

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



**Módulo Correspondencias del Sistema de Control de Documentos del Ministerio
del Poder Popular de la Energía y Petróleo de Venezuela**

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

Autores:

Nilo Cabeza Díaz

Jesús Monterrubio Abstengo

Tutor:

Ing. Dayelis Blanco Hernández

Ciudad de La Habana

Junio de 2009

*El futuro pertenece a quienes creen en la belleza de sus
sueños...*



Anna Eleanor Roosevelt

Declaración de Autoría

Declaramos ser los autores del presente trabajo de diploma y autorizamos a la Universidad de las Ciencias Informáticas para que hagan uso pertinente de este trabajo.

Para que así conste firmamos el presente a los 17 días del mes de junio de 2009.

Firma de Autor

Nilo Cabeza Díaz

Firma de Autor

Jesús Monterrubio Abstengo

Firma Tutora

Ing. Dayelis Blanco Hernández

Agradecimientos

A mi familia por apoyarme en todo momento e impulsarme a lograr este sueño, gracias por su confianza.

A mi novia, por ser tú y estar ahí siempre para mí, gracias por todo.

A mis amigos que los llevo en el corazón y son como hermanos.

A mi compañero de tesis, por su sacrificio y entrega total a pesar de las dificultades.

A mi tutora por su confianza.

Y a todos los que de una forma u otra contribuyeron a que este sueño se hiciera realidad.

JMA

A todas la personas que en algún momento me dieron su apoyo para poder terminar la tesis

A mi novia, por estas en todo momento conmigo.

A mi compañero de tesis, por su sacrificio y entrega total a pesar de las dificultades.

A mi tutora por su confianza.

NCD

Dedicatoria

A mi mamá, mis hermanas y en especial a mi papá que ya no esta conmigo.

A todos mis familiares y a mi novia que la quiero con la vida.

NCD

Resumen

Con el presente trabajo se propone la implementación del Módulo Correspondencias del Sistema de Control Documentos (SCD) del Ministerio de Energía y Petróleo de Venezuela (MENPET) como parte de su estrategia por mejorar los procesos que actualmente se desarrollan en esa institución y en particular procesos relacionados con la gestión de las correspondencias. Su desarrollo está basado en tecnologías libres en cumplimiento con políticas y estrategias de dicho organismo, siendo un sistema multiplataforma, donde algunos elementos presentes en su arquitectura son: el modelo Cliente-Servidor, y sobre una arquitectura en capas utilizando PHP 5 como lenguaje de programación, se implementa el patrón arquitectónico Modelo Vista Controlador (MVC) a través del framework CodeIgniter 1.7, como repositorio de información el sistema de gestión documental Alfresco 3.0 y como metodología de desarrollo RUP. Para ello se hace una valoración crítica del análisis y diseño previamente propuesto, realizándose los cambios pertinentes acorde a los requerimientos del sistema durante el flujo de trabajo de implementación y posteriormente se valida la propuesta de solución a través del desarrollo de un conjunto de pruebas realizadas al módulo implementado con el objetivo de asegurar el correcto funcionamiento del mismo y la calidad requerida para lograr una mayor satisfacción del cliente.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica	6
1.1 Breve acercamiento al concepto de gestión documental	6
1.1.1 Evolución de la Gestión Documental	6
1.1.2 Profundizando en el tema	7
1.1.3 Sistemas de Gestión Documental	7
1.1.4 Alfresco. Propuesta de gestor documental a utilizar	8
1.1.4.1 Beneficios	8
1.1.4.2 Reconocimientos	9
1.1.5 ¿Por qué es necesaria la gestión documental digital?	10
1.1.6 Principales razones para digitalizar	10
1.1.6.1 Beneficios	11
1.1.6.2 Riesgos de la Gestión Documental Digital	12
1.2 Paradigmas de Programación	13
1.2.1 El Paradigma Imperativo	13
1.2.2 El Paradigma Funcional	13
1.2.3 El Paradigma Lógico	14
1.2.4 Programación Orientada a Objetos (POO)	14
1.2.5 La Programación Orientada a Servicios	14
1.3 Frameworks PHP	15
1.3.1 Symfony	16
1.3.2 Zend Framework	16
1.3.3 CodeIgniter	17
1.4 Entornos de Desarrollo para PHP	18
1.4.1 Zend Studio	19
1.4.2 Adobe Dreamweaver	19
1.4.3 Netbeans 6.5 para PHP	20
1.5 Tendencias y tecnologías actuales	21
1.5.1 Uso de software libre	21
1.5.2 Aplicaciones Web	21
1.5.3 Modelo Cliente Servidor	24
1.5.4 Servicios Web	25
1.5.4.1 Protocolos que usan los Servicios Web	25
1.5.4.1.1 XML	25
1.5.4.1.2 SOAP	25
1.5.4.1.3 WSDL	26
1.5.4.1.4 UDDI	26
1.5.5 Arquitectura Orientada a Servicios SOA (Service Oriented Architecture)	26
1.5.6 Lenguaje PHP5	27
1.5.7 Servidor Web Apache	27
1.5.8 Lenguaje XML	28
1.5.9 ECM Alfresco	28
1.5.10 Patrones de arquitectura	29
1.5.10.1 Patrón Modelo-Vista-Controlador	29

1.5.11 RUP	30
Capítulo 2. Descripción y análisis de la solución propuesta.....	32
2.1 Sentar las bases de la arquitectura	32
2.2 Valoración crítica del diseño propuesto por el analista	33
2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados	43
2.3.1 Codelgniter.....	43
2.3.2 Web Services	44
2.3.3 Clase de sesión.....	45
2.4 Descripción de los algoritmos no triviales a implementar.....	45
2.4.1 Búsqueda avanzada	45
2.4.2 Crear la correspondencia entrante.....	46
2.5 Descripción de las nuevas clases u operaciones necesarias.....	47
2.5.1 Clase Controladora	48
2.5.2 Clase Modelo	51
2.5.3 Clases Interfaz	53
Capítulo 3. Validación de la solución propuesta.....	58
3.1 Búsqueda o diseño de los test de unidades que permitan validar la solución propuesta	58
3.1.1 Niveles de prueba	58
3.1.1.1 Nivel de Unidad.....	59
3.1.1.2 Nivel de Integración	60
3.1.1.3 Nivel de Sistema	60
3.1.1.4 Nivel de Aceptación.....	60
3.1.2 Pruebas a realizar en tiempo de desarrollo	61
3.1.3 Método de caja negra	61
3.2 Descripción de los test de caja negra.....	63
3.2.1 CPR 1: Adicionar la correspondencia entrante en el estado “Para Redactar”	63
3.2.2 CPR 2: Adicionar la correspondencia entrante en el estado “Para Revisar”.....	65
3.2.3 CPR 3: Asociar “remitente” a la correspondencia entrante.....	67
3.2.4 CPR 4: Asociar “destinatario” a la correspondencia entrante	69
3.2.5 CPR 5: Asociar “preparado por” a la correspondencia entrante	70
3.2.6 CPR 6: Adjuntar “documento anexo” a la correspondencia entrante.....	72
Conclusiones	76
Recomendaciones	77
Referencias Bibliográficas.....	78
Bibliografía	81
Anexos	85
Glosario de Términos	92

Introducción

La información, constituye un factor clave para el progreso social, político y económico; procesa y genera el conocimiento, ampliando las posibilidades del pensamiento humano. Es de vital importancia para el desarrollo y equilibrio de los sistemas, organizaciones, sectores y países. La organización y planificación necesitan información, así como la toma de decisiones; para ello esta debe ser concreta, precisa, coherente y adaptada a las necesidades que se desean satisfacer.

Los servicios de información y documentación accesibles a través de una intranet, más concretamente mediante servidores web, están aumentando de una forma exponencial. La lógica y evolución de los servicios web desde hace más de diez años ha ido produciendo la sustitución de páginas y documentos estáticos por documentos generados dinámicamente, a merced de la interacción del usuario con la lógica de procesos y flujos de trabajo definida por los creadores del servicio y a la disponibilidad de cada vez mayores repositorios de información. Evidentemente, se ha ido pasando progresivamente de un concepto de publicación de páginas web, bastante simple en su origen, a esquemas más complejos y diferenciados, fundamentados en procedimientos y técnicas basados en la gestión de información que no es más que el uso de tecnologías y procedimientos que permiten la gestión y el acceso unificado a información generada en la organización.(1)

La gestión documental es el conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos en una organización, permitir la recuperación de información, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no tienen utilidad y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía.(2)

En la actualidad, coexisten en el mundo los más diversos sistemas de gestión documental: desde el simple registro manual de la correspondencia que entra y sale, hasta los más sofisticados sistemas informáticos que manejan no sólo la documentación administrativa propiamente tal, venga ella en papel o en formato electrónico, sino que además controlan los flujos de trabajo del proceso de tramitación de los expedientes, capturan información desde bases de datos de producción, contabilidad y otros, enlazan con el contenido de archivos, bibliotecas, centros de documentación y permiten realizar búsquedas sofisticadas y recuperar información de cualquier lugar.(3)

Este es el caso de Alfresco, que actualmente goza de gran prestigio y su utilización se ha masificado considerablemente frente a otros sistemas similares. Además es la alternativa principal de código abierto para la gestión del contenido empresarial. También por ser de código abierto le permite utilizar las tecnologías y las contribuciones de la comunidad del código abierto para conseguir un software de alta calidad produciéndolo más rápidamente con un costo mucho más bajo.

Ante toda esta revolución en el uso de la información, de los sistemas y servicios informáticos, en Venezuela se han tomado medidas para no quedarse rezagada. Entre ellas destaca el decreto No 3.390 de fecha 23/12/2004, gaceta No 38.095 de fecha 28/12/2004, en el cual establece que la Administración Pública Nacional empleará prioritariamente Software Libre desarrollado bajo Estándares Abiertos, en sus Sistemas, Proyectos y Servicios Informáticos.

De ahí que el sistema de control de documentos (SCD) es una parte importante dentro de la estrategia empresarial del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET).

El Ministerio del Poder Popular de Energía y Petróleo de Venezuela (MENPET) fue creado en el año 2005, es la entidad gubernamental encargada de la regulación, formulación y seguimiento de políticas, la planificación, realización y fiscalización de las actividades del Ejecutivo Nacional Venezolano en materia de hidrocarburos y energía en general. Efectúa estudios de mercado y análisis, fijar precios de los productos de petróleo y del servicio eléctrico.(4)

El MENPET posee un Sistema Interno Central (SIC) que permite llevar el control y seguimiento de los procesos de reclutamiento de personal, así como también, mantener un registro de la trayectoria de los trabajadores, registro de estructura de cargos propuestos y aprobados. Este Sistema Interno Central tiene la mayoría de sus aplicaciones migradas al software libre, adicionalmente ofrece formas de integración con los sistemas internos y externos.

Dicho Ministerio ha estado utilizando en sus oficinas desde hace alrededor de 9 años un sistema denominado Sistema de Operaciones Secretariales (S.O.S.) para el control de toda la correspondencia, tanto interna como externa, entendiéndose por correspondencia todo el flujo constante de documentos ya sea en formato duro o digital que se generan diariamente entre las diferentes dependencias y oficinas de la sede del MENPET. Dicho sistema que es una aplicación de escritorio sobre plataforma privativa, fue desarrollado utilizando Borland Delphi 6.0 y Microsoft Access 2000,

además el sistema no cumple con las necesidades reales del Ministerio y no está integrado al Sistema Interno Central que constituye la base de todas las aplicaciones dentro del MENPET.

Además entre las diferentes dependencias y oficinas de la sede del Ministerio del Poder Popular para la Energía y Petróleo existe un constante intercambio de documentos generados por ellas o por entidades externas de los cuales no se lleva un estricto seguimiento y control.

En la actualidad la entidad, no cuenta con una aplicación informática que garantice la gestión de todas las correspondencias tanto entrantes como salientes de manera eficaz. Hoy en día las búsquedas y recuperación de documentos se realizan de modo manual, algunos nunca llegan a digitalizarse y por consiguiente corren el riesgo de pérdida o deterioro, todo esto afecta directamente el desempeño óptimo de los trabajadores en el Ministerio trayendo consigo atraso y poca eficiencia de los mismos y a su vez repercute en la organización, planificación y toma de decisiones en el MENPET.

Enmarcando el siguiente **problema científico**: Necesidad de gestionar eficientemente los procesos documentales del Ministerio del Poder Popular de Energía y Petróleo de Venezuela.

Se toma como **objeto de estudio**: los procesos de gestión documental en el Ministerio del Poder Popular de Energía y Petróleo de Venezuela y se estará profundizando en el **campo de acción**: la gestión de las correspondencias.

El **objetivo general** de la investigación es: Implementar un sistema que gestione eficientemente las correspondencias en el Ministerio del Poder Popular de Energía y Petróleo de Venezuela.

Surgiendo los siguientes **objetivos específicos**:

1. Analizar las tecnologías, herramientas y metodologías a utilizar en la implementación del Sistema de Control de Documentos.
2. Desarrollar la implementación del módulo correspondencias para el Sistema de Control de Documentos del MENPET.
3. Validar la propuesta de solución.

Para dar cumplimiento a los objetivos específicos se trazan las siguientes **tareas**:

1. Estudiar las tecnologías y herramientas necesarias para la implementación del sistema.
2. Seleccionar las tecnologías y herramientas libres necesarias para la implementación del

sistema.

3. Valorar el diseño propuesto inicialmente por el analista.
4. Realizar los cambios necesarios en el diseño propuesto.
5. Estudiar los posibles componentes, clases y librerías que pudieran ser reutilizados para la implementación del sistema.
6. Implementar el módulo correspondencias.
7. Diseñar los test de unidades para la validación de la solución propuesta.
8. Describir los test de unidad y sus valores utilizados.
9. Evaluar la ejecución del test y los resultados obtenidos.

Los métodos de la investigación son:

Métodos teóricos:

- ✓ Análisis - Síntesis.

Este método ha servido para analizar y comprender la teoría y documentación relacionada con nuestro tema de investigación, permitiendo así extraer los elementos más importantes y relacionados con el objeto de estudio.

Métodos Empíricos:

- ✓ Observación.

Se ha utilizado este método sobre otras investigaciones que están estrechamente relacionadas o tienen elementos en común con esta investigación. También se ha observado la situación real que se está investigando, permitiendo acercarse al objetivo final.

Estructuración del contenido

El desarrollo de este trabajo consta de tres capítulos de los cuales se hará una breve descripción a continuación:

En el **Capítulo 1** se abordarán diferentes temas entre los que se encuentran la gestión documental, la evolución de la misma y se profundiza en la importancia de la digitalización. También se abordarán características y conceptos importantes relacionados con los sistemas de Gestión Documental y en especial de Alfresco. También se hará un estudio y una valoración crítica de los diferentes Paradigmas de Programación, Entornos de Desarrollo y Frameworks para PHP. Además se presentará una panorámica general de las tendencias actuales de las tecnologías, herramientas, y metodologías de desarrollo de software utilizadas, teniendo en cuenta la arquitectura general del proyecto al cual pertenece este módulo.

El **Capítulo 2** se centrará en la descripción de la implementación del sistema, para ello, se sentarán las bases de la arquitectura propuesta, se realizará una valoración crítica del diseño propuesto por el analista, donde se estudiarán y explicarán los cambios necesarios para la transición del diseño a la implementación, teniendo en cuenta los requerimientos de la aplicación. Se abordarán cuestiones referentes a los componentes a reutilizar, como clases o librerías previamente implementadas y que fueron utilizadas en la implementación del módulo. Posteriormente se describirán los algoritmos no triviales a implementar, para finalmente describir las nuevas clases y operaciones necesarias para darle solución a la situación problemática.

En el **Capítulo 3** se describirán los niveles de prueba durante las fases de desarrollo de un software, haciendo énfasis en las pruebas a realizar en tiempo de desarrollo y las de nivel de unidad. Se profundizará en el Método de Caja Negra y dentro de este en la técnica de la Partición de Equivalencia. Posteriormente se describirán los test de caja negra y los resultados obtenidos de los mismos que permitirán validar la solución propuesta.

Capítulo 1. Fundamentación teórica

En el presente Capítulo se abordan diferentes temas entre los que se encuentran la gestión documental, la evolución de la misma y se profundiza en la importancia de la digitalización. También se abordan características y conceptos importantes relacionados con los sistemas de Gestión Documental y en especial de Alfresco. También se hace un estudio y valoración crítica de los diferentes Paradigmas de Programación, Entornos de Desarrollo y Frameworks para PHP. Además se presenta una panorámica general de las tendencias actuales de las tecnologías, herramientas, y metodologías de desarrollo de software utilizadas, teniendo en cuenta la arquitectura general del proyecto al cual pertenece este módulo.

1.1 Breve acercamiento al concepto de gestión documental

Desde el inicio de la humanidad, el hombre a través de los diferentes materiales dejó constancia de sus actividades y pensamientos. Durante todos estos siglos cada uno de sus nuevos soportes prestaba ventajas de uso en comparación a los precedidos. En la actualidad, debido a la gran cantidad de información en papel y otros soportes que se generan en las empresas, surge la necesidad de la creación de archivos masivos que permitan almacenar y controlar toda información, además de otras ventajas como por ejemplo la manipulación de la misma en el menor tiempo posible y la perdurabilidad. Toda esta optimización es posible gracias al avance y desarrollo de las tecnologías y las comunicaciones, las cuales favorecen en gran medida el surgimiento de esta nueva óptica de manipulación de la información.

El proceso que mediante el uso de la tecnología gestiona el acceso unificado a la información se le conoce como Gestión Documental y constituye un desafío de muchas organizaciones en sus intentos de lograr controlar todo tipo de información en grandes volúmenes.

1.1.1 Evolución de la Gestión Documental

El término Gestión Documental puede hacer pensar en un montón de papeles y archivos apilados en espera de ser archivados. Si se tiene experiencia con la gestión de documentos digitales podría pensarse en una estación de escaneo y un repositorio de imágenes. Sin embargo, la gestión documental es mucho más que esos conceptos. Nuevas tecnologías están continuamente emergiendo haciendo la gestión documental más funcional y sofisticada, con más capacidades. Con estas evoluciones, la tecnología de la gestión documental se ha convertido más en una solución de

productividad que en una simple solución de conversión de papeles en imágenes.

La Gestión Documental Electrónica está ganando adeptos continuamente pues las tecnologías asociadas a este proceso se han hecho cada vez más fáciles de usar. Primero, la mayoría de los escáneres en el nivel de grupo de trabajo están equipados con la tecnología que evita los múltiples pasos a veces involucrados en el envío de documentos escaneados a otras aplicaciones o dispositivos. También existen algunos escáneres de bajo volumen con software embebido para el tratamiento de imágenes, de esta manera las pequeñas empresas no se ven obligadas a comprar software adicional. Estos programas pueden reconocer malos escaneos así como letras borrosas e imágenes cortadas.

Empresas sin muchos recursos tecnológicos utilizan sistemas integradores o los servicios de un vendedor para configurar y mantener sus soluciones de Gestión Documental; sin embargo, de acuerdo a los avances en el software para la Gestión Documental eso está cambiando. Las soluciones se han convertido más en aplicaciones guiadas por asistentes. Los programas formulan las preguntas en lugar de los usuarios imaginar que preguntas deberían hacerse cuando intentan modificar algo.

Así como el software se ha hecho más fácil de usar también se ha hecho más comprensible. Para manejar varias formas de documentos y mantener la productividad del negocio, el software para la Gestión Documental incluye funcionalidades como la automatización de flujos de trabajo, horarios de retención, revisiones de control y auditorías.

1.1.2 Profundizando en el tema

En una institución u organización se generan diariamente grandes cantidades de de información en papel que se almacenan en enormes archivos requiriendo de condiciones para el almacenamiento, espacio físico en los locales y quedando expuestos a bajos niveles de seguridad. Desde hace poco más de un decenio en casi todos los puntos del planeta se ha valorado la opción de desarrollar e implantar sistemas que gestionan el material digitalizado. La Gestión Documental de manera general es un conjunto de actividades específicas que permiten coordinar y controlar sistemáticamente, todo lo relacionado con la creación, recepción, organización, almacenamiento, preservación, acceso y difusión de los documentos en una organización. O sea no es más que el uso de la tecnología y los procedimientos que permiten la gestión y el acceso unificado a la información generada.

1.1.3 Sistemas de Gestión Documental

Durante siglos, la gestión documental en las organizaciones fue el dominio exclusivo de

administradores, archiveros y bibliotecarios, cuyas herramientas manuales básicas eran los libros de registro, las carpetas, archivadores, cajas y estanterías en que se guardan los documentos de papel (y más tarde los audiovisuales y los documentos en soportes magnéticos u ópticos), los ficheros o kárdex que permiten hacer referencias cruzadas y una larga lista de técnicas de recuperación de información mediante sistemas de codificación y clasificación. Más recientemente se fueron sumando a ellos los informáticos, que son cada vez más necesarios debido a la complejidad y nivel de sofisticación que van alcanzando los sistemas computacionales de apoyo de la actividad administrativa.

