

**Universidad de las Ciencias Informáticas**

**Facultad 15**



**Título:** Herramienta para el análisis de información de los subsistemas de Reclutamiento y Reserva.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Autores:** Liudlemi Pérez Díaz

Ernesto Antonio Ruiz Pérez

**Tutor:** Ing. Yenisleidis Ayala Rosales

**Co-tutor:** Tte. Ing. Jacinto Torres Fernández

Ciudad de La Habana, 2010.

Año 52 de la Revolución



*"Recuerden que el eslabón más alto que puede alcanzar la especie humana es ser revolucionario."*

*Ernesto Che Guevara*

**DECLARACIÓN DE AUTORÍA**

Declaramos que somos los únicos autores de este trabajo y autorizamos al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Liudlemi Pérez Díaz

\_\_\_\_\_  
Cde. Ernesto Antonio Ruiz Pérez

\_\_\_\_\_  
Ing. Yenisleidis Ayala Rosales

**Tutor:** Ing. Yenisleidis Ayala Rosales.

**Breve Currículo:**

*Ingeniero en Ciencias Informáticas. Graduada en julio de 2007. Analista y Líder de gestión de la Línea Capital Humano de la Unidad de Compatibilización e Integración y Desarrollo de Software para la Defensa (UCID) - Universidad de la Ciencias Informáticas.*

**Correo Electrónico:** [yeayala@uci.cu](mailto:yeayala@uci.cu)

**Años de graduado:** 3

**Años de experiencia en el tema:** 4

**Co-tutor:** Tte. Ing. Jacinto Torres Fernández

**Breve Currículo:**

*Ingeniero en Ciencias Informáticas, graduado en julio de 2007. Líder del proyecto Comités Militares de la Unidad de Compatibilización, Integración y Desarrollo de Software para la Defensa (UCID).*

**Correo Electrónico:** [jtfernandez@uci.cu](mailto:jtfernandez@uci.cu)

**Años de graduado:** 3

**Años de experiencia en el tema:** 3

### **Compartidos:**

*Primeramente a la Revolución, sin ella no habiéramos tenido la posibilidad de estudiar en esta universidad.*

*A nuestra tutora Yeni por su paciencia, dedicación, por todo lo que nos has enseñado, por ser como una amiga para nosotros.*

*A Jacinto por ser casi un tutor para nosotros y por la ayuda que siempre encontramos cuando acudimos a él.*

*A Asdrubal y Lisdanay por estar siempre dispuestos a ayudarnos cada una de las de veces que nos hizo falta su ayuda.*

*A Palma por guiarnos en el desarrollo de la tesis, por su experiencia, por su amistad.*

*A todos los compañeros del proyecto en especial a Noel, Alain, Orestes, Doleynis, Arasay que siempre nos apoyaron y nos dieron ánimos para seguir adelante.*

*A todos nuestros compañeros y profesores de toda una vida de estudiante.*

*A Nadín por ser más que una amiga para nosotros, por estar siempre pendiente de nuestras preocupaciones y por brindarnos su experiencia.*

### **De Liudlemi:**

*A mis padres, por su apoyo incondicional y amor infinito, sin ellos no hubiera podido estar aquí en estos momentos.*

*A mi hermana por ser un ejemplo a seguir para mí, por demostrarme que sí se puede y por ser la mejor hermana del mundo.*

*A mis abuelos que de una forma u otra han ayudado a que estudie y me forme como una profesional.*

*A mi tía María Elena por ser como una segunda madre para mí.*

*A mis primos Marlis y Alexander, por ser más que primos hermanos.*

*A mi novio por su apoyo incondicional por darme todo su amor, por estar siempre a mi lado en los buenos y malos momentos.*

*A mi amiga María Cristina, por siempre estar ahí cuando la he necesitado, por ser la mejor amiga.*

*A todas mis compañeras de cuarto a lo largo de estos 5 años en que hemos estado lejos y que han sabido ser como una familia para mí.*

### ***De Ernesto:***

*A mis padres por su apoyo incondicional, por su ejemplo.*

*A la UCI por darme la preparación para formarme como ingeniero informático.*

*A toda mi familia de que siempre me ha apoyado y alentado en los momentos difíciles de la vida de estudiante, en especial a mis abuelos.*

*A todos mis amigos de la UCI, en especial a Noel, Mena, Damián y Alain.*

*Y a mi novia por su comprensión, por alentarme cuando más presionado me sentía, por su amor, su cariño, por que con ella aquí mi vida ha sido mejor.*

### ***De Liudlemi:***

*Dedico este trabajo de diploma a mis queridísimos padres, por siempre guiarme por el buen camino, su enorme sacrificio para que saliera adelante y por confiar en mí en todo momento. A mi maravillosa hermana, por ser un ejemplo para mí y a mi sobrino Leosdany. A todos mis familiares, por todo su apoyo y cariño.*

### ***De Ernesto:***

*Se lo dedico a mis padres, a mis hermanos Amalia y Enrique Alejandro que espero les sirva de ejemplo para que estudien mucho y se superen; y a mis abuelos. A toda mi familia de forma general.*

### **Resumen**

La concepción de una sociedad basada en la información y el conocimiento impone una dinámica constante sobre la práctica y la teoría, tanto de la información como de la comunicación. Los nuevos avances han causado una conmoción en los métodos de procesamiento y recuperación de la información. El análisis de información se centra en el análisis de contenido en un contexto específico, produce información para la toma de decisiones y está condicionada por la calificación, inteligencia y creatividad del analista.

El presente trabajo tiene como objetivo el desarrollo de una herramienta para procesar los datos almacenados de los subsistemas de Reclutamiento y Reserva, que permita el análisis de la información, que contribuye al proceso de toma de decisiones.

La aplicación se desarrolló utilizando el lenguaje de programación PHP, junto a los frameworks ExtJS, ZendFramework y Doctrine mediante un conjunto de herramientas como NetBeans IDE, SQL Manager for PostgreSQL, el servidor web Apache y como navegador web Mozilla Firefox. Para el desarrollo de la herramienta informática se empleó como metodología el Proceso de Desarrollo y Gestión de Proyectos de Software. Los artefactos se generaron usando como lenguajes de modelado la Notación para el Modelado de Procesos de Negocio (BPMN) y el Lenguaje Unificado de Modelado (UML), auxiliados por el Visual Paradigm como herramienta de Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés).

**PALABRAS CLAVE:** Herramienta, Análisis, Toma de decisiones, PHP.



## ÍNDICE DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
1.1. <i>GESTIÓN</i> .....	4
1.2. <i>SOFTWARE DE GESTIÓN</i> .....	4
1.3. <i>HERRAMIENTAS PARA EL ANÁLISIS DE INFORMACIÓN EN EL MUNDO Y EN LA UCI</i> .....	4
1.4. <i>TECNOLOGÍAS</i> .....	7
1.4.1. <i>Arquitectura Cliente-Servidor</i> .....	7
1.4.2. <i>Servidor web</i> .....	8
1.4.3. <i>Aplicaciones web</i> .....	9
1.4.4. <i>Navegador web</i> .....	9
1.4.5. <i>Lenguajes de programación web</i> .....	10
1.4.6. <i>Proceso de desarrollo de software</i> .....	12
1.4.7. <i>Lenguajes de modelado</i> .....	15
1.4.8. <i>Herramientas CASE</i> .....	16
1.4.9. <i>Sistema Gestor de Bases de Datos (SGBD)</i> .....	16
1.4.10. <i>Frameworks</i> .....	17
1.4.11. <i>Control de versiones</i> .....	18
1.5. <i>CONCLUSIONES</i> .....	19
<b>CAPÍTULO 2: CARACTERÍSTICAS DE LA HERRAMIENTA</b> .....	<b>20</b>
2.1. <i>OBJETO DE ESTUDIO</i> .....	20
2.1.1. <i>Situación problemática y problema</i> .....	20
2.2. <i>OBJETO DE AUTOMATIZACIÓN</i> .....	21
2.3. <i>INFORMACIÓN QUE SE MANEJA</i> .....	21
2.4. <i>PROPUESTA DE SISTEMA</i> .....	22
2.5. <i>MODELO CONCEPTUAL</i> .....	23
2.6. <i>DEFINICIÓN DE REQUISITOS</i> .....	24
2.6.1. <i>Requisitos funcionales</i> .....	24
2.6.2. <i>Requisitos no funcionales</i> .....	25
2.7. <i>DESCRIPCIÓN DE REQUISITOS FUNCIONALES</i> .....	28
2.7.1. <i>Requisito funcional Gestionar concepto</i> .....	28

2.7.1.1. Especificación del requisito Adicionar concepto.....	28
2.7.1.2. Especificación del requisito Eliminar concepto.....	30
2.7.2. <i>Requisito funcional Configurar Campos.....</i>	31
2.7.2.1. Especificación del requisito Crear campo.....	31
2.7.3. <i>Requisito funcional Gestionar Recuperaciones.....</i>	32
2.7.3.1. Especificación del requisito Diseñar recuperación.....	33
2.7.3.2. Especificación del requisito Cargar recuperación.....	35
2.7.4. <i>Requisito funcional Gestionar reporte.....</i>	36
2.7.4.1. Especificación del requisito Diseñar reporte.....	37
2.8. <i>CONCLUSIONES.....</i>	39
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA.....</b>	<b>40</b>
3.1. <i>PATRONES UTILIZADOS.....</i>	40
3.2. <i>DISEÑO.....</i>	41
3.2.1. <i>Diseño de clases.....</i>	41
3.2.1.1. Diagramas de clases web.....	41
3.2.1.2. Diagrama de interacción.....	47
3.3. <i>ARQUITECTURA.....</i>	51
3.4. <i>DISEÑO DE DATOS.....</i>	53
3.4.1. <i>Diagrama entidad relación de la BD.....</i>	53
3.5. <i>DEFINICIONES DE DISEÑO.....</i>	53
3.6. <i>TRATAMIENTO DE ERRORES.....</i>	54
3.7. <i>SEGURIDAD.....</i>	54
3.8. <i>INTERFAZ.....</i>	55
3.9. <i>CONCEPCIÓN DE LA AYUDA.....</i>	55
3.10. <i>CONCLUSIONES.....</i>	56
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS DE LA HERRAMIENTA.....</b>	<b>57</b>
4.1. <i>IMPLEMENTACIÓN.....</i>	57
4.1.1. <i>Diagrama de componentes.....</i>	57
4.1.2. <i>Matriz de integración de componentes.....</i>	58
4.1.2.1. Matriz de Integración de Componentes Interna.....	58
4.1.2.2. Matriz de Integración de Componentes Externa.....	59

4.2. PRUEBAS.....	60
4.2.1. Diseño de casos de prueba.....	60
4.3. CONCLUSIONES.....	61
<b>CONCLUSIONES.....</b>	<b>62</b>
<b>RECOMENDACIONES.....</b>	<b>63</b>
<b>BIBLIOGRAFÍA.....</b>	<b>64</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>65</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>67</b>

## **Introducción**

Cuba ha ido desarrollando la ciencia y las tecnologías de la manera en que predijera Fidel hace 50 años que nuestro futuro debería ser necesariamente de hombres de ciencia, de hombres de pensamiento. Durante las dos últimas décadas en nuestro país, el desarrollo tecnológico muestra una convergencia cada vez mayor entre la Informática, las Telecomunicaciones, la Electrónica y la Automatización, proceso que ha devenido una nueva rama del saber denominada Tecnologías de la Información (las cuales son un conjunto de herramientas y métodos empleados para recabar, retener, manipular o distribuir información), de alta incidencia en la modernización y eficiencia de todos los sectores de la sociedad. Dentro de las aplicaciones de las tecnologías de la información están los procesos de análisis de la información, estrechamente ligado a la toma de decisiones.

Las Fuerzas Armadas Revolucionarias (FAR) no ha estado ajena a este empeño nacional por el desarrollo de la informática que contribuye al soporte científico de la defensa nacional, un ejemplo de este desarrollo es la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa (UCID), con el objetivo de fortalecer la capacidad defensiva del país, y para perfeccionar los sistemas de la dirección y el mando a todos los niveles de las FAR. Dentro de la amplia gama de productos desarrollados en la UCID se encuentra DATAFAR, un sistema para la gestión de la información del potencial humano en los Comités Militares, que cuenta con 4 subsistemas: Reclutamiento, Reserva, Medios y Equipos, y Movilización para tiempo de paz y tiempo de guerra; siendo de interés para este trabajo los procesos que se desarrollan en los dos primeros subsistemas.

Actualmente en la aplicación DATAFAR las recuperaciones que se hacen son pocas y de forma estática, o sea, los modelos de presentación de los datos son siempre iguales y están diseñados de antemano. Cuando se hace necesario crear un nuevo tipo de reporte el equipo de desarrollo tiene que dedicar tiempo y esfuerzo adicional por lo cual los usuarios tienen que esperar por la nueva funcionalidad.

La información que brindan los subsistemas Reclutamiento y Reserva es la base del proceso de toma de decisiones, esta depende del entorno ya que los datos obtenidos no serán los mismos en distintas áreas, ejemplo de esto es que en un comité militar municipal no se presenta igual situación demográfica que en un comité militar provincial; también depende del enfoque que se le vaya a dar, puede ser que la decisión esté determinada por la preparación técnico-profesional del personal; y depende del momento en que se

realice, ya sea en tiempo de guerra o tiempo de paz. Con la amplia información existente y con las muchas vías de obtenerla resulta prácticamente imposible saber qué es lo que necesitan los usuarios.

En este momento la información se extrae a partir de datos generales y en muy pocos casos sobre datos específicos, pero es más relevante para el usuario obtener datos específicos que le puedan dar una información más detallada.

Partiendo de esta situación, se identifica el siguiente **problema a resolver**: ¿Cómo facilitar el análisis de información en los subsistemas de Reclutamiento y Reserva?

Se define como **objeto de estudio** los procesos de gestión de la información del potencial humano en los Comités Militares, identificando como **campo de acción** los procesos de análisis de información en los subsistemas de Reclutamiento y Reserva.

De esta forma, se define como **objetivo general** del presente trabajo de diploma, desarrollar una herramienta informática que facilite el análisis de la información de los subsistemas de Reclutamiento y Reserva.

Se plantea la siguiente **idea a defender**: Si se desarrolla una herramienta informática para procesar los datos almacenados en los subsistemas de Reclutamiento y Reserva, entonces se facilitará el análisis de la información que contribuye al proceso de toma de decisiones.

Con vista a dar cumplimiento al objetivo planteado, se definen los siguientes **objetivos específicos**:

- Realizar el diseño teórico de la investigación.
- Elaborar el estado del arte.
- Realizar el diseño de la solución.
- Implementar la herramienta de análisis de la información.
- Realizar pruebas de caja negra a la aplicación.

## **Estructura de los capítulos**

En el **Capítulo 1** se hace un estudio del estado actual de las herramientas utilizadas en el análisis de la información tanto en el mundo como en la universidad. Se explican las tendencias, tecnologías y metodologías usadas en el mundo para el desarrollo de las aplicaciones de gestión y los reportadores.

En el **Capítulo 2** se realiza el modelado del negocio, se obtienen los artefactos correspondientes a la modelación del negocio, se describen las características del sistema y se realiza una descripción de la solución propuesta, especificando los requerimientos funcionales y no funcionales.

En el **Capítulo 3** se realiza el diseño detallado de la solución a través de los diferentes artefactos propuestos por la metodología de desarrollo de software, que darán paso a la implementación. Específicamente se realizan los diagramas de clases web por cada paquete de requisitos y los diagramas de secuencias por cada uno de los escenarios de los requisitos, además se realiza el modelo de datos.

En el **Capítulo 4** se desarrollan los artefactos asociados a la implementación y las pruebas donde se obtuvo el diagrama de componentes que ilustra los fragmentos de software, también se obtienen los diseños de casos de prueba que permiten demostrar que las funcionalidades del software son operativas.

## Capítulo 1: Fundamentación Teórica.

En este capítulo se sustenta teóricamente el desarrollo de este trabajo, presentando el resultado del estudio del estado actual de las herramientas utilizadas en el análisis de la información tanto en el mundo como en la Universidad de las Ciencias Informáticas (UCI). Se explican las tendencias y tecnologías usadas en el mundo para el desarrollo de las aplicaciones de gestión y los reportadores, así como el proceso de desarrollo de software a utilizar elaborado en la UCID.

