**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 6**


**TÍTULO:** Desarrollo de una herramienta informática para buscar y recuperar documentos en el repositorio documental del Departamento Señales Digitales.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS**

**AUTOR:** Yudith Ginarte González.

**TUTOR:** Ing. Rafael L. Cardero Álvarez.

**Ciudad de La Habana, junio, 20 ,2011**

**Año del 53 Aniversario de la Revolución.**

## DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de este trabajo y autorizamos a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.
Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____ _____

**Autor:** Yudith Ginarte González. **Tutor:** Ing. Rafael L. Cardero Álvarez.

## RESUMEN

En el departamento Señales Digitales del centro GEySED se requiere facilitar la búsqueda y recuperación de los documentos existentes en el repositorio documental. En este trabajo de diploma se desarrolla un sistema que resuelve la problemática anterior. Dicho sistema constituye una interfaz web fácil de usar e intuitiva a la poderosa herramienta Swish-e. El sistema además permite controlar el acceso al repositorio documental del departamento, gestionar noticias y gestionar usuarios.

Para construir el sistema se utilizó el lenguaje de programación PHP, el gestor de base de datos PostgreSQL y el framework Symfony, entre otras tecnologías. La metodología que se utilizó fue el Proceso de Unificado de Desarrollo. Se utilizó además el Lenguaje Unificado de Modelado.

La solución lograda es multiplataforma y está basada en software libre, por lo cual se contribuye a la soberanía tecnológica del país. La relevancia técnica de la investigación es que se logra una solución que integra las bondades de la herramienta Swish-e, con otros elementos significativos en el contexto del problema, y es muy fácil de utilizar.

*Palabras claves*: búsqueda, indexación, sistema.

# ÍNDICE

INTRODUCCIÓN ..... 1

CAPÍTULO 1: Fundamentación Teórica de las herramientas para buscar y recuperar documentos en los repositorios documentales ..... 6

1. Introducción ..... 6

1.2 Conceptos asociados al dominio del problema ..... 6

1.3 Objeto de Estudio. ..... 8

1.3.1 Descripción General. ..... 8

1.3.2 Descripción actual del dominio del problema ..... 9

1.4 Análisis de otras soluciones existentes ..... 9

1.4.1 Excalibur RetrievalWare. ..... 10

1.4.2 Copernic Desktop Search ..... 12

1.4.3 Google Desktop Search. ..... 12

1.4.4 Swish-e ..... 13

1.4.5 ¿Por qué Swish-e? ..... 15

1.5 Metodología de Desarrollo de Software. ..... 15

1.5.1 Rational Unified Process (RUP). ..... 15

1.5.2 Extreme Programming (XP) ..... 17

1.5.3 Microsoft Solution Framework (MSF). ..... 18

1.5.4 ¿Por qué RUP? ..... 20

1.6 Arquitectura. ..... 20

1.6.1 Arquitectura en 3 Capas. ..... 20

1.6.2 Modelo Cliente/Servidor. ..... 21

1.6.3 Arquitectura Modelo Vista Controlador (MVC). ..... 22

1.6.4 ¿Por qué arquitectura Modelo Vista Controlador? ..... 23

1.7 Lenguajes usados en el desarrollo del sistema. ..... 23

1.7.1 Lenguaje de Modelado: UML. ..... 23

1.7.2 Lenguajes de Programación y Tecnologías utilizadas ..... 24

1.7.2.1 PHP. ..... 24

1.7.2.2 JavaScript. ..... 27

1.7.2.3 HTML ..... 27

1.7.2.4 ExtJS. ..... 28

1.7.2.5 Ajax. ..... 29

1.8 Herramientas a utilizar para desarrollar la presente investigación. ..... 30

1.8.1 Herramientas CASE ..... 30

1.8.1.1 Visual Paradigm. ..... 31

1.8.1.2 Rational Rose. 31
1.8.1.3 ¿Por qué Visual Paradigm? 32
1.8.2 IDE de desarrollo. 32
1.8.2.1 Zend Studio. 32
1.8.2.2 NetBeans. 33
1.8.2.3 Aptana (Eclipse). 34
1.8.2.4 ¿Por qué NetBeans? 34
1.8.3 Framework. 35
1.8.3.1 Symfony. 35
1.8.3.2 Prado. 36
1.8.3.3 Zend Framework. 36
1.8.3.4 ¿Por qué Symfony? 37
1.8.4 Sistemas Gestores de Bases de Datos 37
1.8.4.1 MySQL. 37
1.8.4.2 PostgreSQL 39
1.8.4.3 Oracle 40
1.8.4.4 ¿Por qué PostgreSQL? 41
1.8.5 Servidor web. 41
1.8.5.1 Servidor Web Apache. 41
1.8.5.2 AOLserver (Servidor web de América Latina Online). 42
1.8.5.3 ¿Por qué Apache? 43
1.9 Conclusiones 43
CAPÍTULO 2: Presentación de la solución propuesta. 44
2. Introducción. 44
2.1 Actor del negocio. 44
2.2 Diagrama de caso de uso del negocio. 44
2.3 Descripción textual de los casos de uso del negocio (CUN). 44
2.3.1 Descripción del CUN Realizar Búsqueda. 44
2.5 Requisitos Funcionales. 45
2.6 Requerimientos no Funcionales. 47
2.7 Descripción de sistema propuesto. 49
2.7.1 Definición de actores. 49
2.7.1.1 Descripción de los actores del sistema. 49
2.7.2 Definición de casos de uso del sistema. 49
2.7.2.1 Diagrama de caso de uso del sistema. 50
2.7.2.2 Descripción del CU Gestionar Usuario. 50
2.8 Conclusiones 52

CAPÍTULO 3: Análisis y diseño de la solución propuesta. ..... 54
3. Introducción. ..... 54
3.1 Análisis. ..... 54
3.1.1 Modelo del Análisis. ..... 54
3.1.2 Diagramas de Clases del Análisis. ..... 55
3.1.2.1 Diagramas de Clases del Análisis CU Gestionar Usuario. ..... 55
3.1.3 Diagramas de Interacción. ..... 55
3.1.3.1 Diagramas de Colaboración del Análisis. ..... 56
3.1.3.1.1 Diagrama de Colaboración del Análisis. CU Realizar Búsqueda. ..... 56
3.1.3.2 Diagrama de Secuencia. ..... 56
3.2.1 Patrones. ..... 57
3.2.1.1 Patrones de Diseño en Symfony. ..... 57
3.2.2 Diagrama de Clases del Diseño. ..... 62
3.2.3 Diseño de la Base de Datos. ..... 63
3.3 Conclusiones ..... 64
CAPÍTULO 4: Implementación y proceso de pruebas al sistema. ..... 65
4. Introducción. ..... 65
4.1 Modelo de implementación. ..... 65
4.1.1 Diagrama de despliegue. ..... 65
4.1.2 Diagrama de componentes ..... 67
4.2 Prueba de Software. ..... 67
4.2.1 Prueba de Caja Negra. ..... 68
4.2.1.1 Partición equivalente. ..... 69
4.2.2 Diseños de Casos de Prueba. ..... 69
4.2.2.1 Diseño de Caso de Prueba del CU Autenticar Usuario. ..... 69
4.3 No conformidades detectadas. ..... 70
4.4 Conclusiones ..... 70
CONCLUSIONES GENERALES ..... 72
RECOMENDACIONES ..... 73
REFERENCIAS BIBLIOGRÁFICAS. ..... 74

## ÍNDICE DE TABLAS

Tabla 1 Descripción de actor del negocio. ..... 44
Tabla 2 Descripción textual del CUN Realizar Búsqueda. ..... 45
Tabla 3 Actores del sistema. ..... 49
Tabla 4 Descripción del CU Gestionar Usuario. ..... 52
Tabla 5 Secciones a probar en el CC Autenticar Usuario. ..... 70
Tabla 6 Descripción de variable del CU Autenticar Usuario. ..... 70
Tabla 7 Matriz de Datos. SC 1 Autenticar Usuario. ..... 70

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Diagrama de casos de uso del negocio. ..... 44
Ilustración 2 Diagrama de Caso de Uso del Sistema. ..... 50
Ilustración 3 Diagramas de Clases del Análisis CU Gestionar Usuario. ..... 55
Ilustración 4 Diagrama de Colaboración del Análisis. CU Realizar Búsqueda. ..... 56
Ilustración 5 Diagrama Entidad Relación. ..... 63
Ilustración 6 Diagrama de Despliegue. ..... 65
Ilustración 7 Diagrama de Componentes. ..... 67

# INTRODUCCIÓN

En la actualidad el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) unido al surgimiento y evolución de las redes de computadores ha propiciado una amplia difusión de información a nivel mundial. Desde los inicios de la actividad humana la información ha sido un baluarte en todo el proceso de comunicación. Ésta ya era valiosa en el pasado, significaba encontrarse en una situación ventajosa respecto a quienes no la tenían. La información ha sido siempre un recurso muy valorado. Desde las primeras civilizaciones se ha archivado, manipulado y creado grandes volúmenes de la misma en bibliotecas, monasterios o por determinadas personas. Pero en el presente su valor se incrementa, por lo que antes, no existía la posibilidad de interrelacionar esa información y de procesarla con rapidez. La explosión de las tecnologías de redes, la aparición de novedosos servicios, el acercamiento entre culturas y las necesidades de comunicación que estas generan, ha provocado que en unos pocos años la información se haya convertido en un eslabón fundamental para el desarrollo de cualquier sociedad. La revolución tecnológica ha redimensionado las relaciones entre los hombres. La información aporta cada vez más conocimientos, considerándose de gran importancia actualmente.

Como resultado del surgimiento de internet, la comunicación se ha extendido por todo el mundo, y el flujo de información es cada vez mayor. En el presente las tecnologías de la información han llegado a ser la figura representativa de nuestra cultura. Detrás de todo este desarrollo tecnológico se encuentra la información como objeto de dicha revolución. La información ha pasado a tener una gran importancia, el flujo y la cantidad de información que se intercambia en formato digital ha crecido de forma exponencial, demandando canales de comunicación rápidos y efectivos. El acceso a esa gran masa de datos requiere soportes potentes y eficaces. Los avances tecnológicos y de las telecomunicaciones han puesto a disposición de los usuarios este valioso tesoro, que es la información.

La existencia de la información digital ha superado a la información analógica. Las TICs facilitan el manejo de la información, su gestión, su análisis, su almacenamiento o su recuperación. Además de que con estas se permite la reusabilidad, interactividad, densidad y virtualidad, facilitando nuevos usos educativos y mejorando las formas de transmisión.

En el país existe la Asociación Cubana de Reconocimiento de Patrones (ACRP[1]) a la que están afiliados especialistas dedicados a diferentes ramas del Reconocimiento de Patrones y algunos a la Minería de Datos. Dicha asociación presenta delegaciones en todas las provincias del país. En la Universidad de las Ciencias Informáticas (UCI) existen afiliados a ella. Entre los fines de la asociación se encuentra el intercambio de información y experiencias, entre profesionales e investigadores. Además de promover, coordinar y desarrollar actividades en temas referentes al Reconocimiento de Patrones, estableciendo relaciones científicas con otras organizaciones nacionales e internacionales, para difundir información sobre el Reconocimiento de Patrones y la Minería de Datos. Los integrantes de la asociación tienen en sus manos la posibilidad de acceder a gran cantidad de información que fluye por toda la red a nivel mundial. La ACRP donó a la UCI, específicamente al Departamento Señales Digitales, perteneciente al Centro de Desarrollo Geoinformática y Señales Digitales (GEySED) de la facultad 6, 200 GB de información de gran importancia para este departamento, la cual se encuentra en formato PDF[2].

Contar con información actualizada es de vital importancia para este departamento, ya que puede ser usada en la investigación. Pero la cantidad de información con que cuenta es tan grande que trae consigo algunos problemas por parte de los miembros del departamento para poner ésta a disposición de los usuarios.

Por lo que existe:

- ✓ Necesidad de usarla: Es información de último momento, y muy importante para ser usada por los estudiantes y profesores del departamento.

- ✓ Necesidad de organizarla para su uso: Se hace necesario que de alguna forma la información pueda ser organizada para que a la hora de acceder a ella no sea difícil o quizás imposible encontrar lo que se busca.

Por ello la dirección del Departamento Señales Digitales se han dado a la tarea de encontrar una herramienta para la indexación, búsqueda y recuperación de documentos, para que estos sean usados

1 Asociación Cubana de Reconocimiento de Patrones: http://acrp.cenatav.co.cu/

2 Formato de documento portátil, es el estándar mundial que le permite capturar y revisar información sofisticada desde cualquier aplicación y en cualquier sistema informático. Gracias al lanzamiento del software Adobe Acrobat X, los PDF son más seguros y dinámicos.

por los estudiantes y profesores en su beneficio. Así contarían con un repositorio documental en el cual podrían buscar y recuperar información, sin la necesidad de buscar en internet. Indudablemente este repositorio se convertiría en una herramienta básica de apoyo en la investigación. En el cual se podrían encontrar información cuya finalidad es la de difundir el conocimiento. Lo que traería consigo mejoras en la productividad y la calidad de los resultados de los miembros del departamento. Esto aporta muchos beneficios, con la herramienta, el consumo de ancho de banda de internet se reduciría, no sería necesaria la búsqueda de información de este tipo en la red de redes, el usuario no consumiría la cuota mensual asignada. Además, en la red no está disponible toda la información con la que cuenta el departamento o hay que comprarla. Téngase en cuenta que parte de la documentación con que cuenta el departamento procede de bases de datos internacionales, por lo que tiene una validez incontable.

Otro factor es que se requiere informar a los miembros del departamento cuando se adicionan nuevas fuentes bibliográficas al repositorio documental. Además, se requiere que a dicho repositorio accedan solamente los usuarios autorizados.

Teniendo en cuenta los argumentos expuestos anteriormente surge como **problema a resolver** en la presente investigación**:** ¿Cómo facilitar la búsqueda y recuperación de la información científico-técnica de interés, existente en el repositorio documental del departamento Señales Digitales?

Como **Objetivo General** se propone**:** Desarrollar una herramienta informática que facilite la búsqueda y recuperación de la información científico-técnica de interés.

Ante el problema expuesto anteriormente se identifica como **Objeto de estudio:** Sistemas informáticos de búsqueda y recuperación de información.

Enmarcado en el **Campo de acción**: Sistemas informáticos de búsqueda y recuperación de información en repositorios documentales.

Siendo la **Idea a defender:** El desarrollo de una herramienta informática que permita la búsqueda y recuperación de la información, facilitaría la búsqueda y recuperación de la información científico-técnica de interés existente en el repositorio documental del Departamento Señales Digitales.

**Objetivos Específicos:**

- ✓ Caracterizar los distintos sistemas de indexación, búsqueda y recuperación de información existente.
- ✓ Desarrollar una herramienta que facilite la búsqueda y recuperación de la información científico-técnica de interés, existente en el repositorio documental del Departamento Señales Digitales.

Para lograr el cumplimiento de estos objetivos se han trazado las siguientes **tareas:**

1. Elaborar un informe sobre el estado del arte de los sistemas para recuperar y buscar documentos.
2. Determinar las tecnologías y herramientas a utilizar para el desarrollo de la herramienta informática.
3. Identificar las características del negocio y del futuro sistema.
   - 3.1 Realizar el modelado del negocio.
   - 3.2 Realizar el levantamiento de requisitos funcionales y no funcionales.
4. Realizar el análisis y diseño de la herramienta informática para buscar y recuperar documentos en el repositorio documental del departamento Señales Digitales.
5. Implementar la herramienta informática para buscar y recuperar documentos en el repositorio documental del departamento Señales Digitales.
6. Diseñar y aplicar las pruebas al sistema.

