

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**Facultad 9**



**Subsistema de Almacenamiento y Gestión  
de Archivos para el producto Captura y Catalogación de  
Medias.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO  
DE INGENIERO EN INFORMÁTICA**

**AUTOR:** Reynier Martínez Pérez

**TUTOR:** Ing. Aneli Valdés Acosta

**Ciudad de La Habana, 6 de julio de 2010**  
**“Año 52 de la Revolución”**



## DEDICATORIA

Dedico este trabajo de forma muy especial a las personas más importantes de mi vida. Primeramente a mis fantásticos padres que con su cariño, confianza y apoyo han permitido que hoy me encuentre realizado profesionalmente.

A la mitad de mi vida, mi hermano del alma, con el cual me hubiese gustado compartir el desarrollo de este trabajo. A Mi gran amor y dueña de mis pensamientos "MAMITA. A mi abuela Maida, Mary, el Caña, al resto de mi familia y mis amigos queridos, además de todas las personas que me desean éxito y prosperidad.

## AGRADECIMIENTOS

Agradecer antes que todo a las dos personas que me dieron la vida: mis padres queridos, gracias a ustedes he podido llegar a convertirme en un hombre de bien, he aquí la muestra de que los esfuerzos no fueron en vano. Gracia por su confianza, amor, entrega y dedicación. Gracias por siempre respetar y apoyar mis decisiones, por mantener nuestra familia unida y por tratar de que cada día sea mejor para todos, gracias por existir.

A mi mujer “Mamita, sin ella no hubiera podido soportar estos 5 largos años. Siempre estaba ahí en los momentos más difíciles con sabios consejos, con alegría, que en algunos momentos era el resultado de un gran esfuerzo, solo para mantener mi estabilidad. Gracias por ser mi inspiración en el camino por la búsqueda del estudiante perfecto. Gracias a ti mi amor por ser tan maravillosa y por permitirme haberte conocido.

A mí querido hermano Reymundito, por haberme acompañado siempre en la vida, incluso hasta esta escuela. Gracias por al menos haberlo intentado. Tu decisión me ayudó a fortalecer mi propósito para poder darles este regalo a nuestros padres, así que este triunfo vale doble para mí. Gracias por ser el mejor hermano del mundo.

A mi abuela Maida por cuidar de mí con tanto cariño desde mis primeros años de vida, por su entrega y dedicación. Has sido para mí, ejemplo de lucha y firmeza. Gracias por todo.

A Mary y el Caña, los mejores suegros del mundo. Gracias por tanto cariño y confianza desde el primer día. Por permitirme cuidar de su tesoro y por haberme acogido cariñosamente en el seno de su familia.

Al resto de mi familia por la preocupación y el interés que han mostrado en mis estudios y mi bienestar. Especialmente mis padrinos que desde muy pequeño han sabido quererme como el hijo que no pudieron tener.

A todos mis amigos, por aceptarme como soy y por permitirme haberlos conocidos. Especialmente resaltar a Yakelin, Dayrón y Pepor, son mis grandes conquistas en amistades durante mi vida en la UCI y

lo mejor que me ha sucedido en todo este tiempo de estudiante universitario. Gracias a ustedes he podido sobrevivir a la soledad por la ausencia de mi hermano. Es por eso que ocuparán siempre un lugar muy importante en mi corazón.

A mis compañeros de grupo, desde la vieja guardia hasta los actuales. Ustedes son lo mejor dentro y fuera del aula. Siempre he creído que dentro de cada uno de ustedes se esconde una gran persona.

Agradecer a mis compañeros de cuarto, en especial a Yumar, Pepe, Olivier, Yosiel mi barbero UCI, el Fale, Yano, Jean, el Vin, Cuza y a YUYI que aunque no es de nuestro cuarto fue donde tuve el placer de conocerla. Gracias a todos ustedes por los buenos momentos.

A mi tutora Aneli, quedaré agradecido infinitamente por la ayuda prestada en el desarrollo de este trabajo, así como a las demás personas que aportaron su grano de arena en el desarrollo de este trabajo: Mónica y Yunior.

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Señales Digitales de la **Universidad de las Ciencias Informáticas** a hacer uso del mismo en su beneficio.

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

**Autor:** \_\_\_\_\_

**Reynier Martínez Pérez**

**Tutor:** \_\_\_\_\_

**Ing. Aneli Valdés Acosta**

**DATOS DE CONTACTO**

<Insertar breve currículum e información de contacto del tutor>

<Insertar breve currículum e información de contacto del co-tutor>

<Insertar breve currículum e información de contacto del consultor>

## RESUMEN

La información juega un papel muy importante en la actualidad ya que es la base para la toma de decisiones. El desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) ha posibilitado que su proceso de gestión sea más eficiente. La llegada de la informática a las esferas de la sociedad como parte de la era de la digitalización, ha permitido encapsular grandes volúmenes de datos en secuencias de bits que son almacenados en archivos informáticos. Según las necesidades y el contenido que portan dichos archivos, cada día alcanzan mayores proporciones, por lo que el proceso de gestión resulta muy complicado. La UCI surgió con el propósito de brindar soluciones de software para la automatización de procesos y convertirse en una de las principales fuentes de ingresos económicos para el país. El departamento de Señales Digitales de la facultad 9 desarrolla una variada gama de productos orientados al trabajo con ficheros multimedia, el producto de Captura y Catalogación de Medias no cuenta con un subsistema que le permita gestionar el almacenamiento de los archivos. Con el propósito de dar solución al problema detectado se ha desarrollado la presente investigación, la misma cuenta con una estructura de 4 capítulos.

**1<sup>ro</sup>:** Contiene una serie de conceptos que resultan imprescindibles para la comprensión de la solución propuesta y para el dominio del problema.

**2<sup>do</sup>:** Resume las tendencias y tecnologías actuales que son utilizadas en el desarrollo de las solución, así como los principales criterios que conllevaron a su selección.

**3<sup>ro</sup>:** Se puede encontrar en detalles la solución propuesta, comenzando por los procesos de negocio, hasta la descripción de las capacidades y cualidades que el sistema debe tener.

**4<sup>to</sup>:** Se tiene una vista del sistema desarrollado en términos de clases de diseños y componentes, así como algunos aspectos generales de la implementación y su distribución física.

## PALABRAS CLAVES

Archivo, captura, catalogación, transcripción, socket.

## ÍNDICE DE FIGURAS

Figura 1: Representación de funcionamiento de un socket.....	16
Figura 2: Imagen representativa de la estructura de RUP. ....	18
Figura 3: Diagrama de Caso de Uso del Negocio.....	33
Figura 4: Diagrama de Caso de Uso del sistema .....	39
Figura 5: Paquete Interfaz (Diseño de clases).....	50
Figura 6: Paquete de Logica de Negocio (Diseño de clases).....	51
Figura 7: Paquete de Acceso a Datos (Diseño de clases). ....	52
Figura 8: Diagrama de clases del diseño (Interacción entre paquetes). ....	53
Figura 9: Diagrama de Despliegue.....	55
Figura 10: Nodo Sistema de Almacenamiento.....	56
Figura 11: Nodo Cliente. ....	56
Figura 12: Nodo Servidor de BD .....	56
Figura 13: Diagrama de componente. Paquete Interfaz. ....	57
Figura 14: Diagrama de Componentes. Paquete Negocio. ....	58
Figura 15: Diagrama de Componentes. Paquete de Acceso a Datos.....	59
Figura 16: Modelo de implementación. ....	60

## ÍNDICE DE TABLAS

Tabla 1: Descripción de Actores del Negocio .....	32
Tabla 2: Descripción de Trabajadores del Negocio .....	32
Tabla 3: Descripción detallada del caso de uso Almacenar Archivo .....	34
Tabla 4: Descripción detallada del caso de uso recuperar Archivo.....	34
Tabla 5: Descripción de actores del sistema .....	39
Tabla 6: Descripción del CU Descargar Archivo.....	40
Tabla 7: Descripción del CU Controlar Servidor. ....	43
Tabla 8: Descripción del CU Gestionar Configuración. ....	46
Tabla 9: Descripción del CU Subir Archivo .....	48

**ÍNDICE DE CONTENIDO**

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 INTRODUCCIÓN .....	5
1.2 GESTIÓN Y ALMACENAMIENTO DE ARCHIVOS .....	5
Sistemas de Gestión de Archivos.....	6
1.3 PROCESOS DE ALMACENAMIENTO Y GESTIÓN DE ARCHIVOS. ....	6
1.3.1 Descripción General .....	6
Almacenamiento virtual de archivos .....	7
Sistemas de almacenamiento.....	7
1.3.2 Concepción del almacenamiento y gestión de archivos en el producto CCM. ....	8
1.3.3 Situación Problemática.....	10
1.4 ANÁLISIS DE SOLUCIONES EXISTENTES .....	11
1.5 CONCLUSIONES .....	15
<b>CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR</b> .....	<b>16</b>
2.1 INTRODUCCIÓN .....	16
2.2 TECNOLOGÍAS QUE SOPORTAN LA SOLUCIÓN.....	16
2.2.1 Socket .....	16
2.3 METODOLOGÍAS DE DESARROLLO .....	17
2.3.1 Metodologías tradicionales.....	17
<i>Proceso Unificado de Desarrollo (RUP)</i> .....	17
2.3.2 Metodologías ágiles .....	19
<i>Programación Extrema (XP)</i> .....	19
<i>Scrum</i> .....	19
2.3.3 Fundamentación de la metodología propuesta: RUP .....	20
2.4 EL LENGUAJE UNIFICADO DE MODELADO (UML) COMO SOPORTE DE LA MODELACIÓN DE LA SOLUCIÓN PROPUESTA .....	20
2.5 SISTEMAS GESTORES DE BASES DE DATOS (SGBD) .....	21
2.5.1 Oracle.....	21
2.5.2 MySQL .....	21

2.5.3 PostgreSQL .....	22
2.5.4 Fundamentación del SGBD propuesto: <i>PostgreSQL</i> .....	22
2.6 FRAMEWORK DE ACCESO A DATO USADO: <i>HIBERNATE</i> .....	23
2.7 LENGUAJES DE PROGRAMACIÓN .....	23
2.7.1 Java .....	24
2.7.2 C++ .....	24
2.7.3 C# .....	25
2.7.4 Python .....	25
2.7.5 Fundamentación del lenguaje propuesto: Java .....	25
2.8 ENTORNOS DE DESARROLLO INTEGRADO ( <i>INTEGRATED DEVELOPMENT ENVIRONMENT</i> ) .....	26
2.8.1 Eclipse .....	27
2.8.2 NetBeans .....	27
2.8.3 Fundamentación del IDE propuesto: NetBeans 6.8 .....	27
2.9 HERRAMIENTAS CASE .....	28
2.9.1 Rational Rose Enterprise .....	28
2.9.2 Visual Paradigm .....	29
2.9.3 Fundamentación de la herramienta propuesta: <i>Visual Paradigm</i> .....	30
2.10 CONCLUSIONES .....	30
<b>CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>32</b>
3.1 INTRODUCCIÓN .....	32
3.2 MODELO DE NEGOCIO .....	32
3.2.1 Actores y trabajadores del negocio .....	32
3.2.2 Procesos de negocio .....	32
3.2.3 Diagrama de Casos de Uso del Negocio .....	33
3.2.4 Descripción textual de los Casos de Uso de Negocio .....	33
3.3 REQUERIMIENTOS FUNCIONALES .....	35
3.4 REQUERIMIENTOS NO FUNCIONALES .....	36
3.5 DESCRIPCIÓN DEL SISTEMA PROPUESTO .....	38
3.5.1 Descripción de los actores .....	38
3.5.2 Casos de Uso del Sistema .....	39
Descripciones de casos de uso del sistema .....	39

3.6 CONCLUSIONES .....	48
<b>CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>49</b>
4.1 INTRODUCCIÓN .....	49
4.2 DIAGRAMAS DE CLASES DEL DISEÑO.....	49
4.2.1 Capa de Presentación .....	50
4.2.2 Capa de Lógica del Negocio.....	51
4.2.3 Capa de Acceso a Datos .....	52
4.2.4 Diagrama general de clases del diseño. ....	53
4.3 PRINCIPIOS DEL DISEÑO .....	53
4.3.1 Estándares de la Interfaz de la aplicación .....	53
4.3.2 Generalidades de la Implementación.....	54
4.3 MODELO DE DESPLIEGUE.....	55
4.5 MODELO DE IMPLEMENTACIÓN .....	57
4.5.1 Diagrama de componentes. Paquete Interfaz. ....	57
4.5.2 Diagrama de Componentes. Paquete Negocio. ....	57
4.5.3 Diagrama de Componentes. Paquete de Acceso a Datos .....	59
4.5.4 Modelo de implementación.....	60
<b>CONCLUSIONES.....</b>	<b>62</b>
<b>RECOMENDACIONES.....</b>	<b>63</b>
<b>BIBLIOGRAFÍA.....</b>	<b>64</b>
<b>GLOSARIO .....</b>	<b>69</b>

## Introducción

En la actualidad la información es la base de toda entidad, sin ella no sería posible desarrollar muchas de las actividades que se encuentran concebidas dentro de estas y sobre las cuales se toman muchas decisiones. Con el desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) y su inserción en diferentes sectores de la sociedad, se ha hecho posible que el procesamiento y gestión de dicha información se realice de manera más eficiente.

La inclusión de la informática en estos sectores ha provocado que grandes volúmenes de información hayan sido digitalizados y que muchos procesos se ejecuten de forma automática. Grandes cúmulos de datos son almacenados en archivos informáticos, estos cada vez alcanzan mayor proporción y necesitan de grandes dispositivos para ser almacenados y por consiguiente la gestión de los mismos se torna cada vez más difícil.

Ante la creciente cantidad de datos que manejan hoy las empresas, los profesionales de las tecnologías de información enfrentan el desafío de administrar datos heterogéneos, guardados en diferentes lugares de las redes corporativas, así como en variedades de dispositivos.

Es por eso que los especialistas de estas tecnologías tienen una mayor necesidad de adquirir las herramientas necesarias para administrar y gestionar la información de una forma segura, eficaz, y que permita ser encontrada rápidamente. El problema del almacenamiento se ha convertido en un negocio, tanto así, que muchas empresas se han dedicado a explotar las tecnologías para fabricar servidores con grandes capacidades de almacenamiento, con el fin de satisfacer las necesidades y alcanzar la supremacía en el mercado. Dentro de las empresas reconocidas que se desatacan por la calidad de sus productos se encuentran HP, Dell, Toshiba, Sony, entre otras.