Los Sistemas de Gestión Documental (SGD) son sistemas dedicados a incrementar el rendimiento y productividad en la gestión de documentos, facilitando el control y optimización de los flujos de información que existen dentro de una compañía. Los SGD actuales son sistemas dinámicos apoyados en tecnologías de flujo de trabajo (workflow) y trabajo en grupo (groupware). (ARSAUTE *et al.*)

1.1.4 Alfresco. Propuesta de gestor documental a utilizar

Inicialmente el cliente hizo la propuesta de que su sistema de control documental fuera implementado utilizando una base de datos empleando el gestor PostgreSQL. El proyecto al cual pertenecemos hicieron un estudio más profundo a nivel mundial sobre las tecnologías y tendencias actuales, en especial las orientadas a los sistemas de gestión documental y archivística y llegaron a la conclusión de que lo más favorable para el desarrollo del módulo era la utilización de Alfresco, que es un sistema de Gestión Documental y es utilizado como software de gestión de documental para documentos, páginas web, registros, imágenes y desarrollo colaborativo de contenido por lo que cumple con las necesidades del módulo a desarrollar, además de ser la alternativa principal del código abierto para la gestión del contenido empresarial en la actualidad.

A continuación algunos reconocimientos y beneficios de utilizarlo:

1.1.4.1 Beneficios (5)

✓ Facilidad de Uso

- Intelligent Virtual File System - Tan sencillo como usar una unidad compartida a través de CIFS, WebDAV o FTP
- Sistema de búsquedas estilo Google y exploración de carpetas estilo Yahoo

✓ **Productividad para el desarrollador**

- Definición de Reglas basadas en aspectos mediante sencillos asistentes.
- Gestión centralizada de Reglas y Acciones en el servidor para todas las Interfaces.

✓ **Colaboración óptima**

- Plantillas de Smart-Space Preconfiguradas. Estructura del Proyecto, Contenido, Lógica, Ciclos de vida.
- Foros .Discusiones sobre Carpetas y Documentos

✓ **Productividad para el Administrador**

- Sencilla instalación en el Servidor sin necesidad de instalación en el Cliente
- Gestión avanzada de la Seguridad de los Contenidos

✓ **Búsqueda Avanzada /Gestión del Conocimiento**

- Contenidos complejos, Atributos, Localización, Tipos de Objetos y Múltiples Taxonomías/Búsqueda por Categorías

✓ **Arquitectura Distribuida**

- Altamente Escalable y Arquitectura orientada a Servicios con tolerancia a fallos

✓ **Open Source**

- Considerable Reducción de costes

1.1.4.2 Reconocimientos (6)

- 2007
 - InfoWorld: Best of Open Source Applications BOSSIE Award Winner
 - Computerworld Honors Program: Finalist
 - Sand Hill Group: Top Software Innovator
 - Network World: Top Ten Enterprise Software Companies to watch
 - Gartner: "Cool Vendors in Content Management, 2007"
 - World Economic Forum: Technology Pioneer of 2007
- 2006

- Red Herring: Red Herring 100 Europe
- EContent: EContent 100
- KM World: Trend-Setting Product Award
- 2005
 - OSBC: Emerging Elite Award

1.1.5 ¿Por qué es necesaria la gestión documental digital?

Antes de valorar la necesidad de gestionar digitalmente grandes volúmenes de documentos es importante precisar que el resultado de la transformación que se realice al papel tradicional será un Documento Digital por lo que se partirá de la explicación más simple del nuevo concepto.

“...Un Documento Digital es, pues, aquél que contiene la información codificada en bits.”

El Documento Digital es, en fin, un producto de información y de comunicación, se crea, se utiliza y tiene su ciclo de vida en un entorno digital. Se trata de un documento dinámico en cuanto puede variar espacial, temporal e informativamente, interactivo en cuanto el usuario/lector posee capacidad de decisión en su lectura/exploración/navegación y adaptativo siendo capaz de variar en consonancia con las necesidades de los usuarios individuales.

1.1.6 Principales razones para digitalizar

La razón de la implementación de un proyecto de digitalización, o más exactamente de la conversión digital de documentos originales no digitales son variadas y pueden solaparse. La decisión de digitalizar puede tomarse con objeto de:

- ✓ Incrementar el acceso: esta es razón principal y la más obvia, cuando se sabe que hay una alta demanda por parte de los usuarios y la organización desea mejorar el acceso a una determinada colección.
- ✓ Mejorar los servicios para un grupo creciente de usuarios proporcionando un acceso de mayor calidad a los recursos de la institución.
- ✓ Reducir la manipulación y el uso de materiales originales frágiles o utilizados intensivamente y crear una “copia de seguridad” para el material deteriorado como libros o documentos quebradizos.
- ✓ Impulsar el desarrollo de recursos cooperativos, compartiendo intereses comunes con otras instituciones para crear colecciones virtuales e incrementar el acceso. El carácter público de los documentos permite que se fortalezcan las relaciones entre instituciones del propio ministerio y

con otros entes externos. De esta forma ahora podrían ser consultados los documentos digitalizados como constancia o evidencia en casos de investigaciones policiales y penales, por citar un ejemplo.

1.1.6.1 Beneficios

Miles de organizaciones alrededor del mundo utilizan a diario sistemas de Digitalización de Documentos en vez de utilizar los sistemas de archivo en papel. La Digitalización de Documentos ofrece una serie de beneficios en comparación con el papel de los cuales se pueden citar:

- ✓ Rápida Recuperación de la Información. El Sistema de Digitalización permite encontrar documentos rápidamente a diferencia de la búsqueda efectuada manualmente con los archivos en papel. Esta facilidad tiene un impacto directo en el incremento de los niveles de servicios pues reduce considerablemente el tiempo de localización y consulta de los documentos archivados en los Registros representando de esta manera una alta satisfacción en los usuarios.
- ✓ Indexado Flexible. El Sistema de Digitalización puede indexar documentos de diferentes maneras simultáneamente sin embargo, la indexación en papel resulta difícil de manejar además de ser mucho más costoso y con una pérdida de tiempo considerable. De esta manera es posible relacionar de múltiples formas los documentos digitalizados estableciendo cuales serían los parámetros y agilizando en todos los casos las consultas a la información.
- ✓ Búsqueda por Texto. Los Sistemas de Imágenes Digitales permiten recuperar archivos mediante el uso de una palabra o frase del documento una habilidad imposible de lograr mediante los registros de papel. Además, la búsqueda puede efectuarse también mediante fecha, libro, tomo, etc.
- ✓ No hay pérdida de Archivos. Los documentos son imágenes que permanecen en sus ubicaciones cuando se visualizan, de esta manera ninguno puede perderse o quedarse mal archivado. Además, mediante la indexación y la opción de búsqueda de texto, se puede encontrar documentos que por algún motivo se encuentren accidentalmente fuera de su ubicación correspondiente. En el uso de documentos de papel, el reemplazo de documentos debido a extravío supone un elevado coste de tiempo y dinero para la administración.
- ✓ Archivo Digital. Con el uso de un Sistema de Digitalización de Documentos se reduce el riesgo de daño o pérdida. Manteniendo los documentos en el sistema informático se protege dicha información de la manipulación de dichos documentos con el consecuente deterioro del papel.

El documento se mantiene en formato electrónico como el primer día, además el sistema no permite la modificación del documento original evitando así, posibles fraudes o alteraciones de la documentación.

- ✓ **Compartir los Archivos.** El Sistema de Digitalización permite compartir los documentos electrónicamente de forma simultánea. Generalmente, los documentos impresos en papel necesitan fotocopiarlos para poderse compartir.
- ✓ **Seguridad.** El Sistema de Digitalización puede garantizar un control más flexible y mejor de los documentos más delicados. El Sistema controla la seguridad de acceso a la documentación ya que solo tendrán acceso a los documentos los usuarios validados para ello, impidiendo el acceso de personal no autorizado.
- ✓ **Ahorro de Espacio.** El Sistema de digitalización ayuda a recuperar espacio físico en la oficina que se había perdido previamente debido a los volúmenes de papel almacenados.
- ✓ **Recuperación en caso de desastre.** El Sistema de Digitalización cuenta con un sistema de seguridad que permite la recuperación de la información en caso de desastres mediante respaldos (backups) realizados periódicamente.
- ✓ **Mayor productividad y mejor servicio al cliente.**

1.1.6.2 Riesgos de la Gestión Documental Digital

La puesta en marcha de un proyecto de digitalización no es todo lo sencillo y beneficioso que ya se ha comentado dentro del presente trabajo. También se incurren en numerosos riesgos que en algunos casos pueden interpretarse como desventajas en dependencia del entorno en el que se plantee la utilización de una solución con esas características. A continuación se relacionan algunos de los principales riesgos a considerar:

- ✓ Está sujeto a la obsolescencia tecnológica y programas informáticos que le dan vida.
- ✓ Depende de hardware y software.
- ✓ Se requiere de un medio de almacenamiento para la recuperación de la información.
- ✓ Altos costos de los medios para mantener la información disponible a largo plazo.
- ✓ Facilidad de modificación de la información.
- ✓ Almacenamiento de la información en sistemas aislados.
- ✓ Falta de legislación que ampare los procesos y métodos para el mantenimiento y conservación de la información.

1.2 Paradigmas de Programación

Podemos decir que, los paradigmas son marcos de referencia que imponen reglas sobre cómo se deben hacer las cosas, indican qué es válido dentro del paradigma y qué está fuera de sus límites. Un paradigma distinto implica nuevas reglas, elementos, límites y maneras de pensar, o sea implica un cambio. Los paradigmas pueden ser considerados como patrones de pensamiento para la resolución de problemas. Desde luego siempre teniendo en cuenta los lenguajes de programación. Representan un enfoque particular o filosofía para la construcción del software. No existe uno mejor que otro, sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro. (7)

1.2.1 El Paradigma Imperativo

El paradigma imperativo debe su nombre al papel dominante que desempeñan las sentencias imperativas. La programación imperativa describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea. La implementación de hardware de la mayoría de computadores es imperativa; prácticamente todo el hardware de los computadores está diseñado para ejecutar código de máquina, que es nativo al computador, escrito en una forma imperativa. Existen varios lenguajes que utilizan este paradigma, como por ejemplo: FORTRAN, COBOL, BASIC, C, Ada, Pascal.

1.2.2 El Paradigma Funcional

La programación funcional tiene como objeto imitar las funciones matemáticas lo más posible. Un lenguaje funcional posee la propiedad matemática de transparencia referencial, lo que significa que una expresión representa siempre el mismo valor. Esto permite razonar sobre la ejecución de un programa y demostrar matemáticamente que es correcto.

Las variables de un lenguaje funcional son como las variables en álgebra. Inicialmente representan un valor desconocido que, una vez calculado, ya no cambia. En un programa funcional, el orden de evaluación de las sub expresiones no afecta al resultado final, por lo tanto las sub expresiones pueden ejecutarse en forma paralela para hacer más eficiente el programa.

1.2.3 El Paradigma Lógico

Una forma de razonar para resolver problemas en matemáticas se fundamenta en la lógica de primer orden. El conocimiento básico de las matemáticas se puede representar en la lógica en forma de axiomas, a los cuales se la agregan reglas formales para deducir cosas verdaderas (teoremas). Los lenguajes que utilizan esta lógica se llaman lenguajes declarativos, porque todo lo que tiene que hacer el programador para solucionar un problema es describirlo vía axiomas y reglas de deducción.

Este concepto de programación lógica está ligado históricamente a Prolog (programación en logique), desarrollado por la universidad de marseille en 1972. Prolog es utilizado para el desarrollo de aplicaciones de Inteligencia Artificial debido a su forma de representar el conocimiento, facilitando las búsquedas en base de datos, la escritura de compiladores, la construcción de sistemas expertos, el procesamiento del lenguaje natural, búsqueda de patrones y programación automática.(8)

1.2.4 Programación Orientada a Objetos (POO)

Es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan *estado* (datos), comportamiento (procedimientos o *métodos*) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar. De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos). A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos (de una misma clase o de distintas), y en consecuencia, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos). Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento.(9)

1.2.5 La Programación Orientada a Servicios

La industria del desarrollo de software se encuentra actualmente en un estadio de transición hacia un nuevo paradigma de programación: la llamada **Arquitectura Orientada a Servicios (SOA)**, por sus siglas en inglés).

Cuando los arquitectos de software descubrieron que debían dejar de pensar en su aplicación como una entidad aislada de la infinidad de aplicaciones existentes desarrolladas en múltiples y diferentes plataformas y lenguajes, y que diseñar aplicaciones ubicuas e interoperables entre sistemas heterogéneos daba a sus productos una ventaja muy grande sobre el resto de sus competidores

Fue así que se comenzó a desarrollar el concepto de Web Services que en resumidas cuentas ofrecían la posibilidad de llamar a un componente de negocios desarrollado en una plataforma X desde una aplicación corriendo en cualquier plataforma, en cualquier parte del mundo, utilizando para ello protocolos estándar como SOAP, XML y HTTP. A partir del éxito y la aceptación obtenida por parte de la comunidad informática, los Web Services han ido evolucionando y se ha llegado a un concepto más general llamado SOA (Service Oriented Application) o programación orientada a servicios.

La programación orientada a servicios es un complemento de la programación orientada a objetos e implementa mejoras a esta última, producto de la experiencia acumulada en la última década, sobre todo en las áreas de la computación distribuida, instalación de una solución e interoperabilidad entre sistemas heterogéneos.

Es importante recalcar que la programación orientada a servicios no reemplaza a **POO**, sino que la complementa introduciendo mejoras.

Una de las diferencias fundamentales entre SOA y POO es la manera en la que ambas definen una “aplicación”. POO determina que una aplicación está compuesta de clases interdependientes, mientras que SOA considera que una aplicación está compuesta por un conjunto de servicios autónomos.(10)

1.3 Frameworks PHP

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírseles. (11)

Los frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes y proporcionan estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener.(12) Los

objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. (11) En general, el término framework, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, se puede considerar como una aplicación genérica incompleta y configurable a la que se le pueden añadir las últimas piezas para construir una aplicación concreta. (11)

1.3.1 Symfony

Symfony es un framework para construir aplicaciones web con PHP. En otras palabras, es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web. (12)

Es uno de los frameworks PHP más populares entre los usuarios y las empresas, ya que permite que los programadores sean mucho más productivos a la vez que crean código de más calidad y más fácil de mantener. Symfony es maduro, estable, profesional y está muy bien documentado. (12)

Emplea el tradicional patrón de diseño MVC para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. (12)

La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista. (12)

Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix y Linux) como en plataformas Windows.

Se puede citar como desventajas que, para su utilización se requiere de la generación de grandes cantidades de código y una configuración exhaustiva de antemano. También que precisa de una estructuración estricta de sus directorios que se establece por consola a la hora de crear módulos.

1.3.2 Zend Framework (13)

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Utiliza para su implementación código 100% orientado a objetos. La estructura de los componentes es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

Aunque se pueden utilizar de forma individual, al combinar los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework para el desarrollo de aplicaciones web que ofrece un gran rendimiento y una robusta implementación del patrón MVC; una capa de abstracción de datos fácil de utilizar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

Otros componentes tales como Zend_Auth y Zend_Acl, proveen ayuda para la autenticación de usuarios contra la mayoría de los proveedores de credenciales existentes. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los web services más populares. Cabe mencionar también el Zend_Config para temas de configuración de aplicaciones web, Zend_Db para tratar con bases de datos, Zend_Search o Zend_Feed entre otros como para la gestión de documentos en PDF, canales RSS etc.

1.3.3 Codelgniter (14)

Codelgniter es un Framework para el desarrollo de aplicaciones para personas que quieran construir sitios usando PHP. Su objetivo es poder hacer los proyectos más rápidos de lo que se pueden hacer, suministrando un rico conjunto de librerías para tareas comunes, con una simple interfaz y estructura lógica para acceder a estas librerías. Codelgniter deja que su creatividad se centre en el proyecto minimizando la cantidad de código necesitado para ejecutar una tarea.

Está liberado bajo la licencia de código abierto Apache-BSD, lo que significa que es totalmente libre y puede ser usado.

Entre sus características se puede encontrar su compatibilidad con PHP 4 y PHP 5, posee soporte para múltiples bases de datos entre las que se pueden mencionar PostgreSQL, MySQL, MSSQL, además de que se lo pueden incluir drivers para otros gestores de bases de datos que se programen, incluye también plantillas, validaciones, no requiere instalación, se puede encontrar una librería con un gran número de clases. Una de las características más interesantes de Codelgniter es el elevado

número de clases que incluye para trabajar con distintos objetos: calendario, bases de datos, correo electrónico, manipulación de imágenes, FTP, lenguaje, tablas, sesiones y compresión ZIP.

A diferencia de otros frameworks, CodeIgniter cuenta con una documentación excelente que permite conocer todos los secretos de este entorno de trabajo. Requiere sólo de unas bibliotecas muy pequeñas. Esto está en contraste severo a muchos frameworks que requieren significativamente más recursos. Las bibliotecas adicionales serán dinámicamente cargadas según la demanda, basada en sus necesidades por un proceso dado, por lo que es bastante rápido.

Puede extenderse fácilmente a través del uso de plugins y bibliotecas de ayuda, o a través de extensiones de clases.

Está basado en el patrón Modelo Vista Controlador (MVC), donde los modelos representan las estructuras de datos y contienen las funciones para insertar y actualizar la base de datos; la vista es la información que es presentada al usuario, normalmente las vistas son páginas web y por último las controladoras son las intermediarias entre los modelos, las vistas y cualquier otro recurso que se necesite para procesar el pedido HTTP y generar la página web.

Muchos de los Framework de PHP ofrecen MVC, pero ninguno es tan ligero y flexible como CodeIgniter. Lo mejor de todo es que este puede ser tan sólo VC (Vista Controlador), es decir no fuerza al usuario a utilizar una BD para un desarrollo. Cabe resaltar que su soporte de Ajax aunque mínimo es lo suficiente para desarrollar aplicaciones vistosas y ágiles y se podría decir sin lugar a dudas que CodeIgniter brinda una muy buena implementación del patrón Active Record, independizando al usuario o programador del sistema en el que esté la BD brindando para ello una amplia gama de funciones que abstraen de sentencias SQL explícitas y evitan las famosas inyecciones SQL.

1.4 Entornos de Desarrollo para PHP

Los Entornos de Desarrollo Integrados, Integrated Development Environment (IDE) son programas que agrupan un conjunto de herramientas que viabilizan la labor de los programadores. Los componentes esenciales que los definen son:

- Editor de texto (código).
- Compilador.
- Intérprete.
- Depurador.

- Herramientas de automatización (completamiento de código y navegación rápida dentro del código).
- Sistema de control de versiones.
- Diseñador de interfaces gráficas.
- Agregación de herramientas externas.
- Soporte para varios lenguajes de programación y plataformas de desarrollo.

No todos poseen las propiedades anteriores, ni las implementan al mismo nivel, pero esas son las más usuales.

1.4.1 Zend Studio

Es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Es orientado a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario.(15)

Algunas de sus características son:

Es compatible con las plataformas GNU/Linux, MAC y Windows. No requiere la instalación previa de PHP ni del entorno de ejecución de Java. Cuenta con un entorno de prueba que agiliza la seguridad de la calidad, integración y las etapas de los procesos. Proporciona las extensiones requeridas para el desarrollo de aplicaciones PHP. Presenta un excelente completamiento de código y coloreado de sintaxis para facilitar el trabajo de los programadores. Inserción automática y emparejamiento de paréntesis y corchetes de cierre. Detección de errores de sintaxis en tiempo real. Posee un manual de PHP integrado. Simplifica el despliegue con la integración de FTP y SFTP lo que permite subir y descargar archivos de forma segura hacia y desde servidores remotos. Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador). Puede conectarse directamente a muchas de las bases de datos utilizadas en el desarrollo de proyectos tales como: IBM DB2/Cloudscape/ Derby/, MySQL, Oracle, Microsoft SQL Server, PostgreSQL y SQLite.(16)

1.4.2 Adobe Dreamweaver (17)

Desarrollado inicialmente por Macromedia (actualmente producido por Adobe Systems) para diseño, desarrollo, publicación y administración de sitios y aplicaciones Web. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

Las versiones originales de la aplicación se utilizaban como simples editores WYSIWYG (*What You See Is What You Get* "lo que ves es lo que obtienes"). Sin embargo, versiones más recientes soportan otras tecnologías web como CSS, JavaScript y algunos Framework del lado servidor.

Además de sus capacidades WYSIWYG, tiene las funciones típicas de un editor de código fuente para la web:

- Un administrador de sitios, para agrupar los archivos según el proyecto al que pertenezcan.
- Un cliente FTP integrado, que permite subir los archivos editados inmediatamente al sitio en Internet.
- Función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP.

1.4.3 Netbeans 6.5 para PHP

NetBeans es un entorno de desarrollo integrado de código abierto ("open source") para desarrolladores. Posee todas las herramientas necesarias para crear aplicaciones web, de escritorio y para teléfonos móviles con los lenguajes de programación Java, C, C++ e incluso lenguajes dinámicos como: PHP, JavaScript, Groovy, y Ruby. Es fácil de usar, instalar y puede ser ejecutado en múltiples plataformas como Windows, Linux, Mac OS X y Solaris.(18)

Características de NetBeans: (19)

- Instalación y actualización más simple.
- Características visuales para desarrollo web con interfaces amigables y completamiento automático de código PHP, así como coloreado de código sintáctico y semántico.
- Mejoras para SOA y UML.
- Permite depurar el código usando Xdebug.

- Soporte para PHP.

Los Entornos de Desarrollo Integrados analizados son muy potentes, poseen disimiles características y ventajas, además de brindarnos un gran soporte para el trabajo con en lenguaje PHP; sin embargo se eligió como IDE a utilizar para el desarrollo del módulo a este último sobre todo por ser su licencia de código abierto, política requerida para su desarrollo.