**Métodos Científicos de la Investigación.**

Los métodos científicos en los cuales se basa la investigación son los métodos teóricos. Estos permiten organizar el trabajo, lo cual posibilita el desarrollo de un amplio conocimiento referente al estado del arte tratado, su evolución en etapas determinadas, así como adquirir de bibliografías elementos importantes para el desarrollo del mismo.

**Métodos Teóricos.**

**Analítico-Sintético:** Permitió analizar teorías y documentos sobre las herramientas de búsqueda y recuperación de documentos existentes y utilizados en la actualidad, buscar sus características fundamentales, ventajas y desventajas, para a partir de ahí seleccionar la herramienta adecuada para su posterior uso.

**Análisis Histórico-Lógico:** Permitió el estudio de las diferentes herramientas de búsqueda y recuperación de documentos existentes actualmente, cuándo fueron desarrollados, así como su evolución en el tiempo, a partir de su investigación en diferentes momentos históricos. Se realizaron comparaciones entre las herramientas encontradas lo que facilitó la obtención de elementos de interés para decidirse por una u otra para utilizar la óptima.

Lo que resta de este documento está organizado cómo sigue.

En el Capítulo 1 “Fundamentación Teórica de las herramientas para buscar y recuperar documentos en los repositorios documentales”, se hace referencia al estado del arte del objeto de estudio de la presente investigación. Se enuncian conceptos que posibilitan un mejor entendimiento de lo planteado en la situación problemática y el marco del problema en sentido general. Además, se realiza un análisis de las tecnologías, metodologías y herramientas actuales, quedando seleccionadas las que se van a utilizar para el desarrollo de la herramienta.

En el Capítulo 2 “Presentación de la solución propuesta”, se presenta la solución propuesta a partir de la realización de un estudio inicial del entorno que encierra el problema a resolver con la misma.Se brinda información referente a los actores y casos de uso del negocio y del sistema, de manera que se pueda entender mejor el producto que se desea obtener. Se identifican los requerimientos funcionales y no funcionales, y se describe el modelo de casos de usos del sistema propuesto.

En el Capítulo 3 “Análisis y diseño de la solución propuesta”, se efectúa el análisis y diseño de la aplicación, tratando aspectos fundamentales como los diagramas de interacción, los diagramas de clases del diseño entre otros.

En el Capítulo 4 “Implementación y proceso de pruebas al sistema”, se dan a conocer las generalidades del flujo de trabajo de implementación y se plasmarán los resultados de las pruebas realizadas al sistema.

# CAPÍTULO 1: Fundamentación Teórica de las herramientas para buscar y recuperar documentos en los repositorios documentales.

## 1. Introducción.

En este capítulo se hará una descripción de los principales sistemas de búsqueda y recuperación de documentos, así como la argumentación sobre los más reconocidos productos que se están utilizando en el mundo. Se llevará a cabo una conceptualización de los términos y procesos más importantes que rodean al problema científico. Además de realizar un análisis de las distintas herramientas y tecnologías existentes, para seleccionar las que serán usadas en el desarrollo del sistema.

### 1.2 Conceptos asociados al dominio del problema.

Dentro de los conceptos clave presentes en esta investigación se encuentra uno que es de vital importancia conocer su significado el cual es el concepto de:

**Documento:** Es una carta, diploma o escrito que ilustra acerca de un hecho, situación o circunstancia. También se trata del escrito que presenta datos susceptibles de ser utilizados para comprobar algo. (definición, 2008)

También se le llama documento a toda aquella información contenida y registrada sobre cualquier soporte material y que es producido, recibido y conservado por las instituciones, organizaciones o personas, durante el desarrollo de sus actividades. Es, por tanto, un testimonio de la actividad humana.(Mailxmail, 2010)

Se le llama documento a todo aquello que contiene información, independientemente de su soporte físico y que abarca todo lo que puede transmitir el conocimiento humano. Documento es todo aquel objeto utilizado por los seres humanos para fijar una información y que actúa como la memoria de actos, acciones, pensamientos, creaciones e imágenes. (Buenastareas, 2011)

Después de haber analizado diferentes conceptos se puede decir que un documento es toda información que se encuentra plasmada en un soporte, yendo desde las antiguas tablillas cuneiformes[3] de Mesopotamia, hasta los correos electrónicos, archivos de audio y videos.

Los documentos contienen grandes cantidades de información la cual es usada para beneficio propio. La **Información** es un conjunto organizado de datos, que constituye un mensaje sobre un cierto fenómeno. La información permite resolver problemas y tomar decisiones, ya que su uso racional es la base del conocimiento. (2008)

La información es un conjunto de datos con significado que estructura el pensamiento de los seres vivos, especialmente, del ser humano. (Definicionabc, 2009)

La información puede entenderse como la significación que adquieren los datos como resultado de un proceso consciente e intencional de adecuación de tres elementos: los datos del entorno, los propósitos y el contexto de aplicación, así como la estructura de conocimiento del sujeto. (Wikilearning, 2007)

En ocasiones se hace necesario recuperar ésta, por lo que el proceso de **recuperación de información** consiste esencialmente en extraer de una colección de documentos aquellos que se ajustan a las especificaciones de una petición determinada. Se trata pues de una comparación sistemática entre los documentos o sus representaciones y la petición o demanda de información.

Un **sistema de información** es el sistema de personas, registros de datos y actividades que procesa los datos y la información en cierta organización, incluyendo manuales de procesos o procesos automatizados. El sistema de información basada en computadora es el campo de estudio de las tecnologías de información. (Mitecnológico)

Se le llama **sistema de recuperación de información** al proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsqueda específicas. Dicha información ha debido de ser estructura antes de su almacenamiento. (Molina, 2004)

[3] Escritura cuneiforme es la que usaron algunos pueblos antiguos de Asia, cuyos caracteres tenían forma de cuña o clavo.(Wordreference, 2005).

Un Sistema de Recuperación de Información en lenguaje natural podría definirse como una interfaz de usuario, más o menos avanzada, cuya función principal es la de recoger la solicitud de información de un usuario, expresada en lenguaje natural (escrito o hablado), procesarla y generar una respuesta adecuada. (Elies)

Se concluye que un sistema de recuperación de información se le llama al proceso donde se accede a una información que haya sido previamente almacenada, mediante una de las herramientas informáticas que permiten establecer criterios de búsqueda específicos, utilizando algoritmos de búsqueda, con el fin de recuperar los documentos más significativos. Ejemplo: las bases de datos y los meta-buscadores.

**Indexación:** La indexación es el proceso de elaboración de estructuras lógicas que permiten a través de términos contenidos en los documentos el acceso de manera más fácil, rápida y sencilla a dichos documentos. Normalmente, los sistemas de recuperación de información utilizan índices de estructura de datos que permiten la búsqueda de información utilizando técnicas de comparación de los enunciados propuestos con términos almacenados en el sistema.

**Procesos de Indexación:** El término proceso de indexación hace referencia al proceso de análisis de documentos para la obtención de una representación concreta de los mismos.

**Indización:** Según la norma ISO 5963 de 1985 la indización es el proceso de describir o representar el contenido temático de un recurso de información. (Raúl Yanquen, 2010). Este proceso da como resultado un índice de términos de indización que será utilizado como herramienta de búsqueda y acceso al contenido de recursos en sistemas de recuperación de información.

## 1.3 Objeto de Estudio.

### 1.3.1 Descripción General.

Con el aumento de las TICs han surgido dos términos que tienen gran relación: la informática y la recuperación de información. La recuperación de información desde un principio ha estado ligada a la ciencia de la informática, no sólo por el uso de un computador y de las TICs como una herramienta de trabajo y que finalmente facilita la recuperación de información, sino porque gran parte de la investigación ha sido orientada al diseño de mejores sistemas de recuperación de información. El problema de la recuperación de información desde el área de la informática, está principalmente en diseñar y construir

índices eficientes para el procesamiento de las consultas de los usuarios con un alto rendimiento, y en el desarrollo de algoritmos de rango que mejoren la calidad de los resultados obtenidos.

### 1.3.2 Descripción actual del dominio del problema.

El acelerado aumento de la cantidad de información a nivel mundial ha propiciado que se incremente la necesidad de desarrollar aplicaciones informáticas, para lograr una búsqueda y recuperación de ésta eficientemente. El departamento Señales Digitales, perteneciente a GEySED tiene como necesidad implantar un sistema, mediante el cual pueda promover la información científica de interés con que cuenta.

## 1.4 Análisis de otras soluciones existentes.

El conocimiento es unos de los bienes más valiosos de una organización, por lo que es de vital importancia acceder a ésta y recuperarla. Sin embargo, con el creciente aumento de la información, la calidad que presentan y su amplia gama de formatos, es todo un reto llegar a tener un acceso a ella de forma eficaz y una captura precisa de lo se está buscando.

El creciente aumento de internet y la tendencia constante hacia sistemas de información distribuidos, imágenes documentales, publicaciones electrónicas y multimedia interactivas, ha obligado a la industria de las tecnologías de recuperación de información a adaptarse a las nuevas demandas.

Un software de recuperación de información debería ser capaz de:

- ✓ Indexar y recuperar eficiente, precisa y automáticamente grandes volúmenes de datos a lo largo de extensiones y entornos diversos

- ✓ Adaptar en todo momento la creciente proliferación de tipos de datos, incluyendo texto estructurado y no estructurado, documentos combinados, fotografías, video en movimiento, gráficos e imágenes especializadas, señales, voces y otros sonidos

- ✓ Posibilitar a las personas encontrar la información requerida fácil y rápidamente sin tener en cuenta su lenguaje o su experiencia

- ✓ Integrar soluciones de arquitectura abierta, adaptable y extensible, dentro de un espectro completo de plataformas, sistemas operativos, redes y aplicaciones diversas

- ✓ Proporcionar, bajo parámetros efectivos de coste-beneficio, los componentes que maximicen un entorno apropiado y eficiente para el desarrollo de las aplicaciones, y la mejora de su productividad.

- ✓ Minimizar los recursos humanos e informáticos necesarios para que aumenten los niveles coste-beneficio y las ventajas competitivas.

### 1.4.1 Excalibur RetrievalWare.

Líder en el mercado desde los 80 en soluciones de búsqueda y recuperación de información. La tecnología Excalibur proporciona con sus herramientas funcionalidades para la gestión de la información basadas en el Proceso de Reconocimiento Adaptativo de Patrones (APRP) y las Redes Semánticas. Así, de forma simultánea es posible acceder a los patrones de información digital y al significado real de las palabras. El objetivo de este es mejorar la forma en que los usuarios recuperan la información almacenada. En lugar de pedir a los usuarios que se adapten a la tecnología, la tecnología se adapta a la forma de pensar y trabajar de las personas.

Los motores de recuperación del conocimiento Excalibur, lo que se denomina comúnmente "la familia Excalibur" son: Excalibur RetrievalWare para texto, Excalibur Visual RetrievalWare para imágenes y Excalibur Screening Room para formatos multimedia. Los productos de Excalibur intentan adaptarse a tipos de datos múltiples y ofrecer soluciones para afrontar los retos provocados por la explosión informativa derivada de internet e intranet. (Montiel, 2002)

Algunas de las **características** de este buscador son:

- ✓ Excalibur RetrievalWare ha sido desarrollada para transformar el amplio espectro de información existente en una organización (papel, ficheros electrónicos, bases de datos, Notes, ofimática, webs) en conocimiento utilizable. Está diseñado para la recuperación rápida de texto tanto estructurado como no estructurado. Procura una captura más rápida y exacta del conocimiento facilitando a los usuarios un acceso instantáneo y unas capacidades de recuperación que recorren todos los

almacenes de información de una empresa, incluyendo internet, intranet o, simplemente documentación en papel.

- ✓ Integra dos nuevas formas de acceder a la información, además de la ya conocida booleana, exacta o tradicional:

    - Patronal: el sistema de patrones APRP, incorporado al motor, descompone cada palabra en un mapa de bits y realiza las búsquedas por medio de una comparación porcentual entre dichos mapas, evitando así cualquier posible error en el reconocimiento de caracteres.

    - Semántica o conceptual: gracias a su red semántica e-lexis, RetrievalWare localiza cualquier texto que contenga una palabra dada, incluyendo sus sinónimos, derivados y palabras relacionadas.

- ✓ Las aplicaciones y componentes de RetrievalWare hacen que el manejo de las herramientas sea fácil y de gran potencia. Los usuarios encuentran con precisión los datos requeridos mientras navegan por bases de datos cada vez más voluminosas, bien en entornos de grupos de trabajo o en internet.

- ✓ Proporciona capacidades de "perfilación" (Difusión selectiva de información) en tiempo real y búsqueda retrospectiva de imágenes documentales, de texto combinado con bases de datos o búsquedas de datos visuales a partir de su contenido.

- ✓ La arquitectura flexible y modular está diseñada para una gran adaptabilidad en entornos Cliente/Servidor y ofrece un set completo de herramientas de desarrollo de la aplicación -desde MS Visual Basic hasta las librerías C.

- ✓ Las plataformas soportadas por RetrievalWare incluyen UNIX y servidores Windows/NT con clientes PC, UNIX y clientes internet.

- ✓ Capacidad para mejorar la productividad y reducir costes en la explotación de los sistemas de gestión, ya que elimina los pre-procesamientos de datos y los costes de aprendizaje del usuario.(Montiel, 2002)

### 1.4.2 Copernic Desktop Search.

Este buscador es un producto de la empresa canadiense Copernic Technologies, es un agente buscador que transfiere una ecuación de búsqueda a un conjunto de buscadores de manera simultánea, recupera las referencias pertinentes y las ordena según el grado medio de relevancia obtenido de cada uno de los buscadores.

Copernic Desktop Search es un buscador de ficheros de todo tipo en el ordenador. Es una herramienta que posee un potente motor de búsqueda de cualquier tipo de archivo. Recorre con rapidez el disco duro en busca de ficheros del tipo: Microsoft Word, Excel, y PowerPoint, PDFs, música en todos sus formatos, imágenes y videos también en todos sus formatos. Realiza las búsquedas tanto en discos duros como en cualquier otra unidad. También busca en los diferentes navegadores de internet que tenga instalados tanto en el historial, favorito, contactos, entre otros. Se puede seleccionar búsqueda por: Títulos y por formatos. (Traducegratis, 2011)

Es un programa que se actualiza constantemente, determinando así la ubicación de todos los ficheros que estén almacenados en el disco duro anexándolos a su potente memoria, permitiendo de esta manera la búsqueda de un modo sencillo y a una velocidad increíble.

Posee una interfaz que es fácil de usar, utiliza un sistema de índice que va agregando cada fichero nuevo a medida que se guarda en el disco. Este índice es personalizable, es decir, puedes recoger en el sólo un tipo determinado de ficheros con la información que más interese tener disponible a la hora de realizar búsquedas. (Bajame, 2007)

### 1.4.3 Google Desktop Search.

Es una aplicación de búsqueda que permite acceder fácilmente a la información almacenada en el equipo y en la web. Con esta se puede buscar información en los mensajes de correo electrónico, archivos, archivos de música y fotografías. Indexa de forma automática prácticamente cualquier tipo de archivo y permite buscar el texto completo de estos. Entre los tipos de archivo que indexa el programa, se incluyen los siguientes: archivos de texto y código fuente, archivos PDF, archivos HTML, documentos OpenOffice.org, archivos de imagen y de música, páginas manuales y páginas de información, nombres de archivos y carpetas, Microsoft Word, Excel y PowerPoint.