### 1.1. Gestión

El término gestión está asociado a las actividades encaminadas a dar solución a un problema cualquiera. Gestionar es también dirigir o administrar una empresa, un proyecto o un negocio. Se entiende también por gestión “la actividad dirigida a obtener y asignar los recursos necesarios para el cumplimiento de los objetivos de la organización” (1).

Entre sus funciones están: planificar que determina el qué se va a hacer, cómo, cuándo, dónde, quiénes van a desarrollar determinada acción; organizar las actividades necesarias para desarrollar los planes; dirigir, función donde se dan instrucciones y se asignan responsabilidades; y controlar, es velar que se cumpla lo planificado con el mayor grado de calidad posible.

### 1.2. Software de gestión

Se definen como software de gestión aquellas aplicaciones diseñadas para sustituir procedimientos que pueden ser comerciales o administrativos y que son normalmente realizados por una persona de forma presencial en una empresa o institución, mediante una aplicación que posibilita la realización de varios procedimientos a un cliente de forma no presencial, y disminuyendo el esfuerzo de los mismos (2).

### 1.3. Herramientas para el análisis de información en el mundo y en la UCI

Los reportadores o generadores de reportes son herramientas informáticas complementarias en el análisis de información ya que proveen una forma transparente al usuario para realizar consultas a la base de datos y obtener información de ella en forma de reporte y presentarlos o exportarlos en muchos formatos. También son conocidos como Sistemas Gestores de Reportes Dinámicos. Muchas empresas los utilizan para recuperar toda la información que necesitan pero no siempre encuentran el software más económico o el que más se adecúe a sus características.

Existen en el mundo variedad de reportadores pero entre los más populares, los más utilizados y los que más funcionalidades brindan están Crystal Report, Oracle Report, iReport, entre otros.

**Crystal Report:** Es una aplicación de inteligencia empresarial utilizada para diseñar y generar reportes, la cual puede conectarse a varias bases de datos y puede utilizarse sobre los sistemas operativos Windows, Linux, Solaris pero aún así pertenece a la familia de software privativo. Su proveedor actual es la compañía SAP, la cual ofrece otros productos como Crystal Reports for Eclipse (3) el cual permite integrar los reportes de Crystal Reports a las aplicaciones desarrolladas en java, y para la plataforma .NET presenta Crystal Reports for Visual Studio .NET. Está orientado fundamentalmente para la recuperación de información en las pequeñas empresas. Con Crystal Report es necesario comprar una licencia para cada PC donde se utilice, además los usuarios finales tienen que tener instalado el visor de reportes de Crystal para ver la información si esta no se encuentra en otros formatos como PDF y HTML.

**Oracle Report:** Es un componente de Oracle Fusion Middleware<sup>1</sup>, es una herramienta de diseño de J2EE 5.0 basado en Oracle Reports Server con la arquitectura de múltiples niveles para acceder a cualquier fuente de datos y generar reportes en varios formatos (4). Presenta varias funcionalidades agregadas como en el tema de la seguridad a la hora de exportar los informes según la autenticación, o sea permite la compartimentación de la información. El principal inconveniente de este como de todos los productos de Oracle pese a su calidad es su precio elevado.

**iReport:** Es un diseñador visual de código abierto para crear informes complejos utilizando la biblioteca JasperReports. Los datos para imprimir, se puede adquirir de cualquier tipo de fuente de datos, incluyendo bases de datos, OLAP XML, archivos XML, CSV, colecciones de JavaBeans o incluso de fuentes personalizadas proporcionadas por el usuario. Tiene un soporte integrado para varios lenguajes de consulta como SQL, HQL, XPath, EJBQL y MDX. iReport ofrece un entorno completo para crear y probar las fuentes de datos, diseño de los informes, vista previa y la exportación en todos los formatos soportados, incluyendo PDF, HTML, XML, Word (RTF), Excel, OpenOffice (ODF) y CSV. iReport ayuda a generar los archivos de Jasper que se pueden desplegar en cualquier aplicación Java (web y de aplicaciones de escritorio) para generar reportes ricos en información dinámica en cualquier formato (5).

---

<sup>1</sup> Es un paquete de productos software de Oracle que abarca desde los portales y administradores de procesos hasta la infraestructura de aplicaciones, las herramientas de desarrolladores, y la inteligencia de negocios.



**JasperReports:** Es una librería para la generación de reportes, su código es java y es libre. Su funcionamiento consiste en crear un XML con las particularidades del reporte. Este XML lo tratan las clases del Jasper para obtener una salida en los formatos PDF, XML, HTML, CSV, XLS, RTF, TXT.

Tanto para iReport como para JasperReport existe muy poca documentación sobre su uso. Es un poco complicado el uso de iReport con JasperReport, ya que hay que tener ciertos conocimientos sobre Java para diseñar los reportes, o sea, que su utilización no está orientada hacia usuarios finales.

Todas estas herramientas tienen en común la variedad de bases de datos a las que se pueden conectar y los formatos en que exportan sus informes.

En la UCI también se han desarrollado reportadores como el Generador de Reportes Dinámicos como componente de PATDSI<sup>2</sup> desarrollado por el centro CENTALAD<sup>3</sup>. Este producto está desarrollado sobre una plataforma web con framework de desarrollo Symfony y ExtJS para la interfaz de usuario, además se utiliza como gestor de base de datos PostgreSQL. Sus informes los presenta en diferentes formatos como HTML, Excel, CSV, TSV y PDF. Puede conectarse a varias fuentes de datos, presenta componentes para diseñar consultas SQL, para diseñar y mostrar los reportes. Actualmente no se ha desarrollado un componente que permita representar sobre mapas la información estadística que se genera, y por ser de uso tan general no implementa la compartimentación de la información.

El mismo está instalado en diferentes proyectos de la Universidad de las Ciencias Informáticas (UCI) como son: (6)

- El ERP Cubano.
- Proyecto Alfa Omega (Facultad 8).
- Proyecto Aduana (Facultad 4).
- Además está instalado en la Oficina Nacional de Estadísticas (ONE) de Cuba.

Otro ejemplo de reporteador en la UCI es el reporteador en Java que se encuentra actualmente en desarrollo. Está basado en la librería para generar reportes JasperReport, puede conectarse a varias fuentes de datos. Es fácilmente integrable en la mayoría de los lenguajes, como por ejemplo, PHP, C#.

---

<sup>2</sup> Plataforma de Ayuda a la Toma de Decisiones y Sistemas Inteligentes

<sup>3</sup> Centro de Tecnologías de Almacenamiento y Análisis de Datos

En la UCID se desarrolló el Sistema de Registro y Control de Cuadros (SRCC) usando lenguaje PHP con framework ExtJS para la interfaz de usuario, este sistema cuenta con un componente llamado Recuperaciones Dinámicas Complejas, esto permite recuperar la información que se gestiona en esta aplicación pero con la particularidad que los usuarios finales son los que configuran sus recuperaciones de una manera sencilla e intuitiva. Una de sus características es que genera la consulta PL/PgSQL que luego ejecuta a partir de la selección de los datos y condiciones que hacen los usuarios en la interfaz, pero este proceso se hace de forma estática sobre una configuración de la base de datos. Una vez diseñada una recuperación permite exportarla en el formato RTF.

En la actualidad existen muchas y buenas herramientas para la generación de reportes, las cuales soportan gráficos, imágenes y sub-reportes; permiten exportar las salidas en diferentes formatos, como PDF, HTML, XML, etc., entre otras características de utilidad. Sin embargo, todas tienen algunos inconvenientes que hacen que no se adecúe a las características del negocio, por ejemplo: usando JasperReport, para modificar el reporte es necesario recompilar el proyecto donde se encuentra; o que son software propietario, lo que implica alto costo de sus licencias, tal es el caso del Crystal Report y Oracle Report. Por otro lado, las opiniones que se han encontrado respecto al sistema libre para la generación de reportes JasperReport no son muy buenas, pues aparte de que no están bien documentados, no son lo suficientemente robustos y flexibles para satisfacer los requerimientos de reportes complejos. Con estas aplicaciones no se le da solución al problema planteado, por tanto se hace necesario desarrollar una herramienta para el análisis de información que facilite la toma de decisiones tomando las ideas fundamentales de cada una de estas para lograr al final un producto que tenga además otras características como que permita compartimentar la información según los niveles de acceso; que exporte información estadística que pueda ser representada geoespacialmente, que sea orientada al usuario, o sea que los usuarios finales puedan diseñar y obtener las recuperaciones sin tener conocimientos de programación o de bases de datos.

## **1.4. Tecnologías**

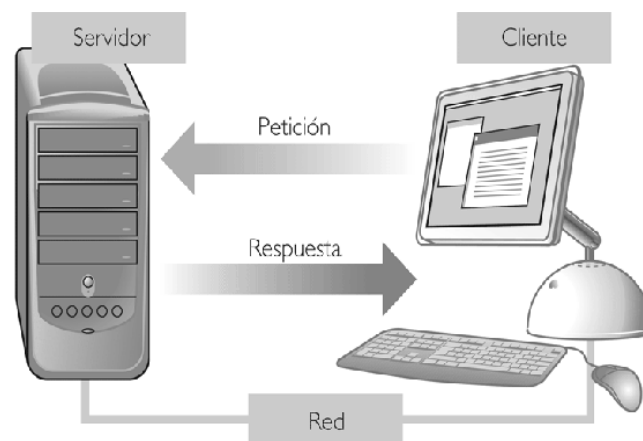
### **1.4.1. Arquitectura Cliente-Servidor**

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que solicita los recursos y servidor al

proceso que responde a las peticiones. Es el servidor el que contiene la parte que debe ser compartida por varios usuarios, y el cliente tiene sólo lo particular de cada usuario. El cliente y el servidor pueden estar en PC diferentes o compartiendo la misma PC. Ver figura 1.1.

En esta arquitectura:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.



*Figura 1.1 Arquitectura Cliente/Servidor*

## 1.4.2. Servidor web

Un servidor web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

**Apache** es un servidor web de código abierto, que se puede obtener gratuitamente aunque no sea totalmente compatible con la versión 2 de la GNU General Public License (GPL) (7), desarrollado para las plataformas de Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. Este software implementa el protocolo HTTP de transferencia de hipertexto. Es flexible, rápido y eficiente. Presenta el inconveniente de no tener una interfaz gráfica para su administración y configuración, aunque esto se compensa con las múltiples funcionalidades que brinda. Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos. Varias compañías importantes de software en el mundo integran a Apache como parte de sus productos.

### 1.4.3. Aplicaciones web

Las aplicaciones web son aplicaciones del tipo cliente-servidor, es decir, los datos y el procesamiento de la información se realizan solo en el servidor y la computadora que utiliza dicha aplicación (cliente), se comunica con el servidor enviando y recibiendo la información por medio de un navegador web (ej. Internet Explorer, Mozilla Firefox) utilizando HTTP como protocolo fundamental. De esta manera un mismo servidor puede atender las peticiones de gran cantidad de clientes al mismo tiempo, por lo cual, todos estos podrán estar compartiendo los mismos datos y utilizando una misma aplicación desde distintos lugares.

### 1.4.4. Navegador web

Un navegador o navegador web es un programa que permite visualizar la información que contiene una página web. El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Las páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen. El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a Internet, se llama navegación; que es de donde se origina el nombre de navegador.

Existen varios navegadores web o *browser* que cumplen las funcionalidades básicas pero entre los más importantes están Mozilla Firefox, Internet Explorer y Netscape Navigator. Mozilla Firefox es hoy en día el más popular, el que más seguridad y rapidez brinda en la navegación, además es software libre, multiplataforma y existe una comunidad internacional que aporta a su constante desarrollo.

## 1.4.5. Lenguajes de programación web

De acuerdo a la arquitectura cliente/servidor la programación web consta de dos grupos de lenguajes, llamados programación del lado del cliente y programación del lado del servidor.

### Del lado del cliente:

**HTML:** Es un lenguaje sencillo estático para el desarrollo de páginas web. Es un lenguaje de marcado de hipertexto, se pueden relacionar unas páginas con otras haciendo posible la navegación web.

### Ventajas:

- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los navegadores.

**Javascript:** Este es un lenguaje interpretado, no requiere compilación. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

El código Javascript puede ser integrado dentro de las páginas web aunque puede estar en ficheros independientes. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (Document Object Model) el cual puede ser manipulado usando Javascript. Ha sido el avance más significativo en el logro de páginas web dinámicas y exactas en cuanto a posición y presentación de su contenido, es un lenguaje robusto y a la vez ligero.

**XML** es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la

lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. (8)

**AJAX** es una técnica de desarrollo web para crear aplicaciones interactivas que se ejecuta en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Permite hacer cambios en las páginas sin necesidad de recargarlas lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. AJAX es válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores.

Al decir de Jesse James Garrett (9), AJAX no es una tecnología, es realmente muchas tecnologías, uniéndose en poderosas nuevas formas. AJAX incorpora:

- presentación basada en estándares usando XHTML y CSS;
- exhibición e interacción dinámicas usando el Document Object Model;
- Intercambio y manipulación de datos usando XML and XSLT;
- Recuperación de datos asíncrona usando XMLHttpRequest;
- y JavaScript poniendo todo junto.

**CSS** es el lenguaje para describir la presentación de páginas web. Esto incluye los colores, el diseño y la información de fuentes, así como la manera de cambiar la presentación de los diferentes tipos de dispositivos, tales como aquellos con pantallas grandes, pequeñas pantallas o impresoras. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje basado en XML. La separación de HTML de CSS hace más fácil mantener el código, hojas de estilo compartir a través de las páginas y páginas de adaptar a diferentes ambientes. Esto se conoce como la separación de la estructura (o contenido) de la presentación. (10)

### **Del lado del servidor:**

**PHP:** Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas.

## Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto, clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

### 1.4.6. Proceso de desarrollo de software

Un proceso de desarrollo de software tiene como objetivo la producción eficiente de un producto de software que satisfaga los requisitos de un cliente con una planificación y una estimación de recursos predecibles.

**El Proceso Unificado de Desarrollo (RUP)** es un proceso de ingeniería de software resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. En este se unifican los mejores elementos de metodologías anteriores. Está preparado para desarrollar grandes y complejos proyectos. Es una metodología orientada a objetos y utiliza el UML como lenguaje de representación visual. RUP se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y ser iterativo e incremental. Está constituido por flujos de trabajo fundamentales: negocio, requerimientos, análisis, diseño, implementación y prueba, los que tienen lugar sobre 4 fases: inicio, elaboración, construcción y transición.

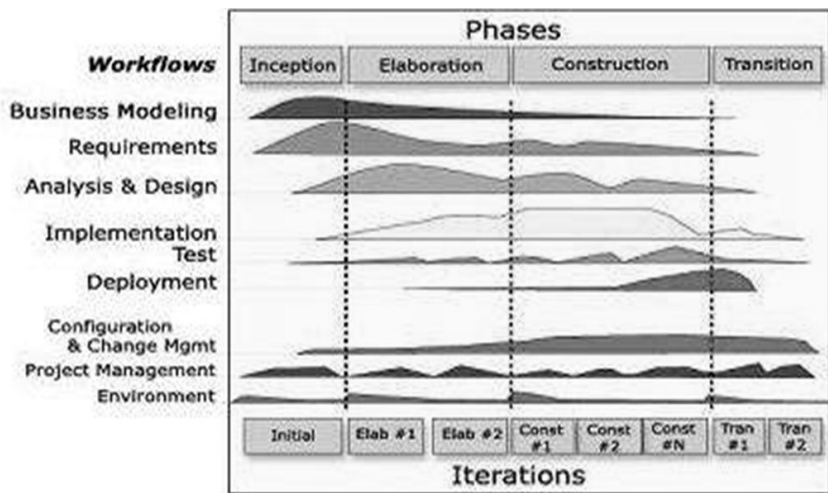
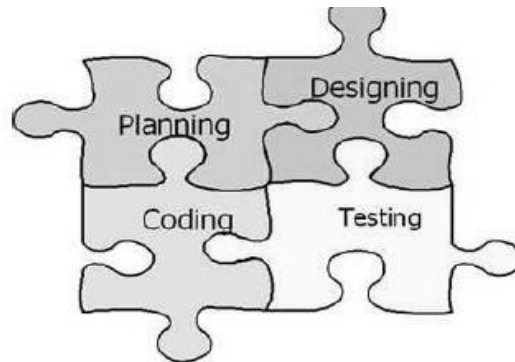


Figura 1.2 Ciclo de vida de RUP

**Extreme Programming (XP)** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en 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 realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (11)

XP es utilizada fundamentalmente para proyectos de corto plazo, con un equipo reducido. Consiste en una programación rápida o extrema donde el usuario final forma parte del equipo. XP también es iterativo e incremental, se basa en las pruebas unitarias continuas, en la reutilización de código, programación en pareja. Cuenta con cuatro fases: Planificación, Diseño, Desarrollo y Pruebas.