1.5 Tendencias y tecnologías actuales

Tendencias y tecnologías actuales posibles a emplear, descritas a continuación.

1.5.1 Uso de software libre

Nuestro país está haciendo un gran esfuerzo para la migración gradual y progresiva hacia el software libre, nuestra facultad es un ejemplo de ello; y como es de esperarse, todas las aplicaciones y proyectos que se realicen en la misma deben estar desarrollados en software libre.

El término de software libre –lo contrario del software patentado- se aplica a las aplicaciones informáticas que están libremente disponibles bajo un acuerdo de licencia pública de manera que cualquiera puede adaptarlos y mejorarlos. Las quejas más comunes sobre los programas de software patentados de Microsoft son sus costos relativamente elevados y que, cuando fallan, sólo Microsoft puede repararlos. Cada vez son más las organizaciones de toda clase que optan por soluciones libres para sus necesidades informáticas, porque los productos libres son generalmente más baratos, más fiables y más fáciles de reparar cuando fallan. Este es el caso del MENPET institución a la cual está dirigido este módulo.

1.5.2 Aplicaciones Web

Son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, etc.) en la que se confía la ejecución al navegador. (20)

La plataforma Web posee una arquitectura Cliente/Servidor. Las aplicaciones Web constituyen un caso más complejo de esta arquitectura, permitiendo manejar lógica de negocio. Éstas se ejecutan en el servidor, encargándose de controlar el estado de dicho negocio y de gestionar los datos almacenados con ayuda de algún Sistema Gestor de Bases de Datos.

Con la introducción de Internet y de la Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web.(21)

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, todo esto evita tener que instalar componentes de software adicionales para hacer uso de ellas. Además, permiten el aprovechamiento de todas las características de Internet. Son fáciles de usar, pues no requieren conocimientos avanzados de computación y su disponibilidad es alta ya que pueden realizar consultas en cualquier parte del mundo donde se tenga acceso a Internet.

Las aplicaciones web tienen varias ventajas sobre los programas de software descargables tradicionales que influyen en que el sistema a desarrollar sea una aplicación Web.

Estas son las principales: (22)

- ✓ **Compatibilidad multiplataforma.** Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- ✓ **Actualización.** Las aplicaciones basadas en web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).
- ✓ **Inmediatez de acceso.** Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y está listo para trabajar sin importar cuál es su configuración o su hardware.
- ✓ **Facilidad de prueba.** Finalmente no habrá más obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones antes de cargar su tarjeta de crédito. Actualmente, especialmente cuando hablamos de software costoso, hay todavía una gran cantidad de funcionalidades y pequeños detalles que no pueden ser totalmente probados descubiertos antes de comprometer dinero en alguna compra total.

- ✓ **Menos requerimientos de memoria.** Las aplicaciones basadas en web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, a esas aplicaciones basadas en web usa en muchos casos la memoria de las computadoras que ellos corren, dejando más espacio para correr múltiples aplicaciones del mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.
- ✓ **Menos Bugs.** Las aplicaciones basadas en web deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en web, todos utilizan la misma versión, y todos los bugs pueden ser corregido tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones basadas en web deberían tener mucho menos bugs que el software de escritorio descargable tradicional.
- ✓ **Precio.** Las aplicaciones basadas en web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios premium como una opción.
- ✓ **Los datos también van online.** Por supuesto con el desplazamiento de las aplicaciones locales a aquellas basadas en web también los datos que creamos y accedemos van a necesitar experimentar profundos cambios. A nadie le gusta no poder acceder a su propio e-mail cuando está de viaje, o poder recuperar un documento particular cuando se conecta desde un ciber café a 15.000 kilómetros de su oficina "Los clientes no deberían almacenar datos; deberían ser como teléfono". De hecho pueden convertirse en teléfonos, o viceversa. Y a medida que los clientes se hagan más pequeños, usted tiene otra razón para no tener sus datos en ellos: algo que usted lleva consigo pueden perderse o ser robado"
- ✓ **Múltiples usuarios concurrentes.** Las aplicaciones basadas en web pueda realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. Las compañías de conferencia web y colaboración online están involucradas algunas transformaciones claves y los usuarios necesitan explorar que significa realmente trabajar efectivamente y coeditar documentos juntos.
- ✓ **Los datos son más seguros.** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se haga cargo

del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, serán la norma más que la excepción, y los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que provee aplicaciones basadas en web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga. Usted puede imaginar que si una compañía comercial pierde los datos de la gente será puesta de rodillas (financieramente) en cuestión de días.

- ✓ **Desarrollar aplicaciones en el lenguaje que usted quiera.** Una vez que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos pueden también ser escritas en prácticamente cualquier lenguaje de programación. Ya que las aplicaciones web son esencialmente una colección de programas más que un simple programa, ellas podrían ser escritas en cualquier lenguaje de programación que esté por ahí. Mientras que para software escritorio usted está limitado a usar el mismo lenguaje que el sistema operativo subyacente este no es el caso cuando la aplicación de software es independiente del sistema operativo.

1.5.3 Modelo Cliente Servidor

La arquitectura cliente-servidor, llamado, modelo cliente-servidor es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificarlas.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes. (23)

En la actualidad se suele hablar de arquitectura de tres niveles, donde la capa de almacenamiento y la de aplicación se ubican en (al menos) dos servidores diferentes, conocidos como servidores de datos y servidores de aplicaciones para lograr una mejor funcionalidad.

Esta configuración se llama una arquitectura de tres-capas: (23)

- ✓ Clientes que interactúan con los usuarios finales.
- ✓ Servidores de aplicación que procesan los datos para los clientes.
- ✓ Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Ventajas de la arquitectura cliente-servidor: (23)

- ✓ Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el

sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).

- ✓ **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- ✓ **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- ✓ Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

1.5.4 Servicios Web

Un servicio Web (Web Service en inglés), no es más que una librería que ofrece determinadas funcionalidades a las que se puede acceder a través de Internet empleando protocolos estándar como XML, SOAP, UDDI o WSDL. Es precisamente el uso de estos protocolos lo que permite que los datos “fluyan” a través de Internet pasando obstáculos como las paredes de fuego que eran insalvables para otras tecnologías con similares propósitos. Estos servicios constituyen una potente herramienta para el desarrollo de aplicaciones distribuidas y en nuestro caso serán de vital importancia a la hora de consumir Alfresco desde nuestra aplicación web.

1.5.4.1 Protocolos que usan los Servicios Web

1.5.4.1.1 XML (Extensible Markup Language): Es un metalenguaje de marcas que permite definir cómo es la información que se transmite. Esto permite una comunicación de datos entre distintos sistemas.

1.5.4.1.2 SOAP (Single Object Access Protocol): Es un protocolo de mensaje liviano basado en XML, usado para codificar los mensajes de Web Services antes de enviarlos por la red. Los mensajes SOAP son independientes de cualquier sistema operativo y protocolo, y pueden ser transportados usando una variedad de protocolos de Internet, incluyendo HTTP, SMTP y MIME. Permite que programas que

corren en diferentes sistemas operativos se comuniquen.

1.5.4.1.3 WSDL (Web Service Description Language): Es un lenguaje en formato XML que define las operaciones que proporciona un servicio, desarrollado conjuntamente por Microsoft e IBM.

1.5.4.1.4 UDDI (Universal Description Discovery and Integration): Es un directorio universal de Servicios Web basado en XML, que permite publicar, localizar y utilizar servicios Web.

1.5.5 Arquitectura Orientada a Servicios SOA (Service Oriented Architecture)

SOA es un concepto de arquitectura de software que da soporte a los requerimientos de software que utilizan servicios Web, formado por servicios de aplicación débilmente acoplados y altamente interoperables y diseñados para enlazar negocios y recursos computacionales. Posibilita la creación y los cambios de los procesos de negocio de forma ágil empleando funcionalidades contenidas en infraestructuras expuestas bajo la forma de servicios Web mediante la composición de nuevos procesos. SOA se puede construir sobre estándares de servicios Web proporcionando mayor interoperabilidad además de ser posible implementarlo utilizando cualquier tecnología basada en servicios, pero en su mayoría, las definiciones de SOA identifican la utilización de servicios Web. Cuando la mayoría de la gente habla de una arquitectura orientada a servicios están hablando de un juego de servicios residentes en Internet o en una intranet, usando servicios web. A estos servicios están ligados SOAP, WSDL y UDDI y aunque por ser SOA orientado a servicios no necesariamente tiene que usarlos, pero se aconseja su utilización.

Ventajas de SOA:

- Minimiza el tiempo de desarrollo, ya que los servicios se pueden reutilizar fácilmente y pueden convertirse en nuevas aplicaciones compuestas.
- Reducen los costos de mantenimiento, los servicios reutilizables reducen el grado de complejidad interna de los servicios.
- Aumenta la calidad de los servicios, porque una mayor reutilización de servicios crea servicios de mejor calidad.
- Rebaja los costos de integración, los servicios estandarizados pueden trabajar en conjunto, permitiendo que aplicaciones distintas se conecten con rapidez y facilidad.
- Reduce el riesgo porque menos servicios reutilizables brindan mayor control, y reducen el riesgo general relacionado con el cumplimiento.

1.5.6 Lenguaje PHP5

Ya que es un lenguaje interpretado de propósito general ampliamente usado, que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML, posee una amplia documentación en su página oficial entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda, permite las técnicas de Programación Orientada a Objetos, no requiere definición de tipos de variables, tiene manejo de excepciones, y es libre, será de gran importancia para desarrollar nuestro módulo.

1.5.7 Servidor Web Apache

Es un software (libre), servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, es el servidor HTTP más usado (empleado en el 70% de los sitios web en el mundo).⁽²⁴⁾ Presenta entre otras las siguientes ventajas: Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera. Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada. Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas. Te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor. ⁽²⁵⁾

1.5.8 Lenguaje XML

XML (o lenguaje de anotación extensible) es desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del lenguaje SGML (sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial), estas las defines el usuario a diferencia de HTML que ya están predefinidas. Es válido aclarar que XML no es ni será nunca un reemplazo del HTML sino un complemento que sirve para manejar la información separada del formato. (26)

Es la base de los Servicios Web, y a pesar de su sencillez aparente, está transformando completamente la creación y el uso de software. Es la solución a un problema de comunicación entre programas de ordenador, pues la información generalmente queda fuertemente ligada al programa con el cual fue creada, y es así como se pierde mucho tiempo en pasar de un formato de definición a otro. El contenido almacenado en un documento XML se puede transferir fácilmente a través de la red.

Se hizo uso de este lenguaje de etiquetas porque: (27)

- ✓ Es facilitador de cargar dinámicamente los contenidos desde un fichero sin necesidad de almacenarlos en la base de datos.
- ✓ Permite estructurar la información y luego aplicarle fácilmente transformaciones para su presentación, lo que ayuda a ganar en tiempo.
- ✓ La mayoría de los navegadores actuales son capaces de entender XML.
- ✓ La comunicación de datos. Cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- ✓ Permite la compatibilidad entre sistemas. Comparte la información de una manera segura, fiable y fácil.

Aunque es una tecnología sencilla tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Por eso la idea de unirlo a PHP y a HTML, lo que permitirá una mayor organización y seguridad en la aplicación, cargar los menús y mensajes dinámicamente.

1.5.9 ECM Alfresco

Alfresco es la alternativa principal del código abierto para la gestión del contenido empresarial. El modelo del código abierto permite a Alfresco utilizar las tecnologías y las contribuciones de la comunidad del código abierto para conseguir un software de alta calidad, produciéndolo más rápidamente con un costo mucho más bajo. Su meta no es sólo proporcionar un código abierto, sino que este supere las características técnicas, funcionalidades y ventajas para la comunidad de usuarios. (28)

1.5.10 Patrones de arquitectura (29)

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Ayudan a especificar la estructura fundamental de una aplicación. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global; por ejemplo, la adaptabilidad de la interfaz de usuario. Los patrones que dan soporte a características similares se agrupan en una misma categoría.

1.5.10.1 Patrón Modelo-Vista-Controlador (Model-View-Controller o MVC) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- ✓ Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es dependiente de cualquier representación de salida y/o comportamiento de entrada.
- ✓ Vista (View): Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- ✓ Controlador (Controller): Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("service requests" en el texto original) para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.

Algunos de sus principales beneficios son:

- Menor acoplamiento
 - Desacopla las vistas de los modelos
 - Desacopla los modelos de la forma en que se muestran e ingresan los datos
- Mayor cohesión

- Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio)
- Las vistas proveen mayor flexibilidad y agilidad
 - Se puede crear múltiples vistas de un modelo
 - Se puede crear, añadir, modificar y eliminar nuevas vistas dinámicamente
 - Las vistas pueden anidarse
 - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual
 - Se puede sincronizar las vistas
 - Las vistas pueden concentrarse en diferentes aspectos del modelo
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales
 - Una vista para cada dispositivo que puede variar según sus capacidades
 - Una vista para la Web y otra para aplicaciones de escritorio
- Más claridad de diseño
- Facilita el mantenimiento
- Mayor escalabilidad

El MVC es un patrón ampliamente utilizado en múltiples plataformas y lenguajes. El sistema se implementó basado en este patrón de arquitectura debido a la flexibilidad, adaptabilidad y confiabilidad que brinda al sistema, pues se distribuye el código de manera organizada en capas, cada una respondiendo a sus funcionalidades correspondientes.

1.5.11 RUP

Constituye la metodología tradicional estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Se caracteriza por ser iterativa e incremental, centrada en la arquitectura y guiada por los casos de uso. Divide el proceso de desarrollo en ciclos de vida, obteniendo un producto al final de cada ciclo, los cuales se dividen en fases que deben de terminar con un hito, dentro de estas fases se encuentran: Inicio, Elaboración, Construcción y Transición. Es una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Unifica los mejores elementos de metodologías anteriores, está preparada para desarrollar grandes y complejos proyectos y utiliza el

UML como lenguaje de representación visual para preparar todos esquemas de un sistema software. De forma general proporciona una guía para ordenar las actividades del equipo de trabajo, dirige las áreas de cada desarrollador por separado y del equipo como un todo, especifica los artefactos que deben desarrollarse y ofrece criterios para el control y la medición de los productos y actividades de proyectos.

Después del estudio realizado de las diferentes variantes actuales en el desarrollo de aplicaciones web, como son los paradigmas de programación a utilizar, los entornos de desarrollo para PHP, los frameworks, se concluye que para llevar a cabo la implementación del sistema, se empleará el lenguaje de programación PHP 5 en su versión 5.2, con el framework CodeIgniter versión 1.7.1, sobre el entorno de desarrollo Netbeans 6.5 para PHP y utilizando como repositorio de información el sistema de gestión documental Alfresco en su versión 3.0 y RUP como metodología de desarrollo, por las facilidades que brindan en su conjunto y que fueron analizadas anteriormente. Todas las herramientas a utilizar tienen en común que son software libre, líderes en su mayoría en este ámbito, por lo que se cumple con las necesidades del cliente.

Capítulo 2. Descripción y análisis de la solución propuesta

El presente capítulo se centra en la descripción de la implementación del sistema, para ello, se sentarán las bases de la arquitectura propuesta, se realizará una valoración crítica del diseño propuesto por el analista, donde se estudiarán y explicarán los cambios necesarios para la transición del diseño a la implementación, teniendo en cuenta los requerimientos de la aplicación. Se abordarán cuestiones referentes a los componentes a reutilizar, como clases o librerías previamente implementadas y que fueron utilizadas en la implementación del módulo. Posteriormente se describirán los algoritmos no triviales a implementar, para finalmente describir las nuevas clases y operaciones necesarias para darle solución a la situación problemática.

2.1 Sentar las bases de la arquitectura

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del sistema a implementar.

Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea. Ahora, como determinar la arquitectura es un poco difícil y muy importante, y lo que se trata es de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño del sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad, siempre teniendo como idea predominante la reutilización.

A continuación se ilustra la arquitectura definida para la implementación del Módulo Correspondencias del SCD para el MENPET.

Capítulo 2. Descripción y análisis de la solución propuesta

Patrones de arquitectura:

Expresan un paradigma fundamental para estructurar u organizar un sistema software. Proporcionan un conjunto de subsistemas o módulos predefinidos, con reglas y guías para organizar las relaciones entre ellos.

En la propuesta de solución se utilizaron los patrones de arquitectura, Modelo Vista Controlador (MVC), y Cliente-Servidor, detallados previamente en el capítulo anterior.

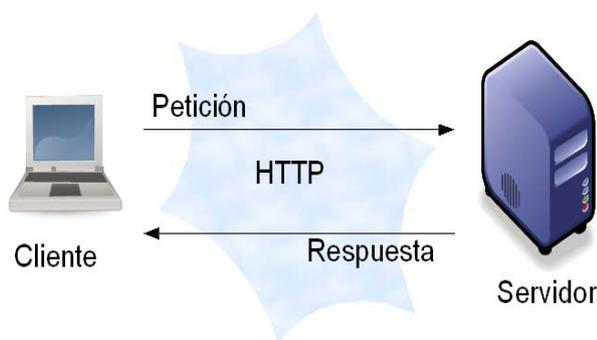


Figura 1. Arquitectura Cliente-Servidor

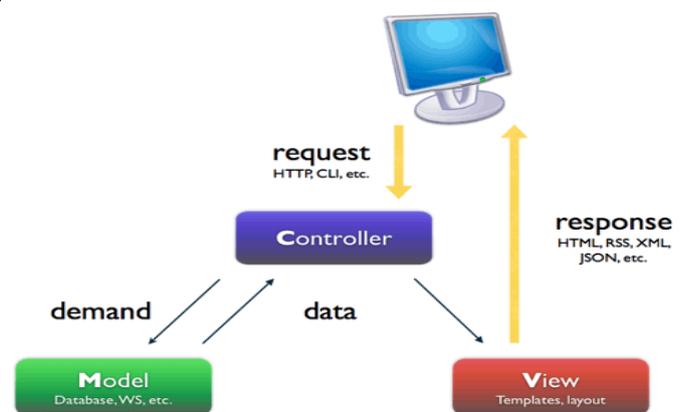


Figura 2. Patrón MVC

2.2 Valoración crítica del diseño propuesto por el analista

Ya habiendo definido la arquitectura a utilizar, se muestran a continuación los diagramas de clases del diseño por estereotipos web propuestos por el analista y la versión final después de realizarle los cambios necesarios acorde a los requerimientos del sistema, durante el flujo de trabajo de implementación. Los cambios obedecen fundamentalmente a la utilización del framework para PHP CodeIgniter, que utiliza el patrón arquitectónico MVC y que repercute en la estructuración de las clases para adecuarse al patrón aplicado.

Pueden citarse entre los principales cambios a la propuesta del analista, que las clases cliente son construidas por una clase en el servidor que contiene la clase controladora del módulo, que es la que tiene funciones que generan las vistas (clases cliente) y dichas vistas envían la información a la misma clase en el servidor, donde en la propia función de la controladora se gestiona el flujo de la información hacia la clase modelo y de esta se retorna la respuesta a la petición solicitada. Además de los cambios generales que se aplicaron a la propuesta del analista, algunos paquetes sufrieron algunos cambios más específicos, que se enuncian a continuación.

Capítulo 2. Descripción y análisis de la solución propuesta

Estos son:

La modificación del paquete Acceso a Datos, que tenía como principal objetivo el uso de funciones almacenadas para la inserción, modificación y eliminación de los registros en la base de datos, por el paquete Servicios de Alfresco con similares objetivos, pero ya brindando este último un conjunto de servicios que permiten consultar, crear, modificar y eliminar el contenido en su propio repositorio. La lógica continúa siendo la misma, con el considerable ahorro de esfuerzo y tiempo al no tener que crear ni validar las consultas a la Base de Datos.

La eliminación del paquete Smarty que definía un conjunto de clases de interfaz en la capa de presentación, con el objeto de lograr que a la hora de implementar el sistema fuese más fácil el trabajo con el diseño que se desee mostrar al usuario. Esto se logra con la utilización del framework CodeIgniter que maneja las clases interfaces o vistas a partir de las clases controladoras de forma similar.

