arquitectura de sistema orientada a componentes.

La unión de ZendExt, Doctrine y ExtJs dio lugar al marco de trabajo Sauxe que contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo.

Sauxe está desarrollado sobre lenguaje de programación PHP y en la capa de presentación utiliza Java script y HTML. Implementa una arquitectura en capas aunque en una de sus capas contenga un modelo vista controlador. En la capa de presentación utiliza ExtJs por la gran gama de componentes que se pueden reutilizar para agilizar el proceso de desarrollo y mostrarle al usuario una interfaz más amigable y funcional. En la capa de negocio utiliza ZendFramework por su nivel de flexibilidad en la integración con la capa superior, inferior y vertical de la arquitectura, otra de las ventajas que brinda es la sencillez a la hora de extender sus componentes, como es el caso de ZendExt (Figura 33 de los Anexos).

1.1.8.1 Componente de configuración actualmente independiente.

La herramienta de actualización consiste en un componente que a través de un fichero compactado con el contenido a actualizar incorpora los cambios a la aplicación, agregándole una versión interna para llevar el control de la Entrega del software. El fichero que contiene los cambios debe cumplir una estructura determinada para poder realizar correctamente el proceso de actualización (9).

El funcionamiento básico del Actualizador radica en 5 requisitos funcionales:

- Establecer conexión: Consiste en permitir la conexión a la base de datos de la aplicación para poder introducir los cambios. Se introduce usuario y contraseña de la base de datos específica de la aplicación que se quiere actualizar.
- Cargar archivo: Consiste en importar el fichero compactado que contiene los nuevos cambios. Se puede importar convencionalmente mediante el fichero .zip o a través de la acción de buscar en línea las actualizaciones ubicadas en un ftp remoto, siempre verificando que sea mayor que la versión instalada.
- Visualizar subsistemas a cambiar: se listan todos los subsistemas a cambiar.
- Actualizar: Consiste a partir de la versión y los cambios a introducir actualizar finalmente la aplicación.
- Cancelar: Permite volver acceder al sistema Cedrux.

1.2 Administración y configuración en aplicaciones para el desarrollo Web.

Los CMS cuentan con una serie de funcionalidades, entre las que se puede señalar, la Administración de la estructura del portal, en donde se realiza la configuración general del sistema. Por tanto se decidió realizar un análisis de los componentes de administración de los CMS genéricos disponibles como Drupal y Joomla. Estos CMS tienen en común que están desarrollados en PHP, MySQL y que se distribuyen como software libre.

1.2.1 Administración en Drupal

La administración en Drupal se lleva a cabo a través del menú de Administración. El menú de Administración se divide en grupos de tareas, teniendo inicialmente las siguientes opciones

principales: Panel de control, Contenido, Estructura, Apariencia, Personas, Módulos, Configuración, Informes, Ayuda.

Drupal cuenta con un archivo que contiene la información necesaria para poder establecer la conexión a la Base de datos, en este archivo se encuentra la sección correspondiente en donde se establece la información necesaria para realizar la configuración de conexión a la base de datos. Además cuenta con opciones adicionales de configuración para la conexión a la base de datos (Figura 36 de los Anexos).

Configuración de la base de datos



✓ Elegir perfil
✓ Elegir un idioma
✓ Verificar requisitos

► Configurar base de datos

Instalar perfil
Configurar traducciones
Configurar sitio
Terminar traducciones
Terminado

Tipo de base de datos *

MySQL, MariaDB o equivalente
 SQLite

El tipo de base de datos donde se almacenarán los datos de Drupal.

Nombre de la base de datos *

bd_curso

El nombre de la base de datos donde se almacenarán los datos de Drupal. Debe existir en el servidor antes de instalar Drupal.

Nombre de usuario de la base de datos *

usuario_curso

Contraseña de la base de datos

► OPCIONES AVANZADAS

Guardar y continuar

Figura 1: Configuración de la base de datos en Drupal.

Las opciones de configuración de la caché que se podrá activar en la configuración de Drupal son (Figura 35 de los Anexos):

- **Caché de páginas para usuarios anónimos.** Almacena en caché las páginas tal y como se mostrarán a los usuarios anónimos.
- **Caché de bloques.** El funcionamiento es similar a la caché de páginas, pero para los bloques del sitio.
- **Mínimo de permanencia en caché.** Es el tiempo mínimo que tiene que pasar una página en caché para que sea renovada por el sistema.
- **Caducidad de las páginas en caché.** Es el tiempo máximo que pasará una página en la caché para luego ser renovada por el sistema.

- **Reunir y comprimir archivos CSS.** Comprime los archivos CSS descargados por los usuarios al cargar una página del sitio.
- **Juntar los archivos JavaScript.** Con esta opción el sistema comprimirá y unificará los archivos JavaScript.
- **Vaciar todas las cachés** permite limpiar y actualizar la caché completa.

Drupal configura sus Actualizaciones activando la opción "**Verificar actualizaciones automáticamente**" el sistema comprueba periódicamente si existen actualizaciones de Drupal o de los módulos y temas instalados, el mismo avisa si existen a través del área de administración. Además Drupal cuenta con la opción "**Recibir avisos por correo electrónico**", que si es activada se recibe en la cuenta de correo de mantenimiento del sitio los avisos de actualizaciones importantes, como las actualizaciones de seguridad (10).

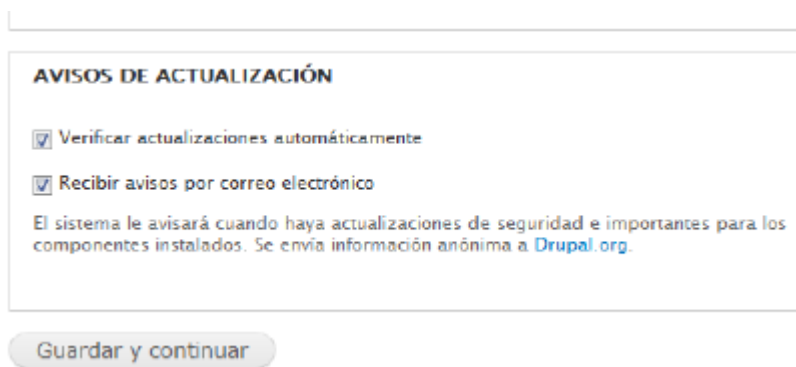


Figura 2: Configuración de las Actualizaciones en Drupal.

1.2.2 Administración en Joomla

Se accede a la zona de administración de Joomla a través del menú correspondiente al backend¹. La configuración de la base de datos en Joomla se realiza en el proceso de la instalación. Joomla permite crear cualquier número de bases de datos en el entorno de un servidor local. En las configuraciones avanzadas aparecen opciones adicionales, como seleccionar si las tablas de cualquier instalación Joomla en esa base de datos serán vaciadas o bien si serán guardadas y marcadas con el prefijo bak.

¹ Área de administración



Figura 3: Configuración de la base de datos en Joomla.

Joomla en sus opciones avanzadas de configuración del panel de administración y configuración permite activar o desactivar el sistema de cacheo, y establecer el tiempo que pasará antes de que el módulo sea "re-cacheado" cuando esté activo. Este CMS cuenta además, con un módulo llamado Caché que permite usar la caché de página. También permite elegir si se desea usar el navegador del cliente para cachear páginas. La caché de página es gestionada por el Plugin de caché del sistema (System Caché Plugin) que viene de serie con Joomla (Figura 34 de los Anexos).

También cuenta con el panel de la caché de módulo y de vista que es diferente a la de página porque sólo guarda copias de ciertas partes de la página.

Mensajería es un componente del núcleo de Joomla y un sistema de mensajería privada para los usuarios del backend. Permite enviar y recibir mensajes a otros usuarios con permisos de acceso al área de administración del sitio, y puede configurar el sistema para que mande un correo electrónico cada vez que se recibe un mensaje nuevo, para que los borre transcurrido un número determinado de días y puedes incluso bloquear la bandeja de entrada.

En el área de las actualizaciones Joomla permite buscar actualizaciones y en caso de encontrar alguna procede a una actualización automática a través de una configuración FTP. El proceso informa al administrador con mensajes de explicaciones durante la actualización del sistema (11).

Una vez realizado el estudio sobre las principales herramientas existentes teniendo en cuenta cada una de sus funcionalidades de administración, se puede concluir que no están acorde con las necesidades del panel de administración que se desea integrar al Sauxe, debido a que los elementos que intervienen en el proceso de configuración de la base de datos de Drupal no son los que requiere la conexión a la base de datos en Sauxe, por otro lado en la configuración de la caché se tuvieron aspectos teóricos de alta validez como la optimización del ancho de banda que mejorará la velocidad de respuestas de las peticiones al servidor. En el caso de las actualizaciones se consideró provechoso el proceso de verificación de actualizaciones de forma automática. Joomla en la configuración de conexión al servidor de base de datos presenta elementos de configuración que son semejantes a los del marco de trabajo Sauxe pero no engloba los necesarios. La configuración de la caché en Joomla no implementa la configuración de varios servidores de caché que es una de las necesidades de la herramienta a desarrollar. De los componentes de mensajería y actualizaciones se asumieron algunos mecanismos que sirven de apoyo al desarrollo de la herramienta aunque no se puedan adaptar debido a la diferencia en la estructura de la arquitectura del Sauxe y las aplicaciones estudiadas. Por tanto, considerando los resultados de la investigación concluida se decide implementar una herramienta para la administración del sistema del marco de trabajo Sauxe.

1.3 Modelo de desarrollo orientado a componentes

Existen dos tipos de metodologías: las tradicionales o pesadas y las metodologías ágiles. Las primeras se centran fundamentalmente en controlar el proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. En el caso de las ágiles se basan en el factor humano y en el producto de software, dando mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Teniendo en cuenta características especiales que presenta la universidad en cuanto a los proyectos productivos en el CEIGE se elaboró el Modelo de desarrollo orientado a componentes.

Fue definido para que se tuvieran en cuenta las necesidades que presentaban cada línea, y los principales riesgos del proyecto. Está basado en buenas prácticas y principios de varias metodologías ya sean ágiles como XP y SCRUM o pesada como RUP, lo cual implica mejoría en aspectos importantes como la planeación, formación y satisfacción del equipo de trabajo.

El modelo de desarrollo orientado a componentes en general propone una solución sencilla y novedosa centrándose en el desarrollo de componentes como base tecnológica con una calidad superior en menor tiempo. Además propone dividir el trabajo y el equipo para lograr una mayor especialización de los desarrolladores. Se caracteriza, por ser (12):

- **Centrado en la arquitectura:** La arquitectura determina la línea base, los elementos de software estructurales a partir de los elementos de la arquitectura de negocio. Interviene en la gestión de cambios y diseña la evolución e integración del producto. La arquitectura orienta las prioridades del desarrollo, resuelve las necesidades tecnológicas y de soporte para el desarrollo.
- **Orientado a componentes:** Las iteraciones son orientadas por el nivel de significancia arquitectónicas de los componentes, los mismos son abstracciones arquitectónicas de los procesos de negocio y requisitos asociados que modelan, el componente es la unidad de medición y ordenamiento de las iteraciones.
- **Iterativo e incremental:** Las iteraciones son planificadas y coordinadas con el equipo de arquitectura, los clientes y la alta gerencia. Cada iteración constituye el desarrollo de componentes, los cuales son integrados al término de la integración, permitiendo de esta manera la evolución incremental del producto.
- **Ágil y adaptable al cambio:** El desarrollo de las partes formaliza solamente las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas. Los clientes y funcionales están involucrados en el proyecto y poseen parte de las responsabilidades del éxito del mismo. Los cambios son conciliados semanalmente, discutidos y aprobados.