Google Desktop empieza a indexar los archivos del equipo inmediatamente después de haberlo instalado. Esta indexación, que tiene lugar una sola vez, se ha diseñado para que coexista perfectamente con el trabajo diario, así que se puede continuar trabajando mientras el sistema efectúa la indexación. En función de los archivos y de los otros elementos que existan en el equipo, este proceso puede durar varias horas. Cuando finaliza, Desktop se asegura de que el índice esté actualizado: agrega los mensajes de correo electrónico que se van recibiendo, los archivos que se actualizan y las páginas web visitadas. Google Desktop para Linux está disponible en varios idiomas. Se adaptará automáticamente al idioma definido en la configuración de idioma de tu equipo. (Google, 2009)

### 1.4.4 Swish-e.

Swish-e[4] (Simple Web Indexing System for Humans – Enhanced) es una máquina de búsqueda que puede indexar rápida y fácilmente directorios y archivos en diferentes formatos y realizar búsquedas sobre los índices generados. Es rápido, flexible, y es un sistema de código abierto para la indexación de las colecciones de páginas web u otros archivos. Se utiliza para poner en un índice colecciones de documentos que se extienden hasta un millón documentos de tamaño e incluye los filtros de la importación para muchos tipos del documento. Este sistema indexa archivos en diferentes formatos tales como: e-mail, PDF, HTML; XML, Microsoft Word, Power Point, Excel y la mayoría de los archivos que se puedan convertir a XML y HTML.

A partir de las instrucciones de un sencillo archivo de configuración, swish-e recorre los directorios y archivos, y genera un índice que puede ser utilizado en cualquier plataforma soportada. A partir del índice, se pueden realizar búsquedas desde la línea de comandos, a través de una librería C y también por medio de una interfaz web implementada en un script de PERL.

Los formatos nativos soportados por swish-e son archivos tipo texto, como html, xml. Por medio de filtros (incluidos) se pueden indexar archivos PDF, gzip, y PostScript. Además, se pueden utilizar otros filtros externos como *GNOME libxml2 parser* para leer archivos de MS-Office. También puede ser utilizado como una alternativa a búsquedas de texto completo implementadas con bases de datos como MySQL.

Algunas de las características que presenta este potente buscador:

[4] Swish-e (Simple Web Indexing System for Humans): **http://swish-e.org/**

- ✓ Indexar rápidamente una gran variedad de formatos de archivos, utilizando filtros.
- ✓ Incluye un *web-spider* para indexar archivos remotos vía HTTP.
- ✓ Puede utilizar programas externos para leer la información de entrada, como por ejemplo, aplicaciones que realizan consultas en bases de datos.
- ✓ Soporta campos o propiedades en los documentos, por ejemplo, etiquetas META, o elementos XML. Estos pueden ser indexados y buscados.
- ✓ Los resultados de las búsquedas pueden regresar un resumen o descripción del documento.
- ✓ Búsquedas inteligentes por medio de soundex y metaphone.
- ✓ Búsqueda por frases, oraciones y comodines.
- ✓ Búsquedas en ligas HTML.
- ✓ Soporta expresiones regulares para seleccionar los archivos a indexar.
- ✓ Los resultados pueden ser ordenados por relevancia, por algún campo o propiedad, en orden ascendente o descendente.
- ✓ Limitar la búsqueda a sólo ciertas etiquetas HTML (Meta, Title, comentarios, entre otras), elementos XML o campos.
- ✓ El archivo índice puede ser utilizado en varias plataformas.
- ✓ Cuenta con una librería para integrar capacidad de búsqueda en aplicaciones, así como un módulo PERL para acceder a la librería por medio de un API.
- ✓ Open Source y gratis. (Webindex, 2006)

Dentro de las principales ***desventajas*** al utilizar el programa Swish-e se encuentran las siguientes:

- ✓ No permite la indización óptima de archivos semi-estructurados que no manejen las marcas de HTML/XML, por ejemplo no se podría indizar archivos del tipo MARC21.

- ✓ No maneja ningún protocolo de interoperabilidad para recuperar la información en forma normalizada (por ejemplo z39.50, OAI-PMH, etc.).

- ✓ Actualización del software: Tal como se puede observar en la página de Swish-e, la última versión (2.5.4) es del 29 de Enero de 2007.(Gómez, 2009)

### 1.4.5 ¿Por qué Swish-e?

Actualmente el país no está en condiciones de afrontar software o productos que estén representados por altos intereses comerciales, por lo que la industria del software conjuntamente con la dirección del país tiene que redimensionar el desarrollo hacia las tecnologías libres. Para que este logre una independencia tecnológica es necesario involucrar todos los esfuerzos posibles dentro de estas tecnologías. Swish-e es una herramienta de tecnología libre, la cual indexa cantidades grandes de documentos, generando un índice por el cual se puede realizar la búsqueda desde la línea de comando. Por lo que es importante el uso de esta en el sistema a desarrollar en la presente investigación, para, por medio de una interfaz web mostrar los resultados de las búsquedas. Este sistema debe permitir una gestión de acceso y contendrá una sesión de noticias.

## 1.5 Metodología de Desarrollo de Software.

En la actualidad el uso de metodologías es un factor importante en el desarrollo de sistemas informáticos. Una Metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. A lo largo del tiempo, gran cantidad de métodos han sido desarrollados, diferenciándose por su fortaleza y debilidad. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software. (Marblestation, 2008)

### 1.5.1 Rational Unified Process (RUP).

El Proceso Unificado es un proceso de desarrollo de software. El Proceso de Unificado es un marco de trabajo genérico que puede especificarse para una gran variedad de sistemas de software, para

diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software. Los verdaderos aspectos definitorios del Proceso Unificado se resumen en: dirigido por casos de uso, centrado en la arquitectura, iterativo, e incremental. Esto es lo que lo hace al Proceso Unificado único.

- ✓ Dirigido por casos de uso: ya que está dirigido a satisfacer las necesidades de los usuarios, es decir; lo que ellos necesitan y desean.

- ✓ Centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.

- ✓ Iterativo e incremental: es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

Estos tres aspectos: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las 3 ideas reduciría dramáticamente el proceso de unificado. (Ivar Jacobson, 2000)

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye con una versión del producto. Cada ciclo consta de 4 fases: Inicio (El objetivo en esta etapa es determinar la visión del proyecto), Elaboración (En esta etapa el objetivo es determinar la arquitectura óptima), Construcción (En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial, o sea, crear el producto) y Transición (El objetivo es llegar a obtener una liberación del producto).

Beneficios que aporta RUP:

- ✓ Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.

- ✓ Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos.

- ✓ Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.

- ✓ Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.

- ✓ Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.

- ✓ Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.

### 1.5.2 Extreme Programming (XP).

XP es una de las metodologías ágiles más utilizadas, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Sourceforge, 2005)

El ciclo de vida ideal de XP consiste de seis fases:

1. Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto.

2. Planificación de la Entrega: En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.

3. Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado.

4. Producción: La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.

5. Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.

6. Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. (Torres, 2003)

### 1.5.3 Microsoft Solution Framework (MSF).

MSF es un marco de trabajo flexible e interrelacionado con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características:

- ✓ *Adaptable:* se ajusta a las necesidades de cada entorno.
- ✓ *Escalable:* puede organizar equipos tan pequeños entre 3 ó 4 personas, así como también, proyectos que requieren 50 personas a más.
- ✓ *Flexible:* es utilizada en el ambiente de desarrollo de cualquier cliente.
- ✓ *Tecnología Agnóstica:* puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. (Sánchez, 2004)

MSF organiza los procesos necesarios para crear y entregar un proyecto, dividiendo el desarrollo de este en 6 fases:

Fase 1: Visión (Visión y Alcance Aprobados). El objetivo de esta fase es obtener una visión del proyecto compartida, comunicada, entendida y alineada con los objetivos del negocio. Además, identificar los beneficios, requerimientos, sus alcances y restricciones; y los riesgos inherentes al proceso.

Fase 2: Planeación (Cronograma de proyecto Aprobado). El objetivo de esta fase es obtener un cronograma de trabajo que cumpla con lo especificado en la fase de Visión dentro del presupuesto, tiempo y recursos acordados. Este cronograma debe identificar puntos de control específicos que permitan generar entregas funcionales y cotas en el tiempo.

Fase 3: Desarrollo (Alcance Completo). Esta fase tiene como objetivo iterativamente de la mano de la fase de Planeación y de la Estabilización versiones de los productos entregables y medibles que permitan ante el cliente probar características nuevas sucesivamente. Esto incluye los ajustes de cronograma necesarios.

Fase 4: Estabilización (Versión Aprobada). Su objetivo es obtener una visión final del producto probada, ajuntada y aprobada en su totalidad.

Fase 5: Instalación (Entrega). Su objetivo es entregar al cliente el producto finalizado en su totalidad. Como garantía se han superado con éxito las etapas anteriores.

Fase 6: Soporte (Entrega Ajustada). Su objetivo es brindar y garantizar al producto durante el tiempo estipulado en el contrato; registrado los reportes de soporte y mantenimiento recibidos, así como los ajustes y versiones ajustadas obtenidas. Esto solo será válido para ajustes que estén dentro de lo descrito en los documentos de la fase Visión. En esta fase es posible identificar características y requerimientos que no fueron tenidos en cuenta y que se salen del alcance de la fase de Visión. Por lo tanto, es probable que se inicie otro proceso con los nuevos requerimientos, generando así un nuevo proyecto y un nuevo inicio de las fases de la metodología.

Este marco de trabajo se puede utilizar para pequeños y grandes proyectos, es un proceso que se basa en la colaboración entre todos los que realizan el proyecto, analiza riesgos y provee de plantillas que ayudan a la hora de la documentación.

### 1.5.4 ¿Por qué RUP?

En correspondencia con el análisis realizado anteriormente sobre las metodologías seleccionadas, se puede evidenciar la presencia de diferentes características para cada una de ellas y la particularidad de poder ser empleadas para diferentes situaciones respectivamente. RUP es perfectamente idónea para proyectos de largo plazo, aunque es caracterizada por ser una metodología compleja. MSF por su parte es una metodología adaptable a proyectos de cualquier dimensión. La metodología XP está destinada a proyectos de corto plazo, se basa en iteraciones pequeñas y está dirigida fundamentalmente a los clientes. Una vez estudiado lo referente a estas metodologías de desarrollo se determinó, por las características, utilizar RUP debido a que aporta todos los elementos para desarrollar aplicaciones grandes y que requieren de mucha documentación, permite desarrollar aplicaciones mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software.

## 1.6 Arquitectura.

### 1.6.1 Arquitectura en 3 Capas.

La arquitectura 3 capas o programación 3 capas consiste literalmente en separar un proyecto en Capa de Presentación, Capa de Negocio y Capa de Datos. Esto permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

Ventajas de esta Arquitectura:

- ✓ El desarrollo se puede llevar a cabo en varios niveles.
- ✓ Desarrollos paralelos (en cada capa).
- ✓ Aplicaciones más robustas debido al encapsulamiento.
- ✓ En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- ✓ Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).

- ✓ Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).

- ✓ Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalabilidad, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad. (Kernelerror, 2009)

### 1.6.2 Modelo Cliente/Servidor.

El modelo Cliente/ Servidor presenta las siguientes características:

- ✓ El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma. Un servidor da servicio a múltiples clientes en forma concurrente. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final. La interrelación entre el hardware y el software está basada en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.

- ✓ Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un sólo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un sólo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final. Además, se constituye como el nexo más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

En conclusión, Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. (Kernelerror)

### 1.6.3 Arquitectura Modelo Vista Controlador (MVC).

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista, interactúa con el Modelo a través de una referencia al propio Modelo.

Este modelo de arquitectura presenta varias **ventajas**:

- ✓ Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado
- ✓ La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación. (Sunsite)

### 1.6.4 ¿Por qué arquitectura Modelo Vista Controlador?

Una vez estudiado lo referente a las arquitecturas se determinó por las características que presenta usar la arquitectura MVC, ya que ésta separa la lógica de negocio (el modelo) y la presentación (la vista) se consigue un mantenimiento más sencillo de las aplicaciones. Además, tendría una organización de código separado por sus funciones.

## 1.7 Lenguajes usados en el desarrollo del sistema.

### 1.7.1 Lenguaje de Modelado: UML.

El UML (Lenguaje Unificado para la Construcción de Modelados) es un lenguaje que permite visualizar, especificar, construir y documentar y/o ser base de documentación de los artefactos de los sistemas de software. Es un sistema de notaciones que, entre otras cosas, incluye el significado de sus notaciones. Su finalidad es describir modelos de sistemas del mundo real y del mundo del software, basados en los conceptos orientados a objetos. (Larman, 1999)

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. UML no es una guía para realizar el análisis y diseño orientado a objetos, es un lenguaje que permite la modelación de sistemas con paradigma orientado a objetos. Este proporciona un vocabulario y reglas para permitir una comunicación, el mismo indica cómo crear y leer los modelos, pero no dice cómo crearlos. UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten, es además un método formal de modelado. Aporta un mayor rigor en la especificación y permite realizar una verificación y validación del modelo realizado. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Una de sus principales ventajas es la notable efectividad y productividad que se consigue en labores de diseño arquitectónico y mantenimiento haciendo uso de UML frente a la realización de las mismas tareas en ausencia de modelos.

Dentro de las principales ventajas que presenta se encuentra el uso de lenguajes visuales facilita su asimilación y entendimiento por parte del equipo de trabajo. Además:

- ✓ El tiempo invertido en el desarrollo de la arquitectura se minimiza.
- ✓ La detección y resolución de errores se agiliza siempre y cuando se haga uso de herramientas adecuadas de diagnóstico y depuración.
- ✓ La trazabilidad y la documentación del proyecto se confeccionan de una forma ordenada y guiada por los casos de uso.

Entre sus principales **desventajas** se encuentran:

- ✓ UML no es un método de desarrollo. No dice cómo pasar del análisis al diseño y de este al código. No son una serie de pasos que te llevan a producir código a partir de unas especificaciones.
- ✓ Diversos desarrolladores también dan nota de que UML es algo impreciso dentro de su notación, por ejemplo, al hacer referencias a un diagrama con servidores, no se sabe si los servidores simbolizados se encuentran operativos, restringidos, pasivos o en otro estado. Por eso se le califica de un poco “inexacto”.
- ✓ Otro problema de UML es que no se presta con facilidad al diseño de sistemas distribuidos. En tales sistemas cobran importancia factores como transmisión y persistencia. UML no cuenta con maneras de describir tales factores. No se puede, por ejemplo, usar UML para señalar que un objeto es persistente o remoto. (Universidad de San Carlos de Guatemala, 2010)

### 1.7.2 Lenguajes de Programación y Tecnologías utilizadas.

#### 1.7.2.1 PHP.

El lenguaje de programación PHP denominado preprocesador de hipertexto, (del inglés “Hypertext Pre-processor”), es un lenguaje libre y multiplataforma. Posee una amplia documentación en su página oficial posibilitando gran comprensión del mismo, se sustenta en la actualidad bajo el paradigma más difundido actualmente en el mundo que es programación orientada a objetos. Incluye también la programación estructurada y servicios web. Presenta excelente integración con todos los motores de base de datos. Cuenta con una biblioteca que trae un conjunto de funciones para realizar cualquier operación (acceso a

base de datos, encriptación, envío de correo, XML, creación de PDF, entre otros). Su código es libre y se sustenta bajo la licencia GPL. PHP, es un lenguaje de programación usado normalmente para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, es la versión libre del sistema equivalente de Microsoft ASP. Es un lenguaje encapsulado dentro de los documentos HTML, de forma que se pueden introducir instrucciones PHP dentro de las páginas. Gracias a esto el diseñador gráfico de la página puede trabajar de forma independiente al programador. Debido a la naturaleza open-source[5] de PHP, si hay algo que actualmente no se pueda hacer en este lenguaje de programación no existe ningún impedimento para que se pueda escribir un módulo o una extensión en código C para extender la funcionalidad y hacer lo que se desee. Esto es posible por la buena documentación de la API que está disponible para todos.

Principales **ventajas** de PHP.

- ✓ Sintaxis cómoda: PHP cuenta con una sintaxis similar a la de C, C++ o PERL. Lo más destacado ocurre a nivel semántico: el chequeo de tipos es poco estricto. Es decir, cuando se crea una variable no se tiene que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador, aunque los errores que se cometen pueden ser muchos más graves y difíciles de corregir al reducirse mucho las posibilidades del intérprete para detectar incompatibilidades entre variables.