Diagrama de Clases del Diseño (Paquete Correspondencia)

Se añadieron las funcionalidades para:

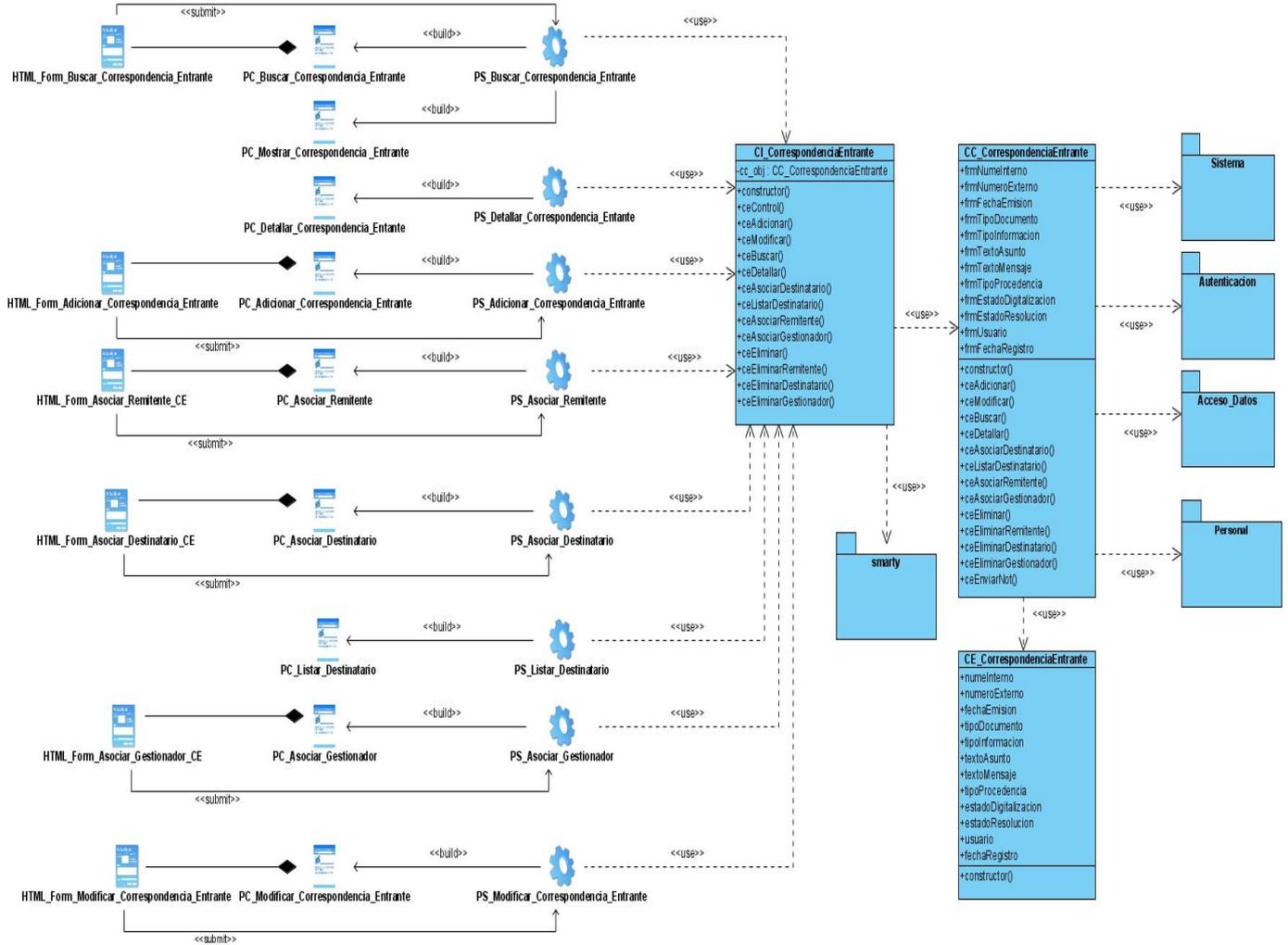
- Buscar Correspondencia Entrantes por Estados: Se añadió la página cliente con un formulario que permite buscar y listar las correspondencias entrantes existentes según estado en que se encuentren (Redacción, Revisión, Listas).
- Buscar Correspondencia Salientes por Estados: Se añadió la página cliente con un formulario que permite buscar y listar las correspondencias salientes existentes según estado en que se encuentren (Redacción, Revisión, Listas).

Algunos formularios fueron sujetos a cambios según peticiones del cliente, estos fueron los de la clase CP_Adicionar_Correspondencia_Entrante y el de CP_Buscar_Correspondencia_Entrante que se le quitaron los campos Estado de Resolución y Estado Digitalización. También se cambió la presentación de la aplicación al acceder a los submódulos de correspondencias entrantes y salientes para que fuera más amigable y funcional para el usuario.

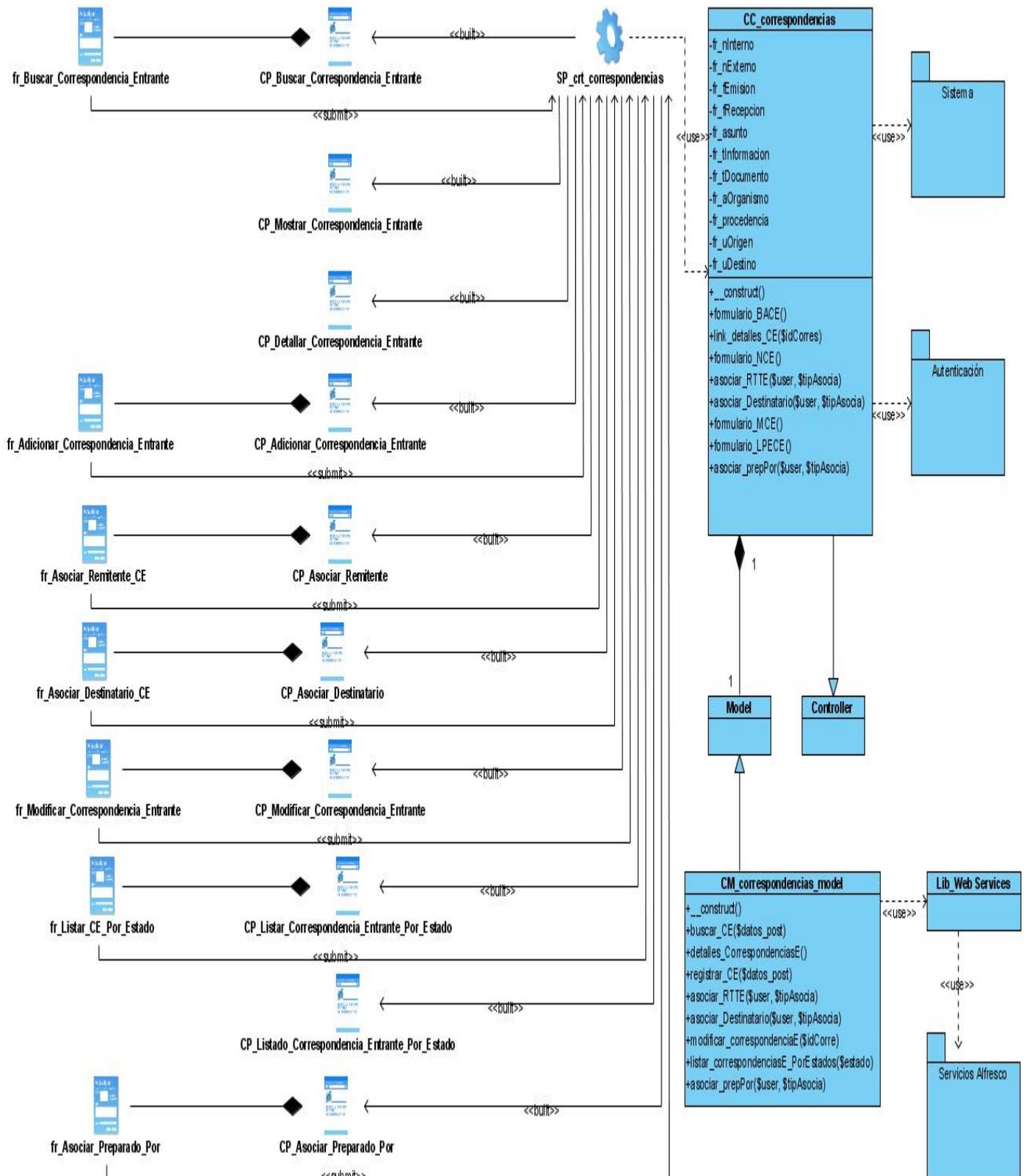
En el presente epígrafe, se analizará solamente el submódulo Correspondencias Entrantes para la valoración crítica, debido a que el submódulo Correspondencias Salientes está sujeto a transformaciones similares y de igual forma fueron modelados por el analista.

Capítulo 2. Descripción y análisis de la solución propuesta

Paquete Correspondencias. Diagrama de Clases Correspondencia Entrante, propuesto por el analista.

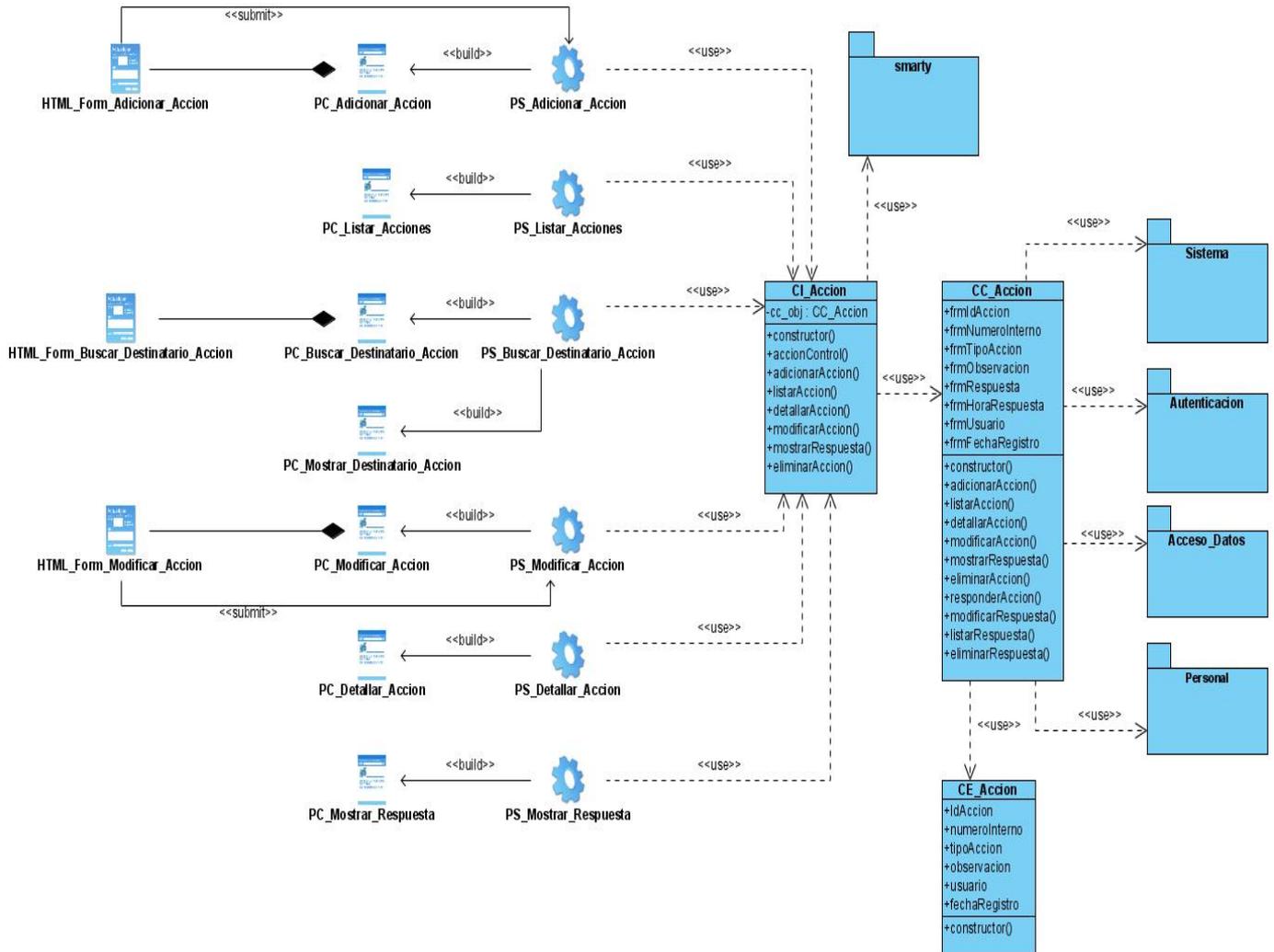


Paquete Correspondencias. Diagrama de Clases Correspondencia Entrante, versión final.

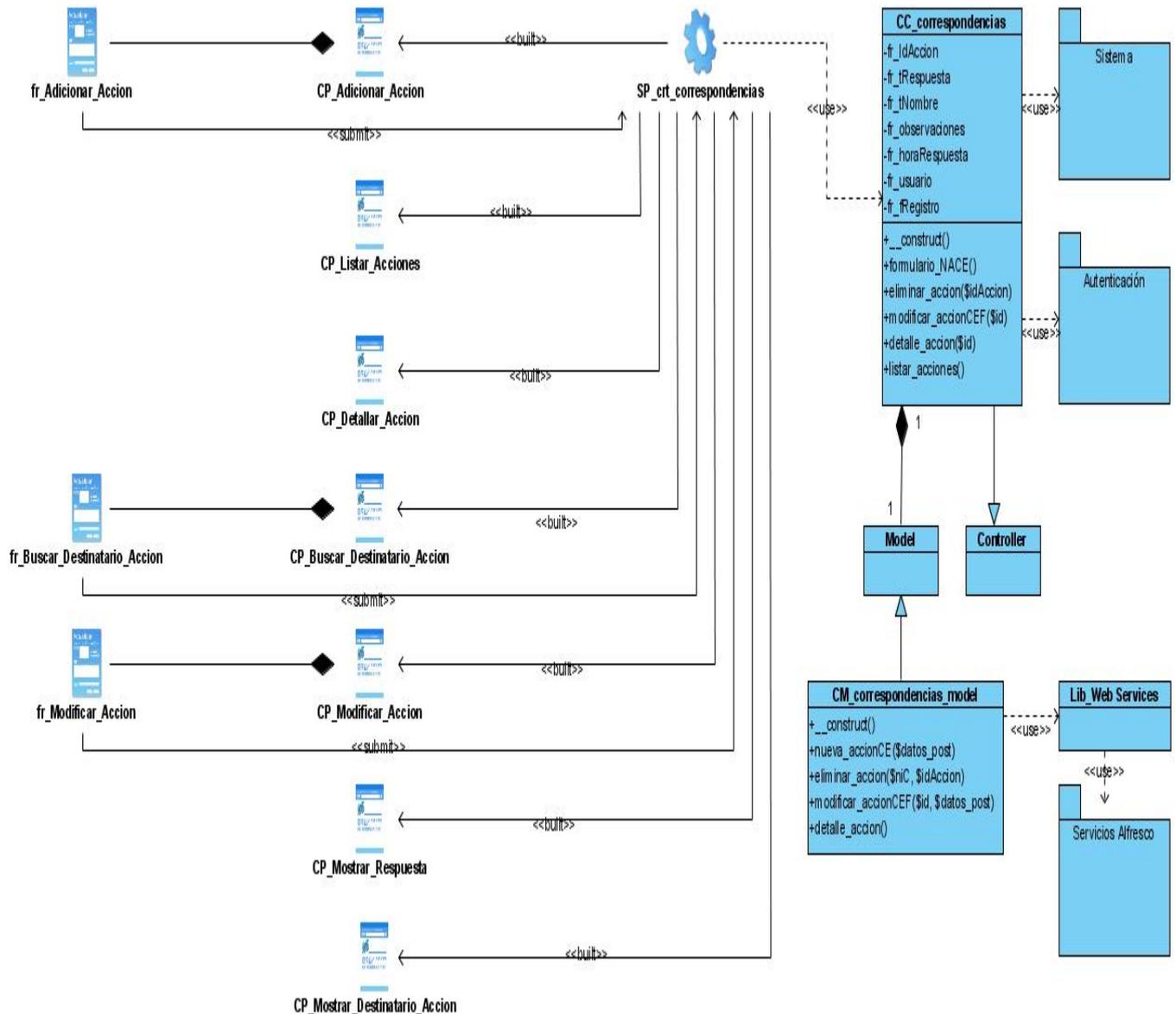


Capítulo 2. Descripción y análisis de la solución propuesta

Paquete Correspondencias. Diagrama de Clases Gestionar Acción, propuesto por el analista.

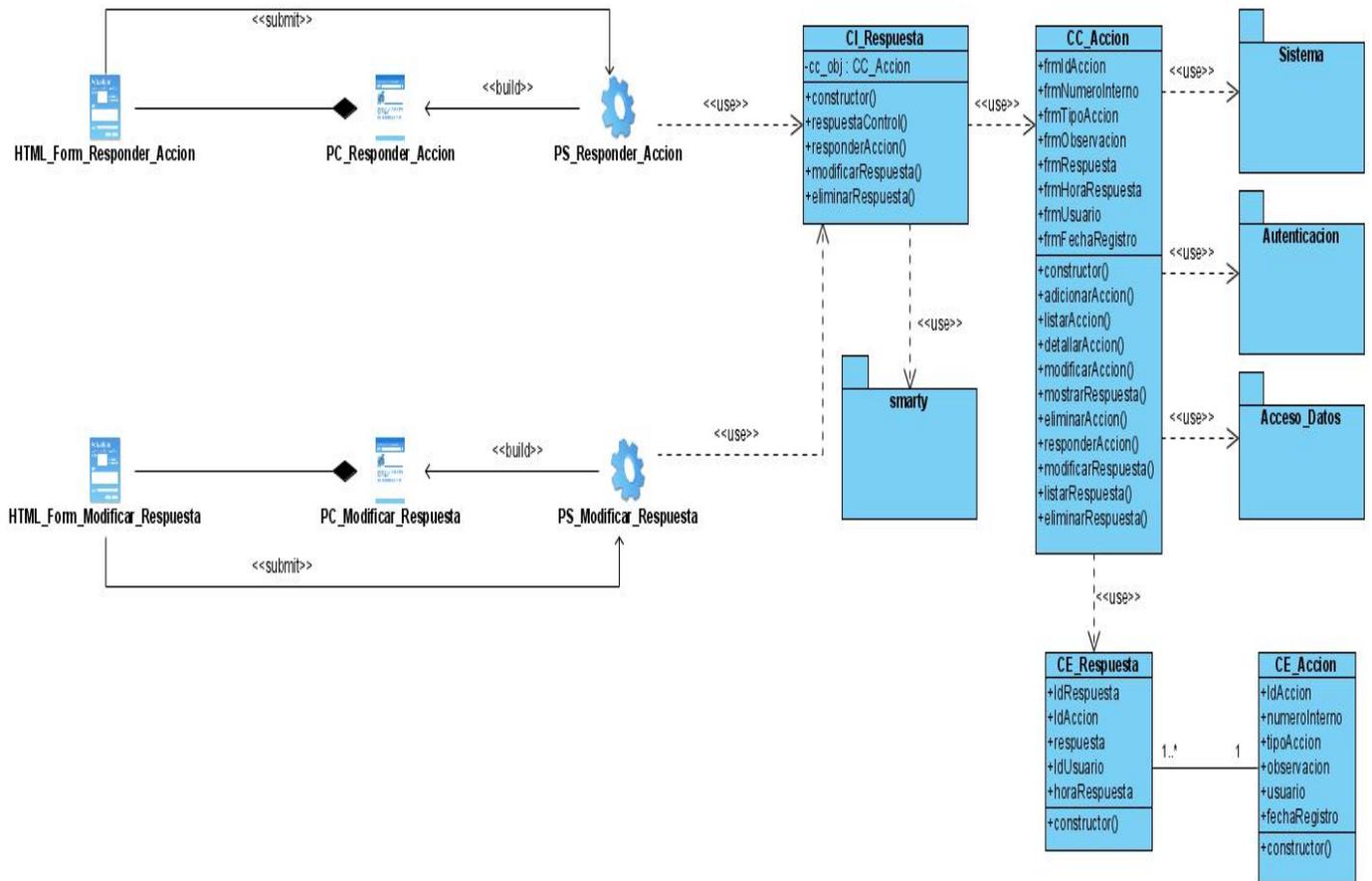


Paquete Correspondencias. Diagrama de Clases Gestionar Acción, versión final.