1.3.1. Modelo de ciclo de vida de los proyectos del CEIGE

El modelo de ciclo de vida que se presenta en la Figura 7 define las fases por las que transitarán los proyectos de desarrollo de software del CEIGE (13):

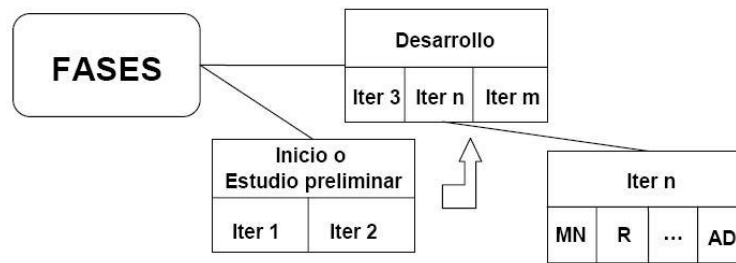


Figura: 4 Fases del ciclo de vida de proyectos del CEIGE.

1.3.2. Descripción de las fases del ciclo de vida de los proyectos

Inicio o Estudio preliminar: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto a un alto nivel, la evaluación de la factibilidad del proyecto y el registro de este. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo para así decidir si se ejecuta o no el proyecto. Los objetivos de la fase son:

- Asegurar la factibilidad del proyecto.
- Establecer un plan para la ejecución del proyecto.

Hitos:

- Plan de desarrollo de software.
- Acta de inicio del proyecto firmada.

Desarrollo: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se refinan los requisitos, se elaboran la arquitectura, el diseño, se implementa y se libera el producto. El objetivo de esta fase es:

- Obtener un sistema que satisfaga las necesidades de los clientes y usuarios finales.

Hito:

- Producto liberado por entidad certificadora de calidad.

En esta fase se ejecutan las disciplinas Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas y Pruebas de liberación (13):

1.3.3. Ciclo de vida de proyectos del CEIGE

El ciclo de vida de los proyectos del CEIGE (Figura 8) tiene en cuenta las actividades de cada una de las fases y áreas de procesos que plantea el nivel dos de CMMI establecido en la UCI. Este abarca el total de acciones que se realizan en las distintas líneas de desarrollo para la elaboración del servicio o producto final, sin embargo, se debe adaptar a las características particulares del proyecto que puede que no ejecute determinada disciplina, así como la elaboración de determinados artefactos.



Figura 5: Ciclo de vida proyectos CEIGE.

En el modelo de desarrollo se generan varios artefactos los cuales son de gran importancia para el desarrollo de un sistema web. Estos artefactos son necesarios para documentar el producto durante el ciclo de vida del desarrollo del software. En cada uno de los flujos de trabajo (Análisis y diseño, Implementación y Prueba) se van elaborando los artefactos correspondientes (13):

Entre los artefactos principales se encuentran:

- Modelo conceptual
- Descripción de los procesos del negocio.
- Descripción de requisitos
- Diagrama de clases del diseño.
- Diseño de casos de prueba
- Diagrama de despliegue.

1.4 Tecnologías utilizadas para el desarrollo

Se realizará un análisis de las tecnologías que se definieron como política de desarrollo del centro (14), ya sean lenguajes de programación como los Frameworks que componen el Sauxe.

1.4.1. Lenguajes de programación.

Los lenguajes de programación son tecnologías que nos permiten crear soluciones de software. Existen innumerables lenguajes, que siguen el concepto de lograr la mayor abstracción posible, y facilitar el trabajo al desarrollador, aumentando la productividad en función del resultado alcanzado.

PHP 5.3.8

PHP (Hypertext Pre-Processor) es un lenguaje de programación usado mundialmente para la creación de sitios web dinámicos. Es una tecnología de código abierto que resulta muy útil para diseñar de forma rápida y eficaz aplicaciones web dirigidas a bases de datos. La implementación principal de PHP es producida por The PHP Group. Publicado bajo la PHP License, considerada por la Free Software Foundation como licencia de software libre.

Es un lenguaje sencillo, de sintaxis cómoda. Su interpretación y ejecución la realiza el servidor en el cual se encuentra almacenada la página y el cliente sólo recibe el resultado de la ejecución. Dispone de una conexión nativa a los principales sistemas de base de datos utilizados actualmente tales como, Postgres, MySQL, Oracle, Microsoft SQL Server, lo cual permite la creación de aplicaciones web muy robustas.

Características generales de PHP (15):

- No es un lenguaje de marcas.
- Presenta compatibilidad de interacción con los servidores de web más populares tales como Apache, XAMPP, FoxServ.
- Incorpora bibliotecas sumamente amplias de funciones integradas para realizar útiles tareas relacionadas con la web.
- Posee una amplia documentación, la cual se destaca por tener todas las funciones del sistema detalladamente explicadas y ejemplificadas en un único archivo de ayuda.
- Es un producto libre, de código abierto, lo que implica que una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.
- Es un lenguaje multiplataforma.
- Permite las técnicas de programación orientada a objetos.
- Posee tratamiento de errores.

Con el estudio de las características del lenguaje PHP se pueden generalizar algunas ventajas de su uso:

- No requiere definición de tipos de variables.
- Las librerías facilitan en gran medida el desarrollo de las aplicaciones.
- Es un producto libre y de código abierto, representa una gran ventaja pues está soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos ya sea Unix, Linux, Windows y Mac OS X.

JavaScript 1.8

Es un lenguaje script e interpretado, es decir, que no requiere compilación, con una sintaxis muy parecida a la del lenguaje C y Java pero más sencillo de utilizar. Este es un lenguaje Case Sensitive² y al contrario que Java, no es exactamente un lenguaje orientado a objetos, puesto que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base que serían los prototipos y extendiendo su funcionalidad. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM³. En JavaScript se puede capturar los eventos dentro de las páginas para ejecutar alguna acción, así como ser responsable de controlar algunas validaciones a nivel de cliente.

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Se utiliza mayormente para incorporar dinamismo, mejorar el aspecto y la funcionalidad de una página web e interactuar con los usuarios. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Mozilla Firefox, Netscape, Opera (16).

²Sensible a la diferencia entre minúsculas y mayúsculas.

³Modelo de Objetos de Documento, DOM siglas en inglés.

XML

El Lenguaje de Marcado Extensible⁴, es un simple formato basado para representar la información estructurada: documentos, datos, configuración, libros, transacciones, facturas, y mucho más. Se derivó de un formato estándar más antiguo llamado SGML (ISO 8879) permitiendo ser más conveniente para la web (17).

El estándar permite definir la gramática de lenguajes específicos, por lo que el XML no es solo un lenguaje en particular, sino también un metalenguaje cuya particularidad más importante es que no posee etiquetas prefijadas con anterioridad, permitiendo describir otros lenguajes de marcado y definir lenguajes de presentación propios en dependencia del contenido del documento.

El XML tiene como objetivos de diseño destacar la simplicidad, generalidad y usabilidad del estándar. Aunque el diseño de XML se centra en los documentos, es ampliamente utilizado para la representación de las estructuras de datos arbitrarios, igualmente se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas, XML se desarrolló para proporcionar una flexibilidad y consistencia que no se podía alcanzar con HTML.

1.4.2. Librerías y marcos de trabajo

Zend Framework 1.9

Es un marco de trabajo de alta calidad para el desarrollo de aplicaciones y servicios web, es de código abierto y está diseñado para PHP.

Zend es un framework basado en la arquitectura MVC (Modelo-Vista-Controlador). Brinda soluciones para construir sitios web modernos, robustos y seguros. Posee un bajo acoplamiento entre sus componentes, lo que posibilita la utilización de los mismos a conveniencia, aunque todos estos en conjunto conforman un potente y extensible framework (18).

Presenta también las siguientes características (19):

⁴ Extensible Markup Language, XML siglas en inglés.

- Módulos para manejar archivos en formato de documento portátil PDF, canales de sindicación de noticias RSS, y presenta el módulo Zend_Service para servicios web que soporta APIs de Amazon, Flickr, Yahoo.
- Módulos para la autenticación que poseen robustas clases, filtrado y validación de entradas de datos.
- Proporciona un sistema de caché dividido en frontend y backend, de forma que se puedan almacenar en caché diferentes datos como resultados de funciones, páginas completas, y que esta información se almacene en archivos, en memoria, en base de datos, etc.
- Incluye objetos de las diferentes bases de datos, pues es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Provee una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad.
- Proporciona una capa de acceso a base de datos, construida sobre PDO pero ampliándola con diferentes características, haciendo extremadamente simple la interacción con estas, sin necesidad de escribir ninguna consulta SQL.
- Permite convertir estructuras de datos PHP a JSON7 y viceversa.

Doctrine Framework 1.2

Doctrine Framework es un potente y completo sistema de Mapas de Relaciones de Objetos (Object Relational Mapper, ORM por sus siglas en inglés) para PHP 5.2 o superior que incorpora una capa de abstracción a base de datos (Database Abstraction Layer). Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases creadas siguiendo las pautas del ORM, a tablas de una base de datos. Posee la habilidad de escribir opcionalmente las preguntas de la base de datos utilizando la programación orientada a objetos debido a que doctrine utiliza el patrón Active Record para manejar la base de datos. Esto les brinda una alternativa poderosa a diseñadores de SQL que mantiene un máximo de flexibilidad sin requerir la duplicación del código innecesario. Cuenta con su propio lenguaje DQL (Doctrine Query Language) a través del que permite crear el modelo en la sintaxis específica que propone Doctrine y luego generar toda la base de datos (20).

ExtJS 3.3.0

ExtJS es un framework de desarrollo de software que facilita las herramientas necesarias para la creación de aplicaciones web. El modelo del componente ExtJS mantiene bien estructurado el código incluso las aplicaciones más grandes pueden mantenerse fácilmente. Se puede extender los componentes predefinidos para satisfacer las necesidades de la solución, y las extensiones pueden ser encapsuladas justamente dentro de los componentes. Como resultado, el equipo de desarrollo puede desarrollar incluso aplicaciones más grandes sin consultar otros códigos. Proporciona una colección enciclopédica de interface de usuario con un elegante tema de inicio. Tener en cuenta que ExtJS 3.3 tiene dos tipos de licencias, Lesser General Public License (LGPL) y la comercial, esta última es obligatoria si se desea obtener soporte.

ExtJS ofrece un rango extraordinario de interface de usuario. Posee una amplia librería con gran variedad de componentes web como ventanas, pestañas, formularios, paneles, barras de herramientas, menús, menús jerárquicos de árbol y más, semejante a aplicaciones de escritorio. Dentro de su librería contiene componentes para el manejo de datos, lectura de XML, lectura de datos JSON. Posee recursos de aprendizaje y una documentación API detallada, y regularmente mantenida.

Su diseño y arquitectura brindan las características esenciales de una librería completa para construir interfaces de usuario intuitivas, amigables y flexibles:

- Puede trabajar por medio de otras librerías JavaScript usando adaptadores.
- Sirve de puente entre las librerías JS más usadas (Prototype, JQuery, YUI⁵). Debido a que se inició como una extensión de YUI ésta presenta una cierta ventaja de compatibilidad respecto a las otras dos.
- Funcionalidades e interfaces controladas por eventos. Esto permite que en cada interacción del usuario con la aplicación, Ext JS intervenga en la presentación de la respuesta que da la aplicación al usuario, ya sea con intervención del servidor web o no.

Estas son las razones por lo que Ext JS es la principal elección de los desarrolladores en el mundo.

⁵ Interfaz de Usuario de Yahoo, YUI siglas en inglés.

Ya sea en Internet Explorer o el último navegador Chrome, las aplicaciones Ext JS tienen el mismo comportamiento, es decir, no importa el navegador donde la aplicación sea ejecutada. Ext JS framework soporta la mayoría de los navegadores incluyendo (21):

- Internet Explorer 6+.
- Safari 3+.
- Opera 9+.
- Mozilla Firefox 1.5+.

1.4.3. Herramientas de desarrollo

En el diseño e implementación de la solución se utilizarán, en sus versiones más recientes, las herramientas definidas por el Departamento de Tecnología del CEIGE.

Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE (por las siglas en inglés de Ingeniería de Software Asistida por Computadora) que utilizando UML (por las siglas en inglés de Lenguaje de Modelado Unificado) como lenguaje de modelado, permite la captura, diseño, gestión y documentación de los artefactos generados durante el proceso de desarrollo de software (22).

Esta herramienta proporciona al equipo de desarrollo de software un lenguaje estándar que facilita la comunicación entre sus integrantes y con el cliente. Además, permite realizar el proceso de ingeniería del software de forma directa (versión profesional) o inversa.

PostgreSQL 8.3

Los servidores de bases de datos surgen producto de la necesidad de manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes de una manera segura.

Los SGBD⁶ proveen herramientas de administración completas que simplifican la tarea de la configuración, seguridad, creación y gestión de bases de datos. Estos sistemas deben

⁶Sistemas de gestión de bases de datos.

proporcionar mecanismos de comunicación con otras plataformas que actúen también como clientes o servidores de datos.

PostgreSQL, inicialmente llamado Postgres, fue creado por el profesor Michael Stonebraker de la Universidad de Berkeley. Es un sistema de gestión de base de datos relacional orientada a objeto que marcó el inicio de las tecnologías objeto-relacional dentro de las bases de datos. Es distribuido bajo la licencia Berkeley Software Distribution (BSD).

PostgreSQL tiene soporte total para disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin que existan bloqueos. Posee un lenguaje procedural propio denominado PL/PgSQL. Garantiza también las dependencias entre objetos, control para transacciones y concurrencia. Brinda soporte SQL/XML, soporte de autenticación, soporte total del modelo relacional de bases de datos, soporte nativo para SSL (23).

Ventajas:

- Se destaca en ejecutar consultas complejas y de gran tamaño. Facilita la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. PostgreSQL siempre tiene en cuenta una característica deseable en las bases de datos: la integridad referencial, gracias a la que se garantiza que una entidad o registro siempre se relacione con otras entidades que existen en la base de datos.
- Cuenta con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno. Fue diseñado para ambientes de alto volumen, por lo que escala fácilmente. Es ejecutado en la mayoría de los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows. Posee una documentación muy bien organizada, pública y libre, es altamente adaptable a las necesidades del cliente.

FireBug 1.11.2

Firebug es una extensión para Firefox que proporciona gran cantidad de herramientas de desarrollo web a su alcance mientras usted navega. Puede editar, depurar, perfilar y monitorear CSS, HTML, JavaScript y HttpRequests en cualquier página web en tiempo real. (24)

Durante el desarrollo de aplicaciones web es muy utilizado por las comodidades que brinda pues los desarrolladores puedan llevar un control del tráfico de información desde el cliente hasta el servidor y desde el servidor al cliente, brindando muchas facilidades en el momento en que es necesario realizar la conexión de la capa de presentación a la capa de negocio.

Características generales (25):

- Inspecciona y edita HTML - Firebug hace que sea fácil de encontrar elementos HTML enterrado profundamente en la página. Una vez que los desarrolladores web que se encuentra lo que está buscando, Firebug les da una gran cantidad de información, y les permite editar el HTML en vivo.
- Edición de CSS - CSS Firebug, dice todo acerca de los estilos en las páginas web.
- Métricas CSS: Firebug puede proporcionar mediciones e ilustraciones en todas las compensaciones, los márgenes, bordes, el relleno y tamaños.
- Supervisa la actividad de red para supervisar el estado de una página web, sobre todo de la rapidez con que se carga, Firebug proporciona la información necesaria para aprender sobre cómo mantener a los sitios web en la forma superior.
- Depura JavaScript: Firebug incluye un potente *depurador JavaScript* que permite a los desarrolladores web detener la ejecución en cualquier momento. Si el código es un poco lento, sólo tiene que utilizar el perfilador de JavaScript para medir el rendimiento y encontrar los cuellos de botella rápido.
- Encuentra errores: Firebug permite a los desarrolladores web saber inmediatamente si hay algo mal en sus sitios web y proporciona información detallada y útil sobre los errores en JavaScript, CSS y XML.
- Explora DOM: Firebug ayuda a encontrar los objetos DOM de forma rápida y luego editarlas sobre la marcha.
- Ejecuta JavaScript: Firebug ofrece una línea de comandos de JavaScript completo con todas las comodidades muy modernas.
- Los registros de JavaScript: Firebug muestra un conjunto de potentes funciones de registro para proporcionar respuestas rápidas.

Apache 2.2.9

Es un software que permite crear en un ordenador, de una sencilla y rápida forma, un servidor de Protocolos de Transferencia de Hipertexto (HTTP), el cual se encarga de transferir los hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El Apache HTTP Server es un servidor robusto, de múltiples características y funcionalidades, una herramienta gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.

Principales características con las que cuenta (26):

- Configuración basada en un poderoso archivo (httpd.conf).
- Soporte del protocolo HTTP y autenticación basada en la web.
- Soporte de host virtuales: Apache es uno de los primeros servidores web en soportar tanto host basados en IP como host virtuales.
- Soporte de scripts PHP: Apache ofrece un amplio soporte de PHP utilizando el módulo mod_php.
- Servidor proxy integrado.
- Estado del servidor y adaptación de registros. Apache le da una gran cantidad de flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador web.

Dadas las características se puede decir que un servidor Apache es altamente configurable y de diseño modular. Posibilita que los administradores de sitios web puedan elegir los módulos que serán incluidos y ejecutados en el servidor. Permite personalizar los mensajes de errores, la creación y gestión de logs⁷, y la negociación de contenido, de este modo es posible tener un mayor control sobre lo que sucede en el servidor. Está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. La aplicación es prácticamente universal pues permite ejecutarse en múltiples sistemas operativos como Windows, Novell NetWare y Mac OS X.

⁷ Es un registro oficial de eventos durante un rango de tiempo en particular.

Subversion 1.4

Antes de existir las herramientas para controlar las versiones de un proyecto, era común que los desarrolladores pasaran trabajo con la gestión de los cambios en el código, de ahí que sea indispensable contar con una herramienta que facilite el control de versiones.

Subversion es una herramienta de control de versiones utilizada principalmente en los proyectos de desarrollo de software. La herramienta es libre y de código fuente abierto, se emplea para manejar los ficheros y directorios a través del tiempo haciendo posible la recuperación en caso de una pérdida inminente, la recuperación se lleva a cabo mediante la salva existente en el repositorio⁸ central. El repositorio es como un servidor de ficheros ordinario, excepto porque guarda todos los cambios hechos a sus ficheros y directorios. Subversion puede acceder al repositorio a través de la red, por lo que es usado por desarrolladores que se encuentran en distintas estaciones de trabajo.

Una característica importante de Subversión es que los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Presenta también las siguientes características (27):

- Mantiene versiones no sólo de archivos, sino también de directorios
- Mantienen versiones de los metadatos asociados a los directorios.
- Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre.
- Atomicidad de las actualizaciones, una lista de cambios constituye una única transacción o actualización del repositorio, esta característica minimiza el riesgo de que aparezcan inconsistencias entre distintas partes del repositorio.
- Soporte tanto de ficheros de texto como de binarios.
- Mejor uso del ancho de banda, pues en las transacciones se transmiten sólo las diferencias y no los archivos completos.

⁸ Es un sitio centralizado donde se almacena y mantiene información, habitualmente bases de datos o archivos informáticos.

Tener en cuenta que Subversion ayuda a medir la calidad y cantidad del trabajo realizado en una unidad de tiempo. Esta visibilidad instantánea nos permite observar la productividad del equipo de trabajo, así como los beneficios a escala administrativa para un líder de grupo.

NetBeans 7.3

NetBeans IDE (Integrated Development Environment) es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. Es de código abierto en constante crecimiento que cuenta con una amplia comunidad de usuarios y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones web, empresariales, de escritorio y aplicaciones móviles utilizando la plataforma Java, pero se puede utilizar para desarrollar en otros lenguajes de programación (28). NetBeans presenta un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación, proponiendo un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS, permitiendo refactorización, búsqueda de usos para CSS y lenguajes de tipo HTML, además admite el completamiento de código y links para atributos de CSS. Por otra parte el editor de PHP es muy ágil y robusto, siendo eficiente en el completamiento de código, reconocimiento de sintaxis y presentando un potente depurador que facilita la programación. Es un producto libre y gratuito sin restricciones de uso (29).

Es un entorno de desarrollo que posee un alto grado de robustez y valor para el usuario final, que se traduce en una considerable reducción en el tiempo de desarrollo, consistencia en la interfaz de usuario, independencia de la plataforma, reusabilidad y confiabilidad (28). Existen además un número importante de módulos para extender este IDE.

1.5 Patrones de Arquitectura

MVC: Modelo-Vista-Controlador (MVC) es una de las arquitecturas de aplicaciones web más comunes. Está conformada como su nombre lo indica, por tres capas, las cuales se describen a continuación (30):

- **Vista:** La renderización de código front-end, frecuentemente llamada nivel de presentación, debería aspirar a producir la salida HTML para el usuario con poco o nada de lógica de aplicación. Como muchas aplicaciones serán internacionalizadas (por

ejemplo no conteniendo cadenas localizadas o información cultural en la capa de presentación), deben usar llamadas al modelo (lógica de aplicación) para obtener la información requerida para suministrar información útil al usuario en su lenguaje y cultura preferido, dirección del script y unidades. Todas las entradas de los usuarios se encuentran redireccionadas hacia los controladores en la lógica de la aplicación.

- **Controlador:** El controlador (o lógica de la aplicación) toma entradas de los usuarios y las dirige a través de varios flujos de trabajo que llaman a los objetos del modelo de la aplicación para extraer, procesar, o almacenar información. Los controladores bien codificados, validan información centralmente en el servidor contra problemas de seguridad comunes antes de pasar la información al modelo de procesamiento y se aseguran que la salida de datos sea segura o en un formato preparado para una salida protegida por parte del código de visualización.
- **Modelo:** La idea es encapsular el trabajo sucio en el modelo de código, en lugar de exponer primitivas. Si el controlador y el modelo se encuentran en diferentes máquinas, la diferencia de rendimiento será asombrosa, por lo que es importante para el modelo ser útil a un nivel alto. El modelo es responsable de la comprobación de datos en contra de las reglas de negocio, y cualquier riesgo residual para el único almacén de datos en uso. Si el modelo almacena los datos en un lenguaje interpretado, como SQL, entonces el modelo se encarga de la prevención de inyección de SQL.

1.6 Pruebas de software

La IEEE⁹ define el concepto de prueba como: “Una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”. Destacando de aquí la importancia que tiene realizarle pruebas al software antes de ser entregado, garantizando así que el mismo tenga la menor cantidad de errores y mejor calidad posible.

⁹ (Institute of Electrical and Electronic Engineers) Instituto de Ingenieros Eléctricos y Electrónicos

1.6.1. Objetivos de la prueba

Las pruebas de software tienen como objetivos detectar errores no encontrados hasta el momento en la aplicación. Se puede asegurar entonces del éxito de las pruebas cuando una vez aplicadas se encuentren errores en el software. Con las pruebas igualmente se pueden observar hasta qué punto el software parece funcionar en concordancia con los requisitos descritos por el cliente; aunque no pueden asegurar la ausencia de defectos, sólo permite descubrir que estos existen (31).

1.6.2. Pruebas de Caja Negra y Caja Blanca

Hay una de dos maneras de probar cualquier producto construido (31):

- Si se conoce la función específica para la que se diseñó el producto, se aplican pruebas, que demuestren que cada función es plenamente operacional, mientras se buscan los errores en cada función.
- Si se conoce el funcionamiento interno del producto, se aplican pruebas para asegurarse de que todas las piezas encajan, es decir que las operaciones internas se realizan de acuerdo con las especificaciones y que se han probado todos los componentes internos de manera adecuada.