No solo la producción de hardware ha demostrado ser una vía importante de desarrollo económico. El desarrollo de software ha demostrado ser un factor importante para la economía de cualquier país o institución, por lo cual nuestra nación ha alcanzado un gran avance en este sector en los últimos años. La creación de la Universidad de las Ciencias Informáticas (UCI), primera universidad construida en Cuba al calor de la Batalla de Ideas, ha tomado un papel protagónico en la impulsión de este sector, aprovechando

las posibilidades que ofrecen las TICs para brindar servicios y desarrollar soluciones de software con el fin de automatizar muchos procesos y proporcionar avances en la economía del país.

Tomando en cuenta los análisis realizados sobre el mundo del audiovisual y apreciando las posibilidades de mercado que tienen los productos de este tipo, y bajo las políticas del software libre como solución más viable para lograr la independencia tecnológica del país, en el Centro de Desarrollo de Geoinformática y Señales Digitales de la Facultad 9 se está desarrollando un producto de software con el fin de capturar y catalogar archivos multimedia<sup>1</sup>, específicamente información de audio y video.

El producto Captura y Catalogación de Media (CCM) se encuentra estructurado por 4 subsistemas:

- ✚ Subsistema de Audio.
- ✚ Subsistema de Video.
- ✚ Subsistema de Catalogación.
- ✚ Subsistema de Administración.

En el desarrollo de las actividades dentro de estos subsistemas se hace imprescindible el uso de las medias como material principal, por lo que se hace necesario contar con un servidor de medias profesional como solución óptima para poder almacenar la información y garantizar la disponibilidad y centralización de esta, facilitando un correcto funcionamiento de las diferentes aplicaciones. El personal que se encuentra al frente del desarrollo del producto ha hecho un análisis de los precios de un servidor de medias capacitado para realizar las operaciones deseadas, concluyendo que, ante la situación económica existente en el país no es posible efectuar la compra de uno de estos productos por el elevado costo que representa. Una inversión de este tipo disminuiría las posibilidades de mercado a la hora de su comercialización debido al incremento de los costos, lo que provocaría un aumento del precio del producto final.

Dados estos antecedentes es necesario contar con una herramienta que cumpla con las expectativas y tendencias actuales en cuanto a tratamiento y gestión de la información se refiere y que brinde a sus usuarios una interfaz simple y agradable que facilite la gestión de archivos de forma rápida y eficaz. Con

---

<sup>1</sup> **Multimedia:** Contenido relacionado con imágenes, audio y video, en numerosos medios de comunicación e informaciones oficiales se puede encontrar por: **Media.**

esta solución se logrará una mayor compatibilidad entre los sistemas, ya que deberá ser implementada bajo estándares y tecnologías libres. Permitirá establecer un protocolo de comunicación propio definido por las necesidades existentes, facilitará la comprensión de su funcionamiento para posibles mantenimientos, disminuirá los costos y el desarrollo del producto CCM no se verá comprometido por la falta de un servidor de medias profesional.

Ante la presente situación se ha determinado como **problema a resolver** la inexistencia de un subsistema para el almacenamiento y la gestión de archivos en el producto Captura y Catalogación de Medias del Centro de Desarrollo de Geoinformática y Señales Digitales de la Facultad 9. Para dar solución al problema detectado se ha planteado como **objetivo general** desarrollar un prototipo funcional de un subsistema para el almacenamiento y la gestión de archivos para el producto Captura y Catalogación de Medias, teniendo como **objeto de estudio** a los procesos de almacenamiento y gestión de archivos.

Se ha definido como **campo de acción** la automatización de los procesos de almacenamiento y gestión de archivos en plataformas libres en el producto Captura y Catalogación de Medias del Centro de Desarrollo de Geoinformática y Señales Digitales. **Defendiendo la idea** de que con la utilización de una aplicación que permita almacenar y gestionar archivos se garantizará el almacenamiento y la gestión de los mismos de manera exitosa y automáticamente en el producto de Captura y Catalogación de Medias del Centro de Desarrollo de Geoinformática y Señales Digitales de la Facultad 9.

Encaminado a alcanzar el objetivo anteriormente planteado se han establecido las **tareas de investigación** mostradas a continuación:

1. Caracterizar los procesos de almacenamiento y gestión de archivos a nivel internacional y nacional.
2. Caracterizar los procesos concebidos en el producto Captura y Catalogación de Medias relacionados con el almacenamiento y la gestión de archivos.
3. Describir el estado del arte de las aplicaciones para el almacenamiento y gestión de archivos.
4. Determinar las tecnologías y herramientas a utilizar para el desarrollo de la aplicación.
5. Analizar y diseñar el sistema de Almacenamiento y Gestión de Archivos.
6. Implementar un prototipo funcional del subsistema que permita el Almacenamiento y la Gestión de Archivos.

La realización de la investigación está dirigida mediante **métodos científicos** que se fueron aplicando durante su desarrollo.

## **Métodos Teóricos**

**Histórico Lógico:** Se usó con el objetivo de conocer la evolución, el estado y las tendencias actuales de los sistemas de gestión y almacenamiento de archivos en el mundo, con los cuales se le podría dar solución al problema expuesto con anterioridad, existente en el producto de Captura y Catalogación de Medias. Se realizó una revisión en diferentes bibliografías, dígame revistas digitales, sitios web corporativos de empresas que se dedican al desarrollo de productos para la captura y catalogación de medias, así como sistemas especializados en la gestión y almacenamiento de archivos, artículos, entre otras fuentes, con el objetivo de determinar las plataformas de desarrollo más comunes, lenguajes de programación más utilizados y las técnicas específicas para la programación de sistemas con requisitos similares a los encontrados en el problema analizado.

**Análisis y síntesis:** Permitió concretar los elementos más importantes relacionados con el proceso de gestión y almacenamiento de archivos que necesita el producto de Captura y Catalogación de Medias. Se realizó el análisis de un gran volumen de documentación, que permitió sintetizar el contenido que da soporte a la propuesta del trabajo a desarrollar.

**Modelación:** Con el objetivo de realizar el análisis y diseño del software y posteriormente la implementación del prototipo funcional.

**Inductivo deductivo:** Fue necesario para poder particularizar lo conocido en función de la situación problemática, ya que se tenían conocimientos muy generales sobre los procesos de almacenamiento y gestión de archivos.

## **CAPÍTULO 1: Fundamentación Teórica.**

### **1.1 Introducción**

En este capítulo se brindará información sobre los procesos de gestión y almacenamiento de archivos en el mundo, así como sus características y tendencias. Se explicará con más profundidad aspectos relacionados con la situación problemática presentada y se analizarán posibles soluciones que puedan contribuir de forma parcial o total a la solución planteada.

### **1.2 Gestión y almacenamiento de archivos**

El tratamiento de la información se convierte en objeto de estudio de las tendencias actuales. Se hace imprescindible que la información esté disponible en todo momento y no esté supeditada al dispositivo físico que proporcione el almacenamiento.

Una de las tendencias con mayor auge en el entorno informático actualmente es la disponibilidad inmediata y permanente de contenidos. Siguiendo esta línea, el almacenamiento de archivos es una tarea indispensable que apunta en esta dirección. Bajo este contexto se desarrolla un sistema de almacenamiento y gestión de archivos, orientado al producto de CCM, que provee almacenamiento básico mediante la interconexión con otros sistemas existentes.

En el proceso de gestión y almacenamiento de archivos se relacionan todo un grupo de conceptos que sería bueno comprender, a continuación se representan una serie de términos que ayudarán a una mejor comprensión del contenido.

#### **¿Qué es un archivo?**

Es un conjunto de datos que se almacenan bajo un determinado formato, que es identificado por un nombre y extensión; puede ser guardado en el disco duro de la computadora o en algún otro medio de almacenamiento como disquete, disco compacto, entre otros **(1)**.

#### **Almacenamiento de Archivos**

Un computador puede almacenar información en varios soportes, como los discos magnéticos y los discos ópticos. También en otros dispositivos como el almacenamiento primario (de manera temporal, RAM<sup>2</sup>) y las tarjetas de almacenamiento portátil con interfaz USB). El sistema operativo se encarga de brindar una vista lógica uniforme para el almacenamiento de la información, proporcionando una abstracción de las propiedades físicas de los elementos de almacenamiento, con el fin de definir una unidad lógica de almacenamiento: **el archivo (2)**.

## Sistemas de Gestión de Archivos

Un **sistema de gestión de archivos** es el software del sistema que proporciona servicios a usuarios y aplicaciones para el uso de archivos. Normalmente la única forma en que un usuario o aplicación puede acceder a los archivos es mediante el sistema de gestión de archivos **(3)**.

### 1.3 Procesos de Almacenamiento y Gestión de Archivos.

#### 1.3.1 Descripción General

En los procesos de descripción, almacenamiento, ordenamiento y clasificación de la información documental son empleadas tecnologías digitales. Las principales ventajas del almacenamiento digital son su capacidad de almacenar, recuperar, difundir, gestionar y distribuir gran cantidad de información de distinta naturaleza en archivos digitales.

Los procesos de almacenamiento digitales ayudan a reducir los costos operativos. El almacenamiento de archivos digitales se utiliza para guardar datos, procesarlos y usarlos como programas o información de cualquier tipo. Dentro de las ventajas que representa el almacenamiento digital se encuentran:

- ✚ Permite duplicar y enviar archivos a través de medios digitales (sin sacarlo de su ubicación).
- ✚ Conserva los archivos intactos a lo largo del transcurso del tiempo. La información no se deteriora ya que se encuentra en archivos digitales.

---

<sup>2</sup> **RAM**: Siglas en inglés de **Random Access Memory (Memoria de Acceso Aleatorio)** y cuyo contenido puede ser leído, borrado o modificado a voluntad.

- ✚ Permite un acceso inmediato a los ficheros según distintos criterios de búsqueda. Por ejemplo, por año, por tipo de documento, por cliente, entre otros.
- ✚ Permite el acceso simultáneo de varios usuarios al mismo archivo digital.
- ✚ Permite establecer criterios de acceso de forma flexible y fácil.

El volumen creciente de información en los ordenadores hace cada vez más difícil la gestión de los archivos informáticos. Ante la heterogeneidad de los datos que son almacenados en los ordenadores surge la necesidad de contar con programas especializados para la gestión de los archivos con el fin de lograr un mejor almacenamiento.

## **Almacenamiento virtual de archivos**

Se denomina así al proceso de almacenar archivos en una red de computadores utilizando al menos dos máquinas, distintas y conectadas entre sí. Una de las máquinas actúa como servidor de la red, que se encarga de almacenar los archivos físicamente. La otra máquina hace el papel usuario cliente que se comunica con el servidor para enviar datos que se encuentran alojados en él. **(2)**

## **Sistemas de almacenamiento**

Los sistemas de almacenamiento se han convertido en un aspecto singular y complejo de la informática que puede enfocarse desde distintos puntos de vista. La tecnología de almacenamiento actual engloba todo tipo de soportes, dentro de los que se pueden encontrar las bibliotecas de cintas y bibliotecas virtuales, entre otros.

La tecnología ha evolucionado a través de los años para satisfacer el incremento exponencial en el volumen de la información almacenada en la actualidad. Los sistemas de almacenamiento en red permiten almacenar y administrar información de manera efectiva al permitir que los clientes y los sistemas de

servidores puedan acceder fácilmente a la información en la red. En los últimos años, los sistemas SAN<sup>3</sup> y NAS<sup>4</sup> han demostrado su excelente fiabilidad.

Teniendo en cuenta que los procesos concebidos en el producto CCM necesitan un almacenamiento eficiente y que de acuerdo al contexto donde se emplee el producto podría ser necesario almacenar grandes cantidades de información, sería importante considerar el uso de una **SAN** como sistema de almacenamiento. Algunas de las ventajas que ofrece este sistema a la solución propuesta son:

- + Almacenamiento compartido, simplifica la administración y añade flexibilidad.
- + Las rutas de almacenamiento son muchas, un servidor puede acceder a uno o "n" discos y un disco puede ser accedido por más de un servidor, lo que hace que aumente el beneficio.
- + Una de las grandes ventajas que también tiene es que proporciona alta disponibilidad de los datos.
- + Compatibilidad con los dispositivos SCSI<sup>5</sup> ya existentes.
- + Conexión directa con dispositivos SCSI como subsistemas RAID<sup>6</sup>, librerías de cintas o torres de CD-ROM.
- + La capacidad de una SAN se puede extender de manera casi ilimitada y puede alcanzar cientos y hasta miles de terabytes.
- + Una SAN permite compartir datos entre varios equipos de la red sin afectar el rendimiento, porque el tráfico de SAN está totalmente separado del tráfico de usuario.

### 1.3.2 Concepción del almacenamiento y gestión de archivos en el producto CCM.

El producto de **CCM** cuenta en su composición con cuatro subsistemas:

---

<sup>3</sup> **SAN:** Por sus siglas en inglés (**Storage Area Network**) o Red de Área de Almacenamiento.

<sup>4</sup> **NAS:** Del inglés (**Network Attached Storage**), tecnología de almacenamiento que comparte la capacidad de almacenamiento de un servidor a través de la red.

<sup>5</sup> **SCSI:** Acrónimo inglés **S**mall **C**omputers **S**ystem **I**nterface (**Sistema de Interfaz para Pequeñas Computadoras**). Es una interfaz estándar para la transferencia de datos entre distintos dispositivos de la computadora.

<sup>6</sup> **RAID:** Del inglés **R**edundant **A**rray of **I**ndependent **D**isks o conjunto redundante de discos independientes, hace referencia a un sistema de almacenamiento que usa múltiples discos duros.

**Subsistema de Audio:** es una solución factible a la problemática de la perdurabilidad de la información para su posterior procesamiento, capaz de grabar el audio de reuniones, encuentros, vistas judiciales, conferencias y demás actividades de repercusión, permite transcribir<sup>7</sup> esta información para una mejor portabilidad.

**Subsistema de Video:** potencia la automatización de los procesos de captura e indexación de medias, dando cobertura a la ejecución de estas actividades a través de una planificación.