*Figura 1.3 Fases de XP*

La UCID propone como guía para la construcción de software, su propio Proceso de Desarrollo y Gestión de Proyectos de Software como resultado de varios años de experiencia utilizando metodologías de desarrollo. Este proceso puede decirse que es una combinación de las características de algunas de las metodologías de desarrollo de software más utilizadas en el mundo, que más se ajustan a las condiciones de trabajo del centro, resultando ser robusta, flexible, ágil y fácil de asimilar por todos los implicados en un proyecto.

El modelo de desarrollo de software propuesto describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Tiene como características fundamentales que es basado en componentes y es iterativo e incremental.

Desarrollo iterativo e incremental: Está compuesto por iteraciones y varias actividades, cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable. En cada iteración se obtiene como resultado un incremento.

Desarrollo basado en componentes: Mayor nivel de reutilización de software. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo. (12)

Según se define en el proceso a utilizar existen cinco fases secuenciales: Inicio, Modelación, Construcción, Explotación Experimental y Despliegue.



Figura 1.4 Ciclo de vida del Proceso de Desarrollo y Gestión de Proyectos de Software

#### 1.4.7. Lenguajes de modelado

**BPMN** (*Business Process Modeling Notation*, Notación para el Modelado de Procesos de Negocio) es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Está dirigido a gerentes, directores, dueños de empresas, ingenieros de procesos, analistas de negocios, analistas de sistemas, administradores de proyectos, responsables de calidad y todo aquel que necesita definir, documentar y hacer más eficientes sus procesos de negocio con el estándar más avanzado y aceptado a nivel internacional. (13)

El objetivo fundamental de utilizar BPMN es dar una notación rápidamente comprensible para todos los responsables en el negocio, desde los analistas, desarrolladores, como los trabajadores del negocio que monitorearan y gestionaran el sistema, además de proveer a los usuarios de una notación de uso libre.

**UML** (*Unified Modeling Language*, Lenguaje Unificado de Modelado) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema.

Lo que se plantea en este trabajo es la coexistencia de ambas notaciones, ya que UML toma un perfil orientado a objetos en el modelado de aplicaciones, mientras que BPMN toma un perfil orientado a

procesos en el modelado de sistemas. BPMN tiene un enfoque en procesos de negocio, UML se enfoca al diseño de software y por lo tanto ambas notaciones son totalmente compatibles entre sí. Las extensiones de UML para el modelado proporcionan algunas otras vistas de la arquitectura de negocio que son más difíciles de observar usando únicamente BPMN. Por ejemplo, la visualización de las responsabilidades de los trabajadores del negocio, la manipulación de las entidades del negocio y la comprensión de los estados asociados a las entidades del negocio. (13)

#### **1.4.8. Herramientas CASE**

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan 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.

**Visual Paradigm for UML (VP-UML)** es una herramienta multiplataforma de modelado visual UML y una herramienta CASE muy potente y fácil de utilizar. VP-UML aporta a los desarrolladores de software una plataforma de desarrollo puntera para construir aplicaciones de calidad mejores y más baratas con rapidez. Aporta una excelente interoperabilidad con otras herramientas CASE y muchos de los entornos IDE líderes del mercado (14).

Soporta el ciclo de vida completo del desarrollo de software: análisis, diseño, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, importar proyectos desde Rational Rose, generar código desde diagramas y generar documentación.

Apostar por el uso de esta herramienta CASE en desarrollo de este trabajo es significativo cuando se tiene en cuenta que es multiplataforma, que implementa las versiones más actuales de UML, que puede obtenerse en su versión gratuita y que es fácil de usar.

#### **1.4.9. Sistema Gestor de Bases de Datos (SGBD)**

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir (15):

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Ejemplo de SGBD son MySQL, Oracle, SQL Server de Microsoft y PostgreSQL.

Algunas de las características deseables en un Sistema Gestor de Base de Datos SGBD son:

- Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdiciar espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

**PostgreSQL** es un sistema de gestión de base de datos objeto-relacional ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es software libre, código abierto y multiplataforma, publicado bajo la licencia BSD<sup>4</sup>. PostgreSQL es un potente motor de bases de datos avanzado, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Permite métodos almacenados, restricciones de integridad, vistas, etc. Con herramientas gráficas como PgAdmin, phpPgAdmin la administración de la base de datos es sencilla.

#### 1.4.10. Frameworks

Un framework es una estructura de soporte definida 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 software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (16)

---

<sup>4</sup> La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*), esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre.

**EXT JS:** es un framework de código limpio, usando tecnología AJAX, escrito en Javascript con soporte para la mayoría de los navegadores, presenta cientos de funcionalidades que hacen de él una herramienta indispensable, ya que junto con el ahorro de tiempo y de líneas de código hace que el usuario se sienta a gusto pues presenta una interfaz amigable y pegajosa. Contiene estilos CSS predefinidos aunque es totalmente configurable. Posee 2 licencias, una comercial y otra Open Source (GPL v3). Posee una gran comunidad y hoy en día es uno de los frameworks de Javascript más usados. (17)

**Zend Framework:** es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. 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 voluntad). (18)

**Doctrine:** es un mapeador objeto relacional (ORM) para PHP 5.2.3 y posteriores que se sienta en la cima de una potente capa de abstracción de bases de datos (DBAL). Uno de sus principales características es la opción de escribir las consultas de base de datos en un objeto propiamente orientado al dialecto SQL llamado Doctrine Query Language (DQL), inspirado por Hibernate HQL. Esto proporciona a los desarrolladores una poderosa alternativa a la que mantiene la flexibilidad de SQL sin necesidad de la duplicación innecesaria de código. (19)

#### 1.4.11. Control de versiones

**Subversion:** Es un sistema de control de versiones libre y de código abierto, es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, como por ejemplo: el código fuente de un programa. Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Esto permite compartir los cambios sobre un mismo conjunto de ficheros. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

#### Herramientas cliente para la comunicación con Subversion

- TortoiseSVN

- RapidSVN

**TortoiseSVN:** es un cliente que se integra con el gestor de ficheros estándar de Windows para acceder y depositar ficheros en cualquier gestor de versiones centralizado Subversion (SVN). TortoiseSVN es un software libre que se puede descargar gratuitamente para su uso sin restricciones. Todos los comandos de Subversion están disponibles desde el explorador de Windows.

**RapidSVN:** es un cliente gráfico que permite manipular repositorios de Subversion. Es una de las alternativas más conocidas para los sistemas GNU/Linux, pero también corre sobre varios sistemas operativos.

## 1.5. Conclusiones

En aras de desarrollar un software para el análisis de información que se ajuste a las características requeridas, se ha realizado en este capítulo un estado del arte a diferentes niveles, tanto en la UCI como en el mundo, llegando a conclusiones de la necesidad que existe actualmente de desarrollar una herramienta para el análisis de información que facilite la toma de decisiones para los subsistemas de Reclutamiento y Reserva. Además se hizo un estudio de las diferentes metodologías, tecnologías, herramientas y lenguajes más utilizados en el mundo a la hora de desarrollar este tipo de aplicaciones, determinando los que serán utilizados para el desarrollo de la aplicación, la decisión de utilizarlas fue tomada en conjunto con la dirección del proyecto siguiendo fundamentalmente la política de uso de herramientas con soporte multiplataforma y licencias de utilización libre. Para cumplir con dichas condiciones las herramientas y tecnologías a utilizar para el desarrollo de la aplicación son: PostgreSQL como gestor de base datos; servidor web Apache; navegador web Mozilla Firefox; PHP como lenguaje de programación del lado del servidor; HTML, XML, CSS, Javascript y AJAX del lado del cliente; como metodología de desarrollo: el Proceso de Desarrollo y Gestión de Proyectos de Software de la UCID con notación BPMN para el modelado del negocio y UML para los demás fases; Visual Paradigm como herramienta CASE; los frameworks a utilizar son ExtJS, Zend Framework, Doctrine; y para la gestión de los cambios se empleará Subversion en el servidor de aplicaciones y como cliente TortoiseSVN.

## Capítulo 2: Características de la herramienta.

En este capítulo se hace una propuesta de solución de la herramienta a desarrollar basado en el problema planteado. Se realiza el modelo conceptual, definiendo los diversos conceptos del dominio del problema, así como las relaciones que se establecen entre estos. Se definen y se especifican los requerimientos funcionales y no funcionales, que deberá tener y cumplir el sistema para satisfacer las necesidades de los usuarios. Se muestran los prototipos de interfaz de usuarios iniciales de la herramienta.

### 2.1. Objeto de estudio

#### 2.1.1. Situación problémica y problema

El empleo del potencial humano incluye un conjunto de tareas con el interés de asignarle a cada ciudadano un lugar en la defensa en correspondencia con la concepción de la guerra de todo el pueblo, los requerimientos del sistema defensivo territorial y las condiciones físicas, psíquicas, políticas y sociales, así como la preparación militar y otras cualidades personales, de modo que pueda ser empleado adecuadamente de acuerdo con las misiones que le sean asignadas para tiempo de guerra y recibir la preparación necesaria.

Los comités militares provinciales y municipales son los órganos profesionales especializados para la organización, control y ejecución de la definición y distribución del potencial humano en la defensa.

Dentro de los procesos de gestión de la información en los CM se encuentran los procesos de análisis de información en los subsistemas Reclutamiento y Reserva, estos no son más que aquellos que se llevan a cabo para procesar la información asociada a los procesos que se realizan durante la actividad de incorporar a los ciudadanos al servicio militar activo y para el completamiento del personal de la unidades a partir de la reserva; sobre esa base se podrán entonces tomar mejores decisiones.

Actualmente los Comités Militares utilizan el sistema DATAFAR para la gestión de la información que manejan, donde las recuperaciones que se hacen son pocas y de forma estática, o sea, los modelos de presentación de los datos son siempre iguales y están diseñados de antemano. Cuando se hace necesario crear un nuevo tipo de reporte el equipo de desarrollo tiene que dedicar tiempo y esfuerzo adicional por lo cual los usuarios tienen que esperar por la nueva funcionalidad.

Con la amplia información existente y con las muchas vías de obtenerla resulta prácticamente imposible saber qué es lo que necesitan los usuarios.

En este momento la información se extrae a partir de datos generales y en muy pocos casos sobre datos específicos, pero es más relevante para el usuario obtener datos específicos que le puedan dar una información más detallada.

Partiendo de esta situación, se identifica el siguiente problema a resolver: ¿Cómo facilitar el análisis de información en los subsistemas de Reclutamiento y Reserva?

### **2.2. Objeto de automatización.**

DATAFAR es una aplicación web que se encarga de la gestión de la información de los procesos de Reclutamiento y Reserva. Este sistema no permite realizar todas las recuperaciones necesarias para el trabajo con los Comités Militares, lo cual constituye el principal objeto de automatización de este trabajo: las recuperaciones de información para los Comités Militares, se requiere poder obtener toda la información que se gestiona.

### **2.3. Información que se maneja.**

La información que se maneja es toda la relacionada con las personas controladas por los comités militares, así como la información asociada a los procesos que se realizan durante el reclutamiento de los jóvenes para el SMA y para el completamiento del personal de las unidades a partir de la reserva, que no es más que:

El comprobante de inscripción: constituye un documento personal e intransferible que el responsable de área de atención (funcionario designado) confecciona a todos los ciudadanos controlados de acuerdo con la Ley, al momento de su inscripción en el registro militar (prerreclutas).

Tarjeta de registro: es confeccionada por el responsable de área de atención (funcionario designado) en el momento de la inscripción, contiene los datos de los ciudadanos registrados militarmente.

Libro control de cargos: Contiene los cargos y datos del registro de los ciudadanos asignados a las unidades militares.

Libro de registro de personal de plantilla: Contiene los cargos y datos del registro de los ciudadanos asignados a las unidades militares y de las MTT.



Libro alfabético de los prerreclutas: tiene como objetivo el control alfabético de los prerreclutas y permite la localización por el área de atención a que corresponda, conocer la situación con relación al cumplimiento del servicio militar activo de una generación en cuestión, reflejar en él certificaciones que legalizan las causas de las bajas del registro como prerreclutas de los jóvenes incluidos en este y emplearse para el control de las mujeres que desean incorporarse al servicio militar voluntario femenino.

Libro de actas del trabajo de reclutamiento (Comisión Médica): Tiene como objetivo dejar constancia de la participación de los miembros de la comisión médica en sus sesiones de trabajo, así como de la relación nominal de los jóvenes examinados y la conclusión de cada caso.

Libro de actas del trabajo de reclutamiento (Comisión de reclutamiento): Tiene como objetivo dejar constancia de la participación de los miembros de la comisión de reclutamiento, sus sesiones de trabajo, así como de la relación nominal de los jóvenes entrevistados y la conclusión de cada caso.

### **2.4. Propuesta de sistema.**

Se propone desarrollar una herramienta para el análisis de información que facilite la toma de decisiones, que permitirá recuperar la información que actualmente se gestiona en DATAFAR de forma dinámica. Podrán generarse reportes tanto estadísticos, como nominales. La solución estará orientada al usuario, o sea, los usuarios finales pueden diseñar y obtener las recuperaciones sin tener conocimientos de programación o de bases de datos.

La herramienta tendrá como fuente de datos la base de datos de DATAFAR y una base de datos propia para el trabajo con la aplicación. Estará compuesta por tres componentes relacionados entre ellos: El componente para la configuración de la fuente de datos servirá a los administradores para configurar los campos por los cuales podrán los usuarios recuperar la información, permitiendo una abstracción sobre los datos reales almacenados; el componente para las recuperaciones donde el usuario podrá seleccionar los campos a obtener y agregar las condiciones por las que desea filtrar y el sistema genera la consulta, la ejecuta y devuelve los datos; y el componente de los reportes donde se podrá configurar el reporte de una manera sencilla y seleccionar el formato a exportarlo.

La seguridad se gestionará usando el subsistema de seguridad SIGIS<sup>1</sup> mediante el cual entre otras funcionalidades se puede gestionar funcionalidades, acciones, roles y usuarios; donde la utilización de las

---

<sup>1</sup> Sistema de Gestión Integral de Seguridad

funcionalidades propias del sistema se especifica para cada rol y a cada usuario se le indica a cual estructura tiene acceso. Para asegurar la compartimentación de la información el sistema internamente asocia el identificador de cada estructura con el cual un usuario accede a la aplicación con el identificador de las estructuras que establece DATAFAR, así se puede filtrar la información que se le mostrará a cada usuario.

### **2.5. Modelo Conceptual**

Este Modelo Conceptual explica los conceptos significativos en el dominio del problema. Puede mostrar: conceptos, asociaciones entre conceptos y atributos de conceptos. Una cualidad esencial que debe ofrecer un Modelo Conceptual es que debe representar cosas del mundo real. El modelo conceptual es una herramienta de comunicación, con la cual se intenta comprender los conceptos importantes y sus relaciones.