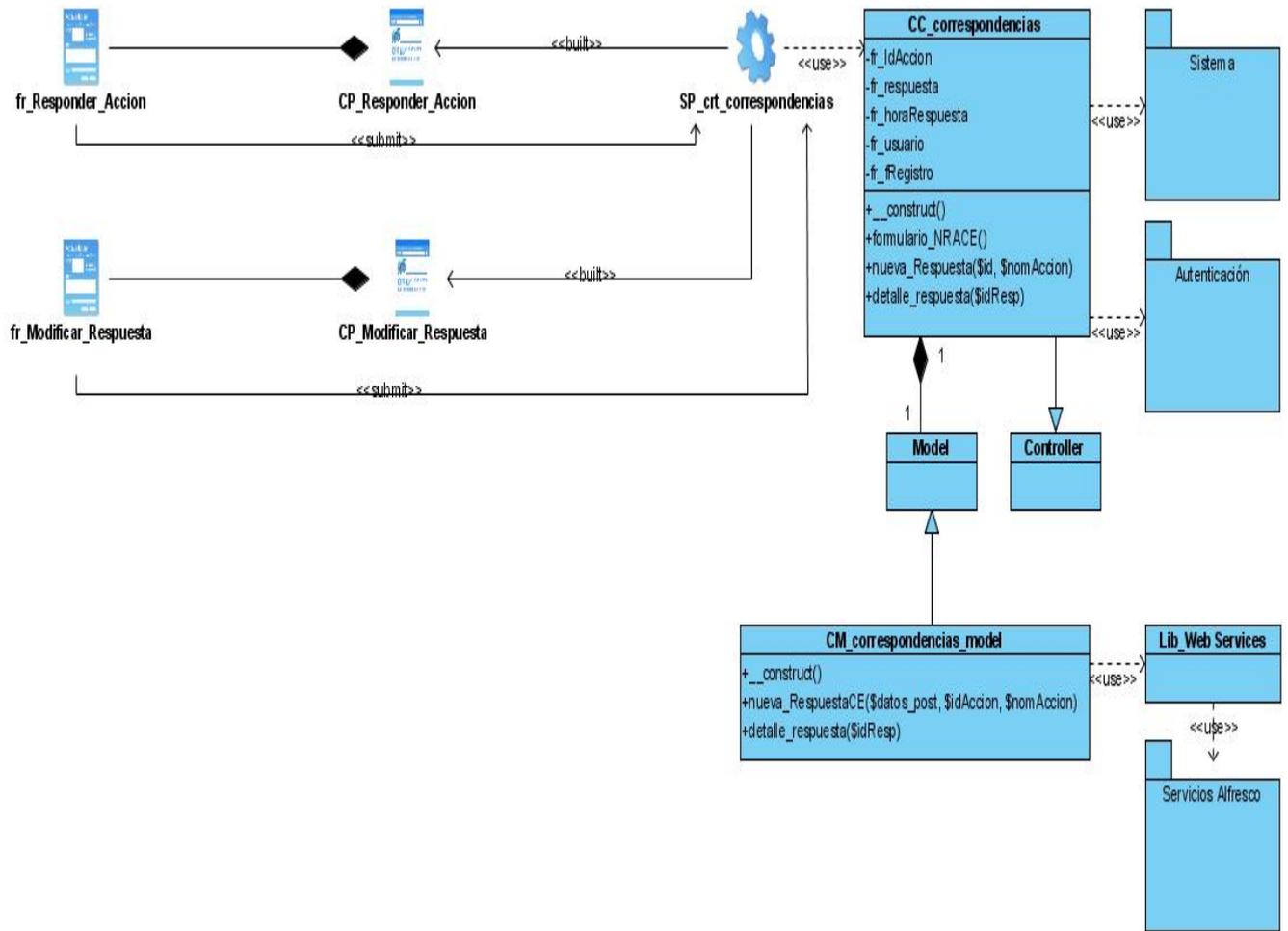


Capítulo 2. Descripción y análisis de la solución propuesta

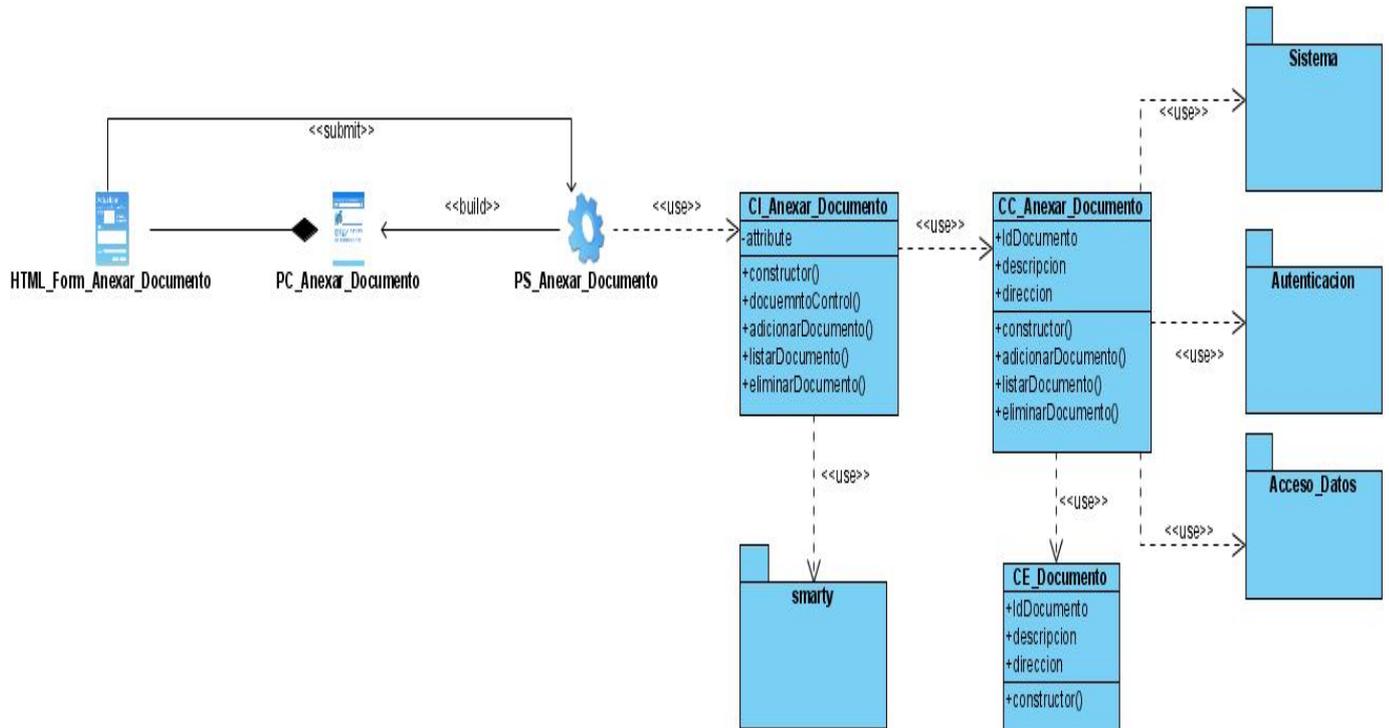
Paquete Correspondencias. Diagrama de Clases Gestionar Respuesta, propuesto por el analista.



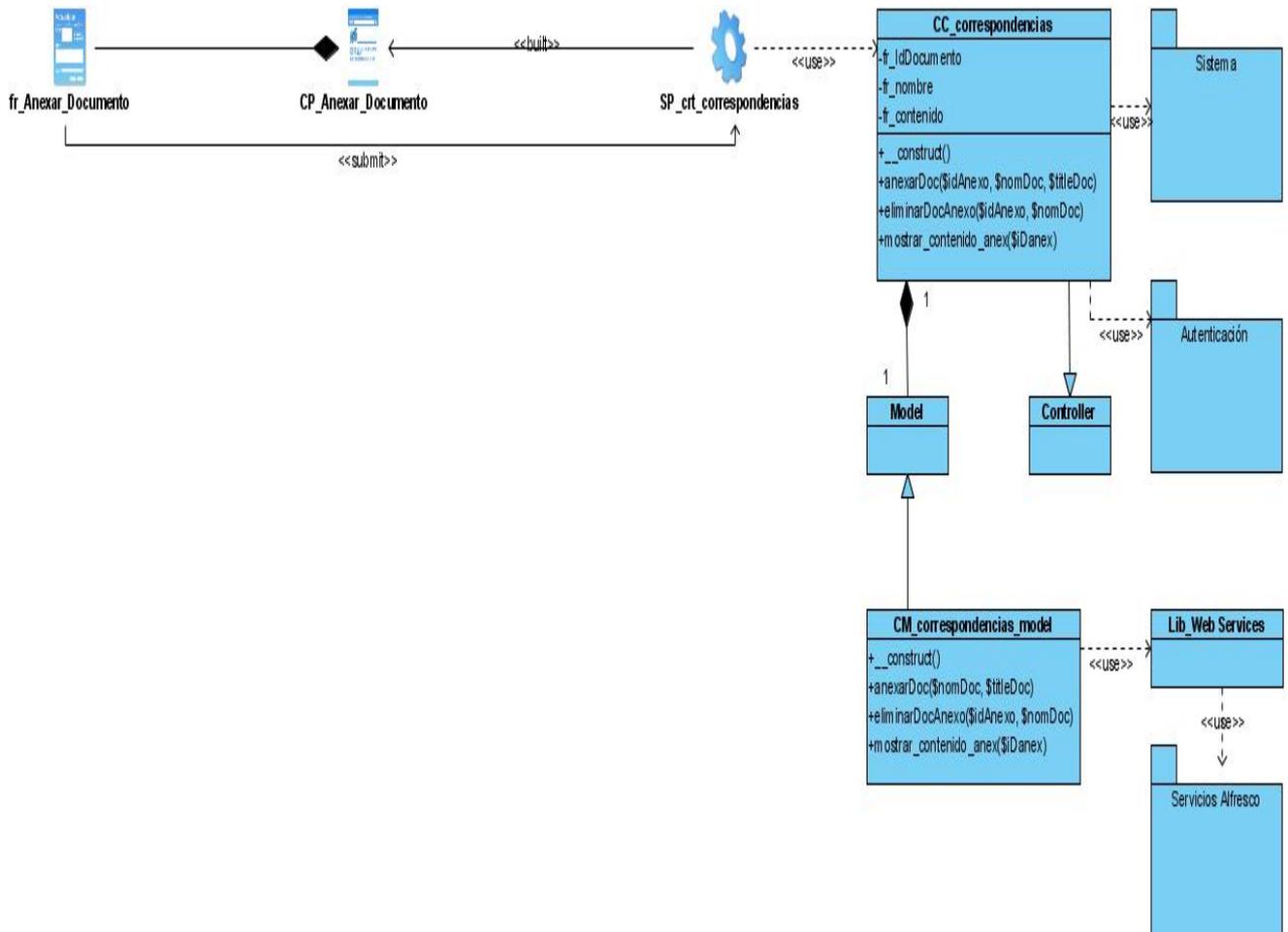
Paquete Correspondencias. Diagrama de Clases Gestionar Respuesta, versión final.



Paquete Correspondencias. Diagrama de Gestionar Documento Anexo, propuesto por el analista.



Paquete Correspondencias. Diagrama de Gestionar Documento Anexo, versión final.



Posterior a la valoración crítica de la propuesta de diseño del analista, que tiene una gran importancia ya que decide qué se asume y qué se modifica para la implementación del sistema, atendiendo a las necesidades reales del mismo, debe analizarse de igual forma la reutilización de componentes o módulos ya existentes, que pueden ser reutilizados, así como las estrategias de integración, el resultado de este análisis se traduce en un avance considerable en la implementación por concepto de ahorro de tiempo y esfuerzo.

2.3 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados

2.3.1 Codelgniter

En el capítulo anterior se analizó y se decidió emplear por sus características y potencialidades para la implementación del módulo, el framework de PHP Codelgniter. Entre las características que lo describen se mencionó que implementa el patrón arquitectónico Modelo-Vista-Controlador (MVC), que se definió utilizar como parte de la arquitectura del sistema y que es una aproximación al software que separa la lógica de la aplicación de la presentación y en la práctica permite que las páginas web contengan mínima codificación ya que la presentación es separada del código PHP. También puede ser fácilmente extendido a través del uso de plugins y librerías asistentes, o a través de extensión de clases o ganchos del sistema.

En el siguiente diagrama se muestra el flujo de datos a través del sistema.

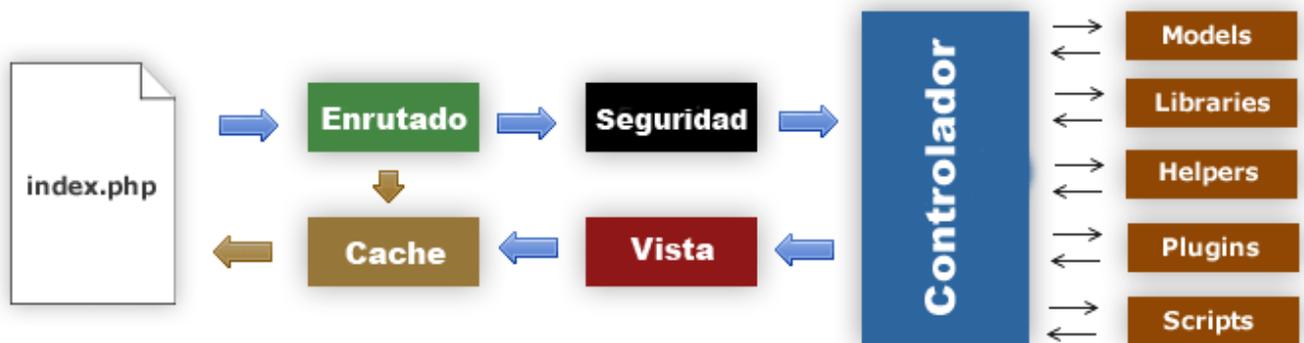


Figura 3. Diagrama de Flujo de la Aplicación

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr Codelgniter.
2. El Router examina la petición HTTP para determinar que debe ser hecho con él.
3. Si un archivo de caché existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.
4. Seguridad. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
5. El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.

Capítulo 2. Descripción y análisis de la solución propuesta

- La Vista finalizada es presentada entonces enviada al explorador web para ser vista. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas.

2.3.2 Web Services

Aprovechando la característica de ser extensible y como parte de la reutilización de códigos empleada, se le agregó la librería Web Services brindada por la comunidad de Alfresco para desde nuestra aplicación hacer uso de ese gestor documental, la siguiente tabla proporciona material de referencia que cubre todos los aspectos de los web services de Alfresco.

Servicio	Descripción	Estado
Autenticación	acceso y salida	Disponible
Repositorio	consulta y el modelo de manipulación	Disponible
Autoría	creación de contenidos colaborativos	Disponible
Clasificación	aplicar las clasificaciones y categorías	Disponible
Control de Acceso	roles, los permisos y la propiedad	Disponible
Acción	gestión de acciones y normas	Disponible
Administración	administración de usuarios, exportación e importación	Disponible
Contenido	manipulación de contenidos	Disponible

Tabla 1. Servicios Web ofrecidos por Alfresco

En este sentido hay que resaltar la amplia y actualizada documentación que existe sobre los Web Services en la propia Wikipedia de Alfresco proporcionando una vía para su mayor comprensión y mejor utilización.

En el [anexo 1](#) se muestran algunos diagramas donde se muestra la integración del repositorio de Alfresco con sistemas externos, donde se destaca los Web Services.

2.3.3 Clase de sesión

Otra clase que se investigó para su reutilización fue la Clase de Sesión que trae incluida el CodeIgniter y que permite mantener el "estado" de un usuario y seguir su actividad mientras navegue por el sitio. La clase de Sesión guarda información de sesión para cada usuario como datos serializados (y opcionalmente encriptados) en una cookie. La clase de Sesión no utiliza las sesiones nativas de PHP. Genera sus propios datos de sesión, ofreciendo más flexibilidad para los desarrolladores. Una vez inicializada, la clase de Sesión corre automáticamente. No hay nada que se necesite hacer para que el comportamiento anterior ocurra. Se podrá trabajar con datos de sesión o incluso agregar sus propios datos a la sesión del usuario, pero el proceso de lectura, escritura y actualización de sesión es automático.

2.4 Descripción de los algoritmos no triviales a implementar

Los algoritmos que se describirán a continuación tienen un nivel de complejidad superior a la media, y sus implementaciones son de vital importancia porque brindan soluciones a problemáticas claves a fin de satisfacer los requisitos funciones de la aplicación como son la búsqueda avanzada y crear la correspondencia entrante.

2.4.1 Búsqueda avanzada

Consiste en mostrar un listado con todas las correspondencias entrantes que contengan los parámetros de búsqueda especificados por el usuario.

Para su implementación fue necesario la creación de un algoritmo que partiendo de la entrada de uno o varios de los siguientes parámetros de búsqueda:

- Número externo
- Fecha de emisión desde
- Fecha de emisión hasta
- Fecha de recepción desde
- Fecha de recepción hasta
- Tipo de información
- Tipo de documento
- Abreviatura del organismo

- Procedencia

construyera la consulta con los parámetros válidos introducidos por el usuario y que ésta contenga la estructura correcta para que pueda ser interpretada por el motor de búsqueda que contiene el Alfresco, para lograr este objetivo fue necesario utilizar para las fechas una función auxiliar llamada `cambia_plequita` que se le pasa como parámetro la variable que contiene la fecha seleccionada por el usuario en el calendario y devuelve la misma fecha pero con el formato adecuado para que pueda ser interpretada correctamente por el motor de búsqueda del Alfresco. Después de construida la consulta, concatenando y validando todos los campos introducidos por el usuario, se crea la sesión alfresco que no es más que el puente para que fluyan los datos entre la aplicación y el repositorio de Alfresco, se le pasa la misma para que el motor genere todas las correspondencias entrantes que coincidan con los campos especificados y los devuelva en un arreglo y estas puedan ser mostradas al usuario siempre y cuando exista alguna.

En el [anexo 2](#) se muestran las vistas de la búsqueda avanzada y de los resultados encontrados en la misma.

2.4.2 Crear la correspondencia entrante

Consiste en registrar la nueva correspondencia entrante en el Alfresco en el estado correspondiente (En Redacción o Revisión).

Para su implementación fue necesario la creación de un algoritmo que verificara el estado (Para Redactar, Para Revisar) al cual se debía crear la correspondencia según la completitud o no de los datos necesarios requeridos para crear la correspondencia entrante, estos datos son:

- Número externo
- Fecha de emisión
- Fecha de recepción
- Datos del remitente
- Datos del destinatario
- Asunto
- Contenido
- Tipo de información

- Tipo de documento
- Abreviatura del organismo
- Procedencia
- Ubicación origen
- Ubicación destino
- Documentos anexos
- Preparado por

para adicionar una nueva correspondencia entrante en cualquier estado lo primero es crear la sesión alfresco, que no es más que el puente para que fluyan los datos entre la aplicación y el repositorio de Alfresco; ahora si falta alguno o algunos de los datos requeridos el algoritmo creará la correspondencia entrante en el estado de “Para Redactar” si esta es la opción que es usuario escoge, para que en otro momento pueda completar el registro de la misma. Para crear la correspondencia en el estado de “Para Revisar” el algoritmo verifica que todos los campos hayan sido introducidos por el usuario de forma correcta, para lograr esto se hace uso de la función desecho que lo que hace es crear un tipo de contenido en el Alfresco o en otras palabras un “Desecho” de la correspondencia entrante que se está registrando, para que si al usuario le faltó algún campo por llenar, cuando el algoritmo le diga que debe llenarlo no se pierdan los datos que había introducido con anterioridad, el mismo algoritmo se encarga de eliminar el “Desecho” del repositorio de Alfresco una vez que el usuario haya completado satisfactoriamente la introducción de los datos de la correspondencia entrante y ésta última se haya registrado satisfactoriamente en el estado correspondiente.

En el [anexo 3](#) se muestran las vistas para crear la nueva correspondencia entrante y el estado en la que es creada según la completitud de los datos introducidos por el usuario.

2.5 Descripción de las nuevas clases u operaciones necesarias

Siguiendo el enfoque MVC del CodeIgniter, las nuevas clases a describir caen en alguna de las siguientes categorías: Controladora, Modelo o Vista (Interfaz).

2.5.1 Clase Controladora

Nombre: correspondencias	
Tipo de clase: Controladora	
Atributos:	Tipo:
-	-
Funciones:	
Nombre:	logout
Descripción:	Permite salir de manera segura del sistema
Nombre:	formulario_NCE
Descripción:	Permite gestionar la creación de la correspondencia entrante
Nombre:	formulario_DCE(\$nInterno)
Descripción:	Permite visualizar la correspondencia entrante con todos sus detalles
Nombre:	formulario_MCE(\$idCorre)
Descripción:	Permite gestionar la modificación de los datos de una correspondencia entrante
Nombre:	formulario_LPECE
Descripción:	Permite gestionar el listado las correspondencias entrantes según estado en que se encuentran
Nombre:	mover_correspondenciasE(\$numeroInterno, \$estado)
Descripción:	Permite gestionar el traslado las correspondencias entrantes de un estado hacia otro
Nombre:	mostrar_contenido(\$ni)
Descripción:	Permite gestionar que se muestre el contenido de una

Capítulo 2. Descripción y análisis de la solución propuesta

	correspondencia entrante
Nombre:	mostrar_contenido_anex(\$iDanex)
Descripción:	Permite gestionar que se muestre el contenido de los anexos de la correspondencia entrante
Nombre:	formulario_BU(\$tipo)
Descripción:	Permite gestionar la búsqueda de usuarios internos o externos según criterio especificado
Nombre:	formulario_BACE
Descripción:	Permite gestionar la búsqueda de las correspondencias entrantes a través de uno o varios criterios especificados
Nombre:	asociar_RTTE(\$user, \$tipAsocia)
Descripción:	Manda para la sesión el usuario que va a ser asociado como remitente de la correspondencia entrante
Nombre:	asociar_Destinatario(\$user, \$tipAsocia)
Descripción:	Manda para la sesión el usuario que va a ser asociado como destinatario de la correspondencia entrante
Nombre:	asociar_prepPor(\$user, \$tipAsocia)
Descripción:	Manda para la sesión el usuario que va a ser asociado como “preparado por” de la correspondencia entrante
Nombre:	anexarDoc(\$idAnexo, \$nomDoc, \$titleDoc)
Descripción:	Manda para la sesión todos los documentos que van a ser anexados a la correspondencia entrante
Nombre:	agregarDoc
Descripción:	Sube a Alfresco el documento a anexar

Capítulo 2. Descripción y análisis de la solución propuesta

Nombre:	eliminarDocAnexo(\$idAnexo, \$nomDoc)
Descripción:	Gestiona la eliminación del nombre del documento de la sesión y de Alfresco
Nombre:	listar_acciones_userCE
Descripción:	Permite gestionar el listado de acciones pendientes de respuestas de las correspondencias entrantes
Nombre:	formulario_NACE()
Descripción:	Permite gestionar la creación de una nueva acción a la correspondencia entrante
Nombre:	eliminar_accion(\$idAccion)
Descripción:	Permite gestionar la eliminación las acciones de la correspondencia entrante
Nombre:	modificar_accionCEF(\$id)
Descripción:	Permite gestionar la modificación las acciones de la correspondencia entrante
Nombre:	detalle_accion(\$id)
Descripción:	Permite gestionar que se muestren los detalles de la acción de la correspondencia entrante.
Nombre:	formulario_NRACE()
Descripción:	Permite gestionar las respuestas a las acciones de la correspondencia entrante
Nombre:	formulario_NTE()
Descripción:	Permite gestionar la creación de un nuevo trabajador externo