**Subsistema de Catalogación:** permite catalogar y organizar eficientemente los distintos materiales de media, con el propósito de poder mantener un control sobre el contenido de los mismos.

**Subsistema de Administración:** posibilita administrar y configurar de forma segura la solución de Captura y Catalogación de Medias del Centro de Desarrollo de Geoinformática y Señales Digitales, automatizando por completo la administración de la solución e integrándose con el resto de los sistemas que la conforman.

Para el correcto funcionamiento de los 3 primeros se hace necesaria la existencia de un servidor de medias, con el fin de mantener centralizada la información y permitir su gestión de una forma más eficiente. En los procesos de captura tanto de audio como de video se hace necesario efectuar el copiado del material capturado a un lugar donde esté disponible para su posterior catalogación en el caso de los videos y transcripción en el caso de los ficheros de audio. En el proceso de transcripción de un audio se genera un fichero con la información del contenido del material, el cual necesita ser almacenado para posibilitar su gestión. Actualmente la información es almacenada localmente en los mismos puestos de trabajo, ya que la adquisición de un servidor profesional capacitado para interactuar con estas aplicaciones se hace muy difícil por los elevados precios que presenta en el mercado.

En el producto de CCM se necesita desarrollar un nuevo subsistema que se comporte como un servidor de medias para dar solución al problema detectado, dando soporte a las aplicaciones de los subsistemas de audio, video y catalogación. El desarrollo será realizado bajo plataformas libres y uno de los valores

---

<sup>7</sup> **Transcribir:** Proceso de escribir en un documento el contenido de un material de audio.

más importantes de esta solución es la creación de sus componentes aprovechando las potencialidades que brindan el lenguaje Java<sup>8</sup> para trabajar con funciones de redes así como garantizar la compatibilidad con los subsistemas de Audio y Catalogación desarrollados bajo el mismo lenguaje, el subsistema de video se encuentra desarrollado en C++<sup>9</sup>.

Como mecanismo de seguridad y para mantener una comunicación eficiente se ha creado un protocolo que debe ser cumplido correctamente por las aplicaciones implicadas para poder lograr un correcto intercambio con el servidor. A continuación se describen los requerimientos necesarios para poder establecer el intercambio con el servidor:

1. Los diferentes módulos que necesiten la comunicación con el servidor necesitarán de una aplicación cliente como interfaz para la interacción con la aplicación servidora. Esta interfaz será portadora de los métodos necesarios para guiar el proceso de comunicación.
2. El usuario que solicite la comunicación deberá estar previamente registrado en el sistema donde se encuentra la aplicación cliente para poder efectuar operaciones de copia y descarga de archivos.

La administración del servidor se efectuará de forma local, permitiendo eliminar físicamente los ficheros. Además permitirá la búsqueda especializada, así como eliminar ficheros que no se hayan copiado correctamente, realizar purgado de los ficheros obsoletos dado una fecha de caducidad, entre otras operaciones.

### 1.3.3 Situación Problemática

Con el aumento del uso de los materiales audiovisuales en diferentes sectores se hace necesaria y cada vez más difícil su distribución y gestión, esto ha provocado un aumento en el desarrollo de aplicaciones que permitan capturar y catalogar medias de forma eficiente. En el departamento de Señales Digitales perteneciente a la Facultad 9 en la Universidad de las Ciencias Informáticas se está desarrollando un software sobre plataformas libres con el propósito de dar solución a esta problemática.

---

<sup>8</sup> **Java:** Lenguaje de programación de alto nivel, libre, multiplataforma y con gran variedad de funcionalidades para el desarrollo de aplicaciones distribuidas.

<sup>9</sup> **C++:** Es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C (17).

CCM está concebido para trabajar con algunos tipos de ficheros multimedia como son los materiales de audio y video, además define operaciones sobre estas medias almacenadas para lograr la gestión según su contenido. Los ficheros relacionados con audio y video tienen la característica de poseer grandes tamaños es por eso que para su almacenamiento y gestión se necesita de capacidades de almacenamiento superiores a las que se utilizan con ficheros de textos. Tomando en cuenta esto fue concebido como solución óptima, el uso de un servidor de archivos profesional con avanzadas tecnologías de almacenamiento y grandes capacidades.

La adquisición de la tecnología requerida para brindar la solución óptima de almacenamiento y gestión implicaría incurrir un gasto de miles de dólares, afectando considerablemente el precio del producto final. Ante la situación económica del país, se ha hecho necesario que la producción sea efectuada con la menor cantidad de costos posibles y es por eso que se ha puesto en desarrollo un subsistema que se comporte como un servidor de medias y que posibilite la interacción de los subsistemas de audio, video y catalogación para dar solución al problema detectado. Esto permitirá abaratar los costos del producto final y tendrá la característica de ser una solución propia de CCM. Permitirá además un nivel de compatibilidad alto entre las aplicaciones encontradas en los diferentes subsistemas, así como gestionar información de dispositivos de almacenamiento externos, permitiendo así el incremento de capacidad.

## **1.4 Análisis de soluciones existentes**

Existen diversas soluciones de software que satisfacen el problema existente en el producto de CCM, aunque estas solo dan una solución parcial, permiten el desarrollo de las actividades de intercambio de ficheros, lo que posibilita que por su funcionamiento se comporte como un servidor de archivos. A continuación se pueden apreciar algunas de estas soluciones, así como sus características.

- **CletuFTP programa web FTP**

Es una aplicación web que utiliza el protocolo FTP para el envío y la recepción de ficheros de grandes dimensiones a lugares remotos, con función de subida y descarga bidireccional o sea en ambos sentidos.

Entre las características de **CletuFTP**, se incluye la posibilidad de limitar el tamaño de los documentos, y definir las extensiones permitidas para el envío, pudiendo así evitar que mediante esta aplicación se instale software malicioso en el servidor, que pueda provocar problemas de funcionamiento. Es una aplicación muy completa, que facilita una serie de funcionalidades que le permite mantener un control, tanto con respecto a un historial de los ficheros que se han enviado a lo largo del tiempo, como mecanismos automáticos de aviso al cliente en el momento en el que se carga un documento al servidor, para poder consultar y descargar.

### ***Las principales funcionalidades de CletuFTP son:***

- ✚ Creación y eliminación de usuarios.
- ✚ Al añadir un fichero, el destinatario recibe un correo con el acceso, nombre de usuario y contraseña actuales asignados, y una descripción del fichero.
- ✚ Eliminación automática de ficheros del servidor pasados unos días (por defecto 20).
- ✚ Posibilidad de eliminar manualmente ficheros del servidor.
- ✚ Avisos en el momento en que el cliente accede por primera vez al fichero que se le ha enviado.
- ✚ Historial de envíos (comentarios y nombre de los ficheros enviados, aunque los ficheros hayan caducado).
- ✚ Espacio total ocupado por los ficheros en el servidor.
- ✚ Espacio para mensajes genéricos, ya sean publicitarios o informativos en los correos de aviso.

### ***Requerimientos mínimos para el uso.***

- ✚ Base de datos MySQL 3.23 o superior
- ✚ Servidor web con PHP 5
- ✚ Espacio suficiente en el disco duro

- **HFS – HTTP File Server**

HFS es un servidor para uso personal muy fácil de usar que emplea el protocolo web http para el envío y recepción de archivos, no necesita instalación y por tanto, con simplemente ejecutar la aplicación e ir arrastrando los archivos dentro de su interfaz se podrán compartir con cualquier persona con tan sólo conocer la dirección IP.

Permite especificar la dirección IP y el puerto de conexión, establecer límites de velocidad y de conexiones simultáneas, seguir los movimientos de las visitas a través de un log en tiempo real y personalizar la página que se visualiza al entrar en tu PC.

### ***Requerimientos de uso:***

Sistema operativo: Win95/98/98SE/Me/2000/NT/XP/2003/Vista/7

- **ProFTPD**

ProFTPD es software para servidor FTP que funciona bajo el sistema operativo GNU Linux, se desarrolló con el fin de tener un servidor FTP seguro y fácil de configurar, y debido a una gran admiración al servidor Web Apache.

Cuando está frente a una correcta configuración suele ser muy estable y seguro. Es un software altamente configurable y está distribuido bajo la ***Licencia Pública General (GPL)*** de **GNU**, se encuentra bien documentado por lo que facilita su configuración y mantenimiento. Su fichero de configuración es muy similar al que posee el servidor Apache y tiene la capacidad de ser configurado con múltiples servidores virtuales. Posee soporte para IP versión 6. Su diseño es modular, lo que permite escribir extensiones como cifrado SSL/TLS, RADIUS, LDAP o SQL como módulos.

- **WS FTP Server**

El Ipswitch WS\_FTP Server con SSH es un servidor de transferencia de archivos y gestión de datos para Windows de carácter profesional altamente seguro, contiene abundantes características y es fácil de administrar. Posee un poderoso control administrativo y alta capacidad, con codificación de 256-bit AES sobre conexiones SSL y SSH, sondeo de integridad de archivos de hasta SHA-512, fuertes políticas de claves que incluyen contraseñas que caducan automáticamente, política de selección de contraseñas que puede ser dictada por el administrador y reglas para intentos de autenticación fallidos, así como automatización relacionada a eventos, registros personalizados y administración Web. **(4)**

Habilita la transferencia de datos y archivos sobre los protocolos FTP, SSL y SSH y puede escalar un sinnúmero de cuentas de usuarios y conexiones concurrentes. En cuanto a sus características técnicas y

funciones más importantes, se destacan la administración remota, una gran variedad de completas opciones de seguridad, la posibilidad de recuperar una descarga interrumpida o pausada, soporte para la creación de certificados SSL, servidores virtuales, bases de datos de usuarios, entre otras. WS\_FTP Server permite añadir usuarios desde la línea de comandos, y funciona como un servicio NT si así se desea.

- **Servidor de archivos de Windows Server**

Es una herramienta integrada a Windows Server que permite montar un servidor de archivos en un ordenador que disponga de este sistema operativo. Está caracterizada por su fácil uso. Permite realizar monitoreo sobre los archivos que se encuentran compartidos, así como crear reglas para impedir la comunicación con algunos de estos. Un criterio para la confección de las reglas puede ser impedir la subida o descarga de ficheros dado una extensión. Posibilita la integración con el directorio activo de Windows.

Aunque las soluciones expuestas anteriormente podrían resolver en parte el problema presente en el producto de CCM, se dependería del uso del sistema operativo Windows en algunos casos, lo que implicaría tener que pagar por su uso. En caso de hacer uso de una de estas soluciones, se verá limitada al uso de un solo sistema operativo, ya sea GNU Linux o Windows en alguna de sus variantes y se contaría con un número reducido de funcionalidades. Las soluciones FTP, podrían ser una vía para alcanzar el objetivo trazado, pero sería algo impropio, no se podrá establecer un protocolo de comunicación lo que comprometería la seguridad de las aplicaciones y con esto la comunicación entre los demás subsistemas que integran la solución de CCM. Todo el sistema se tendría que adaptar a un reducido número de servicios que proporcionan estas soluciones FTP de forma muy general.

La solución propuesta permitirá cumplir con funcionalidades muy propias del producto de CCM dentro de las que se pueden destacar el registro automático del estado del servidor en la Base de Datos, permitiendo disponer de un valor real para la realización de operaciones en los subsistemas Audio, Video y Catalogación, así como mover temporalmente los archivos eliminados a los directorios establecidos por la configuración sostenida en el sistema. Podrá realizar el manejo de las conexiones a un nivel muy bajo, lo que le permitirá manipular las mismas a conveniencia y crear un ambiente de seguridad robusto y

agradable, basado en flujos de mensajes previos para la sincronización de las aplicaciones por lo que funcionará como protocolo de comunicación. Contará con un alto nivel de gestión de configuración que permitirá al administrador del sistema poder abstraerse de los ficheros de configuración, haciendo uso de la aplicación podrá contar con las funcionalidades necesarias para la configuración de la misma.

Teniendo en cuenta las desventajas considerables que implicarían el uso de las soluciones anteriormente analizadas para el producto de CCM, se puede apreciar la importancia de contar con una solución propia que se ajuste a las necesidades de este producto, con el fin de lograr una mayor compatibilidad entre los sistemas que la componen, reforzar el ambiente de seguridad y posibilitar ampliar sus funcionalidades ante cambios futuros. Una nueva funcionalidad que podrá ser implementada en el futuro será un módulo para la realización de salvadas automáticas sin necesidad de disponer de otra aplicación que se encargue de gestionar esta operación, todo esto sin necesidad de comprometer el funcionamiento, así como permitir su portabilidad a diferentes plataformas.

## **1.5 Conclusiones**

Con el estudio de la situación actual del dominio del problema, recogida en este capítulo, se pudo verificar la utilidad y necesidad de un sistema capaz de realizar el almacenamiento y gestión de archivos. Además se expusieron una serie de conceptos que dan soporte al contenido que permitirá una mejor comprensión del dominio del problema, así como la posible incorporación de una SAN al Sistema de Almacenamiento y Gestión de Archivos de CCM para garantizar un almacenamiento efectivo y obtener una mayor calidad en los procesos.

También se puede apreciar las implicaciones que trae el uso de soluciones externas al producto de CCM. Se valoraron algunas aplicaciones que de alguna forma pueden ser una vía de solución al problema planteado, pudiéndose determinar que es necesario desarrollar una solución propia para el producto de CCM, teniendo en cuenta algunos requerimientos como, la seguridad, portabilidad, soporte, flexibilidad a cambios, incorporación de funcionalidades, desarrollo multiplataforma y reducción de costes.

## CAPÍTULO 2: Tendencias y tecnologías actuales a desarrollar.

### 2.1 Introducción

La evolución de la informática ha sido posible por el perfeccionamiento de las herramientas y tecnologías para el desarrollo de software, que constituyen hoy, instrumentos fundamentales para garantizar la calidad de los procesos de desarrollo.

En este capítulo se realiza un estudio de las tecnologías y tendencias actuales para la construcción de aplicaciones de escritorio vinculadas a la gestión y almacenamiento de ficheros. Se abordan temas relacionados con el uso de metodologías de desarrollo, lenguajes de programación y su conexión con gestores de Bases de Datos, así como los sistemas para la realización del análisis y diseño del mismo. El propósito del estudio que se brinda a continuación está basado en dar a conocer cuál de las tecnologías se utilizarán y el por qué.