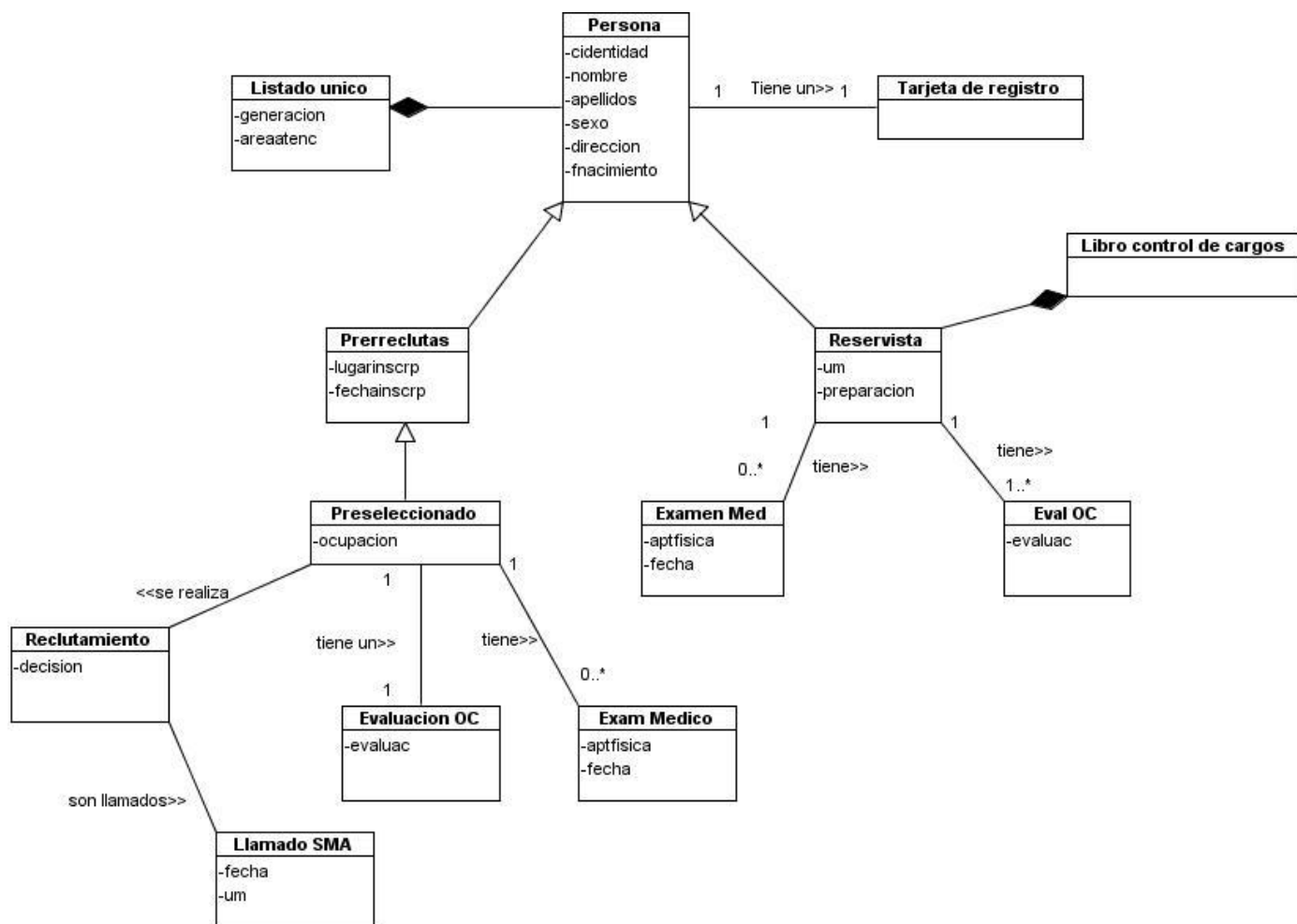


Figura 2.1 Modelo Conceptual

## 2.6. Definición de requisitos

**Requisito:** Una capacidad del software que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar, u otra documentación formal. Los requisitos de sistemas deben mostrar todo lo que el sistema debe hacer. Los requisitos se clasifican en dos tipos: funcionales y no funcionales.

### 2.6.1. Requisitos funcionales.

Los **requisitos funcionales** son capacidades o condiciones que el sistema debe cumplir. Se han definido los siguientes requisitos funcionales:

### **RF1** Gestionar concepto.

RF1.1 Adicionar concepto.

RF1.2 Modificar concepto.

RF1.3 Eliminar concepto.

### **RF2** Configurar campo.

RF2.1 Crear campo.

RF2.2 Eliminar campo.

### **RF3** Gestionar las recuperaciones.

RF3.1 Diseñar recuperación.

RF3.2 Cargar recuperación.

RF3.3 Guardar recuperación.

### **RF4** Gestionar reportes.

RF4.1 Diseñar reporte.

RF4.2 Visualizar un reporte diseñado de antemano.

RF4.3 Eliminar un reporte diseñado de antemano.

#### **2.6.2. Requisitos no funcionales.**

Los **requisitos no funcionales** son propiedades o cualidades que el producto debe tener. Se han definido los siguientes requisitos no funcionales:

#### **Usabilidad**

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. El software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades. Los controles tipo lista de selección que tengan 10 o más opciones emplearán una búsqueda por teclado.

#### **Rendimiento**

Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones. Esto se logra enviando la menor cantidad posible de datos entre el cliente y el servidor, se envían los datos que solo necesita el usuario en ese momento, cuando requiera más datos los solicita en una nueva petición, los datos que se van a utilizar varias veces y se cargan desde el servidor se quedan almacenados temporalmente en el lado del cliente. Uno de los procedimientos que más tiempo requieren en el lado del servidor es la generación del código SQL, por eso se asegura realizarlo una sola vez y cuando se vaya a utilizar la consulta esta estará almacenada en la base de datos, se intercambiará solo el identificador de la consulta.

### **Restricciones de diseño**

Utilizar los estándares establecidos (de diseño de interfaces, de base de datos y de codificación).

### **Requerimiento de Ayuda y Documentación**

El sistema contará con una ayuda, que estará disponible en cada una de las interfaces, de esta forma los usuarios tendrán conocimiento de las funcionalidades del mismo y hacer un mejor uso de estas. Guiando de esta forma al usuario de cómo trabajar en el sistema.

### **Interfaz**

El sistema contará con una interfaz amigable, sencilla, permitiendo que los usuarios finales del mismo sean capaces interactuar con este aún teniendo conocimientos básicos. Asociada al propósito de la organización donde se empleará. Estará diseñado para adaptarse a la resolución del usuario. Empleará imágenes similares a los utilizados en DATAFAR con los cuales los usuarios están familiarizados. Debe poseer una interfaz acorde al desarrollo de las aplicaciones web actuales.

### **Portabilidad**

El sistema será multiplataforma (Linux y Windows fundamentalmente).

### **Políticos Culturales**

El sistema será utilizado en territorio cubano y por las entidades autorizadas por el Ministerio de las FAR. El producto no contendrá palabras en otros idiomas. Se respetarán los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

### **Legales**

- Ley Nro. 75 de la Defensa Nacional.
- Manual para el trabajo de los Comités Militares.
- Manual de los responsables de Áreas de Atención.

### **Seguridad**

El usuario debe autenticarse antes de entrar al sistema. Se garantiza el acceso controlado a la información. Se presentarán las interfaces para cada usuario dependiendo del nivel de acceso a la información. La información de la base de datos del sistema DATAFAR solo podrá verse, no podrá modificarse ni eliminarse.

### **Software**

Para el servidor:

- PHP - v5.2.6 (estable).
- Doctrine-1.2.1 (estable).
- Zend Framework -1.9.7 (estable).
- ZendExt – 1.5.4 (beta).
- PostgreSQL – 8.3 (estable).

Para el cliente:

- Navegador Mozilla Firefox – 2.2 (estable).
- CSS – 3 (estable).
- HTML – 5 (estable).
- Javascript – 1.8.2 (estable).
- ExtJS – 2.2 (estable).
- UCID – 1.1.2 (beta).

### **Hardware**

Para el cliente:

- Requerimientos mínimos: Procesador a 1.40 GHz con 254 Mb (recomendado 512 Mb) de memoria RAM.
- 1 Tarjeta de red.
- 1 Impresora.

Para el servidor:

- Requerimientos mínimos: Procesador a 3GHz y 1Gb de memoria RAM.
- Al menos 40 Gb de espacio libre en disco duro.
- 1 Tarjeta de red.
- 1 UPS.
- 1 Lector de CD.

## 2.7. Descripción de requisitos funcionales

### 2.7.1. Requisito funcional Gestionar concepto.

Este requisito permite adicionar, modificar y eliminar un concepto.



Figura 2.2 Prototipo de IU Gestionar conceptos

#### 2.7.1.1. Especificación del requisito Adicionar concepto.

Conceptos tratados	Conceptos	Atributos
	No procede	No procede

Precondiciones	Precondiciones	Pre-requisito
	El usuario debe estar autenticado.	Autenticar usuario.
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Se selecciona la opción Gestionar conceptos del menú Configuración.</li> <li>2. Se muestra la interfaz Gestionar conceptos.</li> <li>3. Se selecciona la opción Adicionar.</li> <li>4. Se muestran los campos a llenar: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Descripción</li> </ul> </li> <li>5. Se oprime el botón Aceptar, se almacenan los datos del concepto, se abandona la interfaz y se actualizan en el grid.</li> </ol> <p>5.1 En caso de oprimir el botón Cancelar se abandonan las acciones sin realizar ninguna operación.</p> <p>5.2 En caso de oprimir el botón Aplicar se almacenan los datos y se limpian los controles de la interfaz.</p> <p>En caso que se oprima el botón Ayuda se muestra la ayuda de la interfaz.</p>	
<b>Validaciones</b>	<ul style="list-style-type: none"> <li>- El sistema valida los datos según lo descrito en el Modelo Conceptual: <a href="#">Modelo conceptual.doc</a>.</li> <li>- El campo nombre tiene que ser de solo letras con máximo de 20 caracteres.</li> <li>- El campo descripción tiene que ser de solo letras con máximo de 40 caracteres.</li> </ul>	
<b>Post-condiciones</b>	Se ha adicionado el concepto.	
<b>Post-requisito</b>	No procede.	





Figura 2.3 Prototipo de IU Adicionar concepto

2.7.1.2. Especificación del requisito Eliminar concepto.

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	No procede	No procede
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	El usuario debe estar autenticado.	Autenticar usuario.
	Se han adicionado conceptos	Adicionar concepto
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Se selecciona en el grid el concepto que se desea eliminar.</li> <li>2. Se selecciona la opción Eliminar, se muestra una ventana de confirmación. .”¿Está seguro que desea eliminar el concepto?”.</li> <li>3. Se oprime el botón Aceptar, elimina los datos en el grid y sale de la ventana.             <ol style="list-style-type: none"> <li>3.1. En caso de oprimir el botón Cancelar se abandonan las acciones sin realizar ninguna operación.</li> </ol> </li> </ol> <p>En caso que se oprima el botón Ayuda se muestra la ayuda de la interfaz.</p>	
<b>Validaciones</b>	<p>El sistema valida los datos según lo descrito en el Modelo Conceptual:</p> <p><a href="#">Modelo conceptual.doc</a></p>	

<b>Post-condiciones</b>	Se ha eliminado el concepto.
<b>Post-requisito</b>	No procede.

### 2.7.2. Requisito funcional Configurar Campos.

Este requisito permite crear y eliminar campos.

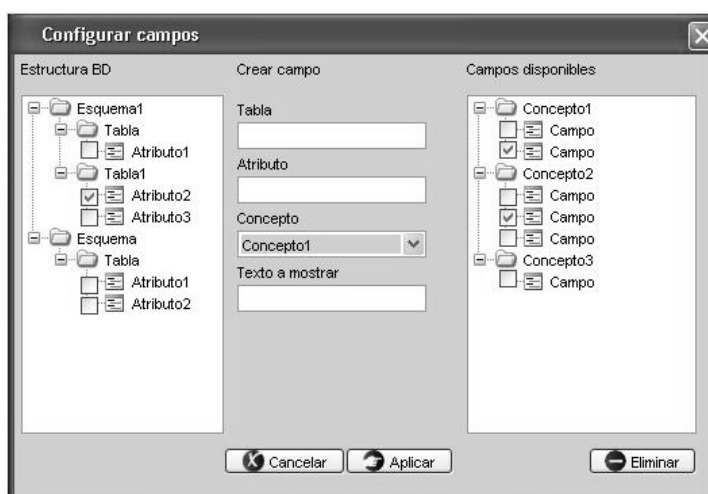


Figura 2.4 Prototipo de IU Configurar campos

#### 2.7.2.1. Especificación del requisito Crear campo.

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Persona	cedentidad, nombre, apellidos, sexo, dirección, fnacimiento
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	El usuario debe estar autenticado.	Autenticar usuario.
	Se han adicionado conceptos.	Adicionar concepto.

<p><b>Descripción</b></p>	<ol style="list-style-type: none"> <li>1. Se selecciona la opción Configurar campos del menú Configuración</li> <li>2. Se selecciona un campo real de la base de datos.</li> <li>3. Se muestran los datos referente a ese campo seleccionado: <ul style="list-style-type: none"> <li>• Tabla</li> <li>• Atributo</li> </ul> </li> <li>4. Se selecciona el concepto al cual va a estar asociado ese campo.</li> <li>5. Se introduce el nombre que va a tener el campo.</li> <li>6. Se oprime el botón Aplicar, se almacenan los datos del campo. <ol style="list-style-type: none"> <li>6.1 Si se oprime el botón Cancelar, el sistema no realiza ninguna operación.</li> </ol> </li> </ol> <p>En caso que se oprima el botón Ayuda se muestra la ayuda de la interfaz.</p>
<p><b>Validaciones</b></p>	<ul style="list-style-type: none"> <li>- El sistema valida los datos según lo descrito en el Modelo Conceptual: <a href="#">Modelo conceptual.doc</a>.</li> <li>- El campo Texto a mostrar no puede estar vacío.</li> </ul>
<p><b>Post-condiciones</b></p>	<p>Se ha creado el campo.</p>
<p><b>Post-requisito</b></p>	<p>No procede.</p>

### 2.7.3. Requisito funcional Gestionar Recuperaciones.

Este requisito permite diseñar, cargar y guardar una recuperación.



Figura 2.5 Prototipo de IU Diseñar recuperaciones

2.7.3.1. Especificación del requisito Diseñar recuperación.

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Persona	Identidad, nombre, apellidos, sexo, dirección, nacimiento
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	El usuario debe estar autenticado.	Autenticar usuario.
	Se han adicionado conceptos.	Gestionar concepto
	Se han creado campos.	Gestionar campo
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Se selecciona la opción Gestionar recuperaciones del menú Recuperaciones.</li> <li>2. Se selecciona los campos por los cual se desea obtener la recuperación.</li> <li>3. Se oprime el botón Adicionar se muestra la interfaz Adicionar</li> </ol>	

	<p>condición, donde se selecciona la condición por la cual se desea filtrar:</p> <ol style="list-style-type: none"> <li>4. Se selecciona el campo.</li> <li>5. Se selecciona la condición.</li> <li>6. Se introduce el valor para la condición.</li> <li>7. Se selecciona un operador lógico.</li> <li>8. Se oprime el botón Aceptar, se adiciona la condición al grid, abandona la interfaz de adicionar condición.</li> <li>8.1 En caso de oprimir Cancelar se terminan las acciones sin adicionar.</li> <li>9. En caso de oprimir Generar se muestra un grid con el resultado de la recuperación.</li> </ol> <p>En caso de oprimir el botón Quitar, si existen condiciones, esta se elimina.</p> <p>En caso de oprimir Nuevo se borran todas las operaciones realizadas, tanto en los campos como en los filtros.</p> <p>En caso que se oprima el botón Ayuda se muestra la ayuda de la interfaz.</p>
<b>Validaciones</b>	<ul style="list-style-type: none"> <li>- El sistema valida los datos según lo descrito en el Modelo Conceptual: <a href="#">Modelo conceptual.doc</a>.</li> <li>- El nombre de la recuperación solo letras hasta 20 caracteres.</li> <li>- La descripción solo letras hasta 30 caracteres.</li> </ul>
<b>Post-condiciones</b>	Se ha diseñado la recuperación.
<b>Post-requisito</b>	No procede.