Al primer enfoque de prueba se le denomina pruebas de caja negra y al segundo pruebas de caja blanca. Las pruebas de caja negra son las que se aplican a la interfaz del software. Una prueba de este tipo examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software. Las pruebas de caja blanca del software se basan en un examen cercano al detalle procedimental. Se prueban las metas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejerciten conjuntos específicos de condiciones.

1.6.3. Justificación de la selección de las pruebas

Para verificar el correcto funcionamiento de la solución se realizarán pruebas de caja negra las cuales se centraran en los requisitos funcionales, asegurando su correcto funcionamiento con la realización de casos de pruebas. Estas pruebas nos facilitarán encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructura de datos o en acceso a base de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Complementando estas pruebas se realizarán pruebas de caja blanca, con la utilización de la técnica del camino básico, con la cual se podrá derivar casos de prueba que:

- Garanticen que se ejercitan al menos una vez todos los caminos independientes de cada módulo
- Ejerciten todas las decisiones lógicas en sus caras verdaderas y falsas.
- Ejerciten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras de datos internas para asegurar su validez.

1.7 Conclusiones

Al finalizar el presente capítulo se puede arribar a las siguientes conclusiones:

- Se construyó el marco teórico de la investigación estudiándose los principales conceptos y definiciones que son esenciales para la investigación, necesarias para lograr un mejor entendimiento del problema tratado en el presente trabajo.
- Existen algunas aplicaciones que incluyen un panel de administración para la configuración de su sistema, debido a sus características y arquitectura, no se ajustan a los requerimientos del marco de trabajo Sauxe, determinándose la necesidad de desarrollar una propuesta que solucione los problemas del marco de trabajo Sauxe.
- Como herramientas y tecnologías para el análisis, diseño e implementación se mantienen las seleccionadas por el equipo de desarrollo del marco de trabajo Sauxe, las cuales permitieron incrementar la productividad, calidad y reducir el tiempo de desarrollo.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

En el presente capítulo se realiza un estudio de la solución propuesta partiendo de la fundamentación teórica y del modelo de desarrollo, se comienza la modelación del negocio describiendo los procesos de negocio. También en este capítulo se explican los principales artefactos de la metodología empleada y se describen las características que debe tener el panel de administración, a través de los requisitos funcionales y no funcionales. A partir de la especificación de los requisitos se elabora el modelo de diseño donde se describen las principales clases del diseño que serán utilizadas posteriormente en la implementación de la herramienta.

2.1. Modelo conceptual

El modelo conceptual tiene como objetivo identificar y explicar los conceptos significativos en un dominio de problema, identificando los atributos y las asociaciones existentes entre ellos.

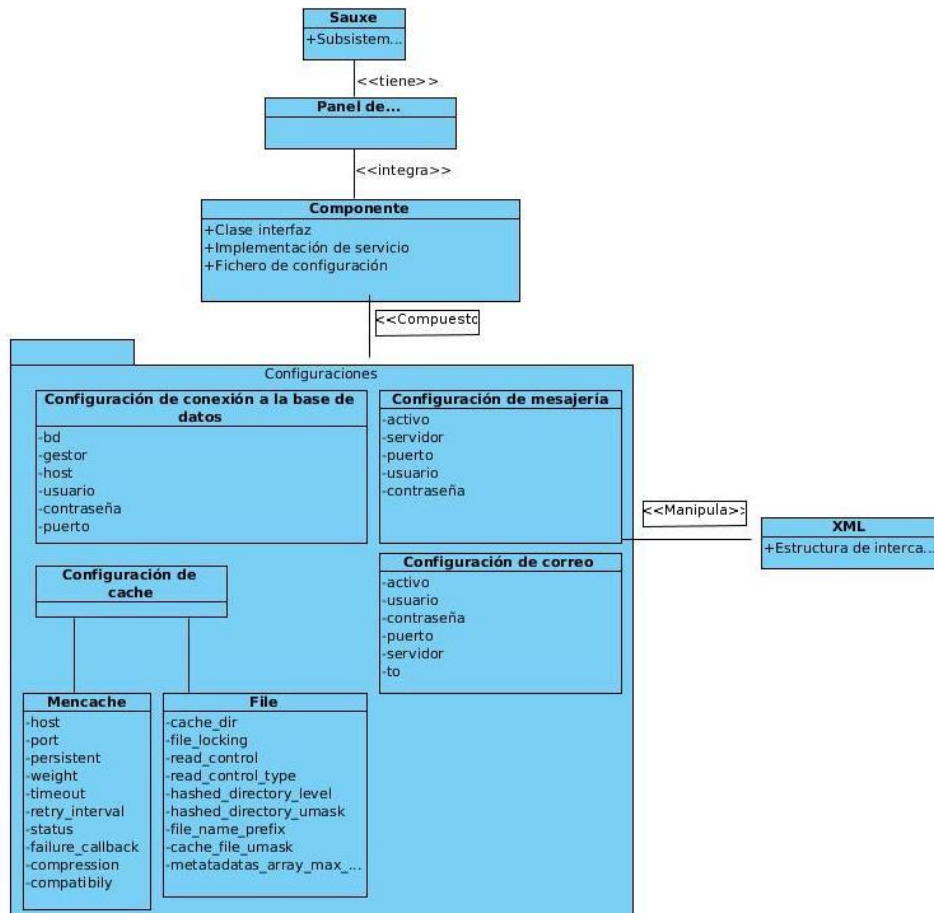


Figura 6: Modelo conceptual.

2.2. Descripción de los procesos del negocio

Un proceso de negocio es un conjunto de actividades o tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. La descripción de estos tiene como objetivo entender el funcionamiento del componente que se desarrolla, comprender los problemas actuales e identificar las mejoras, es una vía natural para la determinación y captura de requerimientos. La descripción de los procesos de negocio hace más viable el paso a las actividades del análisis pues posibilita una comprensión más clara de los procesos en cuestión.

2.2.1. Descripción del proceso: Configuración de archivos del sistema.

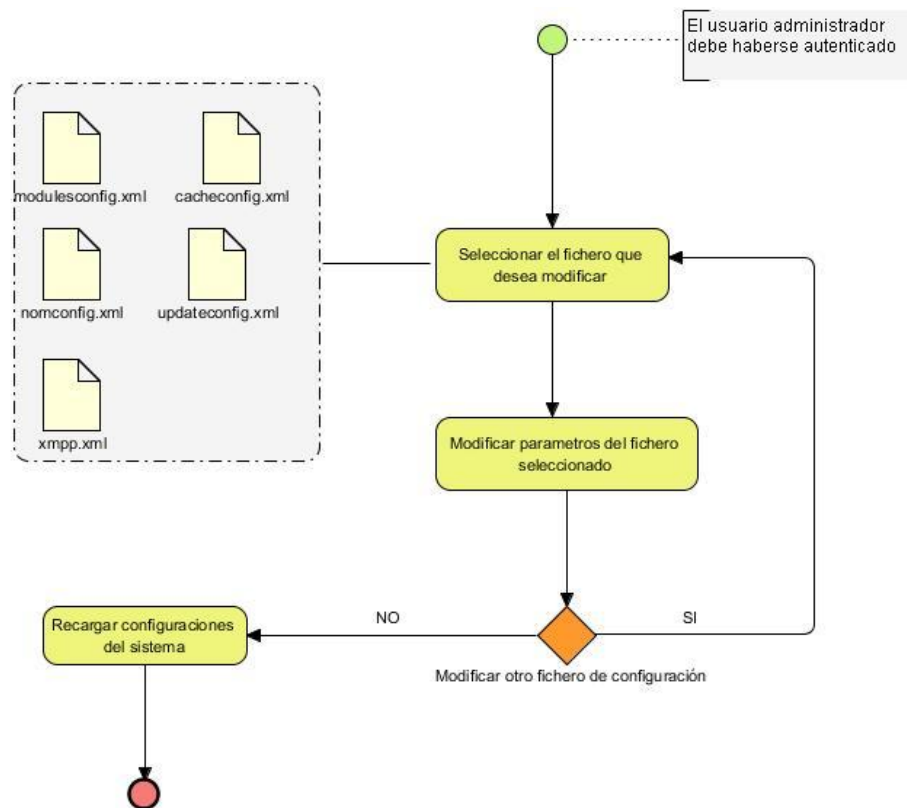


Figura 7: Descripción del Proceso de configuración de archivos del sistema.

En la figura anterior se describe el proceso de configuración de archivos del sistema, previamente para iniciar el proceso el usuario administrador de haberse autenticado, luego es seleccionado el fichero que se desea modificar, seguidamente se modifican los elementos del fichero seleccionado y por último se recargan las configuraciones del sistema o se selecciona un nuevo fichero a modificar.

2.3. Especificación de requisitos

La especificación de requisitos es un proceso básico que garantiza la calidad del software y por lo general es realizado por analistas e interesados en lenguaje natural. En la ingeniería de software, un requerimiento es una necesidad documentada sobre la funcionalidad, el contenido o la forma de un servicio o producto.

2.3.1. Requisitos funcionales

Los requisitos funcionales definen las acciones y funciones que debe de ser capaz de realizar el sistema. Los mismos se centran en qué y cómo se deben de hacer las funciones.

Las técnicas utilizadas para el levantamiento de requisitos fueron las entrevistas, conjuntamente se realizaron talleres, los cuales fueron propicios para desencadenar tormentas de ideas. De las mismas surgieron los requisitos que se muestran y detallan a continuación.

2.3.1.1 Requisitos funcionales del Software

1. RF: Modificar elementos de configuración de la base de datos.
2. RF: Modificar elementos de configuración de correo.
3. RF: Modificar elementos de configuración de mensajería.
4. RF: Modificar elementos de configuración del adaptador de backend Memcached de la caché.
5. RF: Modificar elementos de configuración del adaptador de backend File de la caché.
6. RF: Realizar compresión de JS.
7. RF: Cambiar estado del Sauxe, Offline/Online.

Especificación del requisito Modificar configuración de conexión a la base de datos.

Tabla 1: Descripción del requisito funcional Modificar configuración de conexión a la base de datos.

Precondiciones	El usuario debe estar autenticado. El sistema debe tener los parámetros de configuración registrados.
Flujo de eventos	
Flujo básico	
1	Se selecciona la opción Inicio/ Administración del menú principal.
2	Se muestra la interfaz Gestionar configuraciones del sistema.
3	Se selecciona la opción Configuración de la base de datos.

4	Se introducen los datos correspondientes.
5	Se oprime el botón Aplicar, el sistema modifica los datos de la configuración de la base de datos y muestra el siguiente mensaje de información “Los elementos de la configuración de la base de datos han sido modificados satisfactoriamente.”
5.1	Se oprime el botón Cancelar, el sistema cancela las acciones y cierra la interfaz.
Pos-condiciones	
1	Se han modificados los elementos de configuración de conexión de la base de datos.
Flujos alternativos	
N/A	
Pos-condiciones	
1	N/A
Validaciones	
Se validan los datos según lo establecido en el Modelo conceptual.	
N/A	
Conceptos	Configuración de conexión a la base de datos.
Requisitos especiales	N/A
Asuntos pendientes	N/A

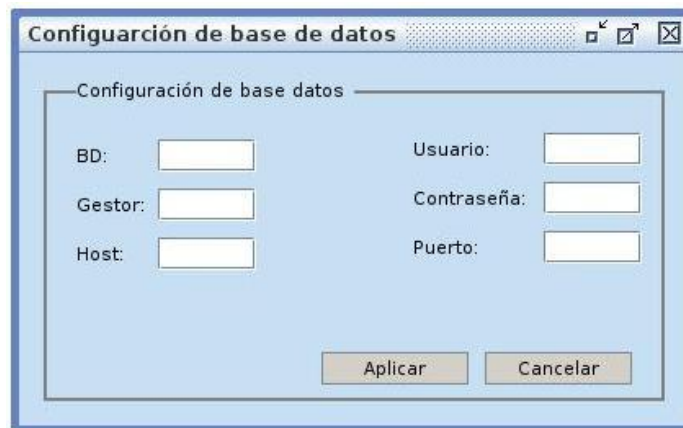


Figura 8: Prototipo de interfaz de configuración de conexión a la base de datos.