- ✓ Soporta objetos y herencia: Tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores. Además, soporta herencia, aunque no múltiple.

- ✓ Ejecución en Servidor: Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página web

[5]Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

con el código HTML resultante de la ejecución del código PHP compatible con todos los navegadores.

- ✓ Se puede incrustar código PHP con etiquetas HTML.
- ✓ Compatibilidad con bases de datos: Amplio soporte para una gran cantidad de bases de datos. Tiene acceso un gran número de gestores de bases de datos: Adabas D, dBase, Empress, Ingress, InterBase, FrontBase, DB2, Informix, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase entre otros.
- ✓ Se puede hacer de todo lo que se pueda transmitir por vía HTTP.
- ✓ Multiplataforma: Funciona tanto en sistemas Unix o Linux con servidor web Apache, como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra.
- ✓ Licencia de software libre: Es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, ni existen límites en su distribución y, es posible ampliarlo con nuevas funcionalidades si así se desea.
- ✓ Extensa librería de funciones: Cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores.

Principales **desventajas** de PHP.

- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La orientación a objetos es aún muy deficiente para aplicaciones grandes.

La aparición de soluciones adecuadas y sencillas hace que PHP se convierta en la mejor opción actual para la multitud de necesidades. Actualmente es uno de los paquetes para programación de internet más utilizados.

#### 1.7.2.2 JavaScript.

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en el interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

#### 1.7.2.3 HTML.

HTML, siglas de Hyper Text Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas". HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. (Deciencias, 2010)

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C[6]. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

[6] El World Wide Web Consortium (W3C) es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica.

HTML es un lenguaje utilizado únicamente para dar estructura a una página web. El estilo de la propia página web vendrá dado por un enlace a una hoja CSS (Hojas de Estilo en Cascada), de las que se hablará aparte. Es decir, que toda etiqueta que defina estilo y no estructura no se podrá considerar perteneciente al lenguaje HTML. (Deciencias, 2010)

Como cualquier página web está soportada por un fichero de texto, se pueden elaborar, editar o modificar directamente con el bloc de notas o el wordpad de Windows 95, aunque existen programas editores de páginas web a la venta como HotDog o Front Page y programas gratuitos de edición básica como Netscape Composer, que permiten editarlas con algunas funciones automatizadas a través de botones y menús. (Nodo50)

#### 1.7.2.4 ExtJS.

ExtJS es una librería JavaScript que permite construir aplicaciones complejas en internet además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX y DHTML. Además, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales.

Esta librería incluye:

- ✓ Componentes UI del alto rendimiento y personalizables.
- ✓ Modelo de componentes extensibles.
- ✓ Un API fácil de usar.
- ✓ Licencias Open Source (GPL) y comerciales.

ExtJS ha marcado la diferencia en la preferencia de muchos desarrolladores. Y no sólo para aplicaciones web sino que ya es útil para productos de alta demanda en el mercado como iPhone y IPAD.

Ventajas:

- ✓ Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos.
- ✓ Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, IE, Safari, Opera etc.).
- ✓ El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.
- ✓ Relación entre Cliente-Servidor balanceado: Se distribuye la carga de procesamiento, permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- ✓ Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir que datos desean transmitir al servidor y viceversa.
- ✓ Comunicación asíncrona. En este tipo de aplicación el motor de renderizar puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario.

Desventajas:

- ✓ Necesidad de una plataforma: Pues de pende del paquete ExtJS para obtener los resultados deseados.
- ✓ Falta de un potente diseñador gráfico: demora la obtención del resultado final al no proveer una forma fácil y rápida de desarrollar.

#### 1.7.2.5 Ajax.

Ajax, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Algunas **ventajas** representativas del uso de AJAX:

- ✓ La experiencia de usuario en la navegación es mucho más rica. Ya no se refresca la página constantemente al interactuar con ella.
- ✓ El tiempo de espera para una petición se reduce. Al hacer un pedido (request) al servidor, no se envía toda la página.
- ✓ Por la misma razón anterior el tráfico al servidor se reduce.

También presenta **desventajas** entre las cuales se encuentran:

- ✓ Falta de integración con el botón retroceder del navegador. Se debe tener en cuenta esto al intentar guardar funcionalidad con este botón.
- ✓ Falta de soporte para todos los navegadores. Aunque esto se va reduciendo, el problema se presenta por la falta de soporte para JavaScript y XMLHttpRequest.
- ✓ Problemas si el usuario ha deshabilitado el uso de JavaScript en su navegador.
- ✓ Demasiado código Ajax hace lento el navegador. A más Ajax, más uso de código JavaScript del lado del navegador, por consiguiente, mayor trabajo para este.(Tarrillo, 2007)

## 1.8 Herramientas a utilizar para desarrollar la presente investigación.

### 1.8.1 Herramientas CASE.

Para el modelado de los artefactos y diagramas generados a lo largo del ciclo de vida del proyecto se hace necesario utilizar una herramienta CASE. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

#### 1.8.1.1 Visual Paradigm.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

#### 1.8.1.2 Rational Rose.

Es una herramienta software para el modelado visual mediante UML de sistemas software. Permite especificar, analizar y diseñar el sistema antes de Codificarlo.

Algunas de las características que presenta Rational Rose son:

- ✓ Mantiene la consistencia de los modelos del sistema software.
- ✓ Chequeo de la sintaxis UML.
- ✓ Generación de documentación automáticamente.
- ✓ Generación de código a partir de los modelos.
- ✓ Ingeniería inversa (crear modelo a partir código).

Costo de la Licencia: Alto.

Cubre todo el ciclo de vida de un proyecto:

- ✓ Concepción y formalización de un proyecto.
- ✓ Construcción de los componentes.
- ✓ Transición a los usuarios y certificación de las distintas fases.

#### 1.8.1.3 ¿Por qué Visual Paradigm?

Se decidió el uso de Visual Paradigm, pues su uso está muy estandarizado a nivel mundial y constituye una herramienta multiplataforma muy madura y acabada. Además, el grupo de desarrollo se encuentra familiarizado con el uso de esta herramienta.

### 1.8.2 IDE de desarrollo.

Un entorno de desarrollo informático es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

#### 1.8.2.1 Zend Studio[7].

Es un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, en lenguaje PHP. 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. Tiene versiones para diferentes sistemas operativos Windows, Linux y MacOS. Consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración hay que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. Contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no sólo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que vaya creando el programador. (Linux-party). Agiliza el trabajo, cuenta con un buen depurador, y presenta infinitas opciones que permiten un desarrollo profesional de las aplicaciones.

[7] *Zend Studio:* http://www.zend.com/en/

**Desventajas**:

- ✓ Requiere Licencia de pago.
- ✓ No incluye editor visual HTML.
- ✓ Un poco complejo.

#### 1.8.2.2 NetBeans.

NetBeans[8] es un entorno de desarrollo multilenguaje, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. Es un producto libre y gratuito sin restricciones de uso. (NetBeans, 2011). Tiene soporte para crear interfaces gráficas de forma visual, crear aplicaciones para móviles, desarrollar aplicaciones web y además estas funcionalidades son ampliables mediante instalación de paquetes adicionales. NetBeans IDE se conoce como la solución más completa para programar en Java. Disponible para Windows, Mac, Linux y Solaris.

Características de NetBeans:

- ✓ Buen editor de código.
- ✓ Soporte para Ruby, JRuby, y Ruby on Rails.
- ✓ Instalación y actualización más simple.
- ✓ Enlazar datos con el Swing GUI.
- ✓ Profiling integrado, profiling “points”.
- ✓ Características visuales para el desarrollo web.

[8] NetBeans: http://www.netbeans.org/community/index.html

- ✓ Creador gráfico de juegos para celulares.
- ✓ Mejoras para SOA y UML.
- ✓ Soporte para PHP.

#### 1.8.2.3 Aptana (Eclipse).

Aptana[9] es un entorno de desarrollo dirigido hacia las aplicaciones web escritas en Ajax/JavaScript. Está basado en Eclipse y se puede encontrar para las tres plataformas mayoritarias (Win, Mac y Linux), ya sea como plugin del mismo Eclipse, o como aplicación por separado. En caso que se quiera que sirva para editar código en PHP lo mejor sería utilizar Aptana como plugin de Eclipse.

Las características de esta IDE son similares a otras IDE más generales: gestión de proyectos, vista outline y vista previa, autocompletado, macros (en este caso, escritos en JS), gestión de documentación, entre otros. Soporta las librerías más populares: Prototype, Scriptaculous, Dojo, MochiKit, Yahoo UI, Aflax, JQuery y Rico. (Linux-party)

**Ventajas**: Permite comprobar la compatibilidad de las funciones con los diferentes navegadores, multiplataforma, sincronización con carpetas locales y remotas, incluye plugins para Eclipse. La principal **desventaja** es el consumo de recursos.

#### 1.8.2.4 ¿Por qué NetBeans?

Después de realizar un análisis de las distintas características de los diferentes IDE se escoge el NetBeans por las ventajas que presenta. Las versiones más recientes permiten integrar el framework Symfony posibilitando desarrollar de forma más eficiente y organizada de las aplicaciones. Por ser usado por una gran parte de los desarrolladores de aplicaciones web del mundo posee mucha documentación en diversos formatos, tanto audiovisuales como en texto. Al poseer una interfaz agradable e intuitiva los desarrolladores se sienten identificados y a gusto. Por todo lo expuesto se propone la utilización de este IDE para el desarrollo de la aplicación propuesta.

[9] Aptana: http://www.aptana.com/

### 1.8.3 Framework.

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

#### 1.8.3.1 Symfony.

Symfony[10] es un framework completo, una biblioteca de clases coherente escrita en PHP. Proporciona una arquitectura, componentes y herramientas para desarrolladores, para crear aplicaciones webs complejas más rápidamente. Symfony se basa en la experiencia. Utiliza la mayor parte de las mejores prácticas de desarrollo web y se integra algunas grandes bibliotecas. Cada vez son mayores los usuarios que se unen a este, y que hace de Symfony el framework PHP más popular. (Symfony-project)

Symfony está diseñado para que se ajuste a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas.
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

[10] Symfony es el mejor framework para crear aplicaciones PHP y la forma más sencilla de aumentar la productividad y calidad de tu trabajo. Symfony ha sido probado con éxito en algunos de los sitios web más grandes del mundo. http://www.symfony.es/

Entre las **desventajas** que presenta Symfony se encuentran:

- ✓ Consumo de memoria.
- ✓ Búsqueda de datos lenta. Esto es cuando se desea realizar una búsqueda de datos muy específica, se la debe hacer en forma manual ya que la utilización de la interfaces PROPEL genera código que puede ser lento.
- ✓ La integración entre diferentes aplicaciones de un mismo proyecto todavía no se encuentra muy depurada en Symfony. Se presentan varios inconvenientes si se desea utilizar un módulo que se encuentra en otra aplicación.

#### 1.8.3.2 Prado.

Prado significa PHP Rapid Application Development Object-oriented, se trata de un framework de desarrollo de aplicaciones web que está basado en objetos y utiliza el lenguaje PHP. Prado es un framework para PHP basado en componentes y en eventos. Inicialmente inspirado en Apache Tapestry[11], la primera versión se realizó para PHP4, pero se reescribió completamente para PHP5. Entre las características que ofrece se encuentran la separación entre la presentación y la lógica, su arquitectura modular configurable, componentes web, internalización y localización, manejo de errores, logs, caché, ACL, prevención de XSS y mucho más. Entre los objetos que esta plataforma de desarrollo ofrece se encuentran: acceso a bases de datos, formularios y controles web y cache entre muchos otros. Además, soporta componentes AJAX, permite personalización y también incorpora seguridad contra XSS y prevención usando cookies.

#### 1.8.3.3 Zend Framework.

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF 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. A menudo, se refiere a este tipo de diseño como "use-at-will" (uso a

[11] Apache Tapestry es un framework open-source para crear aplicaciones dinámicas, robustas y fácilmente escalables en Java.

voluntad). Brinda soluciones para construir sitios web modernos, robustos y seguros. Además, es Open Source.

**Desventajas:**

- ✓ No ofrecen más apoyo que foros, y Google grupos.
- ✓ Curva de aprendizaje: necesidad de aprender nuevas funciones, estructuras y métodos de programación.
- ✓ Dificultad para adaptar el código escrito en PHP tradicional.

#### 1.8.3.4 ¿Por qué Symfony?

Para el desarrollo de la aplicación se decidió usar Symfony ya que es un framework para PHP que permite desarrollar aplicaciones Web de una forma sencilla y cómoda, es escalable y de una potencia más que probada.Permite al desarrollador concentrarse en la lógica de la aplicación y fomenta la consistencia de código entre los desarrolladores. Es estable ya que presenta código base verificado (+4000 pruebas funcionales y de unidades) y cuenta con una numerosa documentación que lo soporta.

### 1.8.4 Sistemas Gestores de Bases de Datos.

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Cualquier operación que el usuario hace contra la base de datos está controlada por el gestor. (Álvarez, 2007)

#### 1.8.4.1 MySQL.

Es un sistema de gestión de bases de datos relacional. Fue creado por la empresa sueca MySQL AB, la cual tiene el derecho de autor del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL[12]de la GNU, aunque MySQL AB distribuye una

[12]Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License* o simplemente sus siglas del inglés GNU GPL. Está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito

versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

En las últimas versiones se pueden destacar las siguientes características principales:

- ✓ El principal objetivo de MySQL es velocidad y robustez.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- ✓ Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- ✓ Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- ✓ Flexible sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- ✓ El servidor soporta mensajes de error en distintas lenguas

Este sistema gestor de base de datos presenta algunas ventajas:

- ✓ Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- ✓ Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

- ✓ Facilidad de configuración e instalación.
- ✓ Soporta gran variedad de Sistemas Operativos
- ✓ Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- ✓ Conectividad y seguridad

Entre las **desventajas** que presenta se encuentra que un gran porcentaje de las utilidades de MySQL no están documentadas y no es intuitivo, como otros programas. (Alma)

#### 1.8.4.2 PostgreSQL.

El PostgreSQL[13] es un poderoso sistema manejador de bases de datos, es decir, un sistema diseñado para administrar grandes cantidades de datos, que tiene la fama de ser el sistema gestor de base de datos de código abierto (Open Source) más avanzado del mundo. Confiable y mantiene la integridad de los datos. PostgreSQL se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo, Linux, varias versiones de UNIX y por supuesto Windows. (PostgreSQL)

A continuación se enumeran las principales características de este gestor de bases de datos:

- ✓ Implementación del estándar SQL92/SQL99.
- ✓ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, entre otros. También permite la creación de tipos propios.
- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, entre otras más.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.

[13] Sistemas Gestores de Bases de Datos (*PostgreSQL*): http://www.postgre.org

- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (Pecos)

El único costo asociado a PostgreSQL es el de conocerlo ya que su código fuente está disponible bajo la más liberal de las licencias del Open Source: la licencia BSD. Bajo esta licencia se tiene la libertad de usar, modificar y distribuir postgres, en productos comerciales o no comerciales, sin costo alguno. El hecho de ser un producto Open Source, sin costos de licencia, convierte al PostgreSQL en una alternativa extremadamente atractiva para las empresas que buscan un ahorro significativo de costos en activos TI.

#### 1.8.4.3 Oracle.

Oracle[14] es una potente herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como las otras bases de datos, por ejemplo, Access, MySQL, SQL Server, entre otras.

Dentro de las ventajas de este sistema gestor está que es el más usado a nivel mundial, puede ejecutarse en distintas plataformas, desde una PC hasta en un supercomputador. El software del servidor puede ejecutarse en multitud de sistemas operativos. Oracle es la base de datos con más orientación hacia internet y presenta un aceptable soporte. (Desarrolloweb, 2002)

El mayor inconveniente de Oracle es su precio. Incluso las licencias de Personal Oracle son excesivamente caras. Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.