Capítulo 2. Descripción y análisis de la solución propuesta

Nombre:	link_detalle_CE(\$idCorres)
Descripción:	Permite mandar para la vista los detalles de la correspondencia entrante
Nombre:	link_modificar_CE(\$idCorre, \$nomCorre)
Descripción:	Permite mandar para la vista los detalles de la correspondencia entrante a modificar

2.5.2 Clase Modelo

Nombre: correspondencias_model	
Tipo de clase: Modelo	
Atributos:	Tipo:
-	-
Funciones:	
Nombre:	registrar_CE(\$datos_post)
Descripción:	Registra una correspondencia entrante
Nombre:	redaccion_CE(\$datos_post)
Descripción:	Crea en el estado de "Redacción" la correspondencia entrante
Nombre:	buscar_CE(\$datos_post)
Descripción:	Busca las correspondencias entrantes a través de uno o varios criterios especificados
Nombre:	modificar_CE(\$idCorre, \$datos_post)
Descripción:	Modifica los datos de la correspondencia entrante especificada
Nombre:	detalles_CorrespondenciasE()
Descripción:	Retorna la correspondencia entrante con todos sus detalles

Capítulo 2. Descripción y análisis de la solución propuesta

Nombre:	listar_correspondenciasE_PorEstados(\$estado)
Descripción:	Lista las correspondencias entrantes según estado especificado
Nombre:	mover_CE(\$numeroInterno, \$estado)
Descripción:	Mueve la correspondencia entrante especificada hacia un estado determinado
Nombre:	nuevo_Usuario_Externo(\$datos_post)
Descripción:	Crea un nuevo trabajador externo
Nombre:	listar_acciones_userCE(\$user)
Descripción:	Lista las acciones pendientes de respuestas de las correspondencias entrantes de un usuario especificado
Nombre:	mostrar_contenido (\$ni)
Descripción:	Muestra el contenido de una correspondencia entrante especificada
Nombre:	mostrar_contenido_anex (\$iDanex)
Descripción:	Muestra los documentos anexados especificados de una correspondencia entrante
Nombre:	buscar_usuarios(\$tipo, \$datos_post)
Descripción:	Busca usuarios internos o externos especificados según criterios dados
Nombre:	dame_user (\$user)
Descripción:	Devuelve el objeto usuario según nombre de usuario especificado
Nombre:	agregarDoc
Descripción:	Sube al repositorio el documento a anexar

Capítulo 2. Descripción y análisis de la solución propuesta

Nombre:	anexarDoc (\$nomDoc, \$titleDoc)
Descripción:	Anexa los documentos especificados a la correspondencia entrante
Nombre:	eliminarDocAnexo (\$idAnexo, \$nomDoc)
Descripción:	Elimina el documento especificado de la sesión y de Alfresco
Nombre:	nueva_accionCE(\$datos_post)
Descripción:	Crea una nueva acción
Nombre:	dame_acciones
Descripción:	Devuelve las acciones asociadas a una correspondencia entrante
Nombre:	detalle_accion
Descripción:	Muestra los detalles de la acción
Nombre:	eliminar_accion(\$niC, \$idAccion)
Descripción:	Elimina una acción determinada
Nombre:	modificar_accionCEF(\$id, \$datos_post)
Descripción:	Modifica los datos de una acción
Nombre:	nueva_RespuestaCE(\$datos_post, \$idAccion, \$nomAccion)
Descripción:	Crea una nueva respuesta a una acción determinada

2.5.3 Clases Interfaz

Nombre: Autenticacion_al_Sistema_view
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz necesaria para autenticarse en el sistema, mediante usuario y contraseña.

Nombre: Descripcion_Modulos_Panel_de_Control_view
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz principal del SCD del Menpet con todos los módulos correspondientes.

Nombre: Descripcion_Modulos_Correspondencia_view
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz de los submódulos correspondencias entrantes y salientes del módulo correspondencias del SCD del Menpet.

Nombre: Partida
Tipo de clase: Interfaz
Descripción General: Muestra las interfaces necesarias para crear una nueva correspondencia entrante, ver las acciones pendientes de respuesta del usuario en el sistema, hacer búsquedas avanzadas de correspondencias entrantes y listar las correspondencias entrantes según el estado en que se encuentren.

Nombre: Gestionar_Documentos_Anexos_view
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz necesaria para anexar uno o varios documentos a la correspondencia entrante.

Nombre: Buscar_Correspondencia_entrante_view
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con el formulario para hacer búsquedas avanzadas de correspondencias entrantes según parámetros especificados por el usuario.

Nombre: Detalle_Correspondencia_Entrante
Tipo de clase: Interfaz

Descripción General: Muestra una interfaz con todos los detalles de la correspondencia entrante.

Nombre: Listado_Correspondencia_Entrante

Tipo de clase: Interfaz

Descripción General: Muestra una interfaz con el listado de correspondencias entrantes generado de la búsqueda realizada.
--

Nombre: Listado_Correspondencia_Entrante_Por_Part
--

Tipo de clase: Interfaz

Descripción General: Muestra una interfaz con los diferentes estados en que pueden encontrarse las correspondencias entrantes para que el usuario seleccione las de cual estado desea buscar.
--

Nombre: Listar_Acciones_User

Tipo de clase: Interfaz

Descripción General: Muestra una interfaz con las acciones pendientes de respuesta que tenga el usuario en el sistema y con las respuestas dadas a anteriores acciones.
--

Nombre: Modificar_CorrespondenciaE

Tipo de clase: Interfaz

Descripción General: Muestra una interfaz con el formulario la correspondencia entrante seleccionada para que el usuario designado le realice los cambios pertinentes y luego mandarla para el estado correspondiente.

Nombre: Nueva_Correspondencia_Entrante_view
--

Tipo de clase: Interfaz

Descripción General: Muestra una interfaz con el formulario para crear la nueva correspondencia entrante.
--

Nombre: Acciones_Opciones

Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con las acciones asociadas a la correspondencia entrante seleccionada y brinda además las opciones de crear una nueva acción, modificar o eliminar una ya existente.

Nombre: Detalle_Accion
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con los detalles de la acción y del registro de dicha acción que está asociada a la correspondencia entrante seleccionada.

Nombre: Detalle_Accion_User
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz al usuario con los detalles de la acción seleccionada y del registro de dicha acción, brindándole además la posibilidad de responder la misma.

Nombre: Detalle_Respuesta
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con los detalles de la respuesta y del registro de la misma.

Nombre: Modificar_Accion
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz del formulario que contiene los datos actuales de la acción seleccionada para que el usuario los pueda modificar a conveniencia.

Nombre: Modificar_Respuesta
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz con el formulario de la respuesta dada a una acción seleccionada para que el usuario la pueda modificar.

Capítulo 2. Descripción y análisis de la solución propuesta

Nombre: Nueva_Accion
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz con el formulario para crear una nueva acción asociada a la correspondencia entrante seleccionada.

Nombre: Nueva_Respuesta
Tipo de clase: Interfaz
Descripción General: Muestra la interfaz con el formulario para dar respuesta a una acción determinada.

Nombre: Pop-up_buscar_remitente_externo_view
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con el formulario para buscar el trabajador externo que envió la correspondencia entrante.

Nombre: Pop-up_buscar_remitente_view
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con el formulario para buscar el trabajador interno que envió la correspondencia entrante.

Después de haber sentado las bases de la arquitectura y de realizar una valoración crítica del diseño propuesto por el analista, se pudo arribar a un diseño acorde con las necesidades del sistema, a dicha propuesta se le realizó un conjunto de cambios que permiten implementar el sistema de forma más eficiente, aunque los cambios no fueron globales porque la estructura básica, dígame las clases y demás elementos que componen el diagrama de clases del diseño, se aceptaron, lo que varió fue la manera de relacionarlos.

También se realizó un análisis de las librerías y clases previamente implementadas a reutilizar, que repercuten en el ahorro de tiempo y código en la implementación por las facilidades que brindan al utilizarlas, entre ellas la estructura del framework de PHP, CodeIgniter y la API de Web Services que brinda Alfresco. De igual forma se describieron algunos de los algoritmos no triviales presentes en la solución, y las nuevas clases y funciones necesarias para solucionar el problema.

Capítulo 3. Validación de la solución propuesta

Una etapa vital en el desarrollo del software son las pruebas, guiadas fundamentalmente al descubrimiento de errores que se puedan encontrar o bien para verificar y revelar la calidad de un producto de software. La prueba del software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

Por tanto el presente capítulo tendrá como objetivo adentrarnos un poco en los diferentes niveles de pruebas existentes, y de ahí realizar una búsqueda o diseño de las pruebas unitarias que permitirán validar la solución propuesta, realizando además una descripción de estas, tomando en consideración el objetivo de la prueba, su alcance, tipo y detalles. Posteriormente se describirán los valores empleados para las pruebas y para concluir una evaluación de su ejecución y los resultados obtenidos.

3.1 Búsqueda o diseño de los test de unidades que permitan validar la solución propuesta

3.1.1 Niveles de prueba

Existen diferentes niveles de pruebas, cada una de ellas se realiza en determinados momentos del ciclo de vida del software, estos son:

- Unidad
- Integración
- Sistema
- Aceptación

Este modelo describe a un nivel alto de abstracción las fases del ciclo de desarrollo en las que se involucran las pruebas y los niveles de las mismas, la figura que se muestra a continuación describe este modelo.

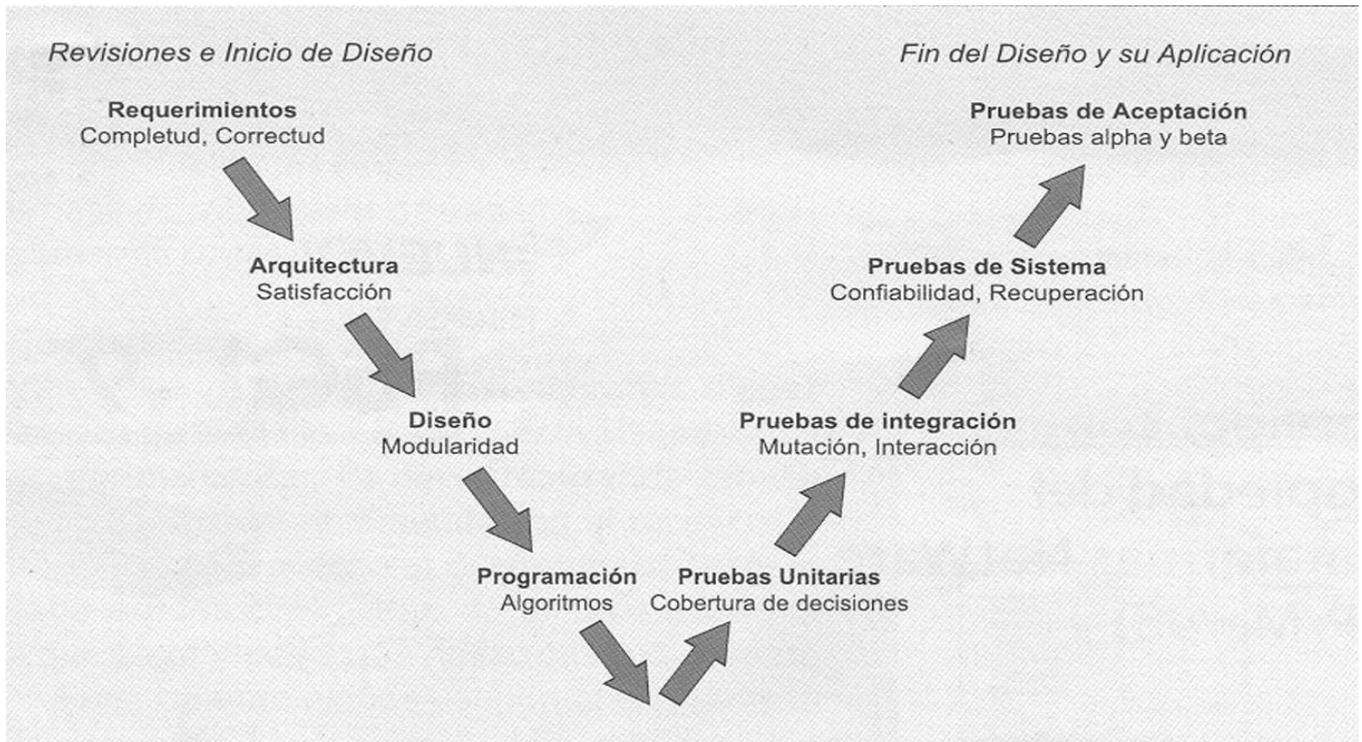


Figura 4. Niveles de prueba con las fases de desarrollo de software

3.1.1.1 Nivel de Unidad

Las pruebas de unidad son realizables a módulos del software, estas se simplifican cuando se diseñan con un alto grado de cohesión. Procede a comprobar las interfaces de usuarios, la estructura de los datos locales, las condiciones límites, los caminos independientes y los caminos de manejo de errores. Dentro de esta podemos encontrar dos fases fundamentales, la fase conocida como informal previa encargada de ejecutar pequeños ejemplos y la fase de prueba sistemática que contiene las técnicas de pruebas de caja negra centrada a la comprobación de las especificaciones funcionales y las técnicas de prueba de caja blanca que requieren para su aplicación del conocimiento de la estructura interna del programa derivadas a partir de las especificaciones internas del diseño o del código.

Los casos de pruebas diseñados para ejecutarse a nivel de unidad deben descubrir errores como:

- Comparaciones entre tipos de datos distintos.
- Operadores lógicos o de precedencia incorrectos.
- Igualdad esperada cuando los errores de precisión la hacen poco probable.
- Variables o comparaciones incorrectas.

- Terminación de bucles inapropiada o inexistente.
- Fallo de salida cuando se encuentra una iteración divergente.
- Variables de bucles modificadas de forma inapropiada.

3.1.1.2 Nivel de Integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción.

El objetivo es coger los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. En el proceso de integración puede derivarse la realización de los casos de uso del sistema, estos describen como interactúan las clases y los objetos y por lo tanto la interacción de los componentes.

Los encargados de diseñar los casos de pruebas de integración consideran como entrada a los diagramas de secuencia de las realizaciones de los casos de uso, pues ahí buscan las combinaciones de entradas, salidas y estado inicial del sistema que dan lugar a escenarios que usan las clases y por lo tanto los componentes que participan en los diagramas. Un caso de prueba se deriva de un diagrama de secuencia, este describe una o varias secuencias del negocio.

Después, se procede a ejecutar el caso de prueba de integración diseñado, creando trazas de ejecución o ejecutándolo paso a paso, a continuación se compara las interacciones actuales con los diagramas de secuencia y de no ser igual se trata de un defecto que puede dar al traste con un error.

3.1.1.3 Nivel de Sistema

Las pruebas de sistemas caen fuera del ámbito del proceso de software y no las realiza únicamente el desarrollador del software. Sin embargo los pasos durante el diseño del software y durante la prueba pueden mejorar enormemente la posibilidad de éxito de las pruebas de sistema.

Las pruebas de sistema principalmente se centran en verificar la interacción de los actores con el sistema, por lo que a menudo los casos de pruebas se obtienen a partir de las descripciones de los casos de uso. Aunque también se le aplican prueba al sistema como un todo.

3.1.1.4 Nivel de Aceptación

La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado pruebas Alfa o Beta para descubrir errores considerando que solo el usuario final puede descubrir.

La prueba Alfa es llevada a cabo por el cliente en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador de observador registrando los errores y los problemas de uso. Estas pruebas se llevan a cabo en un entorno controlado.

Las pruebas Betas se llevan a cabo por los usuarios finales en sus puestos de trabajo, y a diferencia de la prueba Alfa el desarrollador no está presente, así que esta prueba no puede ser controlada por el desarrollador, por lo que el cliente debe registrar todas las inconformidades y tramitárselas al desarrollador.

3.1.2 Pruebas a realizar en tiempo de desarrollo

Pruebas informales o fase de prueba informal, son aquellas pruebas que hace el desarrollador en su puesto de trabajo, tienen como objetivo comprobar que el programa compile y ver que todo está yendo como debiera, normalmente se realizan varios cientos de estas pruebas que básicamente consisten en compilar periódicamente durante el desarrollo y ejecutar para ver el resultado.

Dentro de las pruebas en tiempo de desarrollo se encuentran las pruebas de unidades, estas son pruebas de menor escala y consisten en probar cada uno de los módulos que conforman el programa, cuando estos módulos son extensos o complejos se dividen para probar objetivamente partes más pequeñas, este tipo de pruebas es la más común. El tipo de prueba de caja blanca es característico y de gran importancia dentro de las pruebas de unidades al igual que el de caja negra y no es recomendable sustituir uno por otro ya que un método complementa al otro. A nuestro software se le aplicaron ambos métodos siendo el de Caja Negra el escogido para su descripción y documentación en el presente capítulo porque muestra el correcto cumplimiento de los requisitos funcionales del software implementado.

3.1.3 Método de caja negra

La prueba de Caja Negra se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Capítulo 3. Validación de la solución propuesta

La misma no es una alternativa a las técnicas de prueba anteriormente vistas, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la caja blanca.

Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de inicialización y terminación.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa, según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Para esta prueba, existen varias técnicas, entre ellas están:

1. Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
2. Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más usadas y efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos de pruebas antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar. En este apartado se hará uso de esta técnica para aplicarle a cierto caso de uso crítico del sistema.

Los pasos a seguir para este método son:

- Identificar las clases de equivalencia válidas e inválidas para cada condición de entrada.
- Escribir un nuevo caso de prueba para cubrir tantas clases de equivalencia válidas como sea posible.
- Escribir un nuevo caso de prueba que cubra una y solo una clase de equivalencia inválida.

3.2 Descripción de los test de caja negra

Objetivo del test

Detectar el incorrecto o incompleto funcionamiento, así como los errores de interfaces, rendimiento, errores de inicialización y terminación de caso de uso crítico:

- Gestionar Correspondencia Entrante.

Las pruebas realizadas y descritas a este caso de uso son:

- ✓ Adicionar la correspondencia entrante en el estado “Para Redactar”.
- ✓ Adicionar la correspondencia entrante en el estado “Para Revisar”.
- ✓ Asociar “remitente” a la correspondencia entrante.
- ✓ Asociar “destinatario” a la correspondencia entrante.
- ✓ Asociar “preparado por” a la correspondencia entrante.
- ✓ Adjuntar “documento anexo” a la correspondencia entrante.

3.2.1 CPR 1: Adicionar la correspondencia entrante en el estado “Para Redactar”.

Descripción

El escenario se inicia cuando el usuario necesita adicionar la correspondencia entrante. Selecciona la opción de “Nueva Correspondencia Entrante” en el menú principal del submódulo de correspondencias entrantes y llena parcialmente los datos de la misma, finalizando así el escenario.

Capítulo 3. Validación de la solución propuesta

Flujo central

El flujo central no es más que la interacción usuario y aplicación, paso a paso, el pedido y la respuesta.

- El sistema muestra la interfaz que permite la autenticación al usuario.
- Si el usuario se autentica correctamente, el sistema muestra la interfaz principal del panel de control.
- El usuario selecciona el módulo correspondencias y dentro de él el submódulo correspondencias entrantes.
- Según los permisos que tiene asignado el usuario autenticado el sistema le muestra el panel de control correspondiente.
- El usuario con permisos adecuados selecciona la opción “Nueva Correspondencia Entrante” del menú principal de trabajo.
- El usuario llena algunos o todos los datos de la correspondencia entrante.
- El usuario pulsa el botón “En Redacción”.

Pre condiciones

- Usuario autenticado satisfactoriamente en el sistema.

Tienen que estar presente en el repositorio de Alfresco.

- Grupos de usuarios con los permisos asignados.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de Prueba	Obs
El usuario autenticado tiene los permisos adecuados.	El sistema no muestra la interfaz para crear la correspondencia.	El sistema muestra la interfaz correspondiente.	Satisfactorio	
El usuario autenticado no tiene los permisos adecuados.	El sistema muestra la interfaz para crear la correspondencia.	El sistema muestra la interfaz correspondiente.	Satisfactorio	

Capítulo 3. Validación de la solución propuesta

El usuario selecciona la opción “Nueva Correspondencia Entrante” del menú principal de trabajo.	No se muestra la interfaz de la nueva correspondencia entrante.	Muestra la interfaz de la nueva correspondencia entrante.	Satisfactorio	
El usuario llena algunos de los datos de la correspondencia entrante.		El sistema muestra el siguiente mensaje: “La correspondencia ha pasado al estado En Redacción”.	Satisfactorio	
El usuario llena todos los datos de la correspondencia entrante.		El sistema muestra el siguiente mensaje: “La correspondencia ha pasado al estado En Redacción”.	Satisfactorio	
	El usuario no llena ninguno de los datos de la correspondencia entrante.	El sistema muestra el siguiente mensaje: “Por favor DEBE llenar al menos uno de los campos”.	Satisfactorio	

Tabla 2. Sección: Adicionar la correspondencia entrante en el estado “Para Redactar”

3.2.2 CPR 2: Adicionar la correspondencia entrante en el estado “Para Revisar”.

Descripción

El escenario se inicia cuando el usuario necesita adicionar la correspondencia entrante. Selecciona la opción de “Nueva Correspondencia Entrante” en el menú principal del submódulo de correspondencias entrantes y llena totalmente los datos de la misma, finalizando así el escenario.

Flujo central

- El sistema muestra la interfaz que permite la autenticación al usuario.