2.3.2. Requisitos no funcionales

Los requisitos no funcionales que debe poseer la herramienta a desarrollar son los establecidos por el centro CEIGE al inicio del proceso de desarrollo, a continuación son descritos algunos de los más importantes.

Usabilidad

La herramienta será usada por los desarrolladores los cuales poseen los conocimientos necesarios para trabajar con dicha herramienta.

Rendimiento

Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

Seguridad

Autenticación y Autorización (contraseña de acceso). Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. El acceso se realizará a través del Sistema de Gestión Integral de Seguridad (ACAXIA). La atención al sistema incluyendo el mantenimiento de las bases de datos, así como la salva de la información, se realizará de forma centralizada por el administrador.

Software

Para el cliente:

- Navegador Mozilla Firefox 3.0 o superior.
- Sistema operativo Windows 98 o superior, o Linux.

Para el servidor:

- Sistema operativo Linux en cualquiera de sus distribuciones.
- Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible. Este debe estar configurado con la extensión "pgsql" incluida.
- Un servidor de base de datos PostgreSQL 8.3 o superior.

Hardware

Para el cliente:

- Requerimientos mínimos: Procesador Pentium III a 1GHz con 256Mb de memoria RAM.
- Tarjeta de red.

Para el servidor:

- Requerimientos mínimos: Procesador Pentium IV a 2GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

2.4. Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

2.4.1. GRASP

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones de software para asignar responsabilidades). Para lograr un diseño exitoso en el desarrollo de la investigación se utilizaron algunos de estos patrones:

- **Experto**

Propone asignar una responsabilidad a la clase que tiene la información necesaria para realizar una tarea en específico. Este patrón se evidencia en la clase controladora `ModulesconfigController`, ya contiene la información necesaria para el trabajo con la configuración de la conexión a la base de datos.

- **Bajo acoplamiento**

Este patrón expresa que entre las clases deberán existir pocos vínculos, lo que reduce el impacto del cambio, es decir, que en caso de producirse una modificación en alguna de las clases, se tenga la mínima repercusión posible en el resto de clases. Este patrón se puede observar en general en las interfaces de la solución que implementan clases lo más independiente posible, reduciendo el impacto a los cambios.

- **Alta cohesión**

Asignar una responsabilidad de manera que cada elemento del diseño deba realizar una labor única dentro del sistema. En el diseño de los diferentes módulos de la solución como configuración de mensajería y configuración de caché se puede comprobar el uso de este patrón pues las clases pertenecientes a la capa de la lógica de la aplicación se le asignaron responsabilidades necesarias y bien delimitadas.

- **Controlador**

Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Es un intermediario entre la interfaz de usuario y el núcleo de las clases donde reside la lógica de la aplicación. Un ejemplo del uso de este patrón se observa en la clase `CacheController` debido

a que esta tendrá la responsabilidad de manejar eventos generados por un actor externo, en este caso los relacionados a la configuración de la caché.

2.5. Modelo de diseño

El modelo de diseño es una abstracción del modelo de implementación y el código fuente se emplea fundamentalmente para representar y documentar el diseño. Con el diseño se crea una entrada para la actividad de implementación futura pues se capturan los conceptos de las clases, interfaces y sistemas que serán necesarios para la solución.

2.5.1. Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación, contiene información como clases, asociaciones, atributos, métodos y dependencias. Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas con el objetivo de modelar los aspectos estáticos y definir una solución.

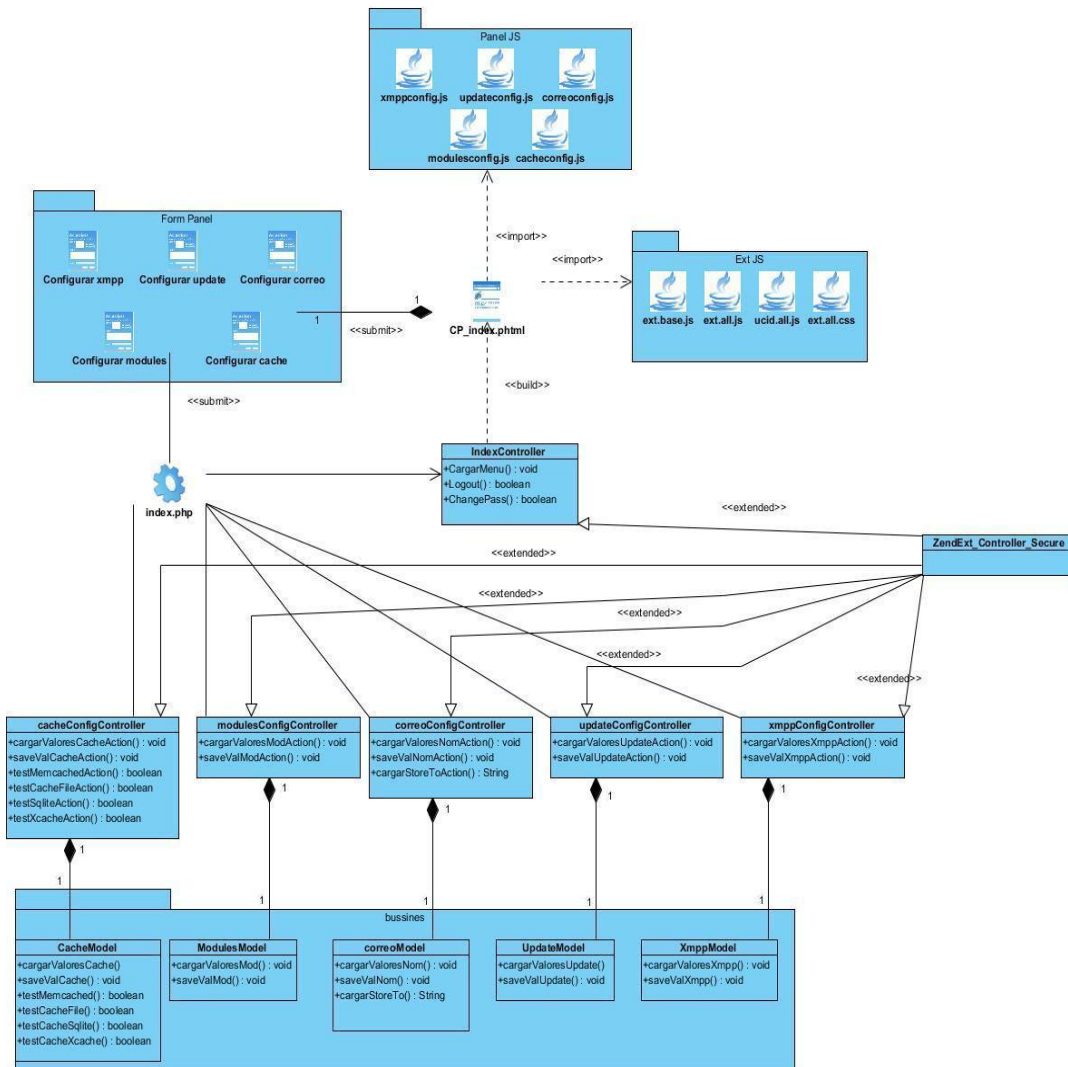


Figura 9: Diagrama de clase de la solución.

Descripción de clases del diseño

Tabla 2: Descripción de clases del diseño.

Clases	Descripción
xmppController	Realiza la lógica del componente de la configuración de mensajería y contiene las funciones para la manejo de datos en los archivos xml.
updateController	Realiza la lógica del componente de actualizaciones y contiene las funciones para

	la manejo de datos en los archivos xml.
correoController	Realiza la lógica del componente de configuración de correo y contiene las funciones para la manejo de datos en los archivos xml.
modulesController	Realiza la lógica del componente de configuración de conexión a la base de datos y contiene las funciones para la manejo de datos en los archivos xml.
cacheController	Realiza la lógica del componente de configuración de la caché y contiene las funciones para la manejo de datos en los archivos xml.
CP_index.phtml	Plantilla HTML encargada de mostrarle al usuario la interfaz correspondiente al componente con la que va a trabajar.
Panel JS	Paquete de formularios JavaScript a través de los cuales se maneja la información que introducen los usuarios del sistema.
Ext JS	Librería JavaScript que utiliza la clase CP_index.phtml.
Index.php	Se encarga de determinar cuál es la clase controladora que le corresponde a cada página cliente.

2.5.2. Diagrama de interacción

El diagrama de interacción, representa la forma en cómo un cliente (Actor) u objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente. Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o del de Casos de Uso. Como diagrama de interacción fue seleccionado el de secuencia, a continuación se muestra el diagrama de secuencia asociado al escenario configurar mensajería.

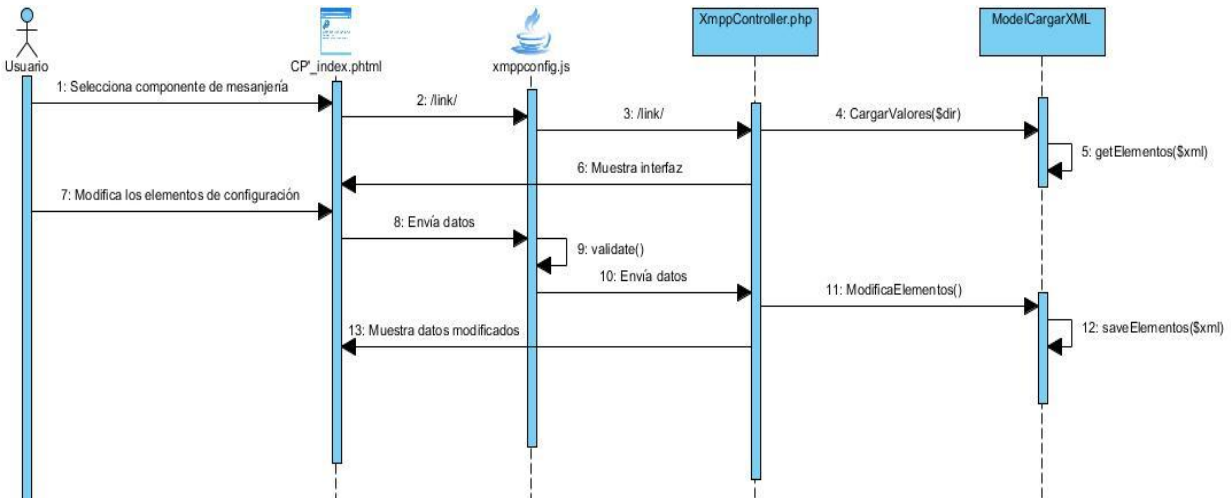


Figura 10: Diagrama de secuencia: Configurar mensajería.

2.6. Conclusiones

Luego de abordadas las temáticas del presente capítulo se arriba a las siguientes conclusiones:

- Con el modelado del negocio se logró un mejor entendimiento del proceso que se desea automatizar, lo que garantizó que la herramienta a desarrollar va a cumplir su propósito.
- La aplicación de las técnicas para la captura de requisitos permitió identificar los requisitos funcionales del sistema.
- El proceso de especificación de requisitos posibilitó que los requerimientos de la herramienta sean especificados y validados, lo cual garantizó estabilidad en la propuesta de solución.
- El uso de los patrones de diseño garantizó estandarizar la solución permitiendo una mayor reutilización del código y solución a problemas de contextos similares en el desarrollo de software.
- El modelo de diseño servirá como entrada a los flujos de implementación y prueba del mismo, permitiendo identificar las principales clases del sistema, las relaciones existentes entre ellas y sus métodos asociados.