14Sistemas Gestores de Bases de Datos (Oracle): http://www.oracle.com/index.html

#### 1.8.4.4 ¿Por qué PostgreSQL?

Para la utilización en el desarrollo de la base de datos se selecciona a PostgreSQL que está licenciado bajo BSD, que utiliza control de versionado concurrente y soporte para la implementación de consultas complejas. Con este SGBD se podrá mantener una base de datos actualizada, confiable y configurable. Además, que es un gestor de base de dato que garantiza una alta seguridad e integridad de los datos, y tiene una arquitectura donde se interrelacionan el nivel lógico, el nivel físico y el nivel externo de las bases de datos, por lo que constituye una potente herramienta para el manejo de los datos.

### 1.8.5 Servidor web.

Un servidor web es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Este servidor se encarga de mantenerse en espera de las peticiones HTTP llevadas a cabo por un cliente. Este realiza una petición al servidor, quien le responde con el contenido solicitado.

#### 1.8.5.1 Servidor Web Apache.

El servidor Apache, es un servidor de páginas web de código abierto multiplataforma y modular, el mismo se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando y qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. Trabaja sobre múltiples plataformas (Unix, Linux, MacOSX, Vms, Win32, OS2) entre otras, soporta CGI, PERL[15], PHP, permite soporte SSL para transacciones seguras. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Apache ha demostrado ser más rápido que muchos otros servidores libres y compite con los mejores servidores comerciales.

[15] PERL es un lenguaje de programación que toma características del lenguaje C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

Es flexible, rápido y eficiente, puede ser ejecutado en cualquier plataforma y adaptado a diferentes entornos y necesidades. Se desarrolla de forma abierta y gracias a que es modular se han desarrollado diversas extensiones.

**Desventajas**:

- ✓ Complejidad - Puede resultar difícil de configurar incluso para tareas sencillas.
- ✓ Formatos de configuración no estándar: Esto dificulta un poco la automatización y el procesamiento de la configuración al no estar basada esta en formatos más soportados como el XML.
- ✓ Falta de integración: Al ser un producto multiplataforma, el servidor no aprovecha al máximo las posibilidades que ofrece el sistema operativo.
- ✓ Administración: Como la mayoría de los programas open-source, uno depende de configurar los archivos a mano o tener que instalarse herramientas adicionales para las tareas de administración.

#### 1.8.5.2 AOLserver (Servidor web de América Latina Online).

El servidor http AOLServer[16] es el servidor web de código abierto libre de América Online, el proveedor de internet con más clientes en el mundo. Es un servidor http de tipo multihebra, basado en TCL, que incluye muchas facilidades de uso orientadas a entornos de gran escala y a sitios web con contenido dinámico. Hay que destacar que todos los dominios y servidores de AOL, que son más de 200 y soportan miles de usuarios simultáneamente y millones de conexiones, funcionan con AOLServer. AOLServer tiene muchos usuarios, gracias a su integración con OpenACS[17], un software de gestión de contenidos de código libre muy potente.

AOLserver se distribuye bajo la licencia AOLserver Public License, que es similar a la de Mozilla (Mozilla Public License). AOLserver fue el primer servidor HTTP en combinar el procesamiento multihilo, con un lenguaje interpretado de serie, y el procesamiento de colas de conexiones persistentes (en inglés:

16 Servidor web de código abierto libre de América Online: http://www.aolserver.com/

[17]OpenACS del inglés *Open Architecture Community System* (Arquitectura Abierta para Sistemas de Comunidades) es un kit de herramientas libre (de código abierto) para el desarrollo rápido de aplicaciones web, con licencia GPL.

Connection Pool) a base de datos. Para los sitios web con bases de datos, esto permitía mejorar el rendimiento hasta cien veces más que la práctica habitual basada en CGI, que abrían una nueva conexión a la base de datos en cada petición de página. Ya hay otros servidores HTTP que consiguen un rendimiento similar con una arquitectura similar, pero AOLserver está varios años por delante de la competencia. (Scripd)

#### 1.8.5.3 ¿Por qué Apache?

Debido a las características expresadas con anterioridad sobre Apache, así como las potencialidades que presenta, se selecciona el mismo para la realización del sistema. Además, es muy fácil de instalar, configurar, está probado y es muy usado por los desarrolladores.

## 1.9 Conclusiones.

En el presente capítulo se ha realizado un análisis de las herramientas de búsqueda y recuperación de información existente, se concluye que aunque son herramientas muy buenas y potentes:

- ✓ Las herramientas Excalibur RetrievalWare, Copernic Desktop Search, Google Desktop Search no constituyen una solución al problema planteado debido a que no posibilitan la gestion de noticias ni la gestión de usuarios.

- ✓ La herramienta swish-e se utilizará como base para elaborar la nueva solución debido a que es rápida, es de código abierto, soporta varios criterios de búsqueda y es capaz de buscar documentos de diferentes formatos.

# CAPÍTULO 2: Presentación de la solución propuesta.

## 2. Introducción.

En este capítulo se describe la solución propuesta, a partir de realizar un estudio inicial del entorno en el cual se enmarca el problema a resolver. Se brinda información referente a los actores y casos de uso del negocio y del sistema, de manera que se pueda entender mejor el producto que se desea obtener. Además, se detallan los requisitos funcionales y no funcionales del sistema propuesto.

### 2.1 Actor del negocio.

| Actor | Descripción |
|---|---|
| Usuario | Es el que lleva a cabo la búsqueda de la información. El Usuario accede al repositorio documental, realiza la búsqueda documento a documento hasta encontrar el de su interés. |

Tabla 1 Descripción de actor del negocio.

### 2.2 Diagrama de caso de uso del negocio.

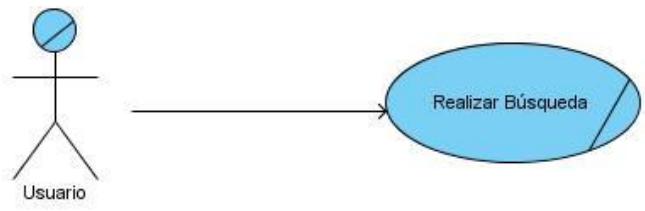


Ilustración 1 Diagrama de casos de uso del negocio.

### 2.3 Descripción textual de los casos de uso del negocio (CUN).

#### 2.3.1 Descripción del CUN Realizar Búsqueda.

| Caso de Uso del Negocio | Realizar Búsqueda |
|---|---|
| **Actores** | Usuario |

| | |
|---|---|
| **Resumen** | El caso de uso se inicia cuando el usuario necesita obtener algún documento. |
| **Casos de Uso asociados** | |
| **Acción del actor** | **Respuesta del proceso de negocio** |
| 1. El usuario realiza la búsqueda. | El usuario entra al repositorio, busca documento a documento, y guarda los deseados. |
| **Otras secciones** | |
| Mejoras propuestas | Se creará una interfaz que permita al usuario realizar una búsqueda de una forma más rápida, mostrando con un nivel de prioridad los resultados de esta. |

Tabla 2 Descripción textual del CUN Realizar Búsqueda.

### 2.5 Requisitos Funcionales.

Los requisitos funcionales describen que debe realizar el software para sus usuarios: aceptar, verificar y registrar datos, transformarlos, presentarlos, entre otras acciones. Estos requisitos quedan recogidos en casos de uso.(Falgueras, 2003)

**RF. 1 Autenticar usuario.**

1.1. Solicitar autenticación.

1.2. Permite acceder a las funcionalidades de acuerdo con los permisos de cada usuario.

**RF. 2 Gestionar Usuario.**

2.1 Modificar usuario.

Mostrar los datos desde el servidor de base de datos (BD).

Seleccionar usuario

Seleccionar elementos a modificar.

Introducir modificaciones.

2.2 Insertar Usuario.

Introducir datos correspondientes al nuevo usuario.

2.3 Eliminar Usuario.

Mostrar los usuarios desde el servidor de BD.

Seleccionar el usuario a eliminar.

Eliminar usuario seleccionado.

**RF. 3 Gestionar Noticia.**

3.1 Modificar Noticia.

Mostrar las noticas existentes.

Seleccionar noticia a modificar.

Introducir modificaciones.

3.2 Insertar Noticia.

Introducir datos correspondientes a la nueva noticia.

3.3 Eliminar Noticia.

Mostrar las noticias desde el servidor de BD.

Seleccionar la noticia a eliminar.

Eliminar la noticia seleccionada.

3.4 Mostrar Noticia.

Mostrar las noticias existentes.

**RF. 4 Realizar Búsqueda.**

4.1 Buscar Documentos.

Introducir elementos a buscar.

4.2 Recuperar Documentos.

Permitir la visualización de los documentos.

4.3 Indexar Documentos.

Ejecutar indexación de documentos.

### 2.6 Requerimientos no Funcionales.

En este epígrafe se hará referencia a los requisitos no funcionales. Los requisitos no funcionales no van asociados a casos de uso concreto y consisten en restricciones impuestas por el entorno y tecnologías, especificaciones sobre tiempo de respuesta o volumen de información tratado por una unidad de tiempo, requisitos en cuanto a interfaces, extensibilidad y facilidad de mantenimiento. (Falgueras, 2003)

**Apariencia o interfaz externa.**

El sistema debe tener una interfaz amigable, donde el usuario pueda realizar acciones con fácil manejo y que posibilite también el buen desempeño en la ejecución de sus funciones.

**Requisitos de Seguridad. Disponibilidad.**

La disponibilidad del sistema debe ser continua con un nivel de servicio de 6 días por 24 horas. Se llevará a cabo un mantenimiento del sistema los fines de semana.

**Hardware.**

La aplicación debe ser compatible con bajas capacidades de hardware, a continuación se muestran datos de estos requerimientos.

- Capacidades de hardware mínimas:

RAM = 512 MB CPU = 1.0 GHz

- Capacidades de hardware recomendadas:

RAM = 1 GB CPU = 2.0 GHz

**Restricciones en el diseño e implementación.**

El sistema estará implementado en lenguaje php5, utilizando como IDE de desarrollo NetBeans. Para la modelación de la arquitectura se utilizará el Visual Paradigm.

**Seguridad.**

El sistema debe tener restricciones de acceso de acuerdo con los roles de cada usuario de manera que sólo puedan ejecutar las acciones para las que tienen permisos.

**Software.**

El sistema debe funcionar sobre Sistema Operativo Linux, a partir de la distribución Ubuntu 8.04.

**Usabilidad.**

El sistema podrá ser utilizado por usuarios que no necesariamente tengan conocimientos informáticos por lo que todas sus funcionalidades deben ser claras, sencillas y accesibles de manera intuitiva.

## 2.7 Descripción de sistema propuesto.

### 2.7.1 Definición de actores.

Los actores representan papeles que la gente (o dispositivos) juegan como usuarios del sistema. Definido más formalmente, un actor es algo que se comunica con el sistema o producto y que es externo al sistema en sí mismo. (Pressman, 2001). Durante el desarrollo de la aplicación se definió un grupo de actores que en un momento determinado desencadenarán un grupo de acciones en el sistema, los mismos se describen a continuación:

#### 2.7.1.1 Descripción de los actores del sistema.

| Actor | Descripción |
|---|---|
| Administrador | Tiene privilegios para autenticarse en el sistema y realizar búsquedas. Tiene la posibilidad de crear, actualizar, eliminar y mostrar noticias. Además de insertar, modificar y eliminar usuarios. Ejecuta la indexación de documentos. |
| Editor | Tiene privilegios para autenticarse en el sistema y realizar búsquedas. Además, tiene la posibilidad de crear, actualizar, eliminar y mostrar noticias. |
| Usuario | Tiene privilegios para autenticarse en el sistema y realizar búsquedas. Además de ver las noticias. |

Tabla 3 Actores del sistema.

### 2.7.2 Definición de casos de uso del sistema.

Los casos de uso del sistema modelan el sistema desde el punto de vista del usuario y son creados durante la obtención de requisitos. En general, un caso de uso es, un texto escrito que describe el papel de un actor que interactúa con el acontecer del sistema. Estos facilitan una descripción de como el sistema se usará. (Pressman, 2001)

### 2.7.2.1 Diagrama de caso de uso del sistema.

El Diagrama de casos de uso del sistema, define las relaciones entre los actores y los casos de uso. Para la realización de la aplicación se definieron un conjunto de acciones que deben ser ejecutadas por distintos actores y que desencadenan un conjunto de operaciones. Las interrelaciones entre las acciones y los actores de la aplicación son agrupadas en el diagrama de casos de uso del sistema que se muestra a continuación.

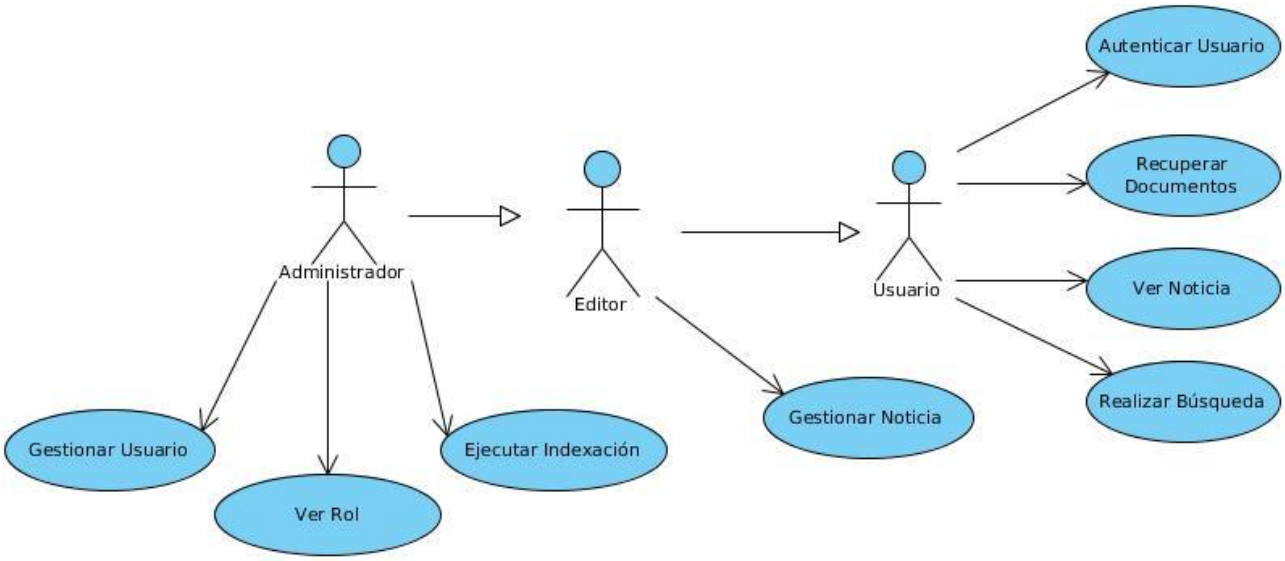


Ilustración 2 Diagrama de Caso de Uso del Sistema.