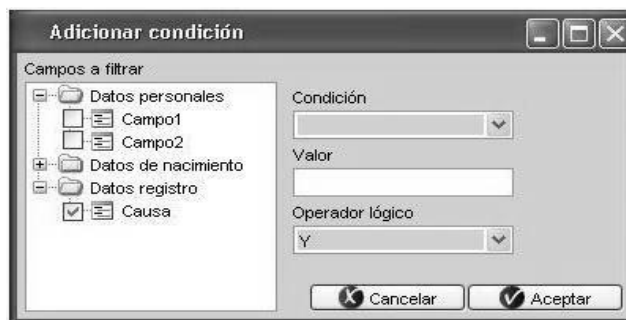


Figura 2.6 Prototipo de IU Adicionar condición

2.7.3.2. Especificación del requisito Cargar recuperación.

Conceptos tratados	Conceptos	Atributos
	Persona	Identidad, nombre, apellidos, sexo, dirección, nacimiento
Precondiciones	Precondiciones	Pre-requisito
	El usuario debe estar autenticado.	Autenticar usuario.
	Se han adicionado conceptos.	Gestionar concepto
	Se han creado campos.	Gestionar campo
	Se han diseñado recuperaciones	Diseñar recuperaciones

<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Se muestra la interfaz Recuperaciones disponibles.</li> <li>2. Se selecciona la recuperación.</li> <li>3. Se oprime el botón Cargar, se muestra la configuración de la recuperación seleccionada.             <ol style="list-style-type: none"> <li>3.1 En caso de seleccionar la opción Eliminar, se muestra una ventana de confirmación. "¿Está seguro que desea eliminar la recuperación?".                 <ul style="list-style-type: none"> <li>• Se oprime Aceptar elimina los datos y sale de la ventana.                     <ol style="list-style-type: none"> <li>3.1.1 En caso de oprimir Cancelar, el sistema no realiza ninguna operación y abandona la ventana.</li> </ol> </li> </ul> </li> </ol> </li> </ol> <p>En caso que se oprima el botón Ayuda se muestra la ayuda de la interfaz.</p>
<b>Validaciones</b>	<ul style="list-style-type: none"> <li>- El sistema valida los datos según lo descrito en el Modelo Conceptual: <a href="#">Modelo conceptual.doc</a></li> </ul>
<b>Post-condiciones</b>	Se ha cargado la recuperación.
<b>Post-requisito</b>	No procede.



*Figura 2.7 Prototipo de IU Recuperaciones disponibles.*

### 2.7.4. Requisito funcional Gestionar reporte.

Este requisito permite diseñar, visualizar y eliminar reporte.



Figura 2.8 Prototipo de IU Gestionar reporte.

2.7.4.1. Especificación del requisito Diseñar reporte.

<b>Conceptos tratados</b>	<b>Conceptos</b>	<b>Atributos</b>
	Persona	Identidad, nombre, apellidos, sexo, dirección, nacimiento
<b>Precondiciones</b>	<b>Precondiciones</b>	<b>Pre-requisito</b>
	El usuario debe estar autenticado.	Autenticar usuario.
	Se han diseñado recuperaciones.	Gestionar recuperaciones.
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Se selecciona la opción Gestionar reporte del menú principal.</li> <li>2. Se oprime el botón la opción Diseñar.</li> <li>3. Se muestra la interfaz Diseñar reporte.</li> <li>4. Se oprime el botón Cargar recuperación.</li> <li>5. Se muestra la interfaz Recuperaciones disponibles.</li> <li>6. Se selecciona la recuperación de la cual se desea hacer el reporte.</li> <li>7. Se muestra los campos de la recuperación seleccionada.</li> <li>8. Se introduce el encabezado con el que saldrá el reporte.</li> <li>9. Se modifica el campo.</li> </ol>	



	<p>10. Se ordena la presentación de los campos</p> <p>11. Se selecciona si se mostrará u ocultará un campo determinado</p> <p>12. Se selecciona un formato de texto para el reporte (HTML, RTF, PDF).</p> <p>13. Se introduce el número de registros por hoja de impresión.</p> <p>14. Se oprime el botón Guardar.</p> <p>15. Se muestra la interfaz Guardar reporte.</p> <p>16. Se introduce el nombre que tendrá el reporte.</p> <p>17. Se oprime el botón Aceptar, se almacena el reporte y abandona las acciones.</p> <p style="padding-left: 40px;">17.1 En caso de oprimir Cancelar se abandonan las acciones sin realizar ninguna operación.</p> <p>En caso de oprimir Vista Previa se muestra una página HTML con la configuración del reporte con los primeros 5 registros de la recuperación.</p> <p>En caso que se oprima el botón Ayuda se muestra la ayuda de la interfaz.</p>
<p><b>Validaciones</b></p>	<ul style="list-style-type: none"> <li>- El sistema valida los datos según lo descrito en el Modelo Conceptual: <a href="#">Modelo conceptual.doc</a>.</li> <li>- El encabezado debe contener hasta 120 caracteres.</li> <li>- La configuración de impresión solo números de 2 dígitos.</li> <li>- El nombre de la recuperación letras hasta 20 caracteres.</li> <li>- La descripción letras hasta 30 caracteres.</li> </ul>
<p><b>Post-condiciones</b></p>	<p>Se ha diseñado el reporte.</p>
<p><b>Post-requisito</b></p>	<p>No procede.</p>



Figura 2.9 Prototipo de IU Diseñar reporte



Figura 2.10 Prototipo de IU Guardar reporte.

## 2.8. Conclusiones

En este capítulo se mostraron los artefactos fundamentales relacionados con la modelación de los procesos de negocio y Requerimientos, como son: el modelo conceptual, definiendo con este los principales conceptos del dominio del problema, así como las relaciones que se establecen entre estos. Se realizó la captura de requisitos a partir del estudio de los procesos negocio existentes y de las entrevistas con los usuarios, donde quedaron definidos los requerimientos funcionales y no funcionales que permitirán determinar las funcionalidades principales del sistema a construir.

### Capítulo 3: Análisis y diseño de la herramienta.

En el presente capítulo se modelarán varios artefactos que tienen lugar durante el diseño detallado del software. En esta etapa se obtienen los siguientes artefactos: diagramas de interacción (específicamente diagramas de secuencia), los diagramas de clases para el diseño web, se realiza además el modelo de datos, mediante el cual se creará la base de datos que soportará las funcionalidades del sistema. El objetivo principal consiste en traducir los requerimientos definidos anteriormente, a una especificación que describe cómo se debe implementar el sistema. Se describen los patrones de diseño utilizados y los principales elementos de la arquitectura.

#### 3.1. Patrones utilizados.

En el diseño, los framework EXT, ZF y Doctrine incorporan algunos de los patrones de diseño más utilizados para la presentación y el manejo de datos. A continuación se brinda una breve explicación del funcionamiento de los mismos.

**Decorator:** Responde a la necesidad de añadir o eliminar funcionalidad a un objeto de forma dinámica y transparente. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. Es una alternativa más flexible para extender la funcionalidad de una clase, de esta forma se añaden atributos o comportamiento adicional a un objeto concreto, no a una clase.

**Factory:** Permite la instancia de objetos en tiempo de ejecución. La ventaja de utilizar este patrón de diseño es que tenemos centralizada la creación de los objetos en una clase que a pedido, nos devuelve instancias de los mismos. Es una forma de tener el control de la creación de objetos. Es llamado el patrón Factory ya que es responsable de "manufacturar" un objeto. (20)

**Front-Controller:** Escucha las peticiones que vienen desde un URL, y se encarga de hacer un *dispatch* al controlador específico llamado, posteriormente llama a la acción deseada por el controlador, en pocas palabras es un *router* de *requests*, y tiene la facilidad de que al todo ser centralizado se puede hacer tareas de levantamiento. (21)

**Table Data Gateway:** Se encarga de generar todo el SQL para acceder a una sola tabla: select, insert, update y delete. También puede tener otros métodos para interactuar con la base de datos. (22)

**Row Data Gateway:** Te permite disponer de un objeto que representa directamente una fila e la base de datos. Todo el acceso a la fila queda oculto tras la interfaz de la clase. (22)

### 3.2. Diseño

El diseño del software permite traducir los requisitos analizados de un sistema, tanto funcionales como no funcionales, en una representación del software, que inicialmente da una visión del mismo y tras posteriores refinamientos sirve como esquema para la implementación del sistema y es, de ese modo, utilizado como una entrada fundamental de las actividades de implementación. (12)

#### 3.2.1. Diseño de clases

En el diseño de clases se resume la definición de las clases que se pueden implementar en el software, se visualizan las relaciones entre ellas, y se muestra gráficamente la interacción de los objetos para comunicarse entre sí. (12)

Como resultado de este proceso se obtienen los diagramas de clases web para cada paquete de requisitos y a partir de los diagramas de clases se realizaron los diagramas de secuencia para cada escenario del requisito.

##### 3.2.1.1. Diagramas de clases web.

El diagrama de clase es el diagrama principal de diseño para un sistema ya que captura la estructura lógica del mismo. Es un modelo estático. En él la estructura de clases del sistema se especifica con relaciones entre clases e interfaces.

Se utilizó un diagrama de clase genérico con el objetivo de describir de forma general el comportamiento de las clases según la arquitectura definida. En un diagrama genérico están presentes las clases de uso común dentro de los frameworks utilizados, clases y/o ficheros genéricos así como las relaciones entre estos. Con esto se evita recargar los distintos diagramas de clases y se facilita el entendimiento de las mismas.

Diagrama de clases genérico de las vistas.

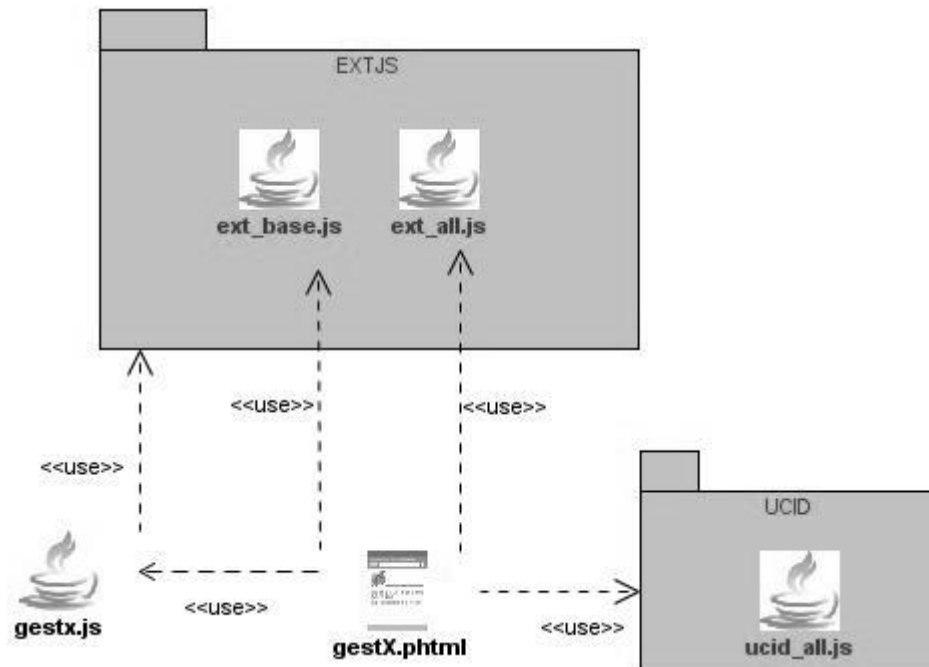
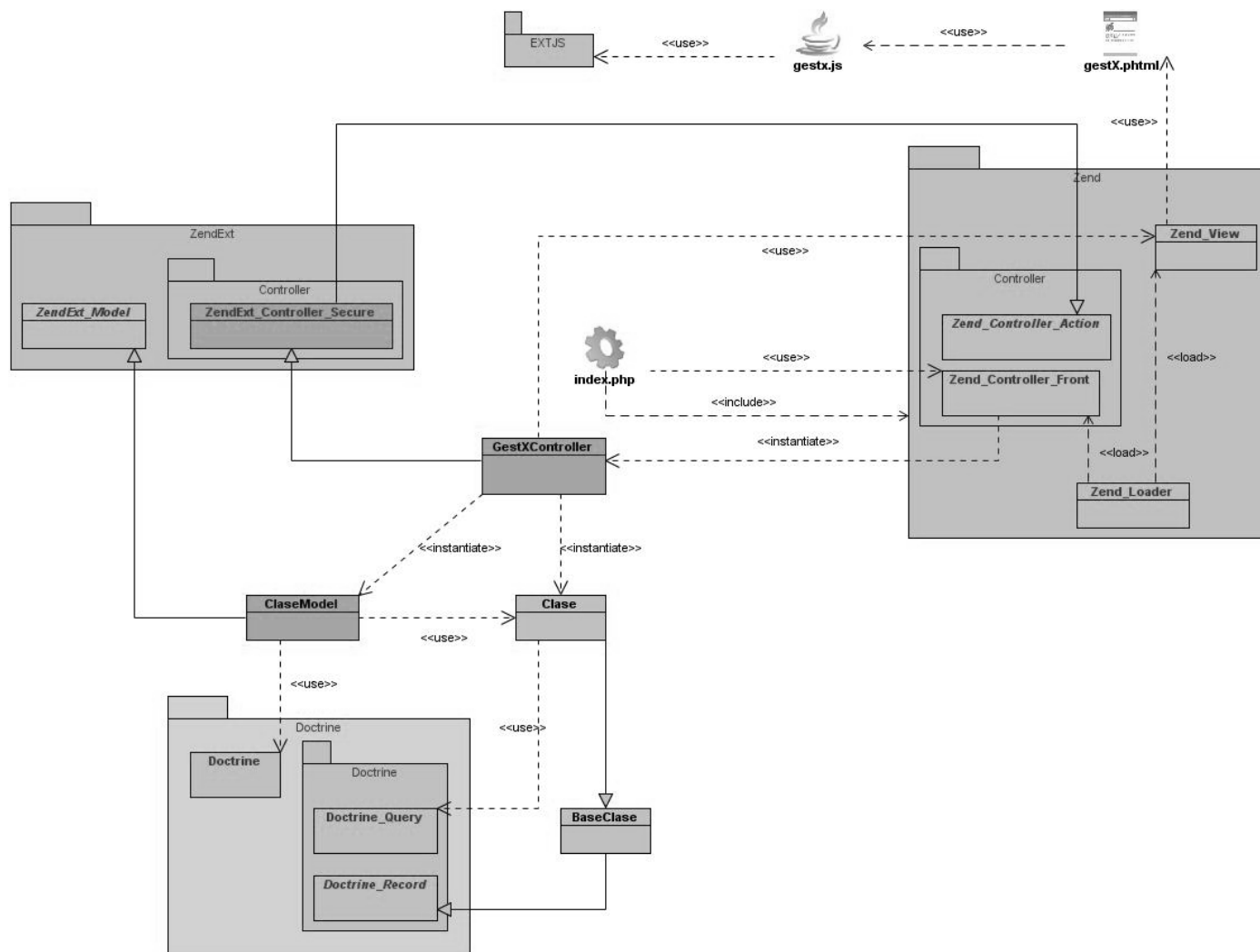


Diagrama de clases genérico.



Seguidamente se representan los diagramas de clases específicos para cada paquete de requisitos.