CAPÍTULO 3: IMPLEMETACIÓN, PRUEBAS Y VALIDACIÓN DE LA SOLUCIÓN

En el presente capítulo se aborda sobre la estructura del marco donde se implementó la herramienta. Se plantean los puntos más importantes en la implementación del panel de administración, se describen los estándares de codificación para una mejor legibilidad del código y también la configuración de la herramienta para su ejecución en un ambiente del mundo real con el diagrama de despliegue. Además en este capítulo se especifican un conjunto de validaciones y pruebas que evalúan la calidad de la herramienta. Este capítulo también está dirigido a la validación de la propuesta de solución, teniendo en cuenta el criterio de especialistas seleccionados para comprobar la validez y adecuación de la propuesta, a partir de encuestas realizadas a los mismos. Se realiza un análisis estadístico para determinar el nivel de concordancia que existe entre los criterios de los especialistas. En aras de determinar el grado de concordancia entre los expertos, con respecto a las evaluaciones se utilizó el coeficiente de Kendall.

3.1. Estructura del Marco de Trabajo

Cumpliendo con las decisiones arquitectónicas del equipo de arquitectura se define una estructura donde van a estar situados cada uno de los subsistemas implementados dentro de la aplicación, de manera que se facilite la organización durante el desarrollo y mantenimiento del sistema, la descripción que a continuación se presenta está enfocada al subsistema administrador.

La carpeta denominada apps es donde se almacenan los controladores y modelos de cada uno de los componentes correspondientes a los subsistemas. En la Figura 14 se muestra el contenido subsistema en cuestión.

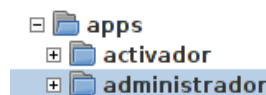


Figura 11: Carpeta correspondiente al Subsistema Administrador.

El subsistema administrador va a contener un conjunto de paquetes que serán especificados a continuación:

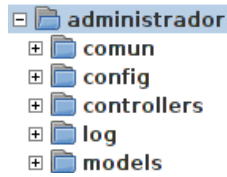


Figura 12: Paquete administrador correspondiente a la carpeta de la aplicación.

En el paquete controllers se hallarán las clases controladoras encargadas de gestionar las funcionalidades del subsistema.

El paquete models contiene dos paquetes para agrupar clases, el paquete bussines contendrá las clases necesarias para el acceso a los datos, el paquete domain debe contener las clases generadas por el ORM Doctrine a partir de cada una de las tablas existentes en la base de datos. En este caso el paquete domain permanecerá vacío pues no se realiza mapeo con el ORM Doctrine.

El paquete models estará estructurado de la siguiente forma:

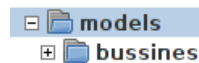


Figura 13: Paquete models correspondiente a la carpeta de la aplicación.

El paquete views contendrá los paquetes idioma y scripts, que se encargan de contener el idioma en que se va a mostrar la aplicación y las páginas clientes respectivamente.

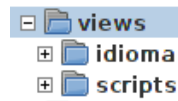


Figura 14: Paquete views correspondiente a la carpeta de la aplicación.

Junto a la carpeta apps antes mencionada debe existir además una carpeta que contiene las vistas de los subsistemas y componentes. Dicha carpeta se denomina web.

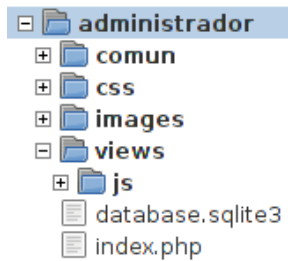


Figura 15: Carpeta de diseño correspondiente al Subsistema Administrador.

Index.php: Este fichero incluye la dirección del archivo de configuración y a través de este inicializa la aplicación para que se carguen en la misma un conjunto de componentes necesarios para su funcionamiento.

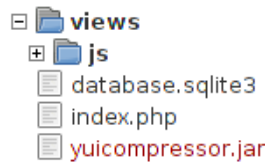


Figura 16: Paquete views correspondiente a la carpeta de diseño.

La carpeta views contendrá el paquete js que se especificará a continuación. El paquete js comprenderá las clases JavaScript necesarias para que el usuario interactúe con el sistema. También la carpeta views contiene archivos necesarios para la realizar la compresión de js.

3.2. Diagrama de despliegue

Los diagramas de despliegue se utilizan para modelar la vista de despliegue estática, además describen la configuración del sistema para su ejecución en un ambiente del mundo real. A continuación se muestra el diagrama de despliegue elaborado en la investigación. (32)



Figura 17: Diagrama de despliegue.

3.3. Estándares de codificación

Un estándar de codificación comprende todos los aspectos relacionados con la generación de código, de tal manera que sea prudente, práctico y entendible para todos los programadores. Un código fuente completo debe reflejar un estilo armonioso y uniforme, como si un único programador hubiera escrito todo el código. El mejor método para lograr que un grupo de desarrolladores mantenga un código de calidad es establecer un estándar de codificación (33).

Los estándares utilizados en la codificación fueron los siguientes:

Notación PascalCasing: Los identificadores, nombres de variables, métodos y funciones que están compuestos por múltiples palabras juntas, iniciarán cada palabra con letra mayúscula.

Notación CamelCasing: Es parecido al Pascal-Casing con la excepción que la letra inicial del identificador debe estar en minúscula.

Nomenclatura de las clases

Los nombres de las clases comenzarán con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing.

El nombre de las clases estará dado según la función que cumplan, es decir, se le agregará al nombre original un prefijo o sufijo que describa el tipo de clase que es la misma. A continuación se listan los prefijos y sufijos determinados para cada tipo de clase.

. Clases controladoras: A las clases controladoras se le adiciona el sufijo “Controller”.

Ejemplo: **XmppController**.

. Clases modelos: A las clases pertenecientes al modelo de negocio se le adiciona el sufijo “Model”.

Ejemplo: **ObtenerModel**.

Nomenclatura de las funciones

Los nombres a emplear para las funciones se escribirán con minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing.

En el caso particular de las funciones de las clases controladoras se les adiciona el sufijo “Action”.

Ejemplo: **cargarValoresAction**.

Nomenclatura de las variables

Se definió que el nombre a emplear para las variables se escribe en minúscula y en caso de ser un nombre compuesto se escriben ambas palabras en minúscula separadas por un guión bajo.

Ejemplo: **cachedir**
retry_interval

Nomenclatura de los comentarios

Se definió el uso de comentarios claros y precisos que ayudarán a una mejor comprensión del código implementado para que se entiendan sus objetivos específicos.

En las clases:

```
/**
 * Nombre de la clase *
 * Descripción *
 * @author *
 * @package *(módulo)
 * @subpackage *(sub módulo)
 * @copyright *
 * @version (versión - parche)
 */
```

Figura 18: Descripción de las clases.

Quedando en la clase controladora **CacheConfigController** de la siguiente manera:

```
/**  
 * CacheConfigController  
 * Clase controladora para la configuración de la cache  
 * @autor Adrián Avila Atencio  
 * @package Administrador  
 * @copyright ERP  
 * @versión 1.0  
 */
```

Figura 19: Ejemplo de comentario en una clase.

En los métodos:

```
/**  
 * Nombre de la función *  
 * Descripción *  
 * @author * (en caso de que no sea el autor de la clase)  
 * @return *(se pone lo que devuelve la función y un comentario)  
 */
```

Figura 20: Descripción de las funciones.

Establecido en la clase controladora **CacheConfigController** de la manera representada en la Figura 24.

```
/**  
 * CargarValoresAction *  
 * Método para cargar los elementos de configuración  
 * de la cache desde un archivo XML.  
 * @autor Adrián Avila Atencio  
 * @return array  
 */
```

Figura 21: Ejemplo de comentario en una clase para describir una función.

Comentarios por líneas

Se utilizan también los comentarios para líneas de código en específico que ayudan a aclarar la complejidad de algunos algoritmos. Para ellos se utiliza `//`, y seguidamente el comentario que describe la línea de código, tal y como se observa a continuación:

```
//Elementos de configuración para MEMCACHED  
$host = (string) $xml->menccache->servers['host'];  
$puerto = (string) $xml->menccache->servers['port'];  
$persitent = (string) $xml->menccache->servers['persistent'];
```

Figura 22: Ejemplo de comentario por líneas de código.

3.4. Métricas de software

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento (34).

En general, la medición persigue tres objetivos fundamentales:

- Ayudar a entender qué ocurre durante el desarrollo y el mantenimiento.
- Permitir controlar qué es lo que ocurre en los proyectos.
- Poder mejorar los procesos y los productos.

A continuación se detallan los atributos que se miden en la validación de la solución:

Responsabilidad: Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.

Complejidad de implementación: Consiste en el grado de dificultad que se tiene al implementar el diseño de una clases determinada.

Reutilización: Consiste en el grado de reutilización que presente una clase o estructura de clase, dentro de un diseño de software.

Acoplamiento: Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.

Complejidad del mantenimiento: Consiste en el grado de esfuerzo necesario a realizar para realizar una corrección, una mejora o rectificación de algún error del diseño del software. Puede influir de manera indirecta, pero fuertemente en los costos y la planificación del proyecto.

Cantidad de pruebas: Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado.

Las métricas escogidas como instrumento para evaluar la calidad del diseño de la herramienta son las siguientes:

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad.

Tabla 3: Atributos de calidad evaluados por la métrica TOC.

Atributos de calidad	Modelo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de la implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Criterios y categorías de evaluación:

Tabla 4: Criterios de evaluación para la métrica TOC.

Atributos	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Complejidad de implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Reutilización	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

Tabla 5: Atributos de calidad evaluados por la métrica RC.

Atributos de calidad	Modelo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Criterios y categorías de evaluación:

Tabla 6: Criterios de evaluación para la métrica RC.

Atributos	Categoría	Criterio
Acoplamiento	Ninguno	0
	Baja	1
	Media	2
	Alta	>2
Complejidad de mantenimiento	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	$>2 \cdot$ Promedio
	Baja	$>2 \cdot$ Promedio
	Media	Entre Promedio y

Reutilización		2*Promedio
	Alta	<=Promedio
Cantidad de pruebas	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio

Resultados obtenidos de la aplicación de la métrica TOC:

La figura 26 muestra los resultados de la métrica Tamaño Operacional de la Clase (TOC), donde en la gráfica se mide el porcentaje de cantidad de procedimientos que contiene una clase. Como se puede observar existe un 21% de clases que contienen entre 6 procedimientos y 10 procedimientos y un 79 % que presentan menos de 6 operaciones, lo que significa un valor aceptable y satisfactorio para la métrica.

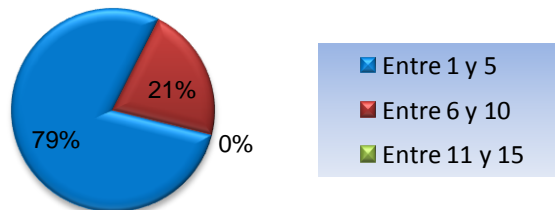


Figura 23: Representación del número de clases por procedimientos de la métrica TOC.

A continuación en la figura 27 se muestra la gráfica que mide el atributo responsabilidad de la métrica, obteniéndose resultados satisfactorios, mostrando una responsabilidad de un 72% y 21% media y baja respectivamente y el resto alta.

Responsabilidad

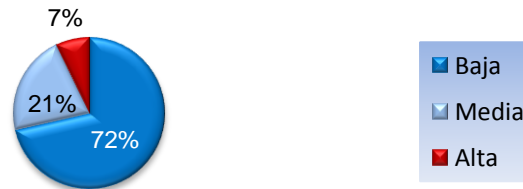


Figura 24: Resultados de la evaluación de la métrica TOC para el atributo responsabilidad.

El atributo que se mide en la gráfica que se muestra la figura 28 después de haberse realizado la medición de la métrica, arrojó también resultados positivos pues el 72% de las clases tiene complejidad baja, el 21% complejidad media y el resto alta.