### 2.7.2.2 Descripción del CU Gestionar Usuario.

| | |
|---|---|
| **Objetivo** | Gestionar los datos de los usuarios. |
| **Actores** | Administrador: (Inicia) Adiciona, modifica y elimina los datos de un usuario en el sistema. |
| **Resumen** | Caso de uso que define funciones de cambio de datos para la cuenta de usuario del sistema, inicia cuando un usuario autentificado como Administrador del sistema solicita adicionar, eliminar o modificar los datos de un usuario, el proceso requiere que los datos de entrada sean válidos. |

| | |
|---|---|
| **Complejidad** | Alta |
| **Prioridad** | Crítico |
| **Precondiciones** | Se ha autenticado como Administrador. |
| **Postcondiciones** | El Administrador gestionó los datos del usuario. |

**Flujo de eventos**

**Flujo básico: Gestionar Usuario.**

| | **Actor** | **Sistema** |
|---|---|---|
| 1. | Solicita cambiar la información de un usuario del sistema. | Permite realizar varias acciones con un usuario:<br>- Insertar Usuario.<br>- Modificar Usuario.<br>- Eliminar Usuario. |
| 2. | | El sistema cambia los datos del usuario y muestra un mensaje que los datos han sido cambiados satisfactoriamente. |

**Flujos alternos**

1a. Los datos son incorrectos.

| | **Actor** | **Sistema** |
|---|---|---|
| 1. | Entra datos no válidos y solicita cambiar la información de un usuario. | El sistema recibe la solicitud y deniega el proceso mostrando un mensaje de datos no válidos. |

**Sección 1: "Adicionar"**

Flujo básico: Gestionar Usuario

| | **Actor** | **Sistema** |
|---|---|---|

| | | |
|---|---|---|
| 1. | Selecciona la opción adicionar. | Muestra el formulario para entrar los datos del usuario a insertar. |
| 2. | Entra los datos. | Adiciona el usuario y muestra un mensaje que el usuario ha sido adicionado correctamente. |
| **Sección 2: "Eliminar"** | | |
| Flujo básico: Gestionar Usuario. | | |
| | **Actor** | **Sistema** |
| 1. | Selecciona el usuario a eliminar. | Elimina el usuario seleccionado y muestra un mensaje que el usuario ha sido eliminado correctamente. |
| **Sección 3: "Actualizar"** | | |
| Flujo básico: Gestionar Usuario. | | |
| | **Actor** | **Sistema** |
| 1. | Selecciona el usuario al cual desea actualizar los datos. | Muestra el formulario con los datos del usuario seleccionado. |
| 2. | Entra los nuevos datos. | Actualiza los nuevos datos y muestra un mensaje que los datos han sido actualizados correctamente. |

Tabla 4 Descripción del CU Gestionar Usuario.

## 2.8 Conclusiones.

Según lo antes expuesto se concluye que:

- ✓ La relación entre los actores y los casos de uso del sistema propuesto demuestran que se obtendrá una solución que permita buscar y recuperar documentos, gestionar usuarios, controlar el

acceso al sistema, gestionar noticias y ejecutar el indexado de los documentos, todo lo cual resolverá los problemas que originaron la investigación.

- ✓ Los tres actores del sistema identificados servirán como base para implementar el control de acceso mediante roles.

# CAPÍTULO 3: Análisis y diseño de la solución propuesta.

## 3. Introducción.

El presente capítulo está dedicado al análisis y diseño a partir de la descripción y definición de las funcionalidades que se trataron en el capítulo anterior. Se definen los diagramas de clases y diagramas de interacción del análisis que se producen como parte del flujo de procesos, lo que permite un mejor entendimiento y una mayor funcionalidad del sistema que se propone. Además de realizar un análisis de los patrones usados por el framework Symfony para llevar a cabo la realización de los diagramas de clases del diseño con los cuales se obtendrá la vista de diseño estática del sistema. También se presentan los Diagrama Entidad-Relación y el Diagrama de Clases Persistentes los cuales forman parte del diseño de la base de datos

### 3.1 Análisis.

#### 3.1.1 Modelo del Análisis.

El modelo de análisis constituye un modelo que se utiliza para obtener una visión del sistema sobre los requisitos funcionales, expresados en un lenguaje técnico, es el resultado de la actividad de analizar los casos de uso. Durante el análisis, se analizan los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar todo el sistema, incluyendo su arquitectura. El objetivo del análisis es comprender perfectamente los requisitos del software.

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. Estos diagramas son los más utilizados en el modelado de sistemas orientados a objetos por constituir el pilar básico del modelado con UML para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). (Software, 2011)

*Entidad:* Modelan información que posee larga vida y que es a menudo persistente.

*Interfaz:* Modelan la interacción entre el sistema y sus actores.

*Control:* Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso. (Software, 2011)

### 3.1.2 Diagramas de Clases del Análisis.

#### 3.1.2.1 Diagramas de Clases del Análisis CU Gestionar Usuario.

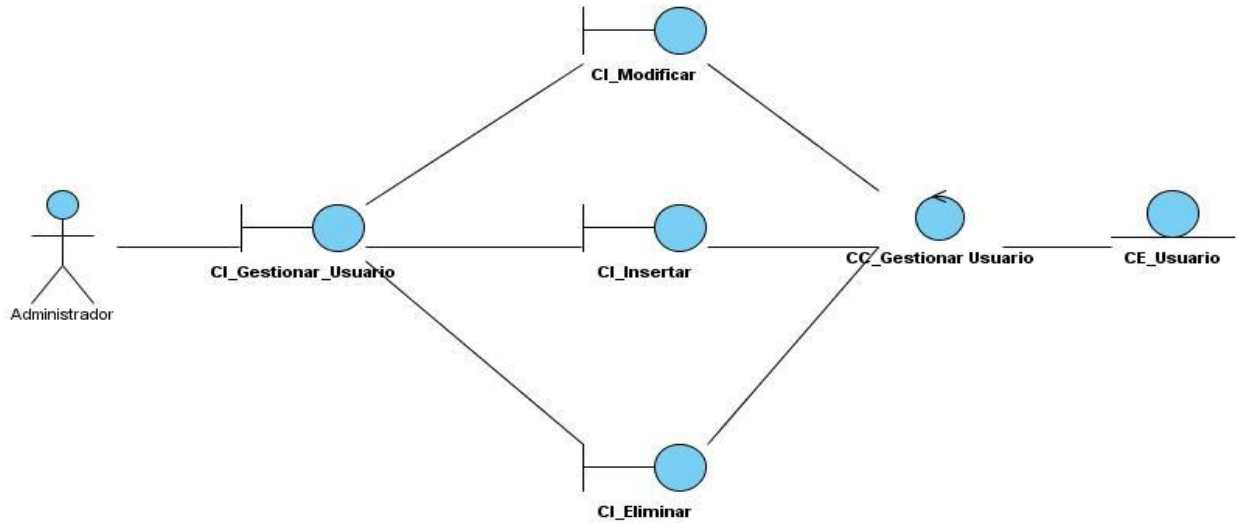


Ilustración 3 Diagramas de Clases del Análisis CU Gestionar Usuario.

### 3.1.3 Diagramas de Interacción.

La ejecución de un software orientado objeto consiste en un encadenamiento de operaciones y cambios de estado de objeto, el cual, a su vez, consiste en que durante la ejecución de una operación o durante una transición se llaman operaciones sobre otros objetos(o sobre el mismo) y se envían señales que provocan otras transiciones. Así, se puede describir el funcionamiento de los casos de uso y de operaciones complejas. En el UML esta acción se lleva cabo mediante los denominados diagramas de interacción. Una interacción es la especificación del comportamiento de un caso de uso u operaciones en términos de secuencia de intercambio de mensajes entre objetos. Estos mensajes contienen estímulos, que pueden ser peticiones de ejecución de operaciones o señales. (Falgueras, 2003). Dentro de los diagramas de interacción se encuentran los diagramas de secuencia y diagramas de colaboración.

#### 3.1.3.1 Diagramas de Colaboración del Análisis.

Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica. Un uso de un diagrama de colaboración es mostrar la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes. Cuando se implementa el comportamiento, la secuencia de los mensajes corresponde a la estructura de llamadas anidadas y el paso de señales del programa.

##### 3.1.3.1.1 Diagrama de Colaboración del Análisis. CU Realizar Búsqueda.

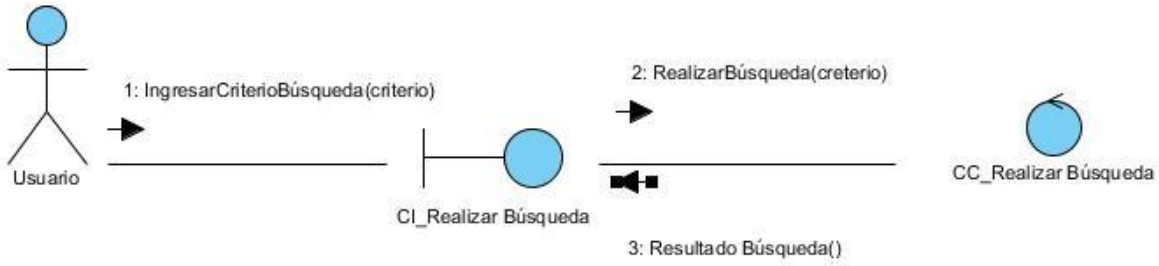


Ilustración 4 Diagrama de Colaboración del Análisis. CU Realizar Búsqueda.

#### 3.1.3.2 Diagrama de Secuencia.

Un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia debajo de la página. La dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración. Cada rol de clasificador se representa mediante una columna vertical-línea de vida. Durante el tiempo que existe un objeto, el rol se muestra por una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble. Se muestra un mensaje como una flecha desde la línea de vida de un objeto a la del otro. Las flechas se organizan en el diagrama en orden cronológico hacia abajo. (Software, 2011).

### 3.2 Diseño.

El diseño es donde manda la creatividad, donde los requisitos del cliente, las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. El diseño crea una

representación o modelo de software, pero a diferencia del modelo de análisis (que se enfoca en la representación de los datos, las funciones y el comportamiento requerido), el modelo de diseño proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema. (Pressman, 2001)

### 3.2.1 Patrones.

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Además, no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. (Gracia, 2005)

Los patrones de diseño pueden usarse durante el diseño del software. Una vez que se ha desarrollado el modelo de análisis, el diseñador puede examinar una representación detallada del problema que debe resolver y las restricciones que impone el problema. (Pressman, 2001)

*Patrones arquitectónicos:* estos patrones definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes del software y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, componentes, subsistemas) de la arquitectura. (Pressman, 2001)

*Patrones de diseño:* estos patrones se aplican a un elemento específico del diseño como un agregado de componentes para resolver algún problema de diseño relaciones entre los componentes o los mecanismos para efectuar la comunicación de componente a componente. (Pressman, 2001)

#### 3.2.1.1 Patrones de Diseño en Symfony.

Los patrones de diseño se dividen en dos grandes grupos los “General Responsibility Assignment Software Patterns” (GRASP) y los “Gang of Four” (GOF).

**Patrones GRASP.**

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, además constituyen un apoyo para entender el diseño y aplica el razonamiento para el diseño de una forma sistemática, racional y aplicable. En Symfony se aplican los cinco patrones que conforman este grupo y son con los que interactúan mayormente tanto diseñadores como programadores. A continuación

se describe la solución que brinda estos patrones, el problema que resuelve y en qué partes del framework se puede evidenciar su uso.

**Alta Cohesión:** Asignar una responsabilidad de manera que la cohesión permanezca alta. ¿Cómo mantener la complejidad manejable? (Larman, 1999) Permite una alta colaboración entre las clases de un sistema. Symfony permite asignar responsabilidades a una clase con una alta cohesión siempre que se realice un trabajo de acuerdo con la estructura planteada por el *framework*, en el cual debe existir una acción para cada una de las plantillas. Las acciones contenidas en cada una de las clases poseen una fuerte relación, teniendo un sentido común y donde su mayor responsabilidad radica en definir las acciones para cada una de las plantillas de un módulo. Permitiendo así que dichas clases sean flexibles y aporten un alto nivel de modularidad a los sistemas que se implementen con el *framework* en cuestión.

**Bajo Acoplamiento:** Asignar una responsabilidad de manera que el acoplamiento permanezca bajo. ¿Cómo soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización? (Larman, 1999) Permite mantener un bajo nivel de dependencias entre las clases que conforman un sistema y una alta probabilidad de que puedan ser reutilizadas. Symfony cumple con este patrón de diseño casi en su totalidad, debido que el nivel de dependencia entre las clases que contienen las acciones es muy bajo por no decir que ninguno. Esto es posible gracias a la sintaxis y la propia estructura del *framework*, dando al traste con el patrón anterior. Las acciones son métodos con el nombre executeNombreAccion de una clase llamada nombreMóduloActions que hereda de la clase sfActions y se encuentran agrupadas por módulos. Lo cual brinda gran nivel de reutilización a cada uno de los módulos que se implemente en una aplicación. Además, si se quiere obtener la reutilización entre acciones, se puede utilizar una sintaxis alternativa para distribuir las acciones en archivos separados. En este caso, cada clase acción extiende de sfAction y su nombre es nombreAccionAction. El nombre del método es simplemente execute. Mientras el nombre del archivo es el mismo que el de la clase a diferencia del anterior que es action.class.php.

**Creador:** Se encarga de asignar a la clase B la responsabilidad de crear una instancia de clase A. ¿Quién debería ser el responsable de la creación de una nueva instancia de alguna clase? (Larman, 1999) Este patrón se puede evidenciar en las clases que contienen las acciones, las cuales son encargadas de instanciar a las clases del modelo que representan las entidades. Su uso también se puede percibir en las clases del modelo, las cuales son generadas automáticamente por cualquiera de los ORM utilizados por el *framework*, y aunque el mismo es transparente al programador, se puede apreciar cuando se debe crear una clase B que agrega, contiene, registra instancias, o tiene los datos de inicialización que se pasarán a

otro objeto A. El caso típico es cuando se desea guardar una tupla en una tabla de la base de datos que tiene relación de uno a muchos con otra tabla, la cual es mapeada como una relación de agregación, donde sólo es necesario crear un sólo objeto específicamente de la clase de uno y esta será la responsable de crear la de muchos.

**Controlador:** Asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. ¿Quién debe ser el responsable de gestionar un evento de entrada al sistema? (Larman, 1999) Este patrón facilita la centralización de actividades, no porque las realice sino porque las delega en otras clases con las que mantiene un modelo de alta cohesión. Todas las peticiones web en el *framework* son manejadas por un sólo controlador frontal, que es el punto de entrada único de toda la aplicación. El controlador frontal es imprescindible en Symfony, se encarga de numerosas tareas entre las que se encuentran: crear la configuración de la aplicación y el contexto; cargar e inicializar las clases del núcleo del *framework*; decodificar la URL de la petición para determinar la acción a ejecutar y los parámetros de la petición; ejecutar los filtros; ejecutar la acción y producir la vista. Las acciones también hacen uso de este patrón conteniendo la lógica de la aplicación; verifican además la integridad de las peticiones y preparan los datos requeridos por la vista utilizando el modelo y definiendo las variables para la vista, por lo que se puede decir que son el corazón de cada aplicación. Este patrón se encuentra ejemplificado en las clases sfFrontController, sfActions, así como todos los archivos actions.class.php que son creados en cada uno de los módulos de la aplicación.

**Experto:** Asignar una responsabilidad al experto en información la clase que tiene la información necesaria para realizar la responsabilidad. ¿Cuál es el principio general para asignar responsabilidades a los objetos? (Larman, 1999) Su uso se encuentra enmarcado fuertemente en el mapeo y abstracción de la base de datos, se puede reflejar siempre que se generan de forma automática ambas capas del modelo, es uno de los patrones más utilizados, debido a que tanto Propel como Doctrine, generan las clases de acceso a datos asignándoles las responsabilidades de ejecutar todas las funcionalidades comunes de las entidades que representan y de la cual poseen información. Es válido mencionar que este patrón suele ser usado por el programador, pero es el propio *framework* quien se encarga de implementarlo a través de los dos ORM de los que se puede hacer uso.

**Patrones GOF.**

A principios de los años 90 con la publicación del libro *Design Patterns*, se establecen veintitrés patrones de diseño GOF. Los patrones GOF según su propósito se clasifican en tres grupos. Los Creacionales tratan la creación de instancias. Los Estructurales tratan la relación entre clases, la combinación clases y la formación de estructuras de mayor complejidad. Mientras que los de Comportamiento tratan la interacción y cooperación entre clases. A su vez según su ámbito se pueden clasificar de Clase, basados en la herencia de clases y de Objetos basados en la utilización dinámica de objetos. Seguidamente se relacionan algunos de los patrones GOF que se pueden encontrar implementados en el interior de Symfony.