### 2.2 Tecnologías que soportan la solución

#### 2.2.1 Socket

Un socket es un punto final de un proceso de comunicación. Es una abstracción que permite manejar de una forma sencilla la comunicación entre procesos, aunque estos procesos se encuentren en sistemas distintos, sin necesidad de conocer el funcionamiento de los protocolos de comunicación subyacentes.

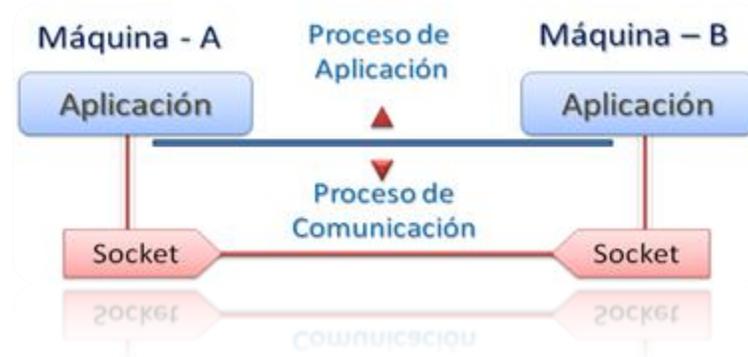


Figura 1 : Representación de funcionamiento de un socket.

**Los socket están definidos por 3 aspectos fundamentalmente:**

- ✚ Un protocolo de comunicación para el intercambio (puede ser TCP o UDP).
- ✚ Un puerto para establecer el punto de comunicación e identificar la aplicación.
- ✚ En caso de uso de TCP/IP, un número de IP para su identificación **(5)**.

El uso de los socket en el desarrollo de la solución propuesta permitirá efectuar la copia de archivos de manera rápida y eficiente. El lenguaje Java proporciona un paquete de clases que minimizan los esfuerzos de su implementación haciéndolo una vía sencilla para la comunicación e intercambio de información entre aplicaciones. Además facilita el uso de un protocolo de comunicación establecido previamente, con el fin de proporcionar una mayor seguridad dentro de la aplicación. Su uso posibilita tener algunas consideraciones que permiten aumentar el nivel de seguridad como son el tiempo de espera para efectuar una transferencia, análisis de direcciones IP y un número máximo de conexiones.

## **2.3 Metodologías de Desarrollo**

Una metodología de desarrollo es un proceso que engloba procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software.

### **2.3.1 Metodologías tradicionales**

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

#### ***Proceso Unificado de Desarrollo (RUP)***

El proceso unificado de desarrollo (RUP) es una metodología que se utiliza para la ingeniería de software, va más allá del análisis y el diseño orientado a objetos, para proporcionar técnicas que soportan un ciclo completo de desarrollo de software. Las 4 P<sup>10</sup>, sustentan el proceso de desarrollo de software. RUP define para cada etapa: el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen. Su meta es asegurar la producción de software con la más alta

---

<sup>10</sup> 4 P: (*proyecto, personas, producto y proceso*).

calidad, que cumpla con las necesidades de los usuarios dentro del cronograma planeado y la inversión prevista.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. Además, utiliza el Lenguaje Unificado de Modelado (UML: Unified Modeling Language) para expresar gráficamente todos los esquemas de un sistema software.

Los aspectos más importantes que definen este Proceso Unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura (7).

**RUP se divide en 4 fases:**

- + Inicio: El objetivo de esta etapa es determinar la visión del proyecto.
- + Elaboración: El objetivo es determinar la estructura óptima.
- + Construcción: El objetivo es obtener la capacidad operacional inicial.
- + Transición: El objetivo es obtener la primera versión del proyecto.

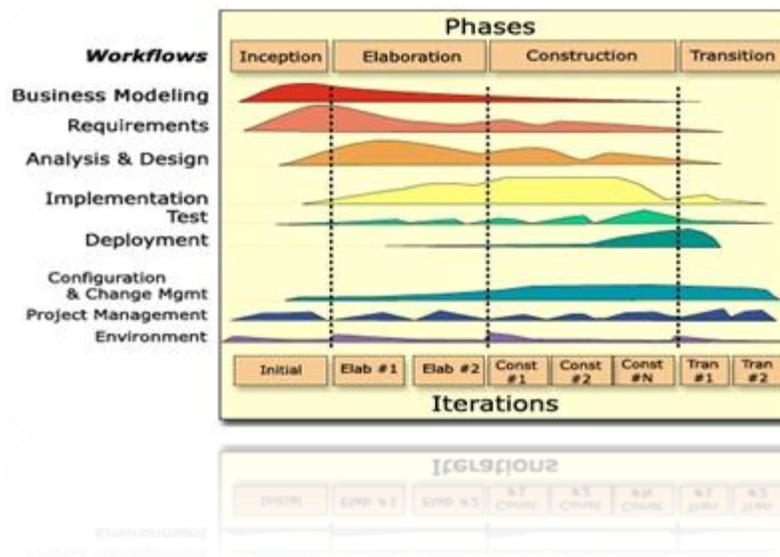


Figura 2: Imagen representativa de la estructura de RUP.

## 2.3.2 Metodologías ágiles

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento).

### *Programación Extrema (XP)*

El primer proyecto se inició 6 de marzo 1996. XP es el más popular de varios procesos ágiles ya que ha demostrado ser muy exitoso en muchas empresas de todos los tamaños e industrias en todo el mundo. Programación Extrema es un éxito porque hace hincapié en la satisfacción del cliente. Faculta a los desarrolladores con confianza para responder a las cambiantes necesidades de los clientes, incluso a finales del ciclo de vida.

XP hace hincapié en el trabajo en equipo, los administradores, clientes y desarrolladores son socios iguales en un equipo de colaboración. Implementa un equipo simple, pero efectivo, medio ambiente propicio para llegar a ser altamente productivo. El equipo se auto-organiza en torno al problema a resolver en la forma más eficiente posible. Permite mejorar un proyecto de software en cinco aspectos esenciales: la comunicación, la sencillez, la retroalimentación, el respeto y el coraje **(8)**.

### *Scrum*<sup>11</sup>

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración **(9)**.

---

**Scrum:** Es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto.

### 2.3.3 Fundamentación de la metodología propuesta: RUP

El Proceso Unificado de Modelado RUP es la metodología que mejor se ajusta a las condiciones del entorno y ofrece una alta capacidad organizativa para grupos de trabajo. La unificación de los mejores elementos de otras metodologías hace que RUP sea de primer nivel. Está preparada para desarrollar proyectos de grandes envergaduras pero posee alta flexibilidad, lo que permite que sea ajustado a cualquier entorno de desarrollo.

Considerando que el sistema a desarrollar formará parte de la solución del producto de CCM se hace necesario ajustarse a los requerimientos expuestos por el mismo, los cuales propones esta metodología para el desarrollo del sistema.

### 2.4 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta

El Proceso Unificado de Desarrollo utiliza Lenguaje Unificado de Modelado (*Unified Modeling Language, UML*) para preparar todos los esquemas de un sistema de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo (10).

Se ha seleccionado UML como soporte de la modelación de la solución propuesta, ya que:

- ✚ UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.
- ✚ El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases y objetos, hasta la implementación y configuración con los diagramas de despliegue.

- ✚ Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

## 2.5 Sistemas Gestores de Bases de Datos (SGBD)

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- ✚ Definición de los datos.
- ✚ Mantenimiento de la integridad de los datos dentro de la base de datos.
- ✚ Control de la seguridad y privacidad de los datos.
- ✚ Manipulación de los datos.

### 2.5.1 Oracle

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. **(12)**

Este SGBD está considerado como uno de los más fuertes y robustos para la elaboración de grandes sistemas donde el principal objetivo sea el manejo fácil y fiable de la información. Hoy Oracle tiene implementaciones sobre plataformas libres pero con los mismos derechos de licencias, que imposibilitan su implantación en soportes de pocos ingresos monetarios.

### 2.5.2 MySQL

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multihilo, multiusuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en software para ser distribuido. MySQL es una marca registrada de MySQL AB. Los usuarios pueden elegir entre usar el software MySQL como un producto de código abierto bajo los términos de la licencia GNU o pueden adquirir una licencia comercial estándar de MySQL AB. **(13)**

## 2.5.3 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD<sup>12</sup> y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilo para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Durante 15 años, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante el desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. **(14)**

## 2.5.4 Fundamentación del SGBD propuesto: *PostgreSQL*

Los sistemas gestores de base de datos por lo general presentan especificaciones muy peculiares de cada uno, por lo que para la implementación de un sistema con manejo de grandes cantidades de datos, la visualización de la información debe ser de fácil acceso.

Para la utilización en el desarrollo de la Base de Datos se selecciona PostgreSQL que está licenciado bajo BSD, que utiliza control de versionado concurrente y soporte para la implementación de consultas complejas. Con este SGBD se podrá mantener una base de datos actualizada, confiable y configurable, así como un alto nivel de integración con las demás aplicaciones que conforman el producto de CCM ya que estas utilizan este SGBD.

---

<sup>12</sup> **BSD:** Del inglés (*Berkeley Software Distribution*) o en español, *Distribución de Software Berkeley*. Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Tiene pocas restricciones por lo que se encuentra muy cerca al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

## 2.6 Framework de acceso a dato usado: *Hibernate*

El desarrollo de software orientado a objetos y el uso de bases de datos relacionales pueden invertir mucho tiempo en los entornos actuales. Hibernate es una herramienta que realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object / relational mapping) o mapeo relacional de objetos y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Hibernate no solo realiza esta transformación sino que proporciona capacidades para la obtención y almacenamiento de datos de la base de datos por lo que reduce el tiempo de desarrollo. Permite desarrollar clases persistentes siguiendo lenguajes orientados a objetos, incluyendo la asociación, herencia, polimorfismo, la composición y las colecciones. Además permite expresar consultas en su propia extensión de SQL portátil (HQL), así como en SQL nativo, o con objetos criterios.

## 2.7 Lenguajes de programación

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático. Un lenguaje de programación permite a un programador especificar de manera precisa sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias **(15)**.

### Los lenguajes de programación se pueden clasificar en 3 grupos según su nivel de trabajo:

- ✚ Lenguajes de maquina: Es el único que entiende directamente la computadora, utiliza el alfabeto binario que consta de los dos únicos símbolos 0 y 1, denominados bits (abreviatura inglesa de dígitos binarios). Fue el primer lenguaje utilizado en la programación de computadoras, pero dejó de utilizarse por su dificultad y complicación, siendo sustituido por otros lenguajes más fáciles de aprender y utilizar, que además reducen la posibilidad de cometer errores. **(15)**

- ✚ Lenguajes de bajo nivel: Proporciona poca o ninguna abstracción del microprocesador de un ordenador. Consecuentemente es fácilmente trasladado a lenguaje de máquina. En general se utiliza este tipo de lenguaje para programar controladores (drivers). **(15)**
- ✚ Lenguajes de alto nivel: Logran la independencia del tipo de máquina y se aproximan al lenguaje natural.

A continuación se caracterizarán algunos de los lenguajes de alto nivel más usados en la actualidad.

## 2.7.1 Java

Java es un lenguaje dinámico simple, orientado a objetos, distribuido, interpretado, robusto, de arquitectura neutra, portable, de alto rendimiento y multihilo. De apariencia similar a C++, pero sin sobrecarga de operadores, herencia múltiple, y aritmética de punteros. Tiene recolector de basura. Primitivas de sincronización basadas en los monitores que se encuentran en Cedar y Mesa. Diseñado para ser seguro, con técnicas de autenticación basada en cifrado de clave pública. Interpretado con una pila a base de máquina virtual. Una interfaz de red permite que el código compilado pueda ser enviado a través de Internet y ejecutado remotamente, lo que permite a los usuarios añadir sus programas a páginas web **(16)**.

## 2.7.2 C++

El C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él, aunque ya se ha dicho que presenta ciertas ventajas. Como lenguaje orientado a objetos se basa en una filosofía completamente diferente, que exige del programador un completo cambio de mentalidad. Las características propias de la Programación Orientada a Objetos (Object Oriented Programming, u OOP) de C++ son modificaciones mayores que sí que cambian radicalmente su naturaleza.

Proporciona nuevas características útiles en diversos contextos como la sobrecarga de operadores. Su expresividad es elevada, pues la sintaxis de clases y objetos permite manipular convenientemente diversas estructuras de datos y operaciones; las excepciones permiten procesar de un modo claro los casos de errores. Es un lenguaje muy difundido y popular por lo que suele ser usado para resolver

diferentes tipos de problemas, desde los más simples hasta los más complejos, su código se puede compilar en diversas plataformas. **(17)**

### 2.7.3 C#

Microsoft C # es un lenguaje de programación diseñado para la construcción de una amplia gama de aplicaciones empresariales que se ejecutan en .NET Framework. Una evolución de Microsoft C y C + +, C # es simple, moderno, seguro y orientado a objetos. El código C # se compila como código administrado, lo que significa que utiliza los beneficios de los servicios de Common Language Runtime. Estos servicios incluyen, la recogida de basura, la mejora de la seguridad, el apoyo y la mejora de versiones.

C # es introducido como Visual C # en Visual Studio. NET suite. Soporte para Visual C # incluye plantillas de proyecto, diseñadores, páginas de propiedades, asistentes de código, un modelo de objetos, y otras características del entorno de desarrollo. La biblioteca de programación de Visual C # es la de .NET Framework. **(18)**

### 2.7.4 Python

Python es un lenguaje de programación dinámico orientado a objetos que pueden utilizarse para diversas formas de desarrollo de software. Ofrece un fuerte apoyo para la integración con otros idiomas y herramientas, posee una gran gama de bibliotecas, y su aprendizaje puede ser en poco tiempo. Python se puede ejecutar en diferentes sistemas operativos como Windows, Linux / Unix, Mac OS X y algunos teléfonos móviles. Este lenguaje también ha sido portado a Java y .NET. Python es distribuido bajo una licencia de código abierto que hace que sea libre de usar, incluso para productos comerciales. **(19)**

### 2.7.5 Fundamentación del lenguaje propuesto: Java

Java posee una curva de aprendizaje muy rápida. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de los desarrolladores lo conoce aunque sea de forma elemental, resulta muy fácil aprenderlo. Los programadores experimentados en C++ pueden migrar muy rápidamente a este lenguaje y ser productivos en poco tiempo. Proporciona una colección de clases para

su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

Está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows NT, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos. **(20)**

Finalmente considerando que la mayoría de las aplicaciones con las que el sistema interactúa están implementadas en este lenguaje y teniendo en cuenta que estos subsistemas necesitan de una aplicación cliente para la conexión con el servidor de archivos. Se hace necesario el uso del lenguaje Java para el desarrollo del sistema de almacenamiento y gestión de archivos, para hacer uso de las potencialidades brindadas por este lenguaje para el trabajo con redes y garantizar un grado de compatibilidad elevado.