Complejidad



Figura 25: Resultados de la evaluación de la métrica TOC para el atributo complejidad.

En la figura 29 se observa que el atributo la reutilización del código presenta resultados satisfactorios al brindar valores por encima del 50%.

Reutilización



Figura 26: Resultados de la evaluación de la métrica TOC para el atributo reutilización.

Resultados obtenidos de la aplicación de la métrica RC:

En la figura 30 se observa el resultado de la métrica Relación entre Clases (RC), donde se mide el porcentaje de cantidad de dependencias que contiene una clase. Como se puede apreciar existe un 93% de clases que contienen solo 2 dependencias y resto solo una dependencia lo que significa un valor aceptable y satisfactorio para esta métrica.



Figura 27: Representación del número de dependencias de las clases de la métrica RC.

La figura 31 muestra el resultado del atributo acoplamiento entre las clases del sistema, el cual presenta un 93% con valor medio y un 7% con bajo, lo que nos indica que las clases presentan una dependencia pobre y es favorable en el diseño de la solución.

Acoplamiento



Figura 28: Resultados de la evaluación de la métrica RC para el atributo acoplamiento.

Por otra parte los resultados del atributo complejidad de mantenimiento son aceptables, con un 100% de complejidad de mantenimiento media, lo que significa poca complejidad para mantener el código.

La medición de la variable cantidad de pruebas, brindó resultados admisibles al presentar un 100% de esfuerzo medio a la hora de realizar pruebas al diseño.

Finalmente se realizó la medición del atributo reutilización, para comprobar el nivel de reutilización de las clase diseñadas, la métrica generó resultados aceptables al indicar que el diseño de la solución posee un 93% y 7% de reutilización de media y alta respectivamente, este resultado es reflejado en la figura 34.

Reutilización

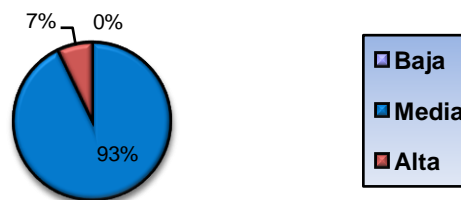


Figura 29: Resultados de la evaluación de la métrica RC para el atributo Reutilización.

3.5. Pruebas de caja blanca

Las pruebas de caja blanca, en ocasiones llamada prueba de caja de cristal, es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba.

Para realizar los casos de prueba primero se procede a enumerar las sentencias del código y a partir del mismo se construye el grafo de flujo asociado.

```
function cargarStoreAction() {  
    $json = array();    (1)  
    $xml = simplexml_load_file($this->_apps_path.'/comun/recursos/xml/emailnotifier.xml'); (1)  
    $to = (string) $xml->conn['to'];    (1)  
    if(strlen($to)>0){    (2)  
        $arreglo= explode(",",$to);    (3)  
        foreach($arreglo as $valor){    (4)  
            $json["to"] = $valor;    (5)  
        }    (6)  
    }  
    echo json_encode(array('root' => $json));    (7)  
}
```

Figura 30: Código fuente de la funcionalidad cargarStoreAction.

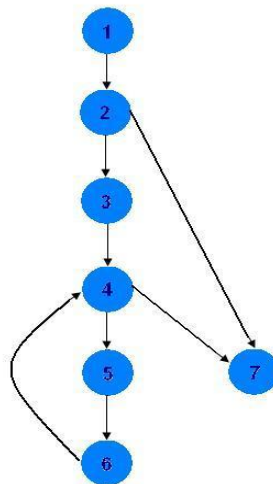


Figura 31: Grafo del flujo asociado a la funcionalidad cargarStoreAction.

Luego de realizar la construcción del grafo se realiza el cálculo de la complejidad ciclomática mediante las tres fórmulas descritas a continuación, las cuales deben arrojar el mismo resultado para asegurar que el cálculo de la complejidad es correcto.

1. $V(G) = (A - N) + 2$

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

$$V(G) = (8 - 7) + 2$$

$$V(G) = 3$$

2. $V(G) = P + 1$

Siendo “P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = 2 + 1$$

$$V(G) = 3$$

3. $V(G) = R$

Siendo “R” la cantidad total de regiones, para cada fórmula “V (G)” representa el valor del cálculo.

$$V(G) = 3$$

El cálculo efectuado utilizando las tres fórmulas antes descritas ha arrojado como resultado el mismo valor en todos los casos, el resultado fue 3, lo que indica que existen 3 posibles caminos por donde el flujo puede circular, y determina el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. Seguidamente se representan los caminos básicos por los que puede recorrer el flujo:

Camino básico #1: 1-2-7

Camino básico #3: 1-2-3-4-5-6-4-7

Camino básico #2: 1-2-3-4-7

Luego de determinar los caminos, se le realiza un caso de prueba a cada uno.

Caso de prueba para el Camino básico #1:

Condición de ejecución: El fichero XML no contiene los datos necesarios.

Entrada: N/A.

Resultados esperados: Se carga el fichero XML vacío.

Caso de prueba para el Camino básico #2:

Condición de ejecución: El fichero XML contiene los datos pero su estructura no es la correcta.

Entrada: N/A.

Resultados esperados: No se realiza la carga de un arreglo con los elementos deseados.

Caso de prueba para el Camino básico #3:

Condición de ejecución: El fichero XML contiene los datos y su estructura es correcta.

Entrada: N/A.

Resultados esperados: Se carga el fichero XML mostrando los datos que contiene.

Al aplicar los casos de prueba descritos anteriormente se comprobó que el flujo de trabajo de la función es correcto pues cumplió con las condiciones necesarias de la prueba.

3.6. Pruebas de caja negra

Las pruebas de caja negra, también denominadas, pruebas de comportamiento, se concentran en los requisitos funcionales del software. Permiten derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa.

Estas pruebas se realizan directamente sobre la interfaz del software entrando una serie de datos para comprobar el correcto manejo de los mismos.

A continuación se muestran resultados obtenidos durante la realización de las pruebas de caja negra. Estas arrojaron resultados satisfactorios a los efectos de la implementación de la propuesta de solución.

Caso de prueba: Modificar elementos de configuración de mensajería.

Condiciones de Ejecución

El administrador del sistema debe estar autenticado correctamente en el sistema.

Tabla 7: Caso de prueba Modificar elementos de configuración de mensajería.

Escenario	Descripción	Variable 1 (Activo)	Variable 2 (Servidor)	Variable 3 (Puerto)	Respuesta del sistema	Flujo central
EP 1.1: Configurar mensajería.	La funcionalidad comienza cuando el Administrador del sistema desea configurar el servidor de correo, luego de la correcta selección de la	V	V	V	El sistema almacena los elementos de configuración n. - Se	Se muestra un interfaz que brinda la posibilidad de modificar los elementos de configuración del
		Checked Unchecked	Lab305-pc7	5222		
		Variable 4 (Usuario)	Variable 5 (Contraseña)			

	funcionalidad en el menú de administración se procede a entrar al sistema los elementos de configuración. La funcionalidad termina cuando el sistema guarda definitivamente la operación correspondiente.	V	V		muestra un mensaje de información "Se guardaron los datos correctamente."	servidor de mensajería. Se muestran además los botones "Aplicar" y "Cancelar". Si se presiona el botón "Aplicar" se modifican los elementos de configuración.
		admin	A@dmin			
EP 1.2: Campos vacíos.		V	V	V	El sistema muestra un mensaje de error para que llene los campos correctamente e "Datos incorrectos".	Se muestra un mensaje al lado de los campos vacíos y se señalan en rojo. Se permite entrar nuevamente los datos solicitados.
		Checked Unchecked	(Vacío)	(Vacío)		
		V	V			
		pcLab	pcPr0yecto			
EP 1.3 Campos incorrectos		V	F	F	El sistema muestra un mensaje de error "Datos incorrectos".	Se muestra el campo incorrecto señalado en rojo.
		Checked Unchecked	pC&%	0		
		V	V			
		admin	A@dm1n			
EP 1.4: Cancelar operación		V	V	V	Se cierra la ventana sin realizar ninguna operación.	Cancela la operación y vuelve a la interfaz previa.
		N/A	N/A	N/A		
		V	V			
		N/A	N/A			

Descripción de variables

Tabla 8: Descripción de variables.

No	Nombre del campo	Clasificación	Valor nulo	Descripción
----	------------------	---------------	------------	-------------

1	activo	Chekcbox	N/A	Activa o desactiva el componente de mensajería.
2	servidor	Campo de texto	No	Nombre del ordenador servidor.
3	puerto	Campo de texto	No	Puerto utilizado por el componente de mensajería.
4	usuario	Campo de texto	No	Nombre del usuario.
5	contraseña	Campo de texto	No	Contraseña del usuario.

A continuación se puede observar el número de no conformidades detectadas durante la ejecución de las pruebas de caja negra.

Tabla 9: Resultados de la prueba por requisitos funcionales.

Requisitos	Nro. NC	Resueltas	%
Modificar elementos configuración de base de datos	1	1	100
Modificar elementos configuración de correo	2	2	100
Modificar elementos configuración de mensajería	2	2	100
Modificar elementos configuración del servidor Memcached	5	4	80
Modificar elementos configuración del servidor File	4	3	75

Tabla 10: Resultados de la prueba de NC.

Tipo NC	Nro.NC	Resueltas	%
Interfaz de usuario	12	12	100
Validación	8	7	87,5
Funcionalidad	4	3	75

Total	24	22	91,7
-------	----	----	------

Luego de ser corregidos los errores encontrados en las pruebas, se pudo comprobar que el flujo de trabajo de las interfaces estaba correcto, debido que cumple con las condiciones necesarias que se habían definido.

3.7. Validación de la solución

Método experto

El método de experto consiste en la selección de un grupo de especialistas que deberán evaluar la propuesta individualmente y de forma anónima. Se trata de llegar a un consenso y analizar los aspectos de discrepancia, permitiendo además que:

- Ningún miembro del grupo de expertos sea influenciado por la reputación de otro de los miembros.
- Un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
- El experto pueda defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.

Para llevar a cabo el método se realizó la selección un grupo expertos de CEIGE que trabajan directamente con las configuraciones del marco de trabajo Sauxe, teniendo en cuenta su disposición a participar en la encuesta y apreciación en el tema.

Los expertos se seleccionaron según los criterios siguientes:

- Dos años de experiencia como mínimo.
- Vinculación al desarrollo de productos informáticos.
- Conocimientos acerca del desarrollo de software.

Elaboración del cuestionario

Una vez seleccionados los expertos, se continúa con la elaboración de la encuesta para la validación de la propuesta, para ello se formularon 11 preguntas dirigidas a valorar los elementos definidos para asegurar la usabilidad y seguridad, durante la configuración del sistema. El cuestionario fue conformado de forma tal que las respuestas fueran categorizadas

en (Muy adecuado (C1), Bastante adecuado (C2), Adecuado (C3), Poco adecuado (C4) y No adecuado (C5). El cuestionario de validación utilizado en la investigación se muestra en la (Tabla 13 de los Anexos).

Resultados del cuestionario

Se analizaron los resultados de la aplicación del cuestionario, donde se evalúa, por parte de los expertos, elementos referidos a la usabilidad y seguridad del sistema.

Tabla 11: Evaluación por preguntas de los expertos.

Preguntas	Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
1	2	3			
2	3	2			
3	3	2			
4	2	2	1		
5	2	3			
6	4	1			
7	2	2	1		
8	2	3			
9	1	4			
10	1	3	1		
11	3	2			

Evaluaciones

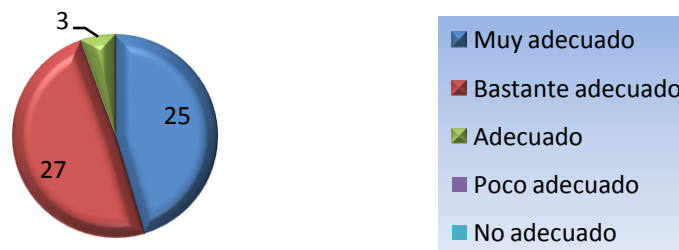


Figura 32: Evaluaciones por categorías.