Diagrama de clases Gestionar concepto

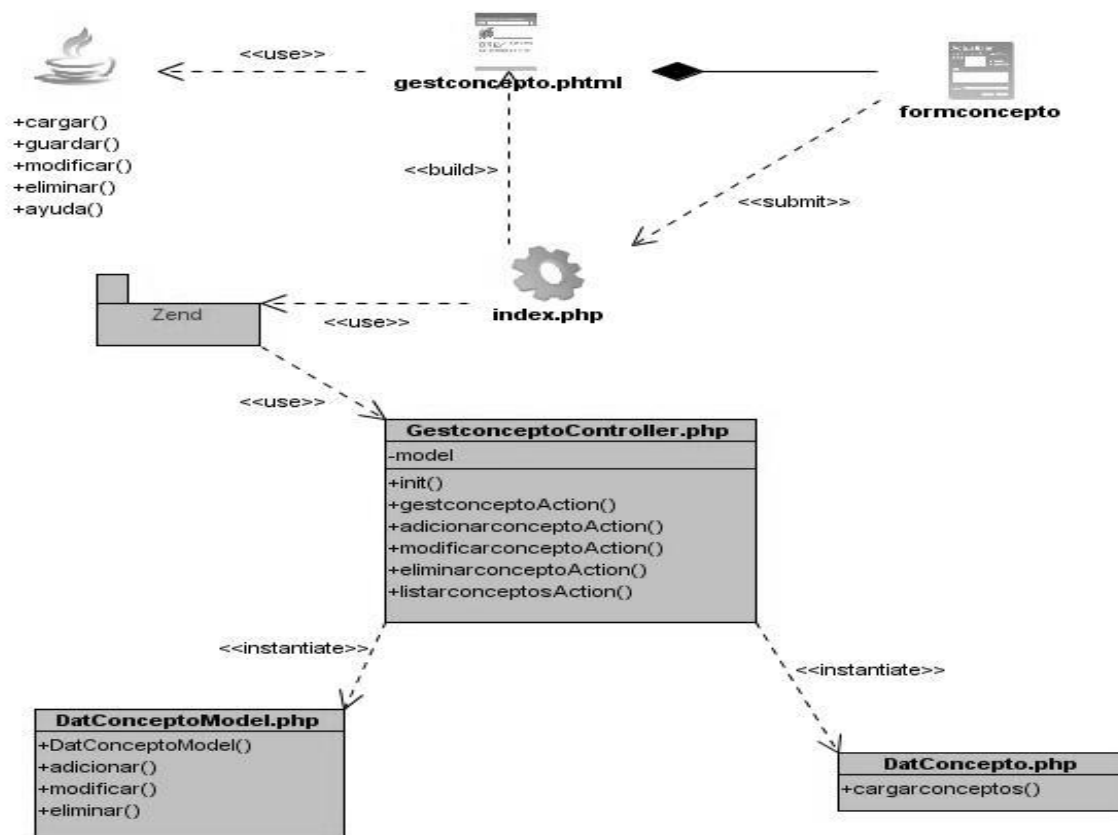


Diagrama de clases Configurar campo

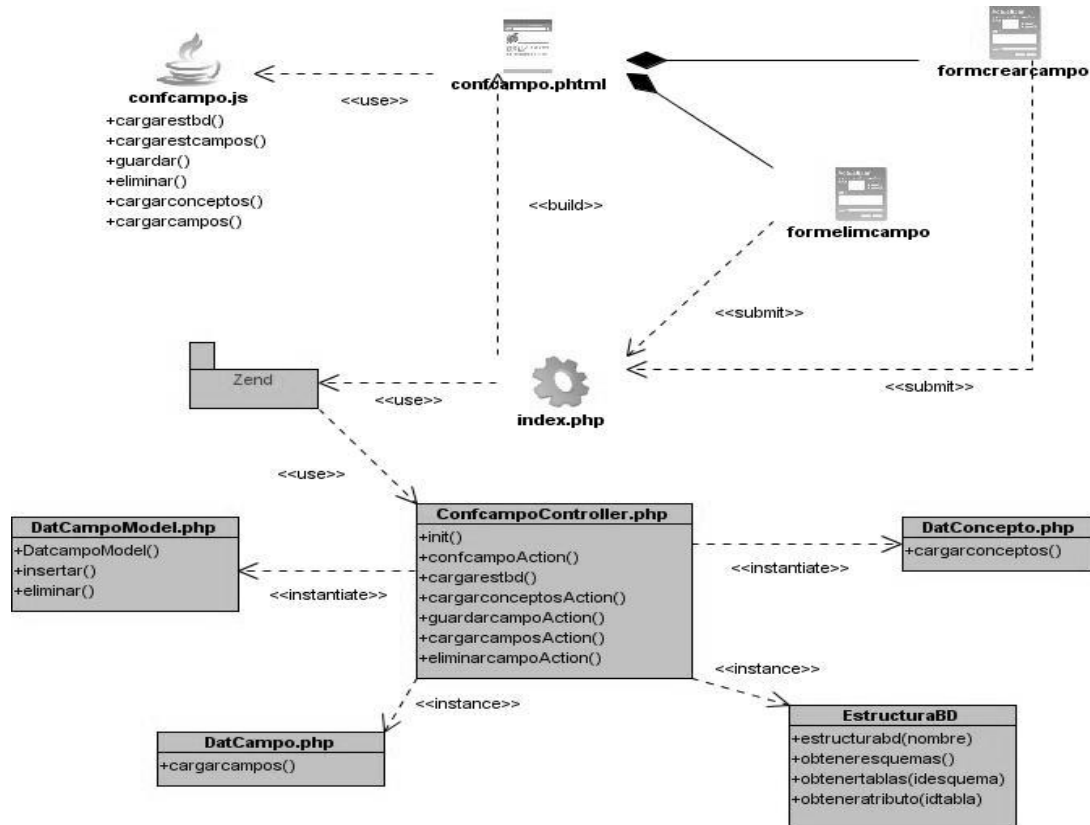
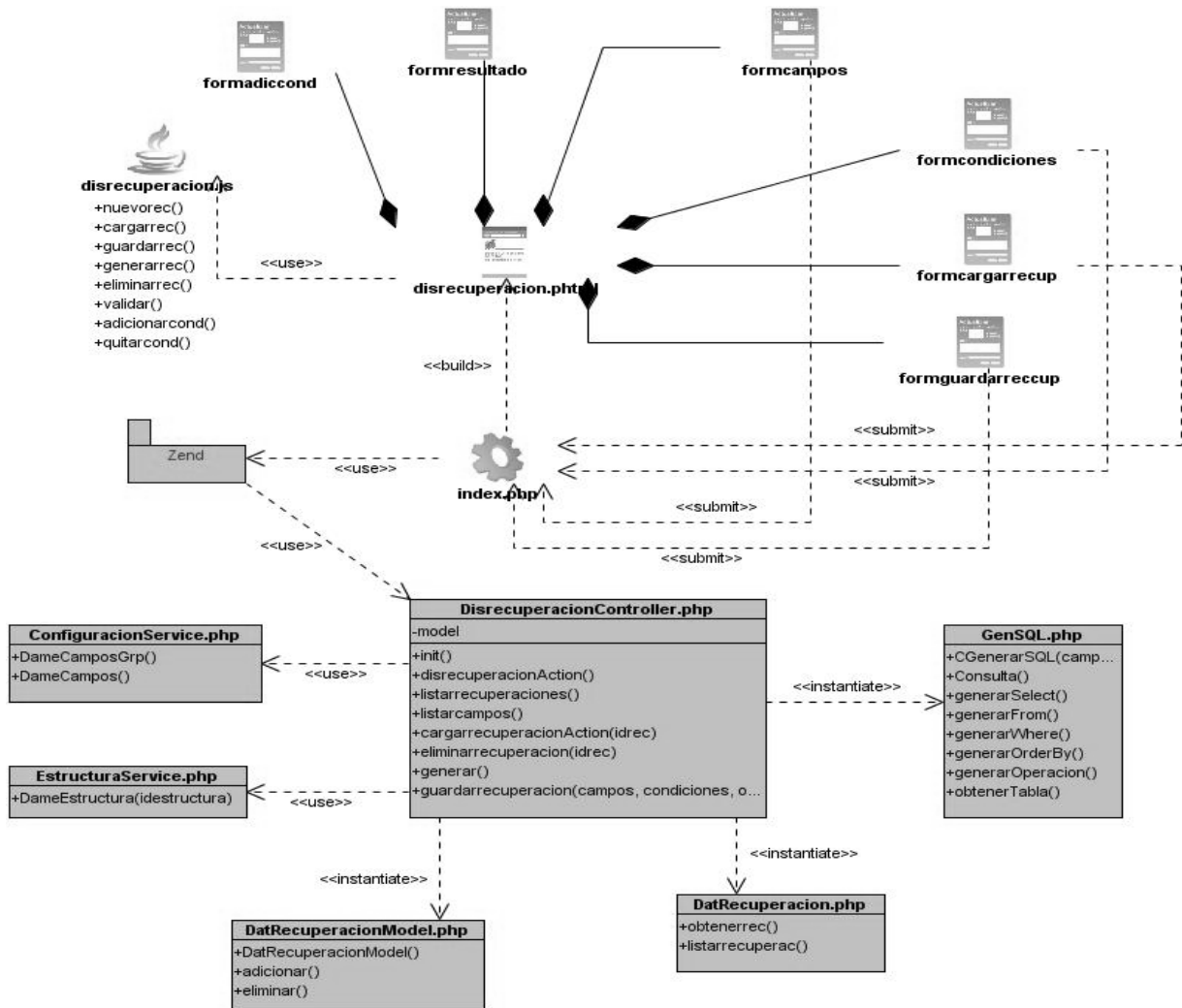




Diagrama de clases Gestionar recuperación



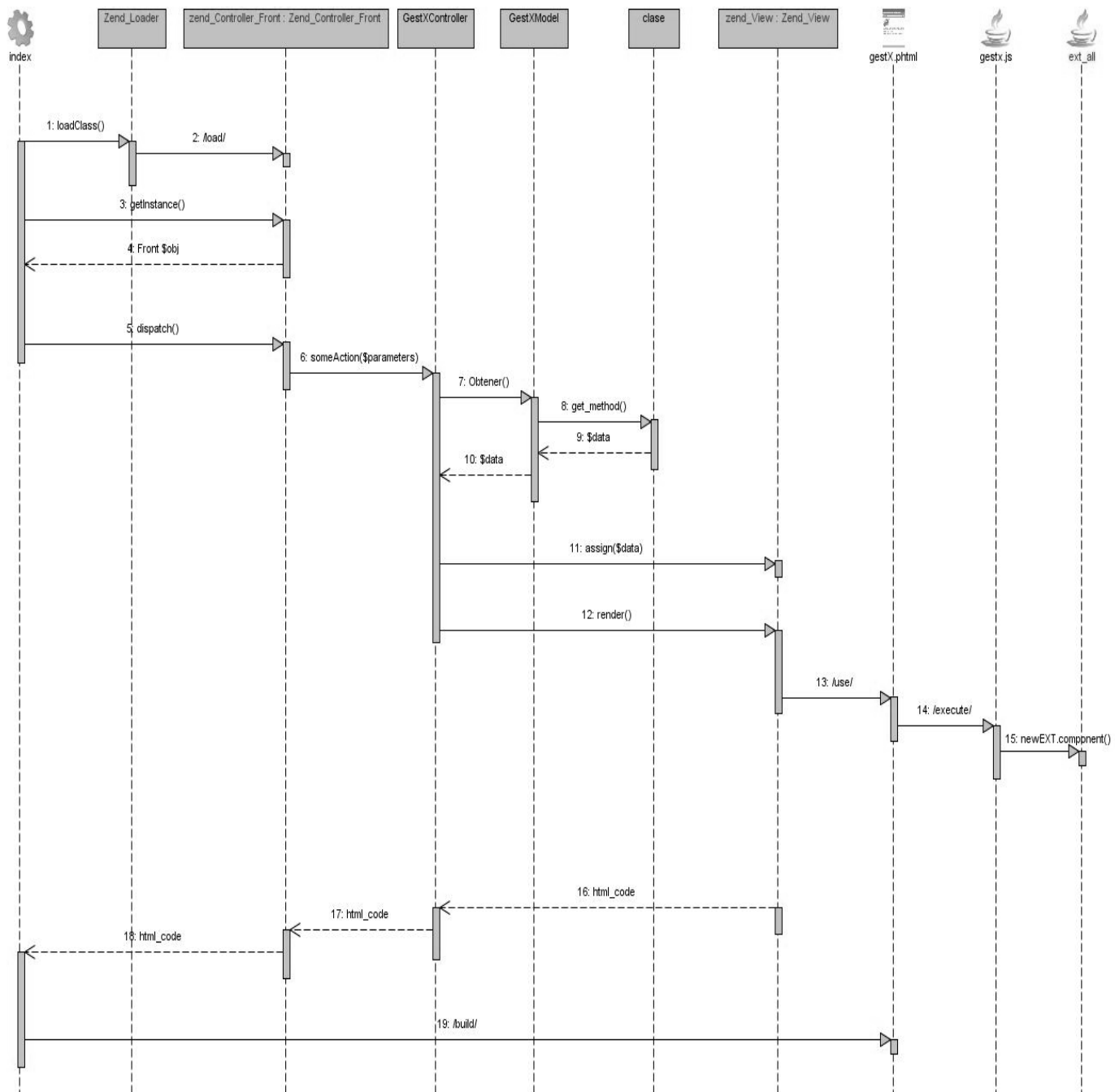


Los diagramas de secuencia muestran el intercambio de mensajes, se centran especialmente en el orden y el momento en que se envían los mensajes a los objetos.

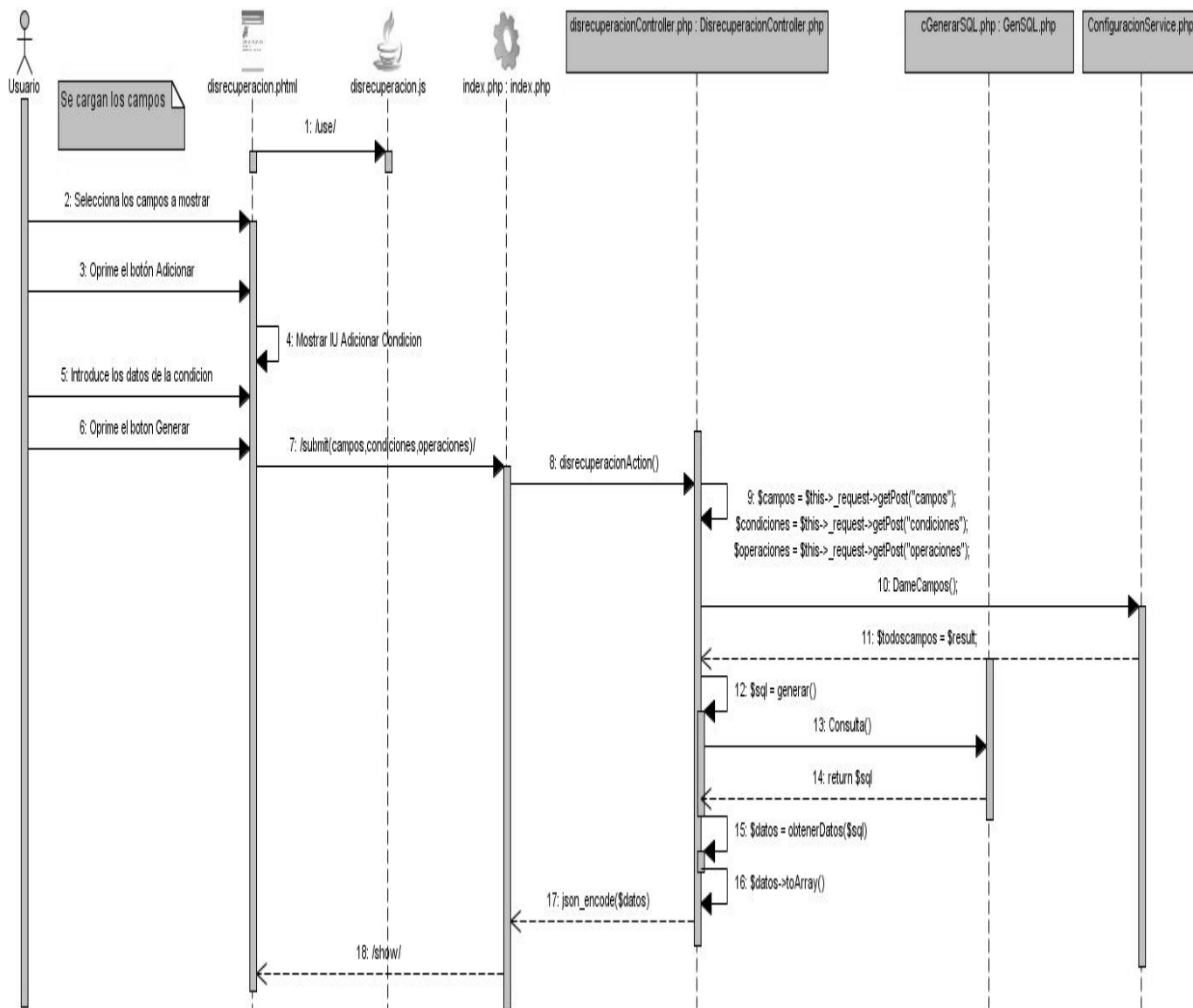
En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

Seguidamente se incorporan algunos de los diagramas de secuencia para los escenarios de los requisitos.