## **2.8 Entornos de Desarrollo Integrado (*Integrated Development Environment*)**

Un Entorno de Desarrollo Integrado (IDE) es un conjunto de herramientas de desarrollo de software para programadores. Dentro de las partes fundamentales que lo integran se encuentra el editor de código, compilador, depurador y un constructor de interfaz gráfica de usuario. Los IDE son creados para desarrollar aplicaciones generalmente en un sólo lenguaje de programación. Sin embargo, hay algunos en los que se puede desarrollar en más de un lenguaje, como son el caso de NetBeans, Eclipse y Visual Studio, entre otros.

## 2.8.1 Eclipse

Eclipse es una comunidad Open Source (**código abierto**), cuyos proyectos se centran en la construcción de una plataforma de desarrollo abierta formada por marcos extensibles, herramientas y rutinas para crear, desplegar y gestionar software en todo el ciclo de vida. El IDE es compatible con varios de los sistemas operativos más conocidos como son, todas las versiones de Windows a partir del 98, GNU Linux y Mac OS X. Es un IDE muy usado en la actualidad por su portabilidad y por la gran cantidad de lenguajes en los que se puede desarrollar por lo que brinda un gran servicio de soporte a través de la web. **(21)**

### Principales ventajas

Mediante plugins se le pueden añadir multitud de funcionalidades, además es libre y tiene una amplia comunidad que ofrece considerable documentación sobre el trabajo con él.

### Principal Desventaja

Entre sus principales desventajas está que consume gran cantidad de recursos del ordenador lo cual no es despreciable cuando se trata de aplicaciones que utilizarán funcionalidades del sistema operativo que ralentizan la máquina.

## 2.8.2 NetBeans

El entorno de desarrollo NetBeans es una herramienta libre y gratuita que permite a los programadores escribir, compilar, depurar y ejecutar programas. Especialmente fue desarrollado para el lenguaje Java pero perfectamente se puede utilizar en otros como C/C++, Ruby, PHP, Perl, entre otros. NetBeans IDE es fácil de instalar y de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, Linux, Mac OS X y Solaris. Además existen un número importante de módulos que posibilitan la extensión de este IDE **(21)**. Aparte de la filosofía de distribución y desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma.

### 2.8.3 Fundamentación del IDE propuesto: NetBeans 6.8

Las versiones más recientes de NetBeans integran el framework Hibernate 3.0 que posibilita un desarrollo más eficiente en los términos de acceso a bases de datos. Por la gran acogida que el mismo posee entre

los desarrolladores de todo el mundo se puede encontrar mucha documentación en diferentes tipos de formatos, tanto textos como audiovisuales. Presenta una interfaz muy agradable y resulta muy cómodo el diseño de interfaces para el lenguaje Java. Por lo cual se propone su utilización para el diseño e implementación de la solución propuesta.

## 2.9 Herramientas Case

Las Herramientas CASE (Computer - Aided Software Engineering: Ingeniería de Software Asistida por Ordenador) son básicamente diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. **(23)**

Las herramientas CASE son un conjunto de métodos, utilidades y técnicas, que facilitan la automatización del ciclo de vida para el desarrollo de sistemas de información.

### ***Las herramientas CASE suelen incluir:***

- ✚ Un diccionario de datos para almacenar información sobre los datos de la aplicación de bases de datos.
- ✚ Herramientas de diseño para dar apoyo al análisis de datos.
- ✚ Herramientas que permitan desarrollar el modelo de datos corporativo, así como los esquemas conceptual y lógico.
- ✚ Herramientas para desarrollar los prototipos de las aplicaciones.

### 2.9.1 Rational Rose Enterprise

Rational Rose Enterprise es el producto más completo de la familia Rational Rose de IBM. Todos los productos Rational Rose, como el *Rose Data Modeler* para el diseño visual de bases de datos, el *Rose Modeler*, para el análisis y la arquitectura de software, entre otros, tienen soporte para modelado UML. Es un software privativo, potente para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad con mayor rapidez **(24)**.

### ***Características adicionales incluidas:***

- ✚ Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en patrones de diseño: Elementos de reusabilidad en software orientado a objeto.
- ✚ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ✚ Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- ✚ La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- ✚ Soporte Enterprise Java Beans™ 2.0.
- ✚ Capacidad de análisis de calidad de código.

Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.

## 2.9.2 Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado ayuda a construir con mayor rapidez aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas de UML y proyectos. **(25)**

La versión 6.4 incluye un paquete de aplicaciones ofreciendo varias herramientas que facilitan el desarrollo de proyectos de cualquier dimensión.

### *Las principales características son:*

- ✚ Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en patrones de diseño: Elementos de reusabilidad en software orientado a objeto.
- ✚ Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.

- ✚ Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- ✚ La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- ✚ Soporte Enterprise Java Beans™ 2.0.
- ✚ Capacidad de análisis de calidad de código.
- ✚ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ✚ Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- ✚ Integración con otras herramientas de desarrollo de Rational.
- ✚ Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

### 2.9.3 Fundamentación de la herramienta propuesta: *Visual Paradigm*

Se escoge esta herramienta para la modelación del sistema por las facilidades que brinda para el diseño UML del ciclo de vida de un proceso de desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Mejora el tiempo de construcción de aplicaciones, posibilitando el modelado de todo tipo de diagramas de clases. Facilita la codificación desde diagramas así como el desarrollo de documentación de una aplicación. Entre sus características se destacan su robustez y su portabilidad. Genera código y realiza ingeniería inversa para diez lenguajes de programación. Además exporta e importa los diagramas con estándar XML y como imágenes. Es multiplataforma y gratis en su edición **Community** por lo que resulta muy utilizado en el desarrollo de proyectos importantes. Posee una amplia bibliografía tanto en materiales audiovisuales como documentación digital. La posibilidad de integrarse con los servidores de control de versiones (Subversion) es una de las características fundamentales por las cuales se propone su uso, especialmente la versión 6.4 Enterprise Edition.

### 2.10 Conclusiones

En este capítulo se encuentra recogido un resumen del estudio realizado sobre las tendencias y tecnologías actuales del desarrollo de software, las cuales son la base teórica y tecnológica de la solución propuesta. Los gestores de bases de datos mejoran la gestión de la información, y permiten mantener un alto nivel de seguridad sobre esta. Concluyendo que el PostgreSQL será el sistema gestor de bases de

datos a utilizar, por ser esta la solución GPL más extendida actualmente entre los desarrolladores de sistemas de cualquier envergadura y por las características expuestas con anterioridad.

Es de vital importancia para la producción de software la selección correcta de la metodología de desarrollo. Aunque las metodologías ágiles gozan de gran aceptación, es RUP la que brinda los mejores elementos para el problema detectado. El Visual Paradigm es la herramienta de modelado que se utilizará en la solución propuesta, por poseer un entorno de trabajo amigable, robusto, además de ser una herramienta que posee una edición libre.

De una gran variedad de lenguajes de programación que existen en la actualidad el Java posee interfaces de programación que facilitan la creación de aplicaciones que interactúan con hardware y potencian el desarrollo de aplicaciones para el trabajo con redes, por lo cual se convierte en la propuesta para el desarrollo de la aplicación. El NetBeans tiene una curva de aprendizaje muy alta por lo cual es muy acogido a nivel mundial y entre sus características más representativas está su portabilidad e integración con frameworks potentes como Hibernate.

## CAPÍTULO 3: Presentación de la solución propuesta.

### 3.1 Introducción

En este capítulo se recogen los principales aspectos de la modelación del negocio: se enuncian las reglas que lo rigen, se describen los actores y trabajadores, se muestra el diagrama de casos de uso del negocio, así como la descripción y los diagramas de actividades de cada uno de ellos. Además se exponen los requerimientos funcionales y no funcionales del sistema y se presenta el diagrama de casos de uso del sistema con la descripción de los casos de uso del sistema que se consideran críticos.

### 3.2 Modelo de Negocio

#### 3.2.1 Actores y trabajadores del negocio

Los actores del negocio son los roles que algo o alguien desempeña en su interacción con el negocio. Puede ser cualquier persona o grupo de ellas, cualquier entidad o sistema que interactúa en los proceso del negocio.

Actor del Negocio	Justificación
Cliente	Puede solicitar almacenar o recuperar un archivo en cualquier momento que lo desee.

Tabla 1: Descripción de Actores del Negocio

Los trabajadores del negocio constituyen una abstracción de personas, grupos de ellas o sistemas externos que actúan en el negocio realizando una o varias actividades.

Trabajadores del Negocio	Justificación
Administrador de archivos	Es el encargado de gestionar todo el proceso de copia y recuperación de archivos, así como mantener el orden de los mismos.

Tabla 2: Descripción de Trabajadores del Negocio

#### 3.2.2 Procesos de negocio

En la organización, los clientes pueden solicitar guardar un archivo que se haya obtenido del resultado de una captura o transcripción, así como solicitar uno ya existente sobre los cuales se necesitan hacer operaciones. Estos clientes son atendidos por el operador de archivos que se encuentra operando en la estación de trabajo donde se almacenan los ficheros.

### 3.2.3 Diagrama de Casos de Uso del Negocio

A continuación se presenta el Diagrama de Casos de Uso del Negocio que representa la interacción entre actores y procesos de negocio.

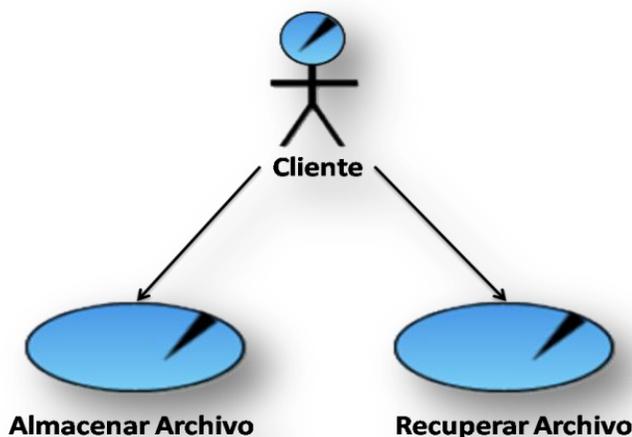


Figura 3: Diagrama de Caso de Uso del Negocio

### 3.2.4 Descripción textual de los Casos de Uso de Negocio

A continuación se presenta la descripción textual de los Casos de Uso del Negocio.

#### Caso de uso Almacenar Archivo

Caso de uso del negocio Almacenar Archivo	
<b>Actores</b>	Cliente
<b>Resumen</b>	El caso de uso se inicia cuando el cliente solicita almacenar un archivo
<b>Casos de uso asociados</b>	-

Acción del actor	Respuesta del proceso de negocio
1. El cliente solicita almacenar un archivo.	1.1 Verifica la identidad del cliente para determinar si tiene la autoridad requerida. 1.2 Se verifica si existe capacidad para almacenar el archivo. 1.3. Si existe se procede a almacenar el archivo.
Otras secciones	Si no existe la disponibilidad se le informa al cliente.
Mejoras propuestas	Se automatizarán las actividades solicitud de almacenamiento emitida por el cliente, así como el proceso de copia de archivos y el proceso de información al cliente.

**Tabla 3: Descripción detallada del caso de uso Almacenar Archivo**

### **Caso de uso Recuperar Archivo**

Caso de uso del negocio Recuperar Archivo	
<b>Actores</b>	Cliente
<b>Resumen</b>	El caso de uso se inicia cuando el cliente solicita un archivo
<b>Casos de uso asociados</b>	-
Acción del actor	Respuesta del proceso de negocio
1. El cliente solicita un archivo	1.1 Verifica la identidad del cliente para determinar si tiene la autoridad de realizar esta operación. 1.2 Buscar archivo. 1.3 Si existe se entrega al cliente
Otras secciones	Si el archivo no se encuentra se le notifica al cliente
Mejoras propuestas	Se automatizarán las actividades de búsqueda y análisis de solicitud del cliente, así como la notificación emitida al cliente.

**Tabla 4: Descripción detallada del caso de uso recuperar Archivo**

Para una mejor comprensión de los procesos de negocio se recomienda la remisión a los Anexos 1 y 2, correspondientes a los diagramas de actividades de los casos de uso expuestos anteriormente.

### **3.3 Requerimientos funcionales**

A continuación se presentan las capacidades o condiciones que el sistema debe cumplir:

#### ***R1. Autenticar usuario.***

El sistema permitirá que los usuarios se autenticuen de forma segura.

#### ***R2. Subir archivos***

El sistema debe permitir la subida de archivos desde otra estación de trabajo.

#### ***R3. Descargar archivos.***

El sistema debe ser capaz de permitir la descarga de archivos desde otro ordenador.

#### ***R4. Controlar el servidor***

**R4.1** Iniciar Servidor

**R4.2** Detener servidor

**R4.3** Reiniciar Servidor

La aplicación servidora debe poseer las características de ser detenido, arrancado y reiniciado en caso que sea necesario y las veces que sea necesaria.

#### ***R5. Visualizar actividad del servidor***

El sistema debe contar con la posibilidad de chequear el estado de las actividades que el servidor de archivos está atendiendo en el momento. Debe mostrar de donde se está accediendo, a qué lugar se accede, quién, en qué momento y sobre que archivos se está llevando a cabo la operación.

#### ***R6. Generar reporte de las actividades del servidor de archivo.***

Se deberá realizar un reporte de las actividades del servidor, donde se deberán apreciar la cantidad de conexiones, número de ficheros descargados, subidos y la cantidad de descargas y subidas fallidas.

#### ***R7. Visualizar el estado de la capacidad del servidor.***

El sistema debe ser capaz de visualizar el estado de los dispositivos de almacenamientos, donde se pueda observar el tamaño total, así como el espacio usado y el que queda disponible.

## ***R8. Listar ficheros***

El sistema debe tener una interfaz donde se visualicen los archivos que se encuentran almacenados en el servidor.