Los resultados arrojados por la encuesta a los expertos fueron satisfactorios, para mayor confirmación se analizan los resultados estadísticamente, se procede a determinar el grado de concordancia entre los expertos, con respecto a las evaluaciones que hicieron. Para ello se determinó utilizar el coeficiente de Kendall (W).

Determinación de la concordancia de los especialistas mediante el coeficiente de Kendall

El coeficiente de Kendall mide el grado de asociación entre varios conjuntos (k) de N entidades. Es útil para determinar el grado de acuerdo entre varios jueces, o la asociación entre tres o más variables. El Coeficiente de Concordancia de Kendall (W), ofrece el valor que posibilita decidir el nivel de concordancia entre los especialistas. El valor de W oscila entre 0 y 1. El valor de uno significa una concordancia de acuerdo total y el valor de cero un desacuerdo total.

Luego de realizar el cuestionario se verificará si existe concordancia entre los expertos mediante la siguiente fórmula (35):

$$(1). W = \frac{12 * S}{K^2 (N^3 + N)}$$

Dónde: W: es el coeficiente de concordancia. S: es la suma de los cuadrados de las desviaciones observadas de la media de S_j (rangos) y se calcula mediante la expresión:

$$(2). S = \sum_{j=1}^n (S_j - \bar{S})^2$$

Dónde: \bar{S} es la suma de los rangos dividido entre la cantidad de preguntas. N: cantidad de preguntas en este caso 11. K: Cantidad de expertos, por tanto K=5.

$$(3). \bar{S} = \frac{\sum_{j=1}^n S_j}{N}$$

A continuación se muestran los cálculos realizados para determinar la concordancia de los expertos:

Tabla 12: Datos obtenidos de la encuesta.

	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Sj
Pregunta 1	5	5	4	4	4	22
Pregunta 2	5	5	4	5	4	23
Pregunta 3	5	4	5	5	4	23
Pregunta 4	5	5	4	3	4	21
Pregunta 5	5	4	4	4	5	22
Pregunta 6	5	5	5	5	4	24
Pregunta 7	5	4	3	4	5	21
Pregunta 8	4	5	4	4	5	22
Pregunta 9	5	4	4	4	4	21
Pregunta 10	5	4	4	4	3	20
Pregunta 11	5	5	4	5	4	23

Sustituyendo los valores en la ecuación (3), se calcula el valor de \bar{S} y se obtiene como resultado:

$$\bar{S} = \frac{242}{11} = 22$$

Luego para determinar la desviación media (S) se sustituye los valores en la ecuación (2) y se obtiene como resultado:

$$S = 14$$

Sustituyendo los valores en la ecuación (1), se obtiene el valor de Kendall:

$$W = \frac{12 * 14}{5^2 (11^3 + 11)}$$

$$W = 0,005$$

Una vez calculado el coeficiente de Kendall (W) es necesario conocer su probabilidad de significancia, para lo cual se debe transformar éste valor en Chi cuadrado de Pearson mediante la fórmula siguiente:

$$X_{cat}^2 = K(N - 1)w$$

Sustituyendo los valores en la ecuación se obtiene:

$$X^2 = 5(11 - 1)0,005$$

$$X^2 = 0,25$$

Para buscar el Chi cuadrado en la tabla de distribución reflejada en la (Figura 37 de los Anexos), se procede a calcular el grado de libertad (gl) donde: $gl = N-1$

Sustituyendo los valores en la ecuación se obtiene: $gl = 11-1=10$

$X^2_{tabla} = X^2_{\alpha,gl}$ donde α es el nivel de significación utilizado para calcular el nivel de confianza. El nivel de confianza es igual a $100(1 - \alpha)$, un $\alpha=0,05$ indica un nivel de confianza de 95 %. Una vez obtenidos los valores se compara $X^2_{tabla} = X^2_{real}$, si $X^2_{tabla} > X^2_{real}$ entonces existe concordancia de criterios entre los expertos.

$$X^2_{tabla} > X^2_{real}$$

$$18,31 > 0,25$$

Por lo tanto se puede afirmar que existe concordancia entre las evaluaciones realizadas por los expertos.

3.8. Conclusiones

Al concluir el presente capítulo se confirma que:

- Las métricas TOC y RC permitieron valorar el diseño propuesto, obteniéndose resultados positivos que benefician la alta reutilización y el bajo acoplamiento entre las clases que fueron diseñadas.
- Los resultados de la realización de pruebas de caja negra evidencian que las entradas de datos a la herramienta se aceptan de forma adecuada y que se producen salidas correctas. Mientras que las pruebas de caja blanca demuestran que las operaciones internas se ajustan a las especificaciones.
- El análisis estadístico de los resultados obtenidos en la encuesta, a través aplicación del Coeficiente de Concordancia de Kendall, demostró la alta probabilidad que tiene la solución de ser efectiva.

CONCLUSIONES GENERALES

Durante el desarrollo del presente trabajo se logró cumplir con los objetivos trazados que abarcaron todas las tareas investigativas propuestas, arribando a las siguientes conclusiones:

- A partir del análisis de las aplicaciones web existentes que cuentan con un panel de administración para la configuración de su sistema, se demostró la necesidad e importancia de desarrollar una herramienta para la administración del marco de trabajo Sauxe.
- La especificación de requisitos permitió que los requerimientos de la herramienta desarrollada fueran especificados y validados, lo que garantizó estabilidad en la propuesta de solución.
- La realización del análisis y diseño de la solución permitió una comprensión de los aspectos relacionados con las funcionalidades y restricciones que debe cumplir la herramienta, igualmente creó una entrada ajustada y un punto de partida para las actividades en el desarrollo de la solución.
- Se desarrolló la solución para la administración del marco de trabajo Sauxe empleando tecnologías, herramientas y técnicas aplicadas como parte de la fundamentación teórica.
- Los resultados de las pruebas estructurales realizadas permitieron comprobar el correcto funcionamiento de la herramienta, la eficiencia y el buen funcionamiento del código implementado.
- Con la utilización del método experto y el análisis estadístico realizado con el Coeficiente de Concordancia de Kendall, se pudo corroborar que la solución propuesta tiene alta probabilidad de acrecentar la seguridad y la usabilidad en el proceso de configuración del sistema, tarea que hasta el momento está siendo realizada por los desarrolladores manualmente.

RECOMENDACIONES

Con vistas a mejorar la solución en futuras versiones se recomienda:

- Implementar las funcionalidades para configurar los adaptadores de backend Xcache y Sqlite que utiliza el componente Zend_Cache.
- Implementar la configuración de varios de servidores de caché para el adaptador de backend Memcached.
- Implementación de las funcionalidades de configuración de los orígenes de recursos del servidor multimedia.
- Continuar perfeccionando la herramienta analizada a partir de los nuevos requisitos que puedan surgir como resultado de su uso.

REFERENCIA BIBLIOGRÁFICA

1. **UVA.** *Introducción a las aplicaciones web.*
2. DDW.DesarrollodeWeb. [En línea] <http://www.desarrollodeweb.com.ar/archivo/25-como-encargar-un-proyecto-web>.
3. El Prisma. Portal para Investigadores y profesionales. *Definición de Administración.* [En línea] http://www.elprisma.com/apuntes/administracion_de_empresas/definicionadministracion/.
4. definicionabc. [En línea] <http://www.definicionabc.com/tecnologia/configuracion.php>.
5. Blog de WordPress.com. . [En línea] http://protejete.wordpress.com/gdr_principal/definicion_si/.
6. HyperText.net. [En línea] http://www.upf.edu/hipertextnet/numero-2/disenio_web.html.
7. **Zaninotto, Fabien Potencier y Francois.** *Symfony la guía definitiva.* s.l. : Apress, 2008. 978-1590597866.
8. **Sánchez, Jordi.** jordisan.net. [En línea] <http://jordisan.net/blog/2006/que-es-un-framework>.
9. **Rodríguez, Ing. Juan Jose Rosales.** *Herramienta de actualización para el Sistema Integral de Gestión Cedrux.*
10. Actualizaciones. [En línea] <http://www.seom.es/cms/actualizar-modulos-en-drupal/>.
11. Cache. [En línea] <http://www.gnuMLA.com/joomla/articulos/tutorial-uso-de-la-cache-en-joomla-15.html>.
12. *Producción, Equipo de. Modelo de Desarrollo orientado a componentes del proyecto ERP-CUBA.*
13. **entidades, Centro de Informatización de gestión de.** *Modelo de desarrollo de software.* 2012.
14. **Pupo, Yanisleydi Cañete.** *Libro de Ayuda del Marco de Trabajo Sauxe, versión 2.0.* Ciudad Habana : s.n., 2010.
15. PHP Group. [En línea] 2001. <http://www.php.net/manual/es/intro-what-is.php>.
16. Territorio PC. [En línea] 2001. http://www.territoriopc.com/javascript/tutorial_javascript_introduccion.php.
17. **Quin., Liam R. E.** W3C. [En línea] <http://www.w3.org/standards/xml/core..>
18. *Secure Programming with the Zend-Framework.* **Esser, Estefan.** Amsterdam : s.n., 2009.
19. [En línea] <http://www.techastico.com/.../zend-framework-una-introduccion>.
20. **Mata, Manel Pérez.** TecnoRetales. [En línea] 3 de Septiembre de 2009. <http://www.tecnoretalles.com/programacion/que-es-doctrine-orm>.
21. **Sencha Inc.** [En línea] 2006. <http://www.sencha.com/products/extjs>.
22. Visual Paradigm website. *Visual modeling tool for building enterprise applications.* [En línea] 2011. <http://www.visual-paradigm.com/product/vpuml/provides/>.
23. **Barrieto, Manuel Sánchez.** QDiario Magazines. [En línea] 6 de Febrero de 2008. <http://www.aplicacionesempresariales.com/version-83-de-postgresql.html>.
24. Firebug | CNET Descargar. [En línea] http://es.download.cnet.com/Firebug/3000-11745_4-10616509.html#ixzz2TM8MoGiW.
25. prmob. [En línea] <http://es.prmob.net/firebug/mozilla-firefox/html-645984.html>.
26. **Kabir, Mohammed J.** *La biblia Servidor Apache2.* s.l. : Anaya.

27. **Ben Collins-Sussman, Brian W. Fitzpatrick, y Michael C. Pilato.** Version Control with Subversion. [En línea] 2002. <http://svnbook.red-bean.com/nightly/es/svn-book.pdf>.
28. **Böck, Heiko.** *The Definitive Guide to NetBeans Platform*. s.l. : Apress, 2009.
29. Netbeans. *symfony.es*. [En línea] <http://www.symfony.es/2009/10/05/netbeans-ya-incluye-soporte-para-symfony..>
30. **otros, Erich Gamma y.** *Design Patterns: Elements of Reusable Object-Oriented Software*. s.l. : Addison-Wesley, 1994. 0-201-63361-2.
31. **Pressman, Roger S.** *Ingeniería de Software*. Sexta Edición.
32. Diagrama despliegue. [En línea] <http://diadspg.blogspot.com/>.
33. Estandares. [En línea] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
34. **Muñoz, Dra. Coral Calero.** *MÉTRICAS DEL SOFTWARE: Conceptos básicos, definición y formalización*. 2006.
35. scielo. [En línea] http://scielo.sld.cu/scielo.php?pid=S2071-00542010000200001&script=sci_arttext.
36. **Grupo Global de Desarrollo de PostgreSQL.** [En línea] 1996. <http://www.postgresql.org/about/press/presskit83.html.es..>