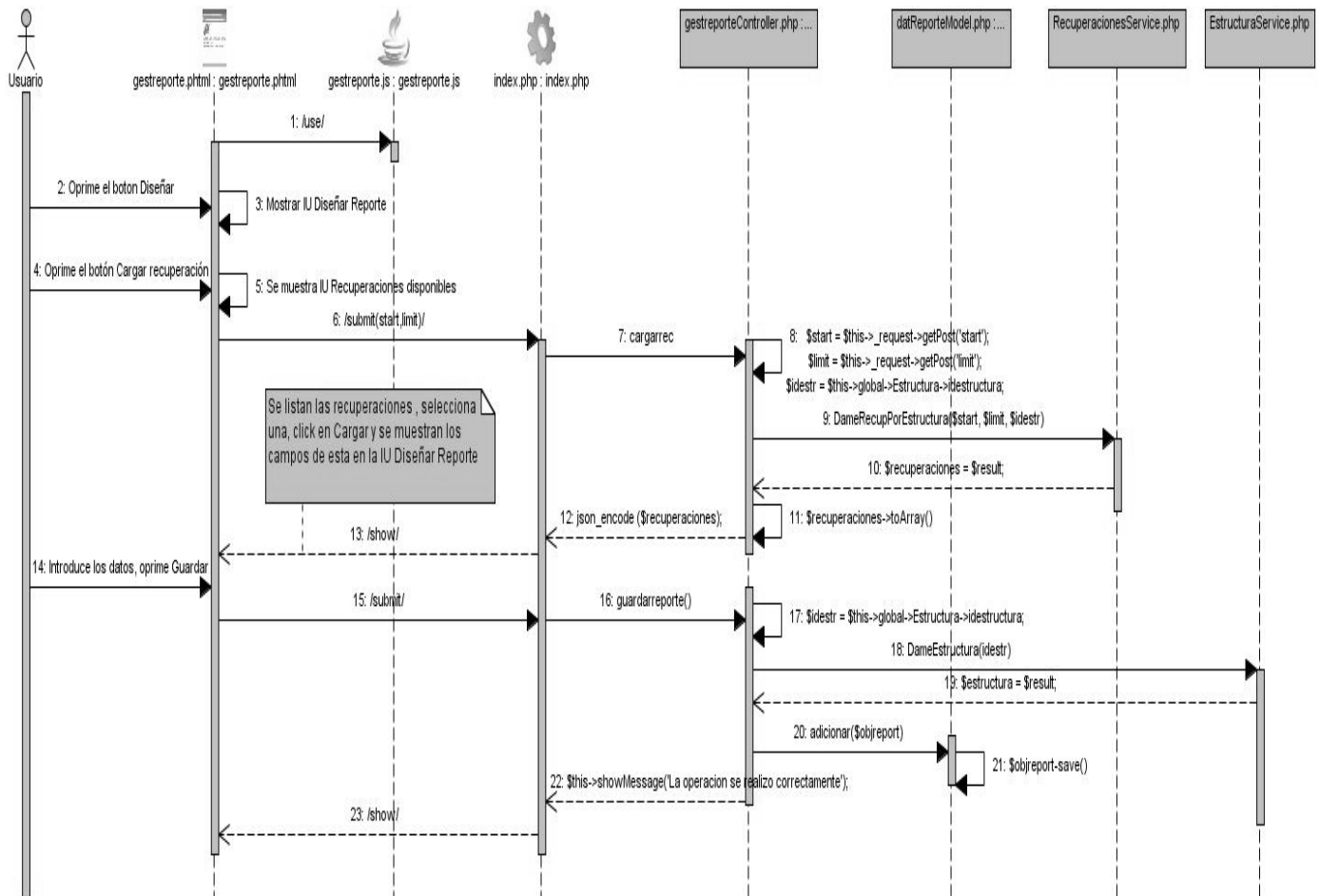
## Diagrama de secuencia genérico



## Diagrama de secuencia Diseñar recuperación.



**Diagrama de secuencia Diseñar reporte.**



### 3.3. Arquitectura

La Arquitectura de Software es la organización fundamental de un sistema representada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (12)

Generalmente en los sistemas se utilizan un conjunto de estilos de arquitectura tales como: Arquitectura basada en objetos, arquitectura basada en servicios (SOA) y arquitectura en n capas, Modelo Vista Controlador (MVC), siendo este último el que se decidió utilizar.

MVC plantea la separación de diferentes clases en dependencia de la función que realizan de modo tal que sea posible manejar dinámicamente la forma en que se procesan solicitudes y se gestiona la manera

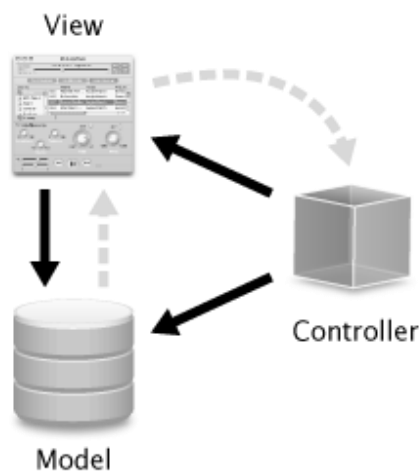
en que se muestran resultados al usuario final. En otras palabras separa la presentación del dominio de la aplicación. (24)

A continuación se brinda una explicación del funcionamiento del patrón MVC que es la base del estilo arquitectónico del mismo nombre.

**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.

**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.

**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. (25)



*Figura 3.1 Patrón Modelo Vista Controlador.*

### 3.4. Diseño de datos

El diseño de datos consiste en la representación lógica y física de los datos persistentes usados para el desarrollo del software.

#### 3.4.1. Diagrama entidad relación de la BD

El diagrama entidad-relación (DER) constituye una guía para el modelado de bases de datos, mediante el cual se visualizan los objetos que pertenecen a la base de datos como entidades las cuales tienen atributos y se vinculan mediante relaciones.

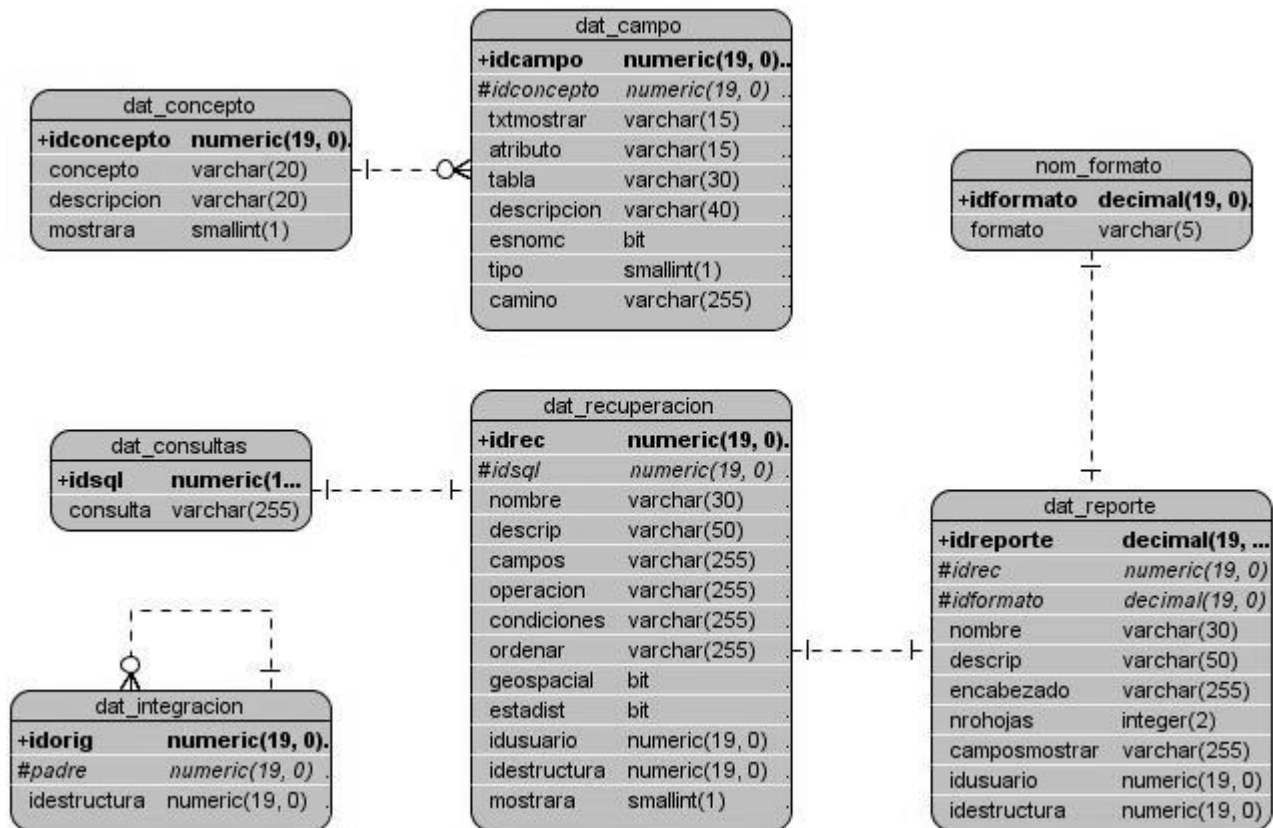


Figura 3.2 Diagrama entidad relación

### 3.5. Definiciones de diseño

El diseño de la aplicación tiene que basarse en el usuario. Es por eso que se deben tener en cuenta aspectos tales como navegabilidad, interactividad, usabilidad y accesibilidad. La aplicación web que se



pretende desarrollar, se diseñará con el objetivo de ayudar al usuario a comprender los contenidos informativos con una presentación cómoda, eficaz y clara, que permita al usuario navegar sin inconvenientes y encontrar rápidamente lo que busca. El usuario podrá acceder solo a la información que le haya sido conferida, en aras de preservar la integridad de la información.

### **3.6. Tratamiento de errores**

El tratamiento de los errores es un paso fundamental para el buen funcionamiento de un sistema, garantizando la armonía y facilidad de uso de la aplicación. El tratamiento de errores se realiza en un primer momento sobre la programación web del lado del cliente, utilizando como lenguaje de programación Javascript, el cual posibilita validar los elementos antes de que el usuario los envíe al servidor, además se encarga de la validación de la entrada de los datos en los formularios, se podrá comprobar que el formato de un campo sea el esperado y se comprobará además que no se sobrepasa la longitud, número de líneas o tamaño de la entrada de datos. De esta forma se reduce la cantidad de transacciones que se efectúan a través del protocolo HTTP y las posibilidades de que se genere un error durante la inserción de datos.

Estas validaciones que se realizarán del lado del cliente no sustituyen a la validación que deberá realizarse, por motivos de seguridad, en la aplicación del servidor que recibe la información, ya que puede darse el caso de que el cliente tenga deshabilitada la ejecución de script del lado del cliente y con esto la validación del lado del cliente no sería ejecutada.

Como parte también del tratamiento de los errores se le brindará una mínima información al usuario sobre el error, para que pueda recuperarse de este.

### **3.7. Seguridad**

Con el objetivo de hacer más seguras y confiables las aplicaciones es que se implementa un conjunto de medidas de seguridad que brindan un entorno seguro para los usuarios que trabajen con la aplicación. En AIDATAFAR se gestiona la seguridad a través de SIGIS<sup>1</sup> mediante el cual entre otras funcionalidades se puede gestionar funcionalidades, acciones, roles y usuarios; donde el acceso a las funcionalidades propias del sistema se especifica para cada rol y a cada usuario se le indica a cual estructura tiene permiso. Aunque esto es necesario, no es suficiente para asegurar la compartimentación de la

---

<sup>1</sup> Sistema de Gestión Integral de Seguridad

información, esto se resuelve internamente en el sistema a nivel de datos asociando el identificador de cada estructura con el cual un usuario accede a la aplicación, con el identificador de las estructuras que establece DATAFAR para su funcionamiento.

Además el sistema está diseñado para que ante una situación de error, brinde la mínima información posible. Para evitar cualquier intrusión, la aplicación web tendrá mecanismos que verifiquen que cualquier dato entrante o saliente es apropiado y esté en el formato que se espera.

### **3.8. Interfaz**

La interfaz es la parte de la aplicación donde interactúa la aplicación con el usuario. El sistema a desarrollar proporcionará al usuario una interfaz amigable, de acceso rápido y fácil de usar, esto se logra poniendo los controles de una forma que el usuario pueda intuir su objetivo, construyendo interfaces que no estén sobrecargadas y diseñadas para dar solución a requerimientos específicos, también se deben mostrar mensajes que informen al usuario de las acciones que se están realizando y siempre en un lenguaje común para este evitando malas interpretaciones. Para el desarrollo de las interfaces se tendrá en cuenta el estándar de diseño de interfaces para las aplicaciones de gestión. La información que sea reflejada en la aplicación web estará libre de errores gramaticales, ortográficos y tipográficos. Los iconos que se mostrarán en la aplicación tendrán correspondencia con la funcionalidad que realizan. Además el usuario podrá acceder a la ayuda sobre el trabajo con el sistema en caso de que tenga dudas con el uso del mismo.

### **3.9. Concepción de la ayuda**

La ayuda es uno de los elementos más importantes para los usuarios a la hora de interactuar con la aplicación, pues esta opción le permite conocer el funcionamiento de la misma, ya que el usuario final, que es para quien está dirigida la aplicación propuesta no tiene un alto dominio del uso de sistemas informáticos. En aras de resolver esta situación existirá una ayuda en línea que ilustrará las funcionalidades del sistema, mostrándose en ella detalladamente cada aspecto de dichas funcionalidades. Esto permitirá que el usuario este en todo momento informado y orientado cuando se encuentre trabajando en el sistema.

### 3.10. Conclusiones

En este capítulo se ha dado una panorámica sobre el diseño del software a realizar. Se obtienen una serie de artefactos claves para la implementación del sistema como son: los diagramas de clases web, los diagramas de secuencia, el modelo entidad relación de la base de datos. Además se describen patrones de diseño utilizados y los principales elementos de la arquitectura.

### Capítulo 4: Implementación y pruebas de la herramienta.

En el presente capítulo se modelarán varios artefactos que tienen lugar durante la implementación del sistema, que empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, de código binario, ejecutables, scripts. Además se obtendrán los casos de prueba con los cuales se podrán demostrar que las funciones del software son operativas.

#### 4.1. Implementación

La implementación comienza con el resultado obtenido del diseño detallado. El objetivo principal de este es desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la implementación son: (12)

- Planificar las integraciones de sistema necesarias en cada iteración. Siguiendo para ello un enfoque incremental.
- Implementar clases, componentes y subsistemas encontrados durante el diseño.
- Integrar componentes.

##### 4.1.1. Diagrama de componentes

Los diagramas de componentes ilustran las piezas del software, controladores embebidos que conformarán un sistema. Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. (26)

Seguidamente se representa el diagrama de componente del sistema.