## ***R9. Borrar archivos***

De los archivos mostrados en el servidor, el sistema debe posibilitar el borrado de los mismos desde la propia interfaz.

## ***R10. Purgar servidor***

El sistema debe permitir realizar un purgado de los elementos del servidor mediante diferentes criterios.

**R10.1** Purgado por fecha de creación.

**R10.2** Purgado por tamaño.

**R10.3** Purgado por extensión.

## ***R11. Configurar servidor***

El sistema debe ser capaz de permitir configurar los parámetros mediante los cuales el servidor prestará su servicio.

**R11.1** Puerto

**R11.2** Límite de conexiones

**R11.3** Tiempo límite de conexión

## **3.4 Requerimientos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Definen características que hacen un producto confiable y atractivo.

***Los requerimientos no funcionales se dividen en varias categorías:***

### **Requerimientos de Software**

#### **Para Linux:**

- Sistema Operativo: Cualquier distribución con soporte para la Java Runtime Environment (JRE) requerida.
- Java Runtime Environment (JRE) V6 jre 1.6.0.30 o superior.

- Gestor de Base de Datos: PostgreSQL 8.3

#### **✚ Para Windows:**

- Sistema Operativo: Windows XP (ServicePack 2 ó 3) o superior.
- Java Runtime Environment (JRE) V6 jre 1.6.0.30 o superior.
- Gestor de Base de Datos: PostgreSQL 8.3.

#### **Requerimientos de Hardware**

- ✚ **Procesador:** mínimo, Dual Core 3.0 GHz.
- ✚ **Memoria RAM:** Mínimo, 2 GB DDR2 o superior.
- ✚ **Tarjeta de Red:** Ethernet a 100 Mbps o superior.
- ✚ **Capacidad de almacenamiento en disco duro:** La capacidad de almacenamiento varía en dependencia de las condiciones donde se desarrolle el producto de CCM. Su desempeño en un ambiente donde la calidad de los materiales audiovisuales sea elevada exige altas capacidades de almacenamiento, no inferior a 5 TB. En caso contrario con 1TB sería suficiente para realizar un gran número de operaciones sin tener que liberar el espacio del ordenador.

#### **Requerimientos de usabilidad**

- ✚ La aplicación debe ser concebida para ser utilizada por personas que tengan conocimientos básicos de informática en el trabajo con aplicaciones de escritorio y en el trabajo con archivos.
- ✚ La aplicación garantizará una conexión segura y eficiente con la base de datos, que tendrá almacenada toda la información del sistema, mejorando esto la gestión de la información y los datos.

#### **Requerimientos de seguridad**

- ✚ Solo tendrá acceso a la aplicación el personal autorizado que se encargará de llevar el control del servidor.
- ✚ Toda la información deberá ser manejada con cuidado, protegiéndola del robo parcial o total.

## Requerimientos de soporte

- ✚ El sistema permitirá ser flexible a modificaciones, permitiendo aumentar su nivel de funcionalidades para brindar mayores prestaciones.

## Requerimientos de Apariencia o Interfaz Externa

- ✚ Las interfaces de la aplicación tendrán los componentes visuales necesarios para las operaciones correspondientes, evitando la sobrecarga de imágenes.
- ✚ El sistema debe mostrar un ambiente profesional, sin información repetida o en exceso.
- ✚ El diseño será sugerente y permitirá al usuario navegar con facilidad e intuición por la aplicación.
- ✚ La aplicación debe ser rápida, flexible y adaptable al entorno.

## Restricciones en el diseño y la implementación

- ✚ El sistema deberá ser desarrollado como una aplicación de escritorio. Permitirá ser funcional en diferentes plataformas y su desarrollo estará soportado bajo plataformas libres.

### 3.5 Descripción del Sistema Propuesto

El sistema será una aplicación de escritorio, útil para el producto CCM que necesita de un servidor de medias para realizar el almacenamiento de los archivos procedentes de los sistemas de Audio, Video y Catalogación. Solo dos usuarios interactúan con el sistema: el Administrador del Servidor y un cliente que realiza las solicitudes de subida o descarga de archivos. El sistema contará con las áreas de trabajos necesarias para facilitar la interacción y dar cumplimiento a los requerimientos funcionales.

#### 3.5.1 Descripción de los actores

Actor	Descripción
Cliente	Es una aplicación que se puede encontrar en el subsistema de Audio, Video o Catalogación, que solicita la subida o descarga de archivos.
Administrador del Servidor	Es el encargado mantener el sistema en correcto estado, manejar las configuraciones del mismo, borra los ficheros que desee, así como detener y reiniciar los servicios en caso

	que sea necesario.
--	--------------------

Tabla 5: Descripción de actores del sistema

### 3.5.2 Casos de Uso del Sistema

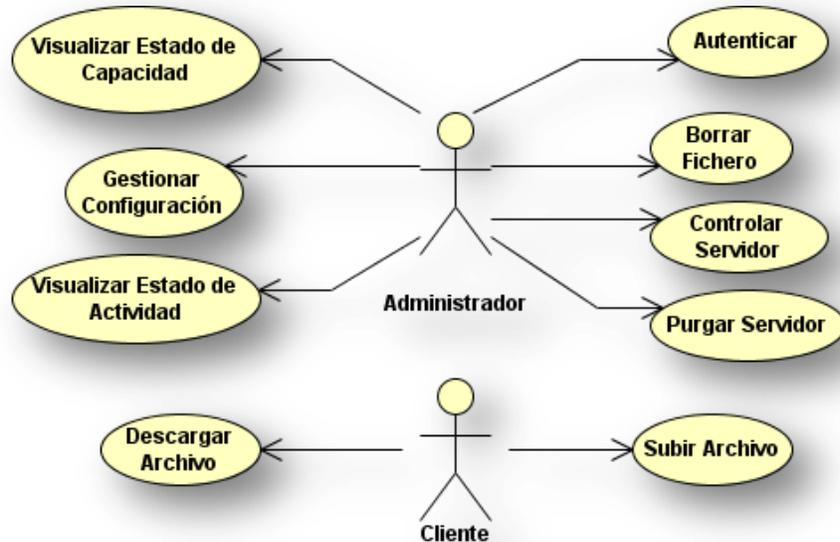


Figura 4: Diagrama de Caso de Uso del sistema

#### Descripciones de casos de uso del sistema

A continuación se describen los casos de uso críticos para el sistema. Su selección está fundamentada teniendo como premisa los requerimientos expedidos por los desarrolladores del producto de CCM, así como determinando las funcionalidades arquitectónicamente significativas para el sistema.

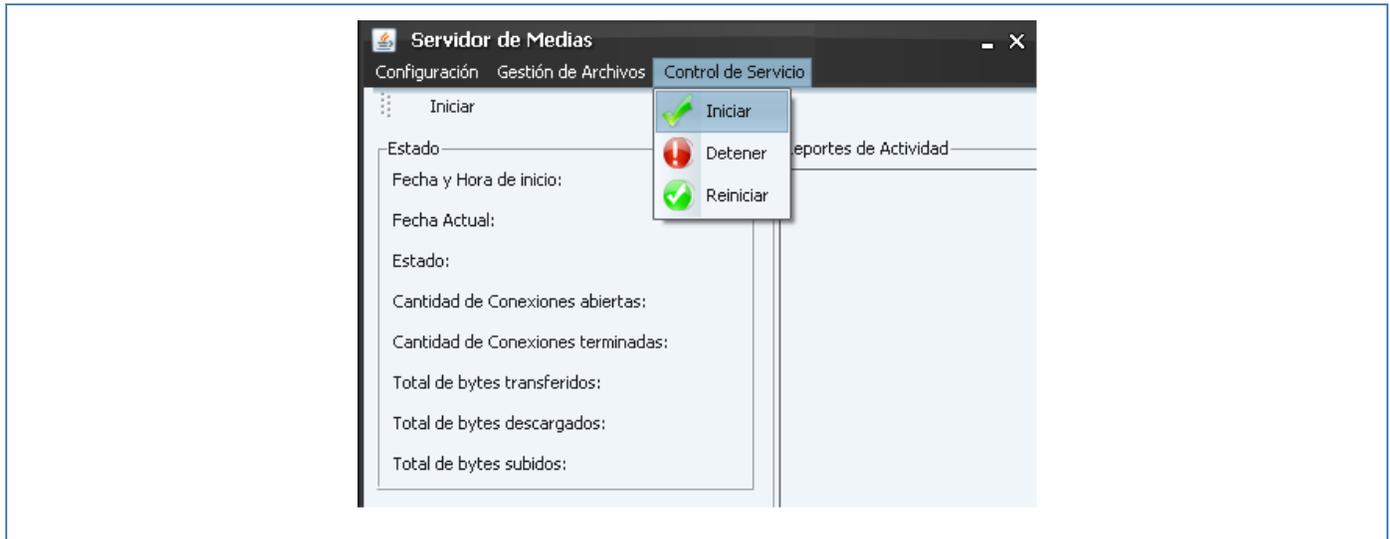
<b>Caso de Uso:</b>	Descargar Archivo
<b>Actores:</b>	Cliente
<b>Resumen:</b>	El caso de uso se inicia cuando el cliente decide descargar un archivo del servidor.
<b>Precondiciones:</b>	

<b>Referencias</b>	RF 3
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El cliente solicita descargar un archivo	1.2 Solicita datos de identificación del usuario que se encuentra registrado en el subsistema.
1. Envía los datos de autenticación.	2.1 Valida los datos enviados (usuario y contraseña). 2.2 Solicita Información del archivo a descargar.
2. Envía los datos del fichero que se desea descargar.	3.1 Verifica la existencia del archivo en el servidor. 3.2 Procede con la descarga del fichero solicitado.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.2 Muestra mensaje de error al no ser un usuario con privilegios para realizar esta operación.
	3.2 Muestra mensaje de notificación de archivo no encontrado.
<b>Poscondiciones</b>	El cliente efectúa la descarga deseada.
	

Tabla 6: Descripción del CU Descargar Archivo.

<b>Caso de Uso:</b>	Controlar Servidor
<b>Actores:</b>	Administrador del servidor
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide iniciar, detener o reiniciar el servidor de archivos.
<b>Precondiciones:</b>	Que el administrador se encuentre registrado en el sistema.
<b>Referencias</b>	RF 4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción de control de servicio en el menú de la ventana principal. Puede ser una de las siguientes opciones: a) Iniciar b) Detener c) Reiniciar	1.1 El sistema ejecuta algunas de las opciones siguientes: a) Si decide Iniciar Servicios ir a la sección (Iniciar Servicios). b) Si decide Detener servicios ir a la sección (Detener Servicios). c) Si decide Reiniciar ir a la sección (Reiniciar Servicios).
<b>Sección "Iniciar Servicios"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador solicita la opción de Iniciar.	1.1 Se inician los servicios. 1.2 Registra estado activo en la Base de Datos. 1.3 Emite información del resultado de la operación.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.3 Emite información de error en la conexión a la Base de Datos.
<b>Poscondiciones</b>	Los servicios quedan iniciados

Sección “Detener Servicios”	
Acción del Actor	Respuesta del Sistema
1. El Administrador solicita la opción de Detener.	1.1 Muestra mensaje de confirmación si desea detener el servicio. 1.2 Se cierran todas las conexiones y se detiene el servicio. 1.3 Emite información del resultado de la operación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.2 Se mantiene todo como se encontraba antes de ser seleccionada esta operación.
<b>Poscondiciones</b>	Los servicios son detenidos.
Sección “Reiniciar Servicios”	
Acción del Actor	Respuesta del Sistema
1. El Administrador solicita la opción de Reiniciar.	1.1 Muestra mensaje de confirmación si desea reiniciar el servicio. 1.2 Se cierran todas las conexiones. 1.3 Emite información del resultado de la operación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.2 Se mantiene todo como se encontraba antes de ser seleccionada esta operación.
<b>Poscondiciones</b>	Los servicios son reiniciados, quedan desconectadas todas las peticiones y liberados los hilos de conexión.



**Tabla 7: Descripción del CU Controlar Servidor.**

<b>Caso de Uso:</b>	Gestionar Configuración
<b>Actores:</b>	Administrador del servidor
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador decide crear, cargar o modificar la configuración del servidor.
<b>Precondiciones:</b>	Que el administrador se encuentre registrado en el sistema.
<b>Referencias</b>	RF 4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El administrador selecciona la opción de gestionar configuración en el menú de la ventana principal.</li> <li>2. Puede realizar alguna de las opciones siguientes:                     <ol style="list-style-type: none"> <li>a) Cargar Configuración</li> <li>b) Crear Configuración</li> <li>c) Editar Configuración</li> </ol> </li> </ol>	<ol style="list-style-type: none"> <li>1.1 Muestra la interfaz de Gestión de Configuración.</li> <li>1.2 El sistema ejecuta algunas de las opciones siguientes:                     <ol style="list-style-type: none"> <li>a) Si decide Cargar Configuración, ir a la sección (Cargar Configuración).</li> <li>b) Si decide Crear Configuración, ir a la sección (Crear Configuración).</li> </ol> </li> </ol>

d) Eliminar Configuración	c) Si decide Editar Configuración, ir a la sección (Modificar Configuración). d) Si selecciona Eliminar Configuración, ir a sección Eliminar Configuración.
<b>Sección “Cargar Configuración”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona la configuración que desea cargar en el listado de configuraciones existentes.	1.1 Carga en la interfaz los datos relacionados a la configuración.
2. Selecciona la opción de cargar.	2.1 Notifica si desea confirmar la operación de establecer la configuración al sistema. 2.2 Notifica confirmación de reinicio de servicios para establecer la configuración. 2.3 Establece la configuración en el sistema.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.3 Mantiene la configuración inicial.
<b>Poscondiciones</b>	La configuración queda establecida en el sistema
<b>Sección “Crear Configuración”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Llena los campos necesarios para crear una configuración en la interfaz Gestión de Configuración.	
2. Acciona la operación de “Crear” para guardar la nueva configuración.	2.1 El sistema valida los datos de la configuración creada. 2.2 Guarda datos de la configuración.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.2 Notifica error en los datos de entrada.
<b>Poscondiciones</b>	Queda creada la una nueva configuración y establecida en el sistema si el

	usuario lo desea.
<b>Sección “Editar Configuración”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona en la lista de configuraciones existentes la configuración que desea editar.	1.1 Carga en la interfaz los datos de la configuración a modificar.
2. Modifica los datos de la configuración y acciona la operación de guardar.	2.1 Valida datos entrados. 2.2 Actualiza los datos de la configuración.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.2 Notifica error en los datos de entrada. Se solicita la entrada de datos correctos.
<b>Poscondiciones</b>	Queda modifican la configuración seleccionada.
<b>Sección “Eliminar Configuración”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Administrador selecciona la configuración que desea eliminar en el listado de las configuraciones existente.	1.1 Llena los campos de la interfaz con los valores correspondientes a la configuración.
2. Acciona la operación de eliminar.	2.1 Muestra confirmación de la operación eliminar. 2.2 Elimina la configuración. 2.3 Actualiza la lista de las configuraciones.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2.2 Todo se mantiene igual. Muestra la interfaz con las configuraciones del sistema existentes.
<b>Poscondiciones</b>	Queda eliminada la configuración seleccionada.