- Si el usuario se autentica correctamente, el sistema muestra la interfaz principal del panel de control.

Capítulo 3. Validación de la solución propuesta

- El usuario selecciona el módulo correspondencias y dentro de él el submódulo correspondencias entrantes.
- Según los permisos que tiene asignado el usuario autenticado el sistema le muestra el panel de control correspondiente.
- El usuario con permisos adecuados selecciona la opción “Nueva Correspondencia Entrante” del menú principal de trabajo.
- El usuario llena todos los datos de la correspondencia entrante.
- El usuario pulsa el botón “Registrar”.

Pre condiciones

- Usuario autenticado satisfactoriamente en el sistema.

Tienen que estar presente en el repositorio de Alfresco.

- Grupos de usuarios con los permisos asignados.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de Prueba	Obs
El usuario autenticado tiene los permisos adecuados.	El sistema no muestra la interfaz para crear la correspondencia.	El sistema muestra la interfaz correspondiente.	Satisfactorio	
El usuario autenticado no tiene los permisos adecuados.	El sistema muestra la interfaz para crear la correspondencia.	El sistema muestra la interfaz correspondiente.	Satisfactorio	
El usuario selecciona la opción “Nueva Correspondencia Entrante” del menú principal de trabajo.	No se muestra la interfaz de la nueva correspondencia entrante.	Muestra la interfaz de la nueva correspondencia entrante.	Satisfactorio	
El usuario llena todos los datos de la		El sistema muestra el siguiente mensaje:	Satisfactorio	El proceso no ocurre de forma

Capítulo 3. Validación de la solución propuesta

correspondencia entrante.		“La correspondencia se ha Adicionado Correctamente”.		instantánea, debido al número de verificaciones que realiza el algoritmo para registrar la correspondencia.
	El usuario no llena ninguno de los datos de la correspondencia entrante.	El sistema muestra el siguiente mensaje: “Por favor DEBE llenar todos los campos”.	Satisfactorio	
	El usuario deja de llenar alguno de los datos de la correspondencia entrante.	El sistema muestra el siguiente mensaje: “Por favor DEBE llenar todos los campos”.	Satisfactorio	
	El usuario no introduce alguna de las asociaciones (remitente, destinatario, preparado por) de la correspondencia entrante.	El sistema muestra el mensaje correspondiente a la asociación o asociaciones que no introdujo ejemplo: “Agregue: Remitente, Destinatario o Quien Preparo la Correspondencia”.	Satisfactorio	

Tabla 3. Sección: Adicionar la correspondencia entrante en el estado “Para Revisar”

3.2.3 CPR 3: Asociar “remitente” a la correspondencia entrante.

Descripción

El escenario se inicia cuando el usuario necesita asociar a la correspondencia entrante el remitente. Selecciona la opción de “Gestionar Remitente” en el formulario de la correspondencia entrante y busca el usuario del trabajador externo que remitió la correspondencia, finalizando así el escenario.

Flujo central

- El sistema muestra la interfaz (Pop_up) que permite buscar y asociar el trabajador externo como remitente a la correspondencia entrante.
- El usuario introduce cualquiera de los datos del trabajador externo y pulsa el botón “Buscar”.
- El sistema muestra los datos del trabajador o trabajadores externos que coincidieron con los parámetros de búsqueda.
- El usuario pulsa la opción de “asociar” del trabajador externo que desea como remitente de la correspondencia entrante.

Pre condiciones

- Usuario autenticado satisfactoriamente en el sistema.

Tienen que estar presente en el repositorio de Alfresco.

- Los datos y usuarios de los trabajadores externos.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de Prueba	Obs
El usuario selecciona la opción “Gestionar Remitente” del formulario de la correspondencia entrante.	No se muestra la interfaz (Pop_up) para buscar el remitente externo.	Muestra la interfaz (Pop_up) para buscar el remitente externo.	Satisfactorio	
El usuario introduce el dato o los datos que conoce del trabajador externo y pulsa “Buscar”.		El sistema muestra el listado de los trabajadores externos que coinciden con los datos especificados.	Satisfactorio	

Capítulo 3. Validación de la solución propuesta

	El usuario no llena ninguno de los datos de trabajadores externos.	El sistema muestra el siguiente mensaje: “Por favor DEBE llenar al menos uno de los campos”.	Satisfactorio	
	El usuario introduce datos incorrectos del trabajador externo.	El sistema muestra el siguiente mensaje: “No se encontró ningún USUARIO que coincidiera con los datos especificados”.	Satisfactorio	

Tabla 4. Sección: Asociar “remitente” a la correspondencia entrante

3.2.4 CPR 4: Asociar “destinatario” a la correspondencia entrante.

Descripción

El escenario se inicia cuando el usuario necesita asociar a la correspondencia entrante el destinatario. Selecciona la opción de “Gestionar Destinatario” en el formulario de la correspondencia entrante y busca el usuario del trabajador interno a quien va dirigida la correspondencia, finalizando así el escenario.

Flujo central

- El sistema muestra la interfaz (Pop_up) que permite buscar y asociar el trabajador interno como destinatario a la correspondencia entrante.
- El usuario introduce cualquiera de los datos del trabajador interno y pulsa el botón “Buscar”.
- El sistema muestra los datos del trabajador o trabajadores internos que coincidieron con los parámetros de búsqueda.
- El usuario pulsa la opción de “asociar” del trabajador interno que desea como destinatario de la correspondencia entrante.

Pre condiciones

Capítulo 3. Validación de la solución propuesta

- Usuario autenticado satisfactoriamente en el sistema.

Tienen que estar presente en el repositorio de Alfresco.

- Los datos y usuarios de los trabajadores internos.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de Prueba	Obs
El usuario selecciona la opción "Gestionar Destinatario" del formulario de la correspondencia entrante.	No se muestra la interfaz (Pop_up) para buscar el destinatario interno.	Muestra la interfaz (Pop_up) para buscar el destinatario interno.	Satisfactorio	
El usuario introduce el dato o los datos que conoce del trabajador interno y pulsa "Buscar".		El sistema muestra el listado de los trabajadores internos que coinciden con los datos especificados.	Satisfactorio	
	El usuario no llena ninguno de los datos de los trabajadores internos.	El sistema muestra el siguiente mensaje: "Por favor DEBE llenar al menos uno de los campos".	Satisfactorio	
	El usuario introduce datos incorrectos del trabajador interno.	El sistema muestra el siguiente mensaje: "No se encontró ningún USUARIO que coincidiera con los datos especificados".	Satisfactorio	

Tabla 5. Sección: Asociar "destinatario" a la correspondencia entrante

3.2.5 CPR 5: Asociar "preparado por" a la correspondencia entrante.

Descripción

El escenario se inicia cuando el usuario necesita asociar a la correspondencia entrante quien preparó la correspondencia. Selecciona la opción de “Gestionar Preparado Por” en el formulario de la correspondencia entrante y busca el usuario del trabajador interno que preparó la correspondencia, finalizando así el escenario.

Flujo central

- El sistema muestra la interfaz (Pop_up) que permite buscar y asociar el trabajador interno que preparó la correspondencia entrante.
- El usuario introduce cualquiera de los datos del trabajador interno y pulsa el botón “Buscar”.
- El sistema muestra los datos del trabajador o trabajadores internos que coincidieron con los parámetros de búsqueda.
- El usuario pulsa la opción de “asociar” del trabajador interno que preparó la correspondencia entrante.

Pre condiciones

- Usuario autenticado satisfactoriamente en el sistema.

Tienen que estar presente en el repositorio de Alfresco.

- Los datos y usuarios de los trabajadores internos.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de Prueba	Obs
El usuario selecciona la opción “Gestionar Preparado Por” del formulario de la correspondencia entrante.	No se muestra la interfaz (Pop_up) para buscar el trabajador interno que preparó la correspondencia entrante.	Muestra la interfaz (Pop_up) para buscar trabajador interno que preparó la correspondencia entrante.	Satisfactorio	

El usuario introduce el dato o los datos que conoce del trabajador interno y pulsa "Buscar".		El sistema muestra el listado de los trabajadores internos que coinciden con los datos especificados.	Satisfactorio	
	El usuario no llena ninguno de los datos del trabajador interno que preparó la correspondencia entrante.	El sistema muestra el siguiente mensaje: "Por favor DEBE llenar al menos uno de los campos".	Satisfactorio	
	El usuario introduce datos incorrectos del trabajador interno que preparó la correspondencia.	El sistema muestra el siguiente mensaje: "No se encontró ningún USUARIO que coincidiera con los datos especificados".	Satisfactorio	

Tabla 6. Sección: Asociar "preparado por" a la correspondencia entrante

3.2.6 CPR 6: Adjuntar "documento anexo" a la correspondencia entrante.

Descripción

El escenario se inicia cuando el usuario necesita adjuntar a la correspondencia entrante uno o varios documentos anexos con su formato determinado. Selecciona la opción de "Gestionar Documento Anexo" en el formulario de la correspondencia entrante y busca el o los documentos que desea anexar a la correspondencia entrante, finalizando así el escenario.

Flujo central

- El sistema muestra la interfaz (Pop_up) que permite buscar y anexar los documentos a la correspondencia entrante.
- El usuario pulsa el botón "Examinar" para cargar el documento anexo y luego pulsa "Agregar a la Lista".

Capítulo 3. Validación de la solución propuesta

- El sistema le va mostrando los documentos que ha agregado a la lista.
- El usuario pulsa la opción de “anexar” el o los documentos que desee anexar a la correspondencia entrante.
- El sistema le va mostrando los documentos que han sido agregados como anexos a la correspondencia entrante.

Pre condiciones

- Usuario autenticado satisfactoriamente en el sistema.
- Que los documentos anexos sean del tipo correcto.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de Prueba	Obs
El usuario selecciona la opción “Gestionar Documentos Anexos” del formulario de la correspondencia entrante.	No se muestra la interfaz (Pop_up) para buscar los documentos anexos.	Muestra la interfaz (Pop_up) para buscar los documentos anexos.	Satisfactorio	
El usuario pulsa el botón “Examinar” para buscar el documento a anexar.	No se muestra la ventana para cargar los documentos.	Muestra la ventana para cargar los documentos.	Satisfactorio	
El usuario busca el documento que desea anexar a la correspondencia entrante y pulsa el botón “Open”.	No muestra la interfaz (Pop_up) con la ubicación del documento cargada.	Muestra la interfaz (Pop_up) con la ubicación del documento cargada.	Satisfactorio	
El usuario pulsa el botón “Agregar a la Lista” para conformar		El sistema muestra el siguiente mensaje: “El Documento fue	Satisfactorio	

Capítulo 3. Validación de la solución propuesta

la lista de documentos que desea anexar.		agregado a la lista” y se muestra en la interfaz (Pop_up) los documentos que ha agregado a la lista para anexar.		
	El usuario pulsa el botón “Agregar a la Lista” para conformar la lista de documentos que desea anexar, pero el tipo de documento no es del formato correcto.	El sistema muestra el siguiente mensaje: “Por favor esa EXTENSIÓN es incorrecta”.	Satisfactorio	
	El usuario pulsa el botón “Agregar a la Lista” sin haber cargado ningún documento.	El sistema muestra el siguiente mensaje: “Por favor escoja el DOC que quiere agregar a la lista”.	Satisfactorio	
El usuario pulsa el ícono “Eliminar” del documento de la lista de documentos para anexar que desea eliminar.		El sistema muestra el siguiente mensaje: ”El Documento (Nombre del Documento) fue eliminado” y es eliminado de la lista de documentos para anexar.	Satisfactorio	
El usuario pulsa el ícono “Anexar” del documento de la lista de documentos para anexar que desea anexar.		El sistema muestra el siguiente mensaje: ”El Documento (Nombre del Documento) fue anexado” y se muestra en la interfaz (Pop_up) los documentos que ha anexado a la correspondencia.	Satisfactorio	
El usuario pulsa el		El sistema muestra el	Satisfactorio	

Capítulo 3. Validación de la solución propuesta

ícono "Eliminar" del documento de la lista de documentos anexados que desea eliminar.		siguiente mensaje: "El Documento (Nombre del Documento) fue eliminado" y es eliminado de los documentos anexados a la correspondencia entrante.		
---	--	--	--	--

Tabla 7. Sección: Adjuntar "documento anexo" a la correspondencia entrante

En este capítulo se realizó un estudio de los principales niveles, métodos y tipos de pruebas que se llevan a cabo durante el ciclo de desarrollo de un software y en especial sobre la técnica de la Partición de Equivalencia del método de Caja Negra que fue la empleada. Se describieron los test de caja negra realizados con esta técnica, y los resultados obtenidos del caso de uso crítico Gestionar Correspondencia Entrante, al cual se le realizaron los siguientes casos de pruebas: Adicionar la correspondencia entrante en el estado "Para Redactar", Adicionar la correspondencia entrante en el estado "Para Revisar", Asociar "remitente" a la correspondencia entrante, Asociar "destinatario" a la correspondencia entrante, Asociar "preparado por" a la correspondencia entrante, Adjuntar "documento anexo" a la correspondencia entrante. Demostrándose que las funcionalidades del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene y que el rendimiento es aceptable, permitiendo validar de esta forma la calidad del producto, aunque lograr un software 100% libre de errores es un poco utópico aún más cuando se trata de sistemas complejos.

Conclusiones

El sistema de control de documentos (SCD) es una parte importante dentro de la estrategia empresarial del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET) y dentro de este, el módulo correspondencias es de vital importancia. Con la implementación de dicho módulo se llegó a las siguientes conclusiones:

- ✓ Se realizó un estudio del arte de las tecnologías actuales para el desarrollo de sistemas web, haciendo énfasis en las tecnologías y herramientas libres, de lo que se decidió utilizar para la implementación del sistema, como lenguaje de programación PHP 5 en su versión 5.2, con el framework CodeIgniter versión 1.7.1, sobre el entorno de desarrollo Netbeans 6.5 para PHP y como repositorio de información el sistema de gestión documental Alfresco 3.0.

- ✓ Se realizó una valoración crítica del diseño propuesto por el analista, que permitió adecuar esta propuesta inicial al diseño final de la aplicación.

- ✓ Se analizó un conjunto de librerías y clases previamente implementadas, logrando su reutilización en la implementación del módulo.

- ✓ Se realizaron un conjunto de pruebas para validar la solución propuesta, enfocándose fundamentalmente en las pruebas unitarias y en especial en la técnica de la Partición de Equivalencia del método de Caja Negra siendo esta última la empleada, haciéndose una descripción de los test de caja negra y de los resultados obtenidos.

Recomendaciones

Los autores del presente trabajo recomiendan:

- Integrar dicho Sistema con el Sistema Interno Central.
- Realizar pruebas de integración al SCD del MENPET una vez terminado.
- Hacer uso de Ajax para hacer más rápidas las búsquedas de contenidos y usuarios.
- Utilizar en las soluciones informáticas que gestionen documentos al Sistema de Gestión Documental Alfresco.

Referencias Bibliográficas

1. TRAMULLAS, J. *Herramientas de software libre para la gestión de contenidos* [Consultado el: 15 de Octubre de 2008]. Disponible en: <http://www.hipertext.net/web/pag258.htm>.
2. WIKIPEDIA, L. E. L. *Gestión Documental* [Consultado el: 3 de Noviembre de 2008]. Disponible en: http://es.wikipedia.org/wiki/Gesti%C3%B3n_documental.
3. ---. *Desarrollo de la Gestión Documental* [Consultado el: 13 de Enero de 2009]. Disponible en: http://es.wikipedia.org/wiki/Gesti%C3%B3n_documental.
4. MENPET. *¿Qué es el Ministerio del Poder Popular para la Energía y Petróleo?* [Consultado el: 10 de Noviembre de 2008]. Disponible en: <http://www.menpet.gob.ve/portalmenpet/secciones.php?option=view&idS=2>.
5. LEMONTECH. *Alfresco Enterprise Content Management* [Consultado el: 9 de Enero de 2009]. Disponible en: <http://www.lemontech.cl/cms/index.php?id=50>.
6. WIKIPEDIA, L. E. L. *Alfresco, Reconocimientos* [Consultado el: 20 de Febrero de 2009]. Disponible en: <http://es.wikipedia.org/wiki/Alfresco>.
7. UNIVERSIDAD TECNOLÓGICA DE TUCUMAN, D. D. S. *PARADIGMAS DE PROGRAMACION* [Consultado el: 10 de Mayo de 2009]. Disponible en: <http://www.frt.utn.edu.ar/sistemas/paradigmas/page22.html>.
8. MONOGRAFÍAS.COM, L. *El Paradigma Lógico* [Consultado el: 10 de Mayo de 2009].
9. MÜLLER, P. *Programación Orientada a Objetos* [Consultado el: 10 de Mayo de 2009]. Disponible en: <http://www.gnacademy.org/text/cc/Tutorial/Spanish/node3.html#SECTION00360000000000000000>.
10. SCHWINDT, A. *Construcción de Sistemas Multiplataforma basados en Servicios* [Consultado el: 10 de Mayo de 2009]. Disponible en: <http://msdn.microsoft.com/es-es/library/bb972254.aspx>.
11. GUTIÉRREZ, J. J. *¿Qué es un framework web?* [Consultado el: 10 de Mayo de 2009]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
12. EGUILUZ, J. *El framework Symfony, una introducción práctica (I parte)* [Consultado el: 10 de Mayo de 2009]. Disponible en: <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.

13. ZENDFRAMEWORKDES.COM. *Capítulo 1. Introducción a Zend Framework, Descripción general* [Consultado el: 11 de Mayo de 2009]. Disponible en: <http://manual.zfdes.com/es/introduction.html>.
14. CODEIGNITER, U. G. V. *Welcome to CodeIgniter* [Consultado el: 10 de Mayo de 2009]. Disponible en: http://codeigniter.com/user_guide/index.html.
15. ALVAREZ, M. A. *Zend Studio* [Consultado el: 11 de Mayo de 2009]. Disponible en: <http://www.desarrolloweb.com/articulos/1178.php>.
16. ZEND.COM. *Zend Studio Features* [Consultado el: 11 de Mayo de 2009]. Disponible en: <http://www.zend.com/en/products/studio/features>.
17. WIKIPEDIA, L. E. L. *Adobe Dreamweaver* [Consultado el: 11 de Mayo de 2009]. Disponible en: http://es.wikipedia.org/wiki/Adobe_Dreamweaver.
18. NETBEANS.ORG. *NetBeans - Development Simplified* [Consultado el: 11 de Mayo de 2009]. Disponible en: <http://www.netbeans.org/features/>.
19. CERDA, F. *Características de NetBeans* [Consultado el: 11 de Mayo de 2009]. Disponible en: http://www.blog.cl/talks/netbeans65es_cl.pdf.
20. WIKIPEDIA, L. E. L. *Aplicación web* [Consultado el: 15 de Febrero de 2009]. Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web.
21. VEGAS, J. *Desarrollo de Aplicaciones Web* [Consultado el: 15 de Febrero de 2009]. Disponible en: <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node17.html>.
22. MASTERNEWMEDIA.ORG. *Beneficios De Las Aplicaciones Basadas En Web* [Consultado el: 15 de Febrero de 2009]. Disponible en: http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.
23. WIKIPEDIA, L. E. L. *Cliente-servidor* [Consultado el: 11 de Febrero de 2009]. Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>.
24. ---. *Servidor HTTP Apache* [Consultado el: 9 de Enero de 2009]. Disponible en: http://es.wikipedia.org/wiki/Apache_http_server.
25. CIBERAULA.COM. *Una Introducción a APACHE* [Consultado el: 9 de Enero de 2009]. Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/.

26. GUERVOS, J. J. M. *Introducción al lenguaje XML* [Consultado el: 10 de Febrero de 2009]. Disponible en: <http://geneura.ugr.es/~jmerelo/xml/>.
27. WIKIPEDIA, L. E. L. *Extensible Markup Language* [Consultado el: 10 de Febrero de 2009]. Disponible en: http://es.wikipedia.org/wiki/XML#Objetivos_y_usos_de_XML.
28. ALFRESCO.COM. *The Open Source Alternative for Enterprise Content Management* [Consultado el: 11 de Febrero de 2009]. Disponible en: <http://www.alfresco.com/es/>.
29. WELICKI, L. *Patrones y Antipatrones: una Introducción - Parte II* [Consultado el: 17 de Febrero de 2009]. Disponible en: <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.

Bibliografía

ALFRESCO.COM. The Open Source Alternative for Enterprise Content Management. Disponible en: <http://www.alfresco.com/es/>.

ALVAREZ, M. A. Zend Studio. Disponible en: <http://www.desarrolloweb.com/articulos/1178.php>.

ARSAUTE, A. S.; G. I. AGUIRRE, et al. Un workflow basado en Java para la gestión documental.

CELIS, J. A. T. Tecnología Híbrida: una excelente alternativa para preservar la historia, 2007.

CERDA, F. Características de NetBeans. Disponible en: http://www.bloog.cl/talks/netbeans65es_cl.pdf.

CHAPIN, K. The Evolution of Document Management, Abril 2006.

CIBERAULA.COM. Una Introducción a APACHE. Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/.

CODEIGNITER, U. G. V. Welcome to CodeIgniter. Disponible en: http://codeigniter.com/user_guide/index.html.

EGUILUZ, J. El framework Symfony, una introducción práctica (I parte). Disponible en: <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.

GROSPE, J. Ibermática 2005. Gestión Documental., 2002.

LAPUENTE, M. J. L. Hipertexto, el nuevo concepto de documento en la cultura de la imagen, 2006.