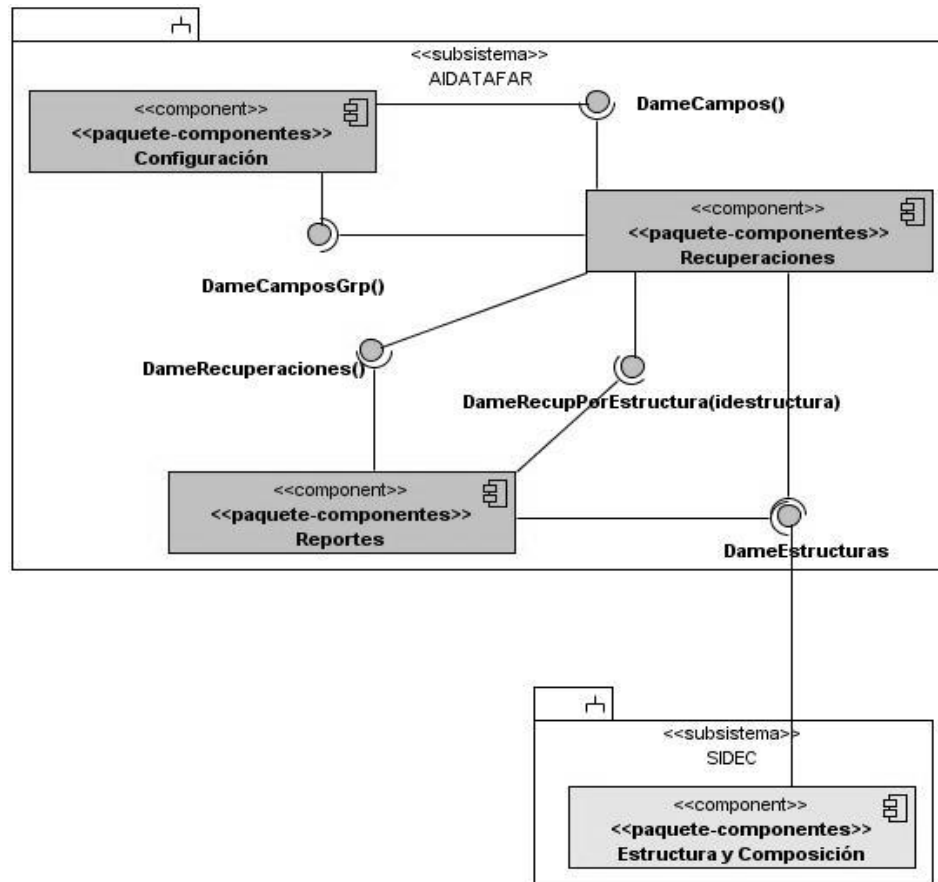


Figura 4.1 Diagrama de componentes

#### 4.1.2. Matriz de integración de componentes

La matriz de integración de componentes contiene todos los componentes definidos en el subsistema, de forma matricial, y en las intercepciones se especificarán los servicios que consume el componente en la vertical del horizontal. Existen dos matrices de integración de componentes, Interna y Externa.

##### 4.1.2.1. Matriz de Integración de Componentes Interna

En la matriz de integración de componentes Interna se especifica la integración entre los componentes internos del sistema.

		Componentes Internos		
Componentes Internos	Configuración	Recuperaciones	Reportes	
Configuración	–	–	–	
Recuperaciones	DameCampos()	–	–	
	DameCamposGrp()	–	–	
Reportes	–	DameRecuperaciones()	–	
	–	DameRecupPorEstructura(idestructura)	–	

**4.1.2.2. Matriz de Integración de Componentes Externa.**

En la matriz de integración de componentes Externa se especifica la integración entre los componentes externos del sistema.

		Componentes externos
Componentes Internos	Estructura y composición (SIDECA)	
Configuración	–	
Recuperaciones	DameEstructuras()	

Reportes	DameEstructuras()
----------	-------------------

### 4.2. Pruebas

Las pruebas constituyen una etapa imprescindible durante el proceso de desarrollo del software, pues permiten detectar y corregir el máximo de errores posibles antes de la entrega al cliente del software desarrollado, por lo que el éxito de las mismas puede mejorar la percepción de calidad del usuario final y lograr su satisfacción. Estas están compuestas por un conjunto de herramientas, técnicas y métodos que hacen operativo el desempeño de un programa.

Es importante considerar que las pruebas de software no garantizan que un sistema esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles para su debida corrección. (12)

Entre los diferentes tipos de pruebas que existen se encuentran: prueba de caja negra y prueba de caja blanca. El tipo de prueba que se utilizó para comprobar la calidad del producto realizado, es el de caja negra. Este se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Dentro de los métodos de caja negra se utilizó la técnica de la Partición de Equivalencia es una de las más efectivas pues permite dividir el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de pruebas. El objetivo de partición equivalente es reducir el posible conjunto de casos de prueba en uno más pequeño, un conjunto manejable que evalúe bien el software.

#### 4.2.1. Diseño de casos de prueba.

El propósito de un caso de prueba es especificar una forma de probar el sistema, incluyendo un conjunto de entradas válidas e inválidas, condiciones de ejecución, resultados esperados y las condiciones bajo las que ha de probarse. Los casos de pruebas deben verificar que el producto cumpla con los requerimientos del usuario y que se comporte como se desea.

Una vez realizados los diseños de caso de prueba, se realizaron las pruebas al sistema, estas se realizaron en varias iteraciones. En la primera iteración se obtuvo varias no conformidades de ellas 5 no significativas y 3 significativas. Las no significativas todas fueron errores gramaticales, por ejemplo en el mensaje de confirmación *¿Está seguro que desea eliminar un concepto?*, le faltaba la tilde. Las

significativas todas fueron errores en las validaciones, ejemplo en el campo nombre que deben entrar solo letras, se permitía introducir otros caracteres. Estos errores fueron corregidos para después realizar otra vez estas pruebas. Se pudo demostrar que el software tiene el mínimo de errores, que sus funcionalidades son operativas y que cumple con los requisitos de los usuarios.

### **4.3. Conclusiones**

En este capítulo se modelaron varios artefactos que tienen lugar durante la implementación del sistema, en términos de componentes, es decir, ficheros de código fuente, de código binario, ejecutables, scripts. Además se realizaron las pruebas de caja negra al sistema a través de los diseños de casos de prueba con los cuales se detectaron varias no conformidades que fueron corregidas, lo cual demostró que las funciones del software son operativas.



### **Conclusiones**

Para la confección del presente trabajo de diploma se estudiaron los procesos que se llevan a cabo con los subsistemas de Reclutamiento y Reserva, llegando de esta manera a la conclusión de la necesidad que existe actualmente de desarrollar una herramienta informática para el análisis de información que facilite la toma de decisiones para estos subsistemas.

Se realizó un análisis de las diferentes metodologías, tecnologías, herramientas y lenguajes más utilizados a nivel nacional e internacional a la hora de desarrollar este tipo de aplicaciones. Se definieron todos los diagramas o artefactos que se encuentran relacionados con cada etapa según el proceso de desarrollo de software.

Se obtuvo como resultado una aplicación web que cumple con los requisitos de los clientes. Por lo anteriormente expuesto se concluye que se han cumplido satisfactoriamente los objetivos trazados para la confección del presente trabajo de diploma; obteniendo como resultado un producto informático que facilita el proceso de toma de decisiones.

### **Recomendaciones**

Le realización del presente trabajo cumplió con el objetivo general propuesto, aunque en el transcurso de su desarrollo surgieron ideas que podrían implementarse en un futuro, de forma que se logre una la aplicación más útil, por lo cual se recomienda:

- Realizar otras pruebas que permitan asegurar la calidad del producto.
- Integrar la herramienta con el sistema DATAFAR, o sea, que los reportes diseñados con esta aplicación puedan ser consultados desde DATAFAR.
- Mejorar la generación dinámica de consultas, para optimizar estas, para un mejor rendimiento del software.

## Bibliografía

**Arias, Yuniel Eliades Proenza. 2009.** Diseño Avanzado de Aplicaciones Web. EXT – Zend Framework y Doctrine. *Revista de Software Libre ATIX*. [Revista]. 09 de 03 de 2009. Vol. 8, pág. 78.

**Cabreja Muñoz, Oscar y Rivero León, Darys Angela. 2009.** *Módulo de potencial humano para el sistema de Comités Militares, procesos de asignación y empleo*. Ciudad de La Habana : s.n., 2009.

**Frederick, Shea, Ramsay, Colin y Blades, Steve. 2008.** *Learning Ext JS*. Birmingham : Packt Publishing Ltd, 2008.

**GESTIÓN-INFORMACIÓN-CONOCIMIENTO. Murray, Pablo. 2002.** 014, Lima, Perú : Universidad Peruana de Ciencias Aplicadas, octubre-diciembre de 2002, Biblios Revista Electrónica de Ciencias de la Información, Vol. 4. 1562-4730.

**Hernández, Lic. Iliana Amabely Silva. 2003.** *Generador Automático de Reportes Dinámicos*. Dpto. de Ingeniería Electrónica Sección de Computación, Centro de Investigación y de Estudios Avanzados del I.P.N. México, D.F. : s.n., 2003. Tesis de maestría.

**López Camino, Michel y Muñoz Vidal, Raydel. 2006.** *Gestor de Recuperaciones Dinámicas para Aplicaciones Web*. Centro de Estudios de Ingeniería de Sistemas, Instituto Superior Politécnico "José Antonio Echeverría". Ciudad de La Habana : s.n., 2006.

**MINFAR. MANUAL PARA EL TRABAJO DE LOS COMITÉS MILITARES.** [Documento]. Ciudad de La Habana : s.n.

**Morell Guerra, Ing. Tte. Persy.** Paquete de Ayuda a la Toma de Decisiones y Sistemas

**UCID. 2010.** *Estándar de codificación para lógica de negocio*. 2010.

—. **2010.** *Estándar de diseño de interfaces para las aplicaciones de gestión*. 2010.

—. **2010.** *Estándar de documentación*. 2010.

—. **2010.** *Estándar de Ingeniería de Software de diseño*. 2010.

—. **2010.** *Estándar de Ingeniería de Software para analistas*. 2010.

—. **2010.** *Estándares para el desarrollo de la UM 1722*. 2010.

—. **2010.** *Políticas y Estándares del Rol Arquitectura de Datos*. 2010.

—. **2009.** *Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión)*. Ciudad de La Habana : s.n., 2009.

**Vesterinen, Konsta y Wage, Jonathan H. 2008.** *Doctrine Manual*. 2008.

### Referencias bibliográficas

1. **Faga, Roberto.** *Cómo profundizar en el análisis de sus costos para tomar mejores decisiones empresariales.* Buenos Aires : s.n., 2000.
2. **Cabreja Muñoz, Oscar y Rivero León, Darys Angela.** *Módulo de potencial humano para el sistema de Comités Militares, procesos de asignación y empleo.* Ciudad de La Habana : s.n., 2009.
3. Soluciones Crystal Reports. *SAP Mexico.* [En línea] [Citado el: 25 de enero de 2010.] <http://www.sap.com/mexico/solutions/sapbusinessobjects/sme/reporting/index.epx>.
4. Oracle Reports Services 11g. *Oracle Iberia.* [En línea] [Citado el: 25 de enero de 2010.] <http://www.oracle.com/technology/products/reports>.
5. **Bermúdez Peña, Rolando, Castro Mecías, Lilian Teresa y Rodriguez Rodriguez, Yaily.** *SOFTWARE PARA LA CREACIÓN Y GENERACIÓN DE REPORTE EN JAVA.* Ciudad de La Habana : s.n.
6. Paquete de Herramientas para la Ayuda a la Toma de Decisiones. *Portal CENTALAD.* [En línea] [Citado el: 25 de enero de 2010.] <http://portal.centalad.prod.uci.cu/projects/show/patdsi>.
7. Various Licenses and Comments about Them. *GNU Operating System.* [En línea] [Citado el: 3 de febrero de 2010.] <http://www.gnu.org/licenses/license-list.html>.
8. **Araujo Pérez, Damaris y González Nuzzo, Yunior Alberto.** *Sistema para el control de Recursos Humanos. Proceso de reclutamiento en los niveles de dirección.* Ciudad de La Habana : s.n., 2007.
9. **Garrett, Jesse James.** Ajax: A New Approach to Web Applications. *Adaptive Path.* [En línea] 18 de febrero de 2005. [Citado el: 1 de febrero de 2010.] <http://adaptivepath.com/ideas/essays/archives/000385.php>.
10. HTML & CSS. *World Wide Web Consortium.* [En línea] [Citado el: 1 de febrero de 2010.] <http://www.w3.org/standards/webdesign/htmlcss>.
11. **Letelier, Patricio y Penadés, M<sup>a</sup> Carmen.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] [Citado el: 2 de febrero de 2010.] <http://www.willydev.net/descargas/masyxp.pdf>.
12. **UCID.** *Proceso de Desarrollo y Gestión de Proyectos de Software (1ra Versión).* Ciudad de La Habana : s.n., 2009.
13. Curso práctico de Modelado de Negocios con UML y BPMN. *Milestone Consulting.* [En línea] [Citado el: 5 de febrero de 2010.] <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.
14. Ingeniería del Software de Gestión. [En línea] 21 de julio de 2007. [Citado el: 18 de enero de 2010.] <http://wwwdi.ujaen.es/asignaturas/isg/Software.html>.

15. Sistema Gestor de base de datos SGBD. *Garbage Collector*. [En línea] 1 de noviembre de 2004. [Citado el: 3 de febrero de 2010.] [http://www.error500.net/garbagecollector/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_dato.html](http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_dato.html).
16. CodeBox. *Glosario*. [En línea] <http://www.codebox.es/glosario>.
17. Ext JS: Cross-Browser Rich Internet Application Framework. *Ext JS*. [En línea] [Citado el: 5 de febrero de 2010.] <http://extjs.com/products/extjs/>.
18. Introducción al Zend Framework. [En línea] 13 de julio de 2006. [Citado el: 5 de febrero de 2010.] <http://sentidoweb.com/2006/07/13/introduccion-al-zend-framework.php>.
19. Doctrine and symfony frameworks collaborating. *Doctrine*. [En línea] [Citado el: 6 de febrero de 2010.] <http://www.doctrine-project.org/blog/doctrine-and-symfony-frameworks-collaborating>.
20. Patrón Factory. *Comunidad de Programadores*. [En línea] [Citado el: 3 de 05 de 2010.] <http://lawebdelprogramador.com/>.
21. Web 2.0 Development. [En línea] 31 de 10 de 2007. [Citado el: 03 de 05 de 2010.] <http://web2development.blogspot.com/>.
22. **Artola, Luis**. *Programania*. [En línea] 03 de 06 de 2008. [Citado el: 03 de 05 de 2010.] <http://www.programania.net>.
23. **Guerrero, Luis A**. UML - Diagramas de interacción. [Presentación].
24. **Arias, Yuniel Eliades Proenza**. Diseño Avanzado de Aplicaciones Web. EXT – Zend Framework y Doctrine. *Revista de Software Libre ATIX*. [Revista]. 09 de 03 de 2009. Vol. 8, pág. 78.
25. **Catalani, Exequiel**. Arquitectura Modelo/Vista/Controlador. [En línea] 20 de 08 de 2007. [Citado el: 03 de 05 de 2010.] <http://exequielc.wordpress.com>.
26. *Esparx systems, Plataforma avanzada de modelado y diseño*. [En línea] [Citado el: 03 de 05 de 2010.] <http://www.sparxsystems.com.ar>.

### Glosario de términos

**API:** interfaz de programación de aplicaciones, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**Autenticación:** la autenticación es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse.

**Código abierto:** (en inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente.

**CSV:** son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla.

**C#:** es un lenguaje orientado a objetos moderno, simple y poderoso. Fue desarrollado por Microsoft y posteriormente estandarizado. Fue creado para desarrollar aplicaciones orientadas a objetos. Incorpora las características de un lenguaje de última generación.

**EJBQL:** (*Enterprise JavaBeans Query Language*), es un lenguaje de consulta de bases de datos.

**Frameworks:** ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez. Los frameworks son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos.

**HQL:** (*Hibernate Query Language*, lenguaje de consulta para Hibernate), es completamente orientado a objetos y comprende nociones de herencia, polimorfismo y asociación.

**IDE:** Un entorno de desarrollo integrado, es un programa informático compuesto por un conjunto de herramientas de programación.

**Inteligencia Empresarial:** conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

**Java:** es un lenguaje de programación de alto nivel, orientado a objetos desarrollado por Sun Microsystems.

**JavaBeans:** son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

**MDX:** (*MultiDimensional eXpressions*) es un lenguaje de consulta para bases de datos OLAP.

**Modelo OSI:** El modelo de referencia de Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection*).

**ODF:** (*Open Document Format*), es un formato de fichero estándar para el almacenamiento de documentos.

**PDF:** (acrónimo del inglés *Portable Document Format*, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

**PL/SQL:** Lenguaje de programación incrustado en Oracle y PostgreSQL, soporta todas las consultas y manipulación de datos que se usan en SQL.

**PL/pgSQL:** (*Procedural Language/PostgreSQL Structured Query Language*) es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas.

**RTF:** formato de texto enriquecido.

**SQL:** lenguaje de consulta estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales.

**TXT:** archivos de texto plano, son aquellos que están compuestos únicamente por texto sin formato, sólo caracteres.

**XLS:** extensión para los archivos de hoja de cálculo Microsoft Excel.

**XPath:** (*Path Language*) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.