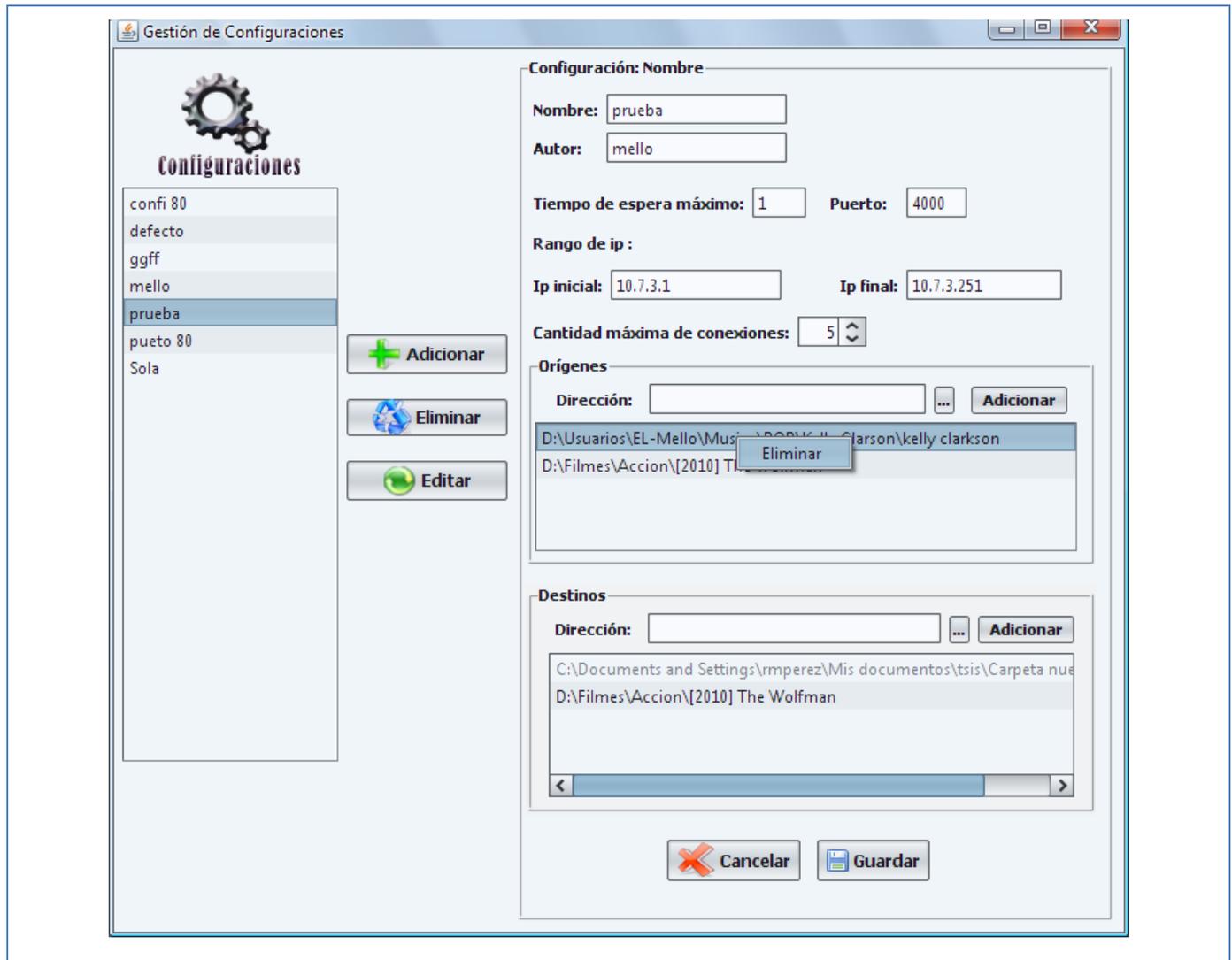


Tabla 8: Descripción del CU Gestionar Configuración.

<b>Caso de Uso:</b>	Subir Archivo
<b>Actores:</b>	Cliente
<b>Resumen:</b>	El caso de uso se inicia cuando el cliente decide copiar un archivo del servidor.
<b>Precondiciones:</b>	
<b>Referencias</b>	RF 2
<b>Prioridad</b>	Crítico

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor del sistema envía sus credenciales de autenticación.	1.1. El sistema verifica las credenciales. 1.2. El sistema solicita tamaño de la Media a subir.
1. El actor del sistema envía el tamaño de la Media a subir	2.1. El sistema calcula el tamaño de todas las medias que esta copiando, más el tamaño que recibió y determina si tiene espacio suficiente para la transferencia. 2.2. Envía confirmación de transferencia.
3. El actor del sistema comienza la transferencia de la Media.	3.1. El sistema recibe los datos de la transferencia de la Media. 3.2. Al terminar la transferencia se envía la confirmación de terminación.
4. Recibe la confirmación de fin de transferencia y cierra la conexión.	4.1. Libera el hilo de la conexión.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Acción del Actor</b>
	2.2. Se emite mensaje de error porque no existe el espacio necesario para la transferencia de la Media.
3. Recibe mensaje de error, busca un nuevo servidor activo y vuelve a la actividad 1.	
	3.2. Se emite mensaje de error porque por algún motivo la transferencia se interrumpió. 3.3. Se libera el hilo de la conexión. 3.4. Se eliminan los datos transferidos.
<b>Poscondiciones</b>	El archivo es guardado correctamente en el servidor



Tabla 9: Descripción del CU Subir Archivo

### 3.6 Conclusiones

En este capítulo se ha presentado la descripción del negocio. Además fueron enumerados los requisitos funcionales y no funcionales, mostrando las capacidades y condiciones que el sistema debe reunir. Se ha modelado el sistema propuesto contando con el diagrama de casos de uso y la descripción detallada de los mismos. Todo esto ofrece una visión general de la estructura y función de la solución que se propone.

## CAPÍTULO 4: Construcción de la solución propuesta.

### 4.1 Introducción

En este capítulo se abordan temas fundamentales para la construcción del sistema propuesto como los diagramas de clases del diseño como punto de partida para el Flujo de Trabajo de Implementación. Se plantean además las principales pautas de diseño que se siguieron en la interfaz gráfica de la aplicación. También son explicados varios aspectos referidos a la implementación del sistema, explicando las características de la aplicación y sus funcionalidades.

### 4.2 Diagramas de Clases del Diseño

El propósito del diseño constituye transformar los requerimientos en lo que será el sistema y desarrollar una arquitectura robusta. En el diagrama de clases del diseño se tiene una visión del sistema en términos de clases, que se encuentran estructuradas por atributos y métodos. Se muestra como quedaría implementada toda la aplicación en términos lógicos.

El sistema ha sido diseñado teniendo en cuenta el patrón en capas que define una organización jerárquica tal, que cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior **(25)**, estableciéndose así la capa de presentación, la capa de lógica del negocio y la capa de acceso a datos.

A continuación se presenta el diagrama de clases del diseño. Para un mejor entendimiento se muestran las clases por capas y posteriormente la interacción general de las mismas.

## 4.2.1 Capa de Presentación

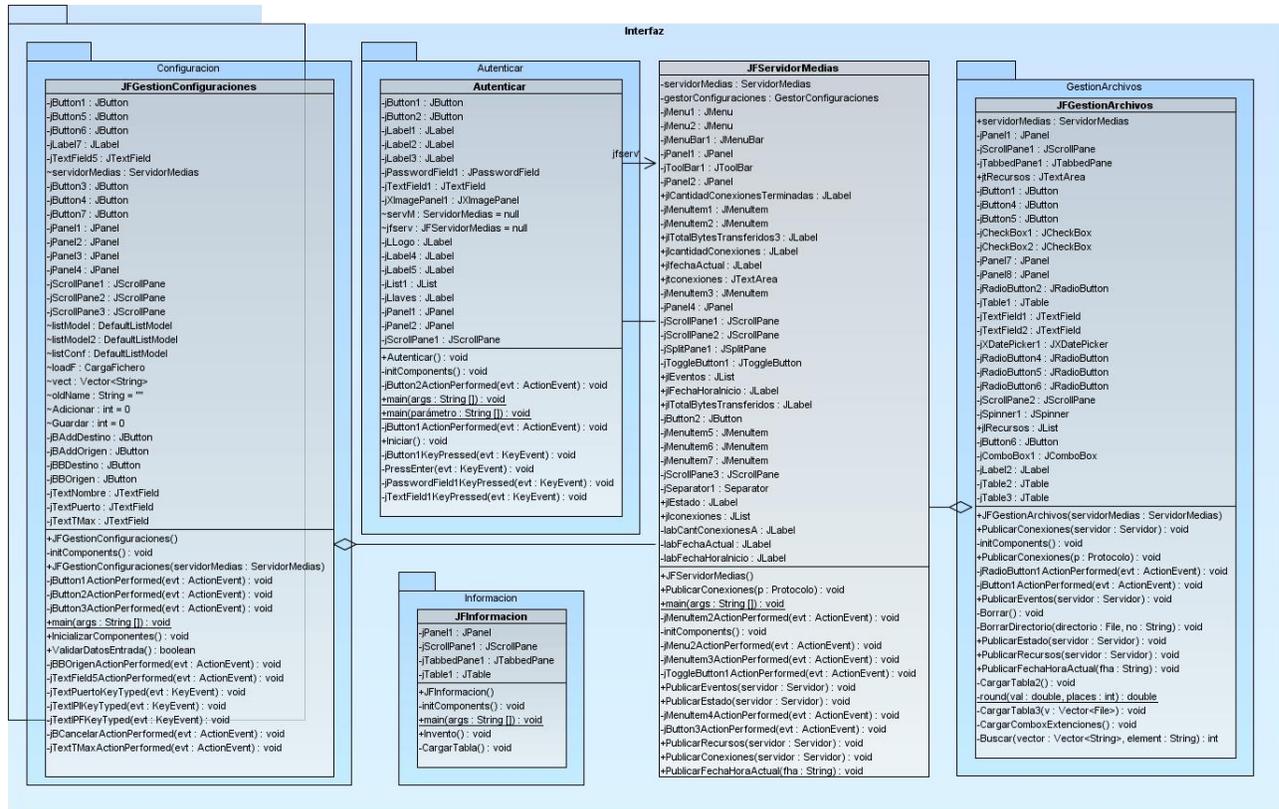


Figura 5: Paquete Interfaz (Diseño de clases).



### 4.2.3 Capa de Acceso a Datos

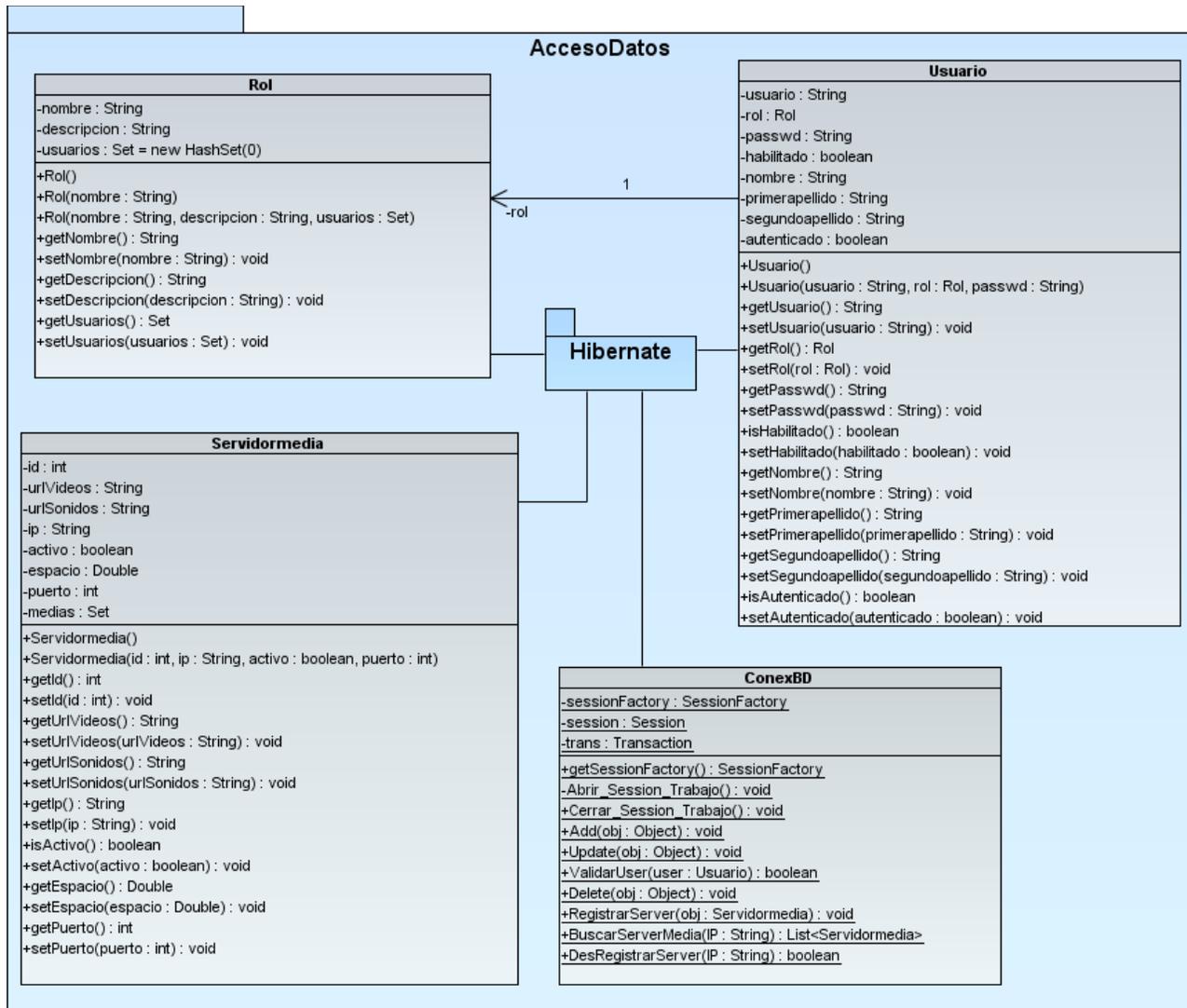


Figura 7: Paquete de Acceso a Datos (Diseño de clases).