**Solitario (*Singleton*):** Patrón creacional a nivel de objetos. Su propósito es garantizar que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma. (scribd, 2010) Este patrón es utilizado en la clase sfContext la cual almacena una referencia a todos los objetos que forman el núcleo de Symfony. Debido que se encarga de enrutar todas las peticiones que se realizan a la aplicación.

**Fábrica Abstracta (*Abstract Factory*):** Patrón creacional a nivel de objetos. Su propósito es proporcionar una interfaz para la creación de familias de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. (scribd, 2010) Su aplicación se puede evidenciar en la clase sfContext utilizada cuando el *framework* necesita crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea, de esta forma, se simplifica la decisión abstrayéndola dentro de una clase especializada.

**Decorador (*Decorator*):** Patrón estructural a nivel de objetos. Su propósito es añadir responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. (scribd, 2010) Es aplicado a las vistas donde el contenido de la plantilla se integra en el *layout*, o si se mira desde el otro punto de vista, el *layout* decora la plantilla. De manera que el *layout* contiene el código HTML y los elementos comunes a todas las páginas, mientras las plantillas se encargan de mostrar el resultado de las acciones. Las clases sfView, sfFilter, y sfWidgetFormSchemaDecorator implementa dicho patrón.

**Fachada (*Facade*):** Patrón estructural a nivel de objetos. Su propósito es proporcionar una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La fachada satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas.

(scribd, 2010) Se evidencia en la clase sfConfig, la cual crea un objeto que proporciona métodos estáticos para poder acceder a los parámetros de configuración desde cualquier punto de la aplicación.

**Acción (*Command*):** Patrón de comportamiento a nivel de objetos. Su propósito es encapsular en un objeto la acción que satisface una petición, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. (scribd, 2010) Su aplicación se evidencia en la clase sfFrontWebController. Esta clase es la encargada de determinar cuál módulo y acción deben responder a las solicitudes de los usuarios. Encapsula las peticiones en forma de objetos permitiendo así parametrizar los clientes utilizando distintas peticiones, encolar las peticiones y ofrecer la posibilidad de deshacer las operaciones.

**Cadena de Responsabilidades (*Chain of Responsibility*):** Patrón de comportamiento a nivel de objetos. Su propósito es proporcionar a más de un objeto la capacidad de atender una petición, para así evitar el acoplamiento con el objeto que hace la petición. Se forma con estos objetos una cadena, en la cual cada objeto satisface la petición o la pasa al siguiente. Su aplicación se encuentra reflejada en la clase sfEventDispatcher, la cual rige el funcionamiento casi completo del *framework*.

- ✓ **Patrón de arquitectónico Modelo Vista Controlador.**

El patrón de arquitectura conocido como Modelo-Vista-Controlador (MVC), separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario; es decir separa en tres capas diferentes los datos de una aplicación, la interfaz de usuario, y la lógica de control:

*Modelo*: Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

*Vista*: Esta capa maneja la visualización de la información, es decir, que presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario.

*Controlador*: Esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos[18], usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases.

[18] Es un evento de alto nivel generado por un actor externo; es un evento de entrada externa.

(Colectivo, 2008-2009). Symfony realiza su propia implementación del MVC, para ello toma lo mejor de esta arquitectura logrando que el desarrollo de las aplicaciones sea más rápido y sencillo.

Normalmente, el controlador realiza muchas tareas en las aplicaciones web por lo que suele tener mucho trabajo. Por este motivo el controlador de Symfony se divide en un controlador frontal que es único para cada aplicación y las acciones que incluyen el código específico del controlador de cada página. El controlador frontal ofrece un punto de entrada único para toda la aplicación y se encarga de las tareas comunes como son, el manejo de las peticiones del usuario, de la seguridad y cargar la configuración de la aplicación entre otras tareas similares.

### 3.2.2 Diagrama de Clases del Diseño.

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Los diagramas de clases también son la base para los diagramas de componentes. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. Después de realizar un estudio del framework Symfony, donde se analizaron los elementos principales de su arquitectura se ha decidido sólo representar los elementos considerados esenciales, con los que se van a relacionar el programador para la implementación de cada una de las funcionalidades, los cuales se describen a continuación.

**Modelo, Vista, y Controlador:** representan paquetes lógicos en correspondencia con la arquitectura del framework.

**Doctrine:** representa de manera lógica el ORM (por sus siglas en inglés *"Object-Relational Mapping"*) seleccionado para la implementación del nuevo sistema por el uso de versiones más actuales del framework.

**Symfony:** subsistema estereotipado que representa todos los patrones de diseño inmersos en el interior del mismo, así como las clases y elementos que permiten su funcionamiento.

**index:** este elemento es único y representa el controlador frontal, es creado de manera automática para toda la aplicación, todo el flujo de trabajo pasa a través del mismo, considerándolo de suma importancia para el funcionamiento del framework y de ahí el motivo de su representación.

**[Nombre]Actions:** representa las nuevas clases que se deben crear para contener las acciones que controlan cada una de las funcionalidades del nuevo módulo.

### 3.2.3 Diseño de la Base de Datos.

El correcto funcionamiento de la aplicación depende también, del buen diseño y funcionamiento de la base de datos (BD) a utilizar. En este epígrafe se muestran el Diagrama de Entidad Relación (DER) y el Diagrama de Clases Persistentes (DCP) correspondiente al diseño de la BD.

El DER representa la realidad de la problemática identificada a través de las entidades y de los enlaces que rigen la unión de las mismas y que constituyen la relación del modelo.

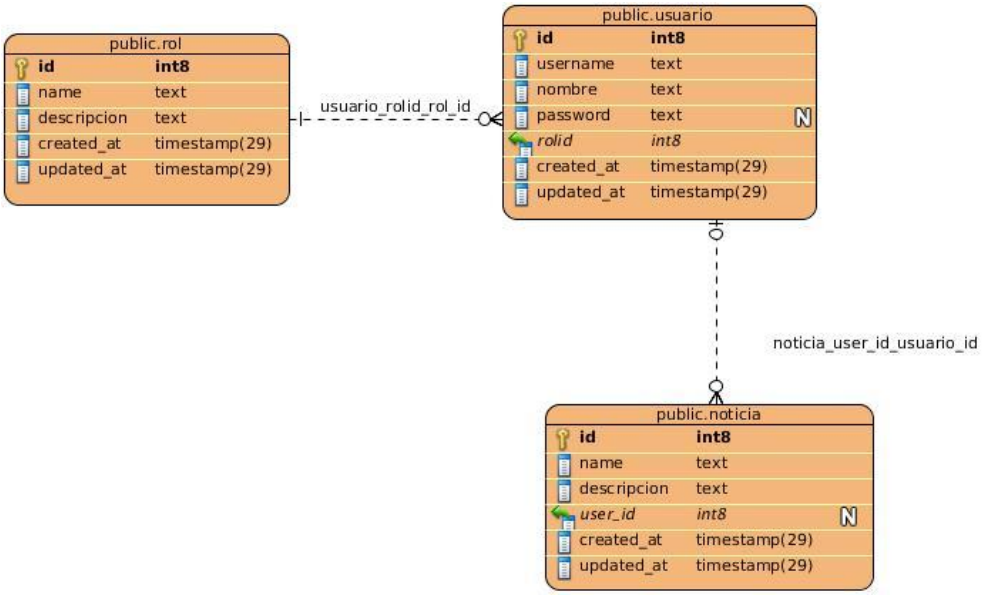


Ilustración 5 Diagrama Entidad Relación.

El DCP muestra las relaciones existentes entre aquellas clases que implementan las entidades del problema de negocio, manteniendo su valor en un espacio y tiempo determinado.

### 3.3 Conclusiones.

Se concluye que:

- ✓ La utilización del framework Symfony en la construcción del sistema permitirá obtener una solución multiplataforma, flexible y que se le puedan añadir nuevas funcionalidades sin necesidad de detenerla.

- ✓ Los diagramas de clases del análisis y del diseño permiten comprender el funcionamiento interno del sistema, lo cual es útil para futuras labores de mantenimiento. Además, constituyen una guía a seguir para ejecutar de la mejor manera posible la fase de implementación.

# CAPÍTULO 4: Implementación y proceso de pruebas al sistema.

## 4. Introducción.

En este capítulo se describe cómo los elementos del modelo del diseño se implementan en términos de componentes. Además de llevar a cabo el proceso de pruebas al sistema, en el cual se expresan las diferentes pruebas que se le pueden realizar a un software. Además de realizar el diseño de los casos de prueba, obteniendo un resultado de estas pruebas realizadas a la aplicación.

### 4.1 Modelo de implementación.

En la implementación se comienza con el resultado del diseño e implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. En el flujo de trabajo de diseño se propone crear un plano del modelo de implementación, por lo que sus últimas actividades están vinculadas a la creación del modelo de despliegue. El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo con los nodos específicos en el modelo de despliegue.

#### 4.1.1 Diagrama de despliegue.

Los diagramas de despliegue describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software. Además de describir la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos.

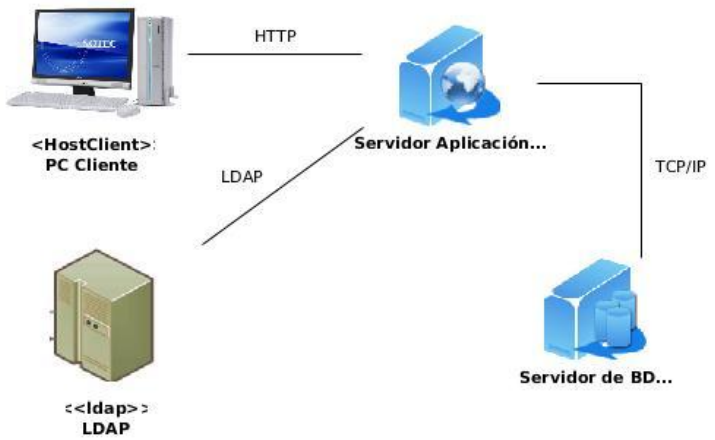


Ilustración 6 Diagrama de Despliegue.

A continuación la descripción de los componentes y protocolos presentes en el diagrama de despliegue.

PC Cliente: Nodo que representa una computadora desde la cual se puede acceder a la aplicación de búsqueda y recuperación de documentos mediante un navegador web, preferentemente Mozilla Firefox[19].

Servidor de BD: Nodo en el cual se estará ejecutando el servidor PostgreSQL con la base de datos del sistema.

Nodo Servidor de Aplicación Web Apache: Este nodo constituye el servidor donde se encuentra hospedada la aplicación para buscar y recuperar documentos. Además, almacena el repositorio de documentos.

Servidor LDAP: Este nodo constituye un servidor LDAP para la autenticación.

*Protocolos:*

LDAP: Protocolo Ligero de Acceso a Directorios que hacen referencia a un protocolo a nivel de aplicación el cual permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

HTTP: Permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MINE[20].

TCP/IP: Son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet, un sistema de protocolos que hacen posibles servicios Telnet, FTP, E-mail, y otros entre ordenadores que no pertenecen a la misma red.

[19] Mozilla Firefox es un navegador web libre y de código abierto.

[20] El tipo MIME (extensiones multipropósito de correo en Internet) es un estándar propuesto por los laboratorios *Bell Communications* en 1991 para ampliar las posibilidades del correo electrónico al incluir la posibilidad de insertar documentos (imágenes, sonidos y texto) en un mensaje.

### 4.1.2 Diagrama de componentes.

Los Diagramas de componentes son diagramas que muestran un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Los diagramas de componentes son usados para modelar el código fuente, la base de datos física, la versión ejecutable y bibliotecas.

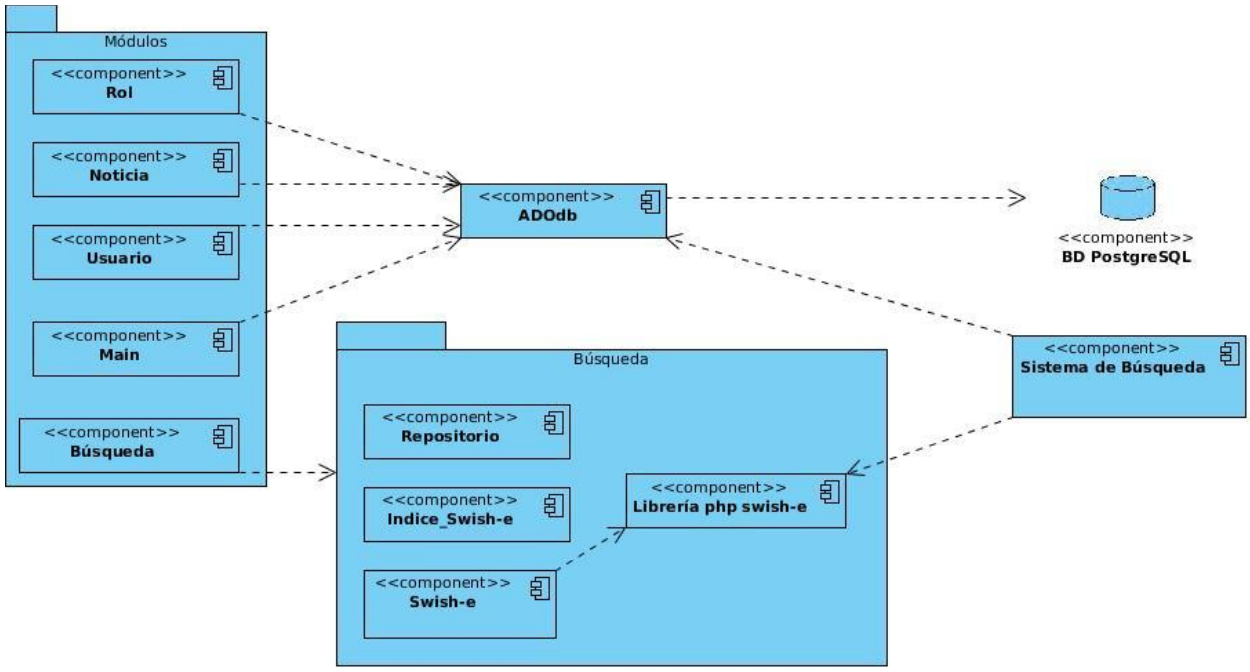


Ilustración 7 Diagrama de Componentes.

## 4.2 Prueba de Software.

Concretamente la prueba de software se puede definir como una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, registrándose los resultados obtenidos. El objetivo de las pruebas no es asegurar la ausencia de defectos en un software, únicamente puede demostrar que existen defectos en el software. El objetivo es pues, diseñar y ejecutar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.(Moreno, 2005)

- ✓ **Pruebas de Aceptación.**

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento

esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

**Técnicas de Prueba:** Como se ha indicado anteriormente, las técnicas de evaluación dinámica o prueba proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas. Estas técnicas se agrupan en:

- ✓ Técnicas de caja blanca o estructural, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.

- ✓ Técnicas de caja negra o funcional, que realizan pruebas sobre la interfaz del programa aprobar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.(Moreno, 2005)

### 4.2.1 Prueba de Caja Negra.

La técnica de prueba escogida para aplicarla a la aplicación es la prueba de Caja Negra la cual 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. La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. (Pressman, 1998)(Beiser, 1995).

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están: (Pressman, 2000).

- ✓ 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.

- ✓ Técnica del Análisis de Valores Límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

- ✓ Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

#### 4.2.1.1 Partición equivalente.

Una partición equivalente es una técnica de prueba de Caja Negra que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada.(Software, 2011)

### 4.2.2 Diseños de Casos de Prueba.

#### 4.2.2.1 Diseño de Caso de Prueba del CU Autenticar Usuario.

- ✓ **Descripción General.**

Caso de uso que define funciones de control de acceso al sistema, inicia cuando un usuario introduce una identidad y solicita entrar al sistema, el proceso requiere que la identidad sea válida para posibilitar el acceso a la interfaz principal del sistema.