GUERVOS, J. J. M. Introducción al lenguaje XML. Disponible en: <http://geneura.ugr.es/~jmerelo/xml/>.

GUTIÉRREZ, J. J. ¿Qué es un framework web? Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.

LEMONTECH. Alfresco Enterprise Content Management. Disponible en: <http://www.lemontech.cl/cms/index.php?id=50>.

MACLLWAINE, J.; J. M. COMMENT, et al. Directrices para Proyectos de Digitalización de colecciones y fondos de dominio público, en particular para aquellos custodiados en bibliotecas y archivos, 2002. 121.

MASTERNEWMEDIA.ORG. Beneficios De Las Aplicaciones Basadas En Web. Disponible en: http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.

MENPET. ¿Qué es el Ministerio del Poder Popular para la Energía y Petróleo? Disponible en: <http://www.menpet.gob.ve/portalmenpet/secciones.php?option=view&idS=2>.

MONOGRAFÍAS.COM, L. El Paradigma Lógico.

Montes, D, LR. Análisis y Modelado del sistema para el proceso de digitalización de documentos de Registros y Notarias del MPPRIJ de la República Bolivariana de Venezuela, junio 2007.

MÜLLER, P. Programación Orientada a Objetos. Disponible en: <http://www.gnacademy.org/text/cc/Tutorial/Spanish/node3.html#SECTION00360000000000000000>.

NETBEANS.ORG. NetBeans - Development Simplified. Disponible en: <http://www.netbeans.org/features/>.

PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico. Ciudad de la Habana: 2005. Vol.1.

SCHWINDT, A. Construcción de Sistemas Multiplataforma basados en Servicios. Disponible en: <http://msdn.microsoft.com/es-es/library/bb972254.aspx>.

TRAMULLAS, J. El producto de información digital. , 2003.

---. Herramientas de software libre para la gestión de contenidos. Disponible en: <http://www.hipertext.net/web/pag258.htm>.

UNIVERSIDAD TECNOLÓGICA DE TUCUMAN, D. D. S. PARADIGMAS DE PROGRAMACION.
Disponible en: <http://www.frt.utn.edu.ar/sistemas/paradigmas/page22.html>.

VEGAS, J. Desarrollo de Aplicaciones Web. Disponible en:
<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node17.html>.

WELICKI, L. Patrones y Antipatrones: una Introducción - Parte II. Disponible en:
<http://msdn.microsoft.com/es-es/library/bb972251.aspx>.

WIKIPEDIA, L. E. L. Gestión Documental. Disponible en:
http://es.wikipedia.org/wiki/Gesti%C3%B3n_documental

---. Desarrollo de la Gestión Documental. Disponible en:
http://es.wikipedia.org/wiki/Gesti%C3%B3n_documental.

---. Alfresco, Reconocimientos. Disponible en: <http://es.wikipedia.org/wiki/Alfresco>.

---. Adobe Dreamweaver. Disponible en: http://es.wikipedia.org/wiki/Adobe_Dreamweaver.

---. Aplicación web. Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web.

---. Cliente-servidor. Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>.

---. Servidor HTTP Apache. Disponible en: http://es.wikipedia.org/wiki/Apache_http_server.

---. Extensible Markup Language. Disponible en:
http://es.wikipedia.org/wiki/XML#Objetivos_y_usos_de_XML.

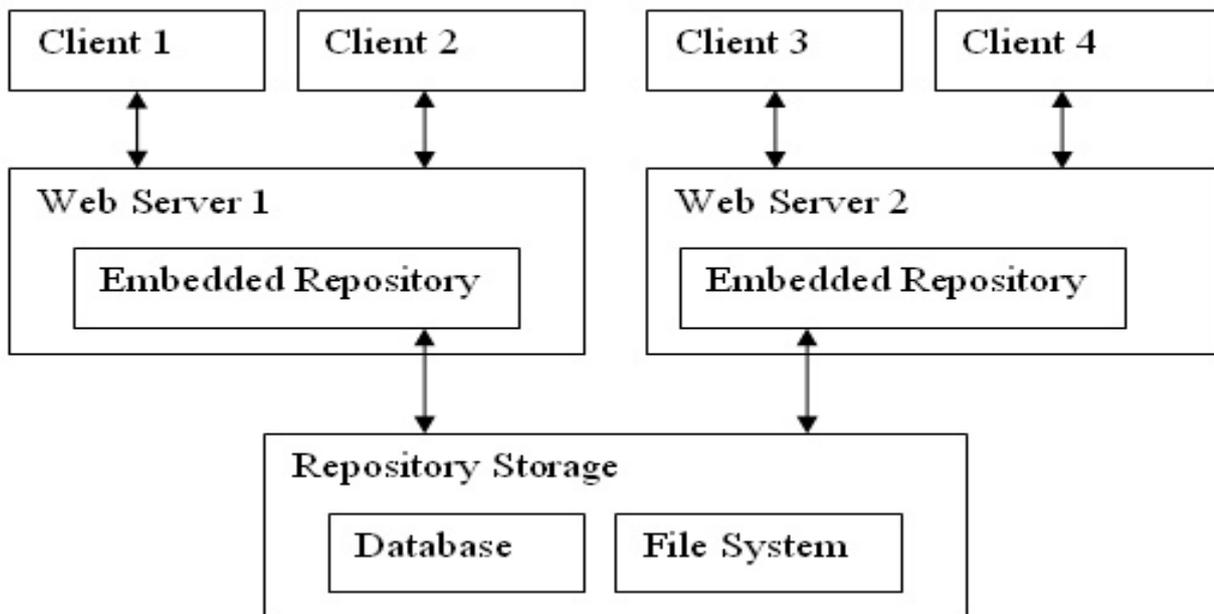
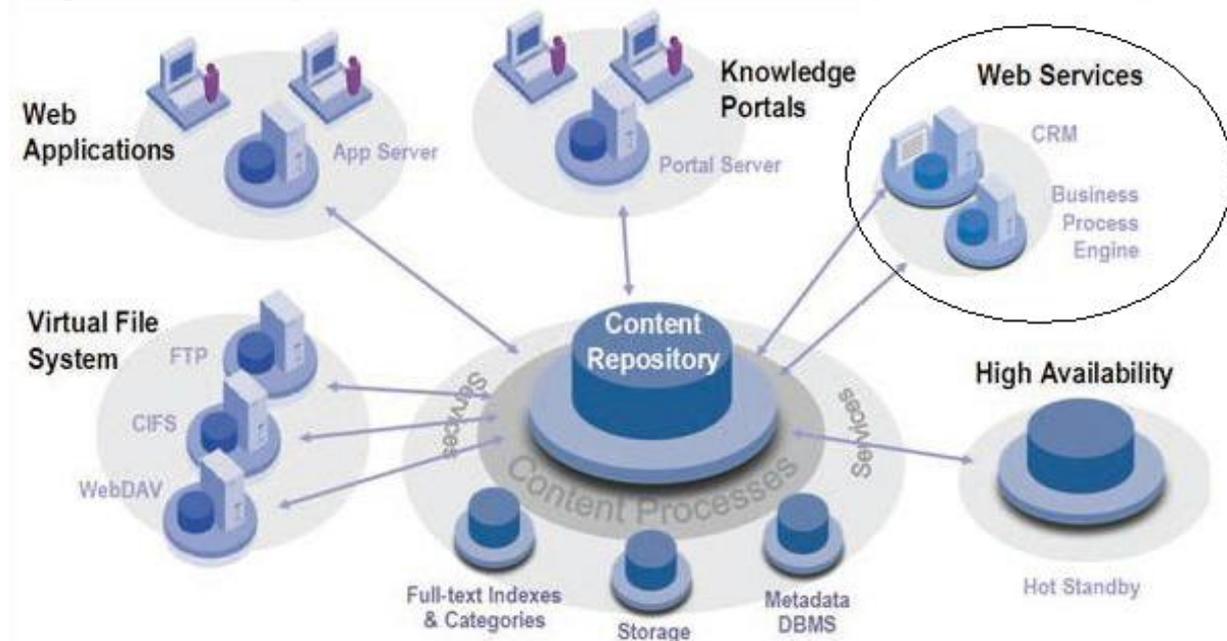
ZENDFRAMEWORKDES.COM. Capítulo 1. Introducción a Zend Framework, Descripción general.
Disponible en: <http://manual.zfdes.com/es/introduction.html>.

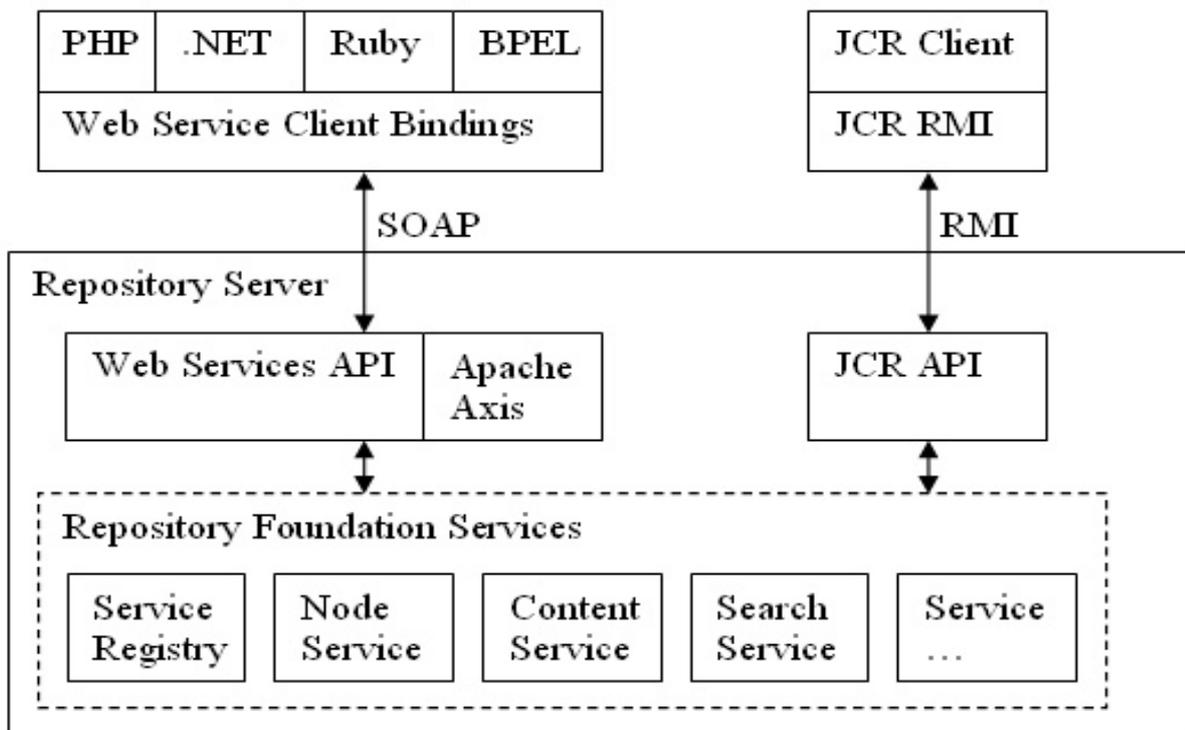
ZEND.COM. Zend Studio Features. Disponible en:
<http://www.zend.com/en/products/studio/features>.

Anexos

Anexo 1

Diagramas de integración de los Web Services con el repositorio de Alfresco





Anexo 2

Interfaz de las búsquedas avanzadas

Usted esta logueado como Administrator Salir ▲

SISTEMA DE CONTROL DE DOCUMENTOS

Panel de Control :: Correspondencias :: Correspondencias Entrantes

Buscar Correspondencias

Número Externo:

Fecha de Emisión: Desde: Hasta:

Fecha de Recepción: Desde: Hasta:

Tipo de Información: -- Seleccione --

Tipo de Documento: Oficio

Abreviatura del Organismo: -- Seleccione --

Procedencia: Internacional

Acciones:

Sistema de Control de Documentos. 4 de junio del 2009, NetBeans (6.5)

Interfaz del resultado de la búsqueda avanzada

Usted esta logueado como Administrator Salir ▲

SISTEMA DE CONTROL DE DOCUMENTOS

Panel de Control :: Correspondencias :: Correspondencias Entrante

Listado de Correspondencias Entrantes

Número Interno	Fecha de Emisión	Fecha de Recepción	Organismo	Tipo de Documento	Procedencia	Original Digital	Copia Digital	Estado de la Correspondencia
2385	08/06/2009 5:24 PM	15/06/2009 5:24 PM	PDVSA	Oficio	Internacional	Sí	Sí	Para Revisar
2386	15/06/2009 5:25 PM	11/06/2009 5:25 PM	MENPET	Oficio	Internacional	Sí	Sí	Para Revisar
2387	22/06/2009 5:26 PM	11/06/2009 5:26 PM	MENPET	Oficio	Internacional	Sí	Sí	Para Revisar

Sistema de Control de Documentos. 4 de junio del 2009, NetBeans (6.5)

Anexo 3

Interfaz de la nueva correspondencia entrante que va a ser adicionada en el estado de REDACCIÓN

DE CONTROL DE DOCUMENTOS

Panel de Control :: Correspondencias :: Correspondencia Entrante

Nueva Correspondencia

Número Externo:	<input type="text" value="1232"/>
Fecha de Emisión:	<input type="text"/>
Fecha de Recepción:	<input type="text"/>
Datos del Remitente:	<input type="button" value="Gestionar"/>
Datos del Destinatario:	<input type="button" value="Gestionar"/>
Asunto:	<input type="text" value="Para Redacción"/>
Contenido:	<div style="border: 1px solid gray; height: 100px;"></div>
Tipo de Información:	<input type="button" value="Nota Informativa"/>
Tipo de Documento:	<input type="button" value="Oficio"/>
Abreviatura del Organismo:	<input type="button" value="-- Seleccione --"/>
Procedencia:	<input type="button" value="-- Seleccione --"/>
Ubicación de Origen:	<input type="text"/>
Ubicación Destino:	<input type="button" value="-- Seleccione --"/>
Documentos Anexos:	<input type="button" value="Gestionar"/>
Preparado Por:	<input type="button" value="Gestionar"/>
Acciones:	<input type="button" value="Registrar"/> <input type="button" value="En Redacción"/> <input type="button" value="Valores Por Defecto"/> <input type="button" value="Cancelar"/>

Sistema de Control de Documentos. 4 de junio del 2009, NetBeans (6.5)

Interfaz mostrando el estado al que ha pasado la nueva correspondencia entrante

Panel de Control :: Correspondencias :: Correspondencia Entrante

Nueva Correspondencia **La correspondencia ha pasado al estado En Redacción**

Número Externo:	<input type="text"/>
Fecha de Emisión:	<input type="text"/> 
Fecha de Recepción:	<input type="text"/> 
Datos del Remitente:	<input type="button" value="Gestionar"/>
Datos del Destinatario:	<input type="button" value="Gestionar"/>
Asunto:	<input type="text"/>
Contenido:	<div style="border: 1px solid gray; height: 100px;"></div>
Tipo de Información:	-- Seleccione -- 
Tipo de Documento:	-- Seleccione -- 
Abreviatura del Organismo:	-- Seleccione -- 
Procedencia:	-- Seleccione -- 
Ubicación de Origen:	<input type="text"/>
Ubicación Destino:	-- Seleccione -- 
Documentos Anexos:	<input type="button" value="Gestionar"/>
Preparado Por:	<input type="button" value="Gestionar"/>
Acciones:	<input type="button" value="Registrar"/> <input type="button" value="En Redacción"/> <input type="button" value="Valores Por Defecto"/> <input type="button" value="Cancelar"/>

Sistema de Control de Documentos. 4 de junio del 2009, NetBeans (6.5)

Interfaz de la nueva correspondencia entrante que va a ser adicionada en el estado de REVISIÓN

Panel de Control :: Correspondencias :: Correspondencia Entrante

Nueva Correspondencia

Número Externo:	<input type="text" value="324"/>
Fecha de Emisión:	<input type="text" value="16/06/2009"/> 
Fecha de Recepción:	<input type="text" value="17/06/2009"/> 
Datos del Remitente:	<input type="button" value="Gestionar"/>
Datos del Destinatario:	<input type="button" value="Gestionar"/>

Asunto:

Contenido:
BLA BLA BLA
BLA BLA BLA
BLA BLA BLA

Tipo de Información:	<input type="button" value="Creación o Modi"/>
Tipo de Documento:	<input type="button" value="Punto de Cuenta"/>
Abreviatura del Organismo:	<input type="button" value="PDVSA"/>
Procedencia:	<input type="button" value="Nacional"/>
Ubicación de Origen:	<input type="text" value="UCI"/>
Ubicación Destino:	<input type="button" value="Dir PDVSA"/>
Documentos Anexos:	<input type="button" value="Gestionar"/>
Preparado Por:	<input type="button" value="Gestionar"/>
Acciones:	<input type="button" value="Registrar"/> <input type="button" value="En Redacción"/> <input type="button" value="Valores Por Defecto"/> <input type="button" value="Cancelar"/>

Sistema de Control de Documentos. 4 de junio del 2009, NetBeans (6.5)

Interfaz mostrando que la correspondencia entrante ha pasado al estado de REVISIÓN

Panel de Control :: Correspondencias :: Correspondencia Entrante

Nueva Correspondencia **La Correspondencia se ha Adicionado Correctamente**

Número Externo:	<input type="text"/>
Fecha de Emisión:	<input type="text"/> 
Fecha de Recepción:	<input type="text"/> 
Datos del Remitente:	<input type="button" value="Gestionar"/>
Datos del Destinatario:	<input type="button" value="Gestionar"/>
Asunto:	<input type="text"/>
Contenido:	<div style="border: 1px solid gray; height: 100px;"></div>
Tipo de Información:	-- Seleccione -- 
Tipo de Documento:	-- Seleccione -- 
Abreviatura del Organismo:	-- Seleccione -- 
Procedencia:	Seleccione 
Ubicación de Origen:	<input type="text"/>
Ubicación Destino:	-- Seleccione -- 
Documentos Anexos:	<input type="button" value="Gestionar"/>
Preparado Por:	<input type="button" value="Gestionar"/>
Acciones:	<input type="button" value="Registrar"/> <input type="button" value="En Redacción"/> <input type="button" value="Valores Por Defecto"/> <input type="button" value="Cancelar"/>

Sistema de Control de Documentos. 4 de junio del 2009, NetBeans (6.5)

Glosario de Términos

Alfresco

Sistema de Gestión Documental.

Clase

Es una abstracción sobre un conjunto de objetos que tienen en común estructura y comportamiento.

Correspondencia

Documentación que se envía o recibe en formato duro, no se considerarán los correos electrónicos como correspondencia.

Correspondencia entrante

Es la que se recibe en el MENPET.

Correspondencia saliente

Es la que se envía desde el MENPET.

Memorándum

Correspondencia para oficinas internas (dentro del MENPET) que estén a un mismo nivel de dirección.

Módulo

Encapsula un conjunto de funciones que debe realizar el sistema, las cuales son agrupadas por tener características muy similares y se definen en la etapa de diseño.

Oficio

Correspondencia para organismos externos. Se envía desde el MENPET para otros ministerios o entes adscritos. No circula dentro del MENPET.

Patrón

Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Punto de cuenta

Correspondencia para los despachos (con el Ministro del MENPET y el Presidente de Venezuela solamente) con el objeto de la aprobación de recursos económicos.

Punto de información

Correspondencia para los despachos (con el Ministro del MENPET y el Presidente de Venezuela solamente) con el objeto de revisión de información.

Sistema de Control de Documentos (SCD)

Herramienta que permitirá llevar el control y seguimiento de la correspondencia entrante y saliente de todas las oficinas, así como también de las notas de remisión, minutas y documentos digitales.

Sistema de Operaciones Secretariales (SOC)

Nombre del sistema de control de documentos actualmente en uso por el MENPET.

PHP

Lenguaje de programación interpretado usado para la creación de páginas Web dinámicas.

HTML

HyperText Markup Language. Lenguaje estándar para la publicación de documentos Web.

Hoja de estilo

Conjunto de instrucciones que permiten dar determinados aspectos de formato y presentación de los contenidos: tipo, fuente y tamaño de letras, justificación del texto, colores, fondos, entre otros.

Motor de búsqueda

Servicio de la Web que permite hacer consultas a las bases de datos mediante palabras clave y muestra los contenidos donde estas aparecen.

Pop-up

Tipo de ventana flotante que aparece sobre la que se está navegando. Esto permite visualizar mensajes de alerta, separar los contenidos o mostrar una serie de datos para impresión.

Sistema Interno Central (SIC)

Sistema desarrollado internamente en la Oficina de Tecnología e Informática del MENPET, el cual permite llevar el control y seguimiento de los procesos de reclutamiento de personal, así como también, mantener un registro de la trayectoria de los trabajadores y de la estructura de cargos propuestos y aprobados. Adicionalmente, ofrece formas de integración con los sistemas internos y externos.

Aplicación Web

Aplicación que es accedida a través de la Web.

Usuario Administrador

Usuario con capacidad de administrar toda la información existente en el sistema.

Usuario Aprobador

Usuario con capacidad de aprobar las correspondencias.

Usuario Auditor

Usuario con capacidad de realizar auditorías a las correspondencias.

Usuario Invitado

Usuario con privilegios básicos en el sistema.

Usuario Redactor

Usuario con capacidad de registrar correspondencias.

Usuario Revisor

Usuario con capacidad de revisar las correspondencias registradas por el Redactor.