#### 4.2.4 Diagrama general de clases del diseño.

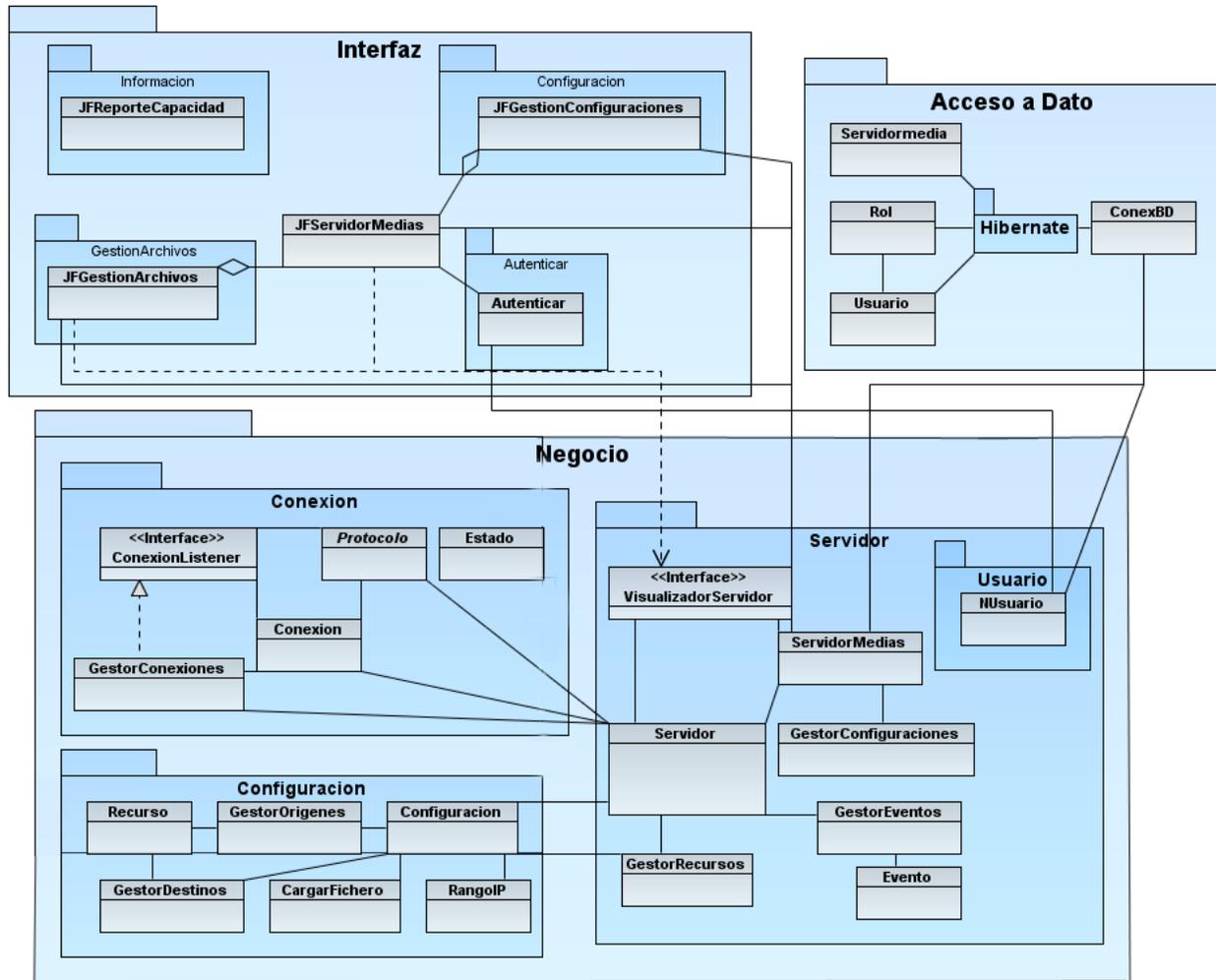


Figura 8: Diagrama de clases del diseño (Interacción entre paquetes).

### 4.3 Principios del Diseño

#### 4.3.1 Estándares de la Interfaz de la aplicación

El diseño de la interfaz de una aplicación constituye uno de los procesos más importantes en el desarrollo de software, pues a través de esta el usuario se comunica con el sistema. En su desarrollo se deben tener en cuenta diversos elementos para lograr interfaces cómodas, intuitivas y amigables que favorecen la usabilidad y alcancen un alto grado de aceptación.

La interfaz gráfica de la aplicación fue concebida con un ambiente sencillo, intuitivo y de fácil navegación para el usuario. Los colores han sido convenientemente utilizados dada la funcionalidad y objetivo del sistema, considerando que la aplicación no necesita de una interfaz tan cargada pues funcionará básicamente como un servidor de archivos, se han empleado en su diseño una combinación de colores claros y oscuros, con predominio del color negro como matiz logrando una vista agradable a los usuarios.

Con el fin de ubicar al usuario en las funcionalidades que brinda el sistema, las mismas se han estructurado convenientemente en diferentes vistas, accesibles desde la barra de menú, teniendo en cuenta en cada uno de los casos los principios fundamentales de diseño como: el balanceado, buscando el equilibrio entre los ejes de coordenadas, la simetría y la regularidad de los elementos ubicados.

### 4.3.2 Generalidades de la Implementación

La aplicación forma parte de un sistema distribuido, contará con un cliente en cada una de las estaciones de trabajo que necesitan comunicarse con la misma para realizar operaciones de descarga y copia. No posee base de datos propia pero sí realiza operaciones de autenticación en la base de datos correspondiente al producto de CCM del cual forma parte. Esta base de datos está soportada sobre PostgreSQL y para efectuar la conexión a la misma se utiliza el driver JDBC.

Otro elemento significativo en la implementación es la interfaz gráfica, debido a que las interfaces de usuarios que genera el NetBeans resultan poco amigables, para hacer el sistema más amigable se efectuó el uso de la funcionalidad "LookandFeel" que implementa, con la utilización de la librería Substance 5.0, la misma implementa un gran número de clases de tipo LookandFeel la cual permitió generar las interfaces del sistema.

En la implementación del sistema se utilizó para el acceso a datos el Framework Hibernate que facilita el mapeo de las tablas de la base de datos, permitiendo tratar las mismas como objetos y sincronizarlas.

Para la gestión de la seguridad del sistema y la transferencia de ficheros a través de la red desde las estaciones clientes se hace uso de algunas de las clases pertenecientes al paquete **java.net** como son las clases Socket y ServerSocket, entre otras.

Las clases han sido nombradas de manera sencilla representando de manera intuitiva las funcionalidades que contienen y el paquete al que pertenecen, ejemplo: **JFGestorArchivos**, donde **JF** indica que pertenecen a una clase interfaz **JFrame**.

Los métodos de las clases también han sido nombrados de forma tal que indiquen la función que realizan, siendo escritos con todas las palabras necesarias en un nombre seguido, donde se marca cada una de ellas con letra mayúscula, ejemplo: **ValidarUsuario()**.

### 4.3 Modelo de Despliegue

El Diagrama de Despliegue muestra las relaciones físicas entre los componentes de hardware y software en el sistema. Se representa como un conjunto de nodos unidos por conexiones de comunicación y los protocolos empleados en el los procesos de intercambio de información, donde un nodo puede contener instancias de componentes de software, objetos y procesos que representan manifestaciones del código en tiempo de ejecución.

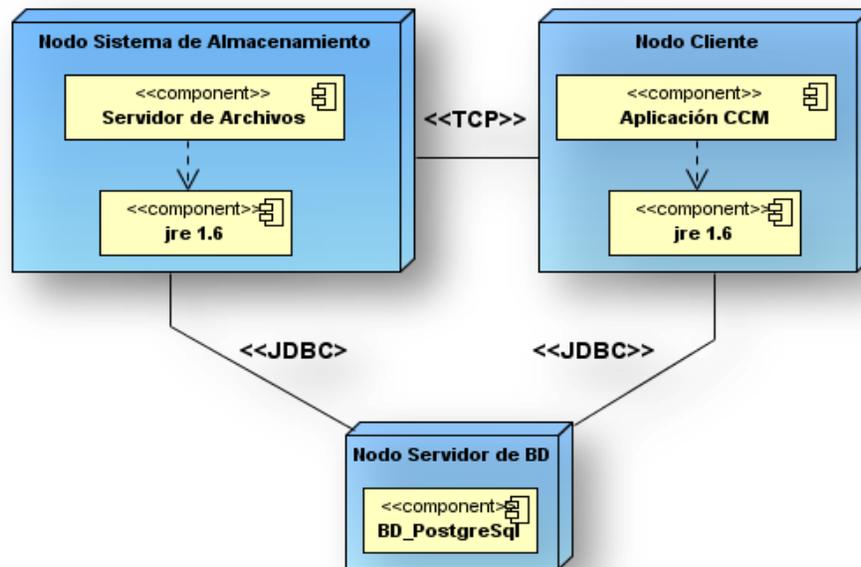


Figura 9: Diagrama de Despliegue.

A continuación se describirán los nodos físicos representados en el diagrama de despliegue:

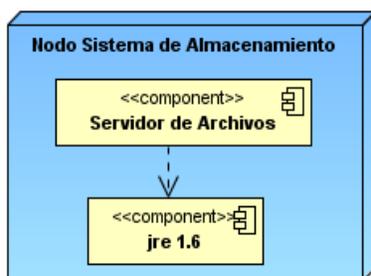


Figura 10: Nodo Sistema de Almacenamiento

**Nodo Sistema de Almacenamiento:** Constituye el nodo servidor de archivos cuyo propósito es gestionar y proveer los materiales que otras estaciones necesitan para trabajar.

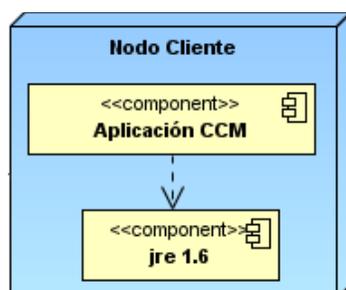


Figura 11: Nodo Cliente.

**Nodo Cliente:** Es una estación de trabajo en la cual se encuentra corriendo una aplicación de CCM, puede ser una estación de Audio, Video o Catalogación.



Figura 12: Nodo Servidor de BD

**Nodo Servidor de BD:** En este nodo se estará ejecutando el servidor de base de datos PostgreSQL.

## Descripción de los protocolos de comunicación:

**JDBC:** *Java Database Connectivity* o *Conectividad de la Base de Datos de Java* es una API<sup>13</sup> para la programación Java que define cómo un cliente puede tener acceso a una base de datos. Proporciona métodos para consultar y actualizar datos. JDBC está orientado a bases de datos relacionales.

**TCP/IP:** Protocolo de comunicación entre las PC clientes con el servidor de archivos.

<sup>13</sup> API: Del inglés (*Application Programming Interface*) o *Interfaz de Programación de Aplicaciones*.

### 4.5 Modelo de Implementación

El modelo de implementación representa la programación de la aplicación en términos de componentes. Describe la organización de los datos, archivos, ejecutables, código fuente y directorios de acuerdo a los mecanismos disponibles para su estructuración en el entorno de implementación. Fundamentalmente, se describen las relaciones que existen entre los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. En este artefacto se describe cómo se implementan los componentes que lo conforman, acoplándolos en paquetes organizados en capas y jerarquías, señalando las dependencias entre los mismos.

#### 4.5.1 Diagrama de componentes. Paquete Interfaz.

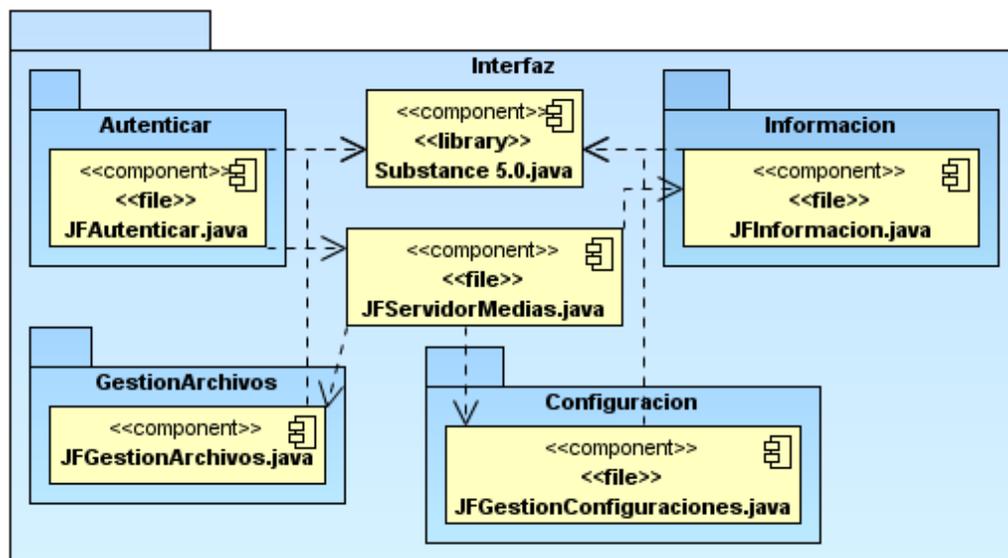


Figura 13: Diagrama de componente. Paquete Interfaz.

#### 4.5.2 Diagrama de Componentes. Paquete Negocio.