- ✓ **Condiciones de Ejecución.**

Debe existir un usuario que desee acceder a la aplicación.

- ✓ **Secciones a probar en el caso de uso.**

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|---|---|---|---|
| SC 1:"Autenticar Usuario." | EC 1.1: Autenticar Usuario con éxito. | El usuario entra una identidad válida, compuesta por un usuario y una clave y solicita entrar al sistema. El sistema recibe la solicitud y comprueba si la identidad es válida verificando que esté compuesta por los datos requeridos y que existe en el sistema, este posibilita el acceso a la interfaz principal asignando la sesión según el rol del usuario. | Interfaz Autenticación. |

| | | | |
|---|---|---|---|
| | EC 1.2: Autenticar Usuario incorrectamente. | Un usuario entra una identidad no válida y solicita entrar al sistema. El sistema muestra un mensaje de error. | Interfaz Autenticación. |

Tabla 5 Secciones a probar en el CC Autenticar Usuario.

- ✓ **Descripción de variable.**

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| *1* | usuario | Campo texto. | No. | Se le introduce el usuario. |
| *2* | password | Campo texto. | No. | Se le introduce la contraseña. |

Tabla 6 Descripción de variable del CU Autenticar Usuario.

- ✓ **Matriz de Datos**

**SC 1 Autenticar Usuario.**

| ID del escenario | Escenario | Variable 1 Usuario | Variable 2 Contraseña | Respuesta del Sistema | Resultado de la Prueba |
|---|---|---|---|---|---|
| EC 1.1 | Autenticar Usuario con éxito. | V | V | El sistema posibilita el acceso a la interfaz principal. Asigna la sesión según el rol del usuario. | Satisfactorio. |
| EC 1.2 | Autenticar Usuario incorrectamente. | V<br>I | I<br>V | Muestra un mensaje de error | Satisfactorio. |

Tabla 7 Matriz de Datos. SC 1 Autenticar Usuario.

### 4.3 No conformidades detectadas.

Caso de uso Ejecutar Indexación: Lleva cabo la indexación de los documentos satisfactoriamente, pero nunca muestra un mensaje avisando de que el indexado de los documentos ha finalizado.

### 4.4 Conclusiones.

Se concluye que:

- ✓ Se obtuvo un sistema para buscar y recuperar documentos que permite: aprovechar fácilmente vía web las funcionalidades ofrecidas por la herramienta Swish-e; gestionar usuarios; gestionar noticias y controlar el acceso al repositorio documental del departamento Señales Digitales.

- ✓ Se diseñaron casos de prueba, se ejecutaron, y se comprobó que el sistema cumple con los requisitos planteados, por lo cual está listo para resolver la problemática que originó la investigación.

# CONCLUSIONES GENERALES

Después de haber analizado los resultados obtenidos con la elaboración del presente trabajo y la culminación de las funcionalidades del Sistema de Búsqueda y Recuperación de Información, se llegaron a las siguientes conclusiones:

- ✓ Se desarrolló la versión 1.0 del Sistema de Búsqueda y Recuperación de Información, el cual facilita la búsqueda y recuperación de la información en el repositorio de documentos del departamento Señales Digitales.

- ✓ El sistema desarrollado constituye una interfaz web a la potente herramienta Swish-e, que facilita la interacción con las funcionalidades ofrecidas por dicha herramienta, y la extiende mediante la gestión de noticias y la gestión de usuarios.

- ✓ El sistema desarrollado está basado en software libre por lo cual contribuye a la soberanía tecnológica del país.

## RECOMENDACIONES

Sobre la presente investigación el autor recomienda:

- ✓ Incorporar las siguientes funcionalidades al sistema:
  - Permitir la autenticación en caso de no existir conexión con el LDAP.
  - Permitir la indexación de ficheros que estén fuera del servidor.
  - Mostrar estado en la indexación de los documentos.
  - Permitir la búsqueda con operadores booleanos combinados con términos (" ", +, -, *).
  - Crear un módulo de configuración para que el usuario escoja que operadores va a utilizar para realizar la búsqueda.
- ✓ Diversificar el uso del sistema en la universidad.

## REFERENCIAS BIBLIOGRÁFICAS.


1. [En línea] [Citado el: 28 de noviembre de 2010.] http://www.http-peru.com/postgresql.php.
2. **2008.** [En línea] 2008. [Citado el: 22 de noviembre de 2010.] http://definicion.de/informacion/.
3. [En línea] http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema09.pdf.
4. **Adobe. 2009.** Archivos PDF. [En línea] 14 de 07 de 2009. [Citado el: 26 de 2 de 2011.] http://www.adobe.com/es/products/acrobat/adobepdf.html.
5. **Alma, Enríquez Toledo.** [En línea] [Citado el: 26 de noviembre de 2010.] http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf.
6. **Álvarez, Sara. 2007.** Sistemas gestores de bases de datos. [En línea] 2007. [Citado el: 27 de noviembre de 2010.] http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html.
7. **Bajame. 2007.** Copernic Desktop Search 2.02. [En línea] 2007. [Citado el: 24 de noviembre de 2010.] http://www.bajame.net/Copernic-Desktop-Search-2_02.htm.
8. **Beiser. 1995.** 1995.
9. **Buenastareas. 2011.** Concepto De Documento Y Documento de Archivo. [En línea] 2011. [Citado el: 22 de noviembre de 2010.] http://www.buenastareas.com/ensayos/Concepto-De-Documento-Y-Documento-De/360149.html.
10. **Colectivo. 2008-2009.** Arquitectura y Patrones de diseño. 2008-2009.
11. **Craig, Lerman. 1999.** *UML y Patrones.* México : Prentice-Hall Hispanoamericana, 1999.
12. **Deciencias. 2010.** Características del lenguaje HTML. [En línea] 2010. [Citado el: 3 de diciembre de 2010.] https://belenus.unirioja.es/~guprado/pagweb/carachtml.html.
13. **—. 2010.** Introducción al lenguaje HTML. [En línea] 2010. [Citado el: 2 de diciembre de 2010.] http://www.deciencias.net/disenoweb/elaborardw/paginas/intro_html.htm.
14. **definición. 2008.** [En línea] 2008. [Citado el: 23 de noviembre de 2010.] http://definicion.de/documento/.
15. **Definicionabc. 2009.** [En línea] 2009. [Citado el: 21 de noviembre de 2010.] http://www.definicionabc.com/tecnologia/informacion.php.
16. **Desarrolloweb. 2002.** Qué es Oracle. [En línea] 2002. [Citado el: 28 de noviembre de 2010.] http://www.desarrolloweb.com/articulos/840.php.
17. **Elena, Raja Prado. 2007.** Casi todas las pruebas del software. [En línea] 2007. http://www.sistedes.es/TJISBD/Vol-1/No-4/articles/pris-07-raja-ctps.pdf. 4.
18. **Elies.** Sistema de recuperación de Información en Lenguaje Natural. [En línea] [Citado el: 21 de noviembre de 2010.] http://elies.rediris.es/elies12/cap31.htm.
19. **Falgueras, Benet Campderrich. 2003.** *Ingeniería de Software*. s.l. : Editorial UOC, 2003.
20. **Gómez, Laureano Felipe. 2009.** *MEMORIAS DE PRÁCTICAS EN EL USO DEL INDIZADOR SWISH-E* . 2009.
21. **Google. 2009.** Funciones de Google Desktop. [En línea] 2009. [Citado el: 22 de noviembre de 2010.] http://desktop.google.com/es/linux/features.html.
22. **Gracia, Joaquin. 2005.** Patrones de diseño.Anális y Diseño. Ingeniería de Software. [En línea] 27 de 5 de 2005. http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php.
23. **Ivar Jacobson, Grady Booch y James Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000.

24. **Kernelerror.** [En línea] [Citado el: 13 de octubre de 2010.] http://kernelerror.net/programacion/php/arquitectura-3-capas/.
25. **—. 2009.** Arquitectura 3 Capas. [En línea] 6 de 2 de 2009. [Citado el: 21 de octubre de 2010.] http://kernelerror.net/programacion/php/arquitectura-3-capas/.
26. **Larman, Craig. 1999.** *UML y Patrones Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999.
27. **Linux-party.** [En línea] [Citado el: 28 de noviembre de 2010.] http://www.linux-party.com/modules.php?name=News&file=article&sid=4436/los-siete-mejores-ide-para-php..
28. **Mailxmail. 2010.** Documento. Conceptos y tipos de documentos. [En línea] 2010. [Citado el: 23 de noviembre de 2010.] http://www.mailxmail.com/curso-archivistica-manejo-organizacion-archivos-manual-archivero/documento-concepto-tipos-documentos.
29. **Marblestation. 2008.** [En línea] 2008. [Citado el: 15 de noviembre de 2010.] http://www.marblestation.com/?p=644.
30. **Mitecnológico.** Concepto de Sistema de Información. [En línea] [Citado el: 23 de noviembre de 2010.] http://www.mitecnologico.com/Main/ConceptoSistemaInformacion.
31. **Molina, Mería Pinto. 2004.** Búsqueda y recuperación de Información. [En línea] 2004. [Citado el: 23 de noviembre de 2010.] http://www.mariapinto.es/e-coms/recu_infor.htm.
32. **Montiel, Patricia Marín. 2002.** excalibur. [En línea] 2002. [Citado el: 1 de octubre de 2010.] http://personales.upv.es/ccarrasc/doc/2001-2002/excalibur/excalibur.htm#_Toc9859546.
33. **Moreno, Ana. 2005.** *Técnicas de Evaluación de Software.* 2005.
34. **NetBeans. 2011.** Bienvenido a NetBeans. [En línea] 2011. http://netbeans.org/index_es.html.
35. **Nodo50.** Lenguaje html. [En línea] [Citado el: 3 de diciembre de 2010.] http://www.nodo50.org/manuales/internet/13.htm.
36. **Pecos, Daniel.** [En línea] [Citado el: 28 de noviembre de 2010.] http://danielpecos.com/docs/mysql_postgres/x15.html.
37. **Pressman. 1998.** 1998.
38. **Pressman, Roger S. 2001.** Ingeniería de Software. *Ingeniería de Software. Un enfoque práctico*. s.l. : Mc Graw Hill, 2001.
39. **Ramón, Tania Cerezo.** Análisi de la herramienta Google Desktop. [En línea] [Citado el: 1 de octubre de 2010.] http://personales.upv.es/ccarrasc/doc/2004-2005/BuscadoresWeb/trabajosrpDEFINITIVO.htm#_Toc104899909.
40. **Raquel, Mariana. 2009.** *Análisis Cualitativo y Cuantitativo de Herramientas de Entorno Visual para Desarrollo Web en PHP aplicado a la EPEC.* Ecuador : s.n., 2009.
41. **Raúl Yanquen, JORGE FORERO,ANDREA PEÑA. 2010.** Tecnología de Redes Universidad de la Salle Sistema de Información. [En línea] 19 de 5 de 2010. [Citado el: 15 de 2 de 2011.] http://tecnologiaorredes.blogspot.com/2010_05_19_archive.html.
42. **Sánchez, María A. Mendoza. 2004.** [En línea] 2004. [Citado el: 25 de noviembre de 2010.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
43. scribd. 2010. *Patrones "Gang of Four"*. Madrid : Universidad Politécnica de Madrid, 2010.
44. **Scripd.** Conceptos Básicos del servidor Web. [En línea] [Citado el: 29 de noviembre de 2010.] http://www.scribd.com/doc/7826492/Conceptos-basicos-Del-Servidor-Web.

45. **Software, Colectivo de Autores Asignatura Ingeniería de. 2011.** Entorno Virtual de Aprendizaje. [En línea] 2011. [Citado el: 15 de 2 de 2011.] http://eva.uci.cu/mod/resource/view.php?id=35381.
46. **Sourceforge. 2005.** [En línea] 2005. [Citado el: 22 de noviembre de 2010.] http://issi.dsic.upv.es/publications/archives/f-1069167248521/actas.pdf.
47. **Sunsite.** tutorial de Java. Arquitectura MVC. [En línea] [Citado el: 1 de diciembre de 2010.] http://sunsite.dcc.uchile.cl/java/docs/JavaTut/Apendice/arq_mvc.html.
48. **Symfony-project.** [En línea] [Citado el: 28 de noviembre de 2010.] http://www.symfony-project.org/.
49. **Tarrillo, Sergio. 2007.** [En línea] 2007. http://geeks.ms/blogs/sergiotarrillo/archive/2007/01/09/8420.aspx.
50. **Torres, Patricio Letelier. 2003.** [En línea] 2003. [Citado el: 15 de noviembre de 2010.] http://oness.sourceforge.net/proyecto/html/ch05s02.html.
51. **Traducegratis. 2011.** Descargar Copernic Desktop Search gratis. [En línea] 2011. [Citado el: 4 de octubre de 2010.] http://descargar.traducegratis.com/es_soft_v_i29412/KOSTENFREIER-DOWNLOAD-VON-COPERNIC-DESKTOP-SEARCH--SPANISCH.htm.
52. **Universidad de San Carlos de Guatemala. 2010.** UML. [En línea] 2010. http://es.scribd.com/doc/2080534/UML.
53. **Webindex. 2006.** [En línea] 2006. [Citado el: 5 de octubre de 2010.] http://www.webindex.com.mx/2006/07/04/swish-e-motor-de-busqueda/.
54. **Wikilearning. 2007.** El concepto de informacion y sus aplicaciones. [En línea] 2007. [Citado el: 23 de noviembre de 2010.] http://www.wikilearning.com/articulo/el_concepto_de_informacion_y_sus_implicaciones/8333-2.
55. **Wordreference. 2005.** Diccionario de la lengua española. [En línea] 2005. [Citado el: 26 de 2 de 2011.] http://www.wordreference.com/definicion/cuneiforme.
56. **Xavier, Ferré Grau.** Principios Básicos de Usabilidad para Ingenieros Software. [En línea] http://is.ls.fi.upm.es/xavier/papers/usabilidad.pdf. 28660.

# BIBLIOGRAFÍA CONSULTADA.


1. **Craig, Lerman. 1999.** *UML y Patrones*. México : Prentice-Hall Hispanoamericana, 1999.
2. **Falgueras, Benet Campderrich. 2003.** *Ingeniería de Software*. s.l. : Editorial UOC, 2003.
3. **Gómez, Laureano Felipe. 2009.** *MEMORIAS DE PRÁCTICAS EN EL USO DEL INDIZADOR SWISH-E* . 2009.
4. **Ivar Jacobson, Grady Booch y James Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.
5. **Larman, Craig. 1999.** *UML y Patrones Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999.
6. **Moreno, Ana. 2005.** *Técnicas de Evaluación de Software*. 2005.
7. **Pressman, Roger S. 2001.** Ingeniería de Software. *Ingeniería de Software. Un enfoque práctico*. s.l. : Mc Graw Hill, 2001.
8. **Raquel, Mariana. 2009.** *Análisis Cualitativo y Cuantitativo de Herramientas de Entorno Visual para Desarrollo Web en PHP aplicado a la EPEC*. Ecuador : s.n., 2009.
9. scribd. 2010. *Patrones "Gang of Four"*. Madrid : Universidad Politécnica de Madrid, 2010.
10. **Software, Colectivo de Autores Asignatura Ingeniería de. 2011.** Entorno Virtual de Aprendizaje. [En línea] 2011. [Citado el: 15 de 2 de 2011.] http://eva.uci.cu/mod/resource/view.php?id=35381.
11. **Universidad de San Carlos de Guatemala. 2010.** UML. [En línea] 2010. http://es.scribd.com/doc/2080534/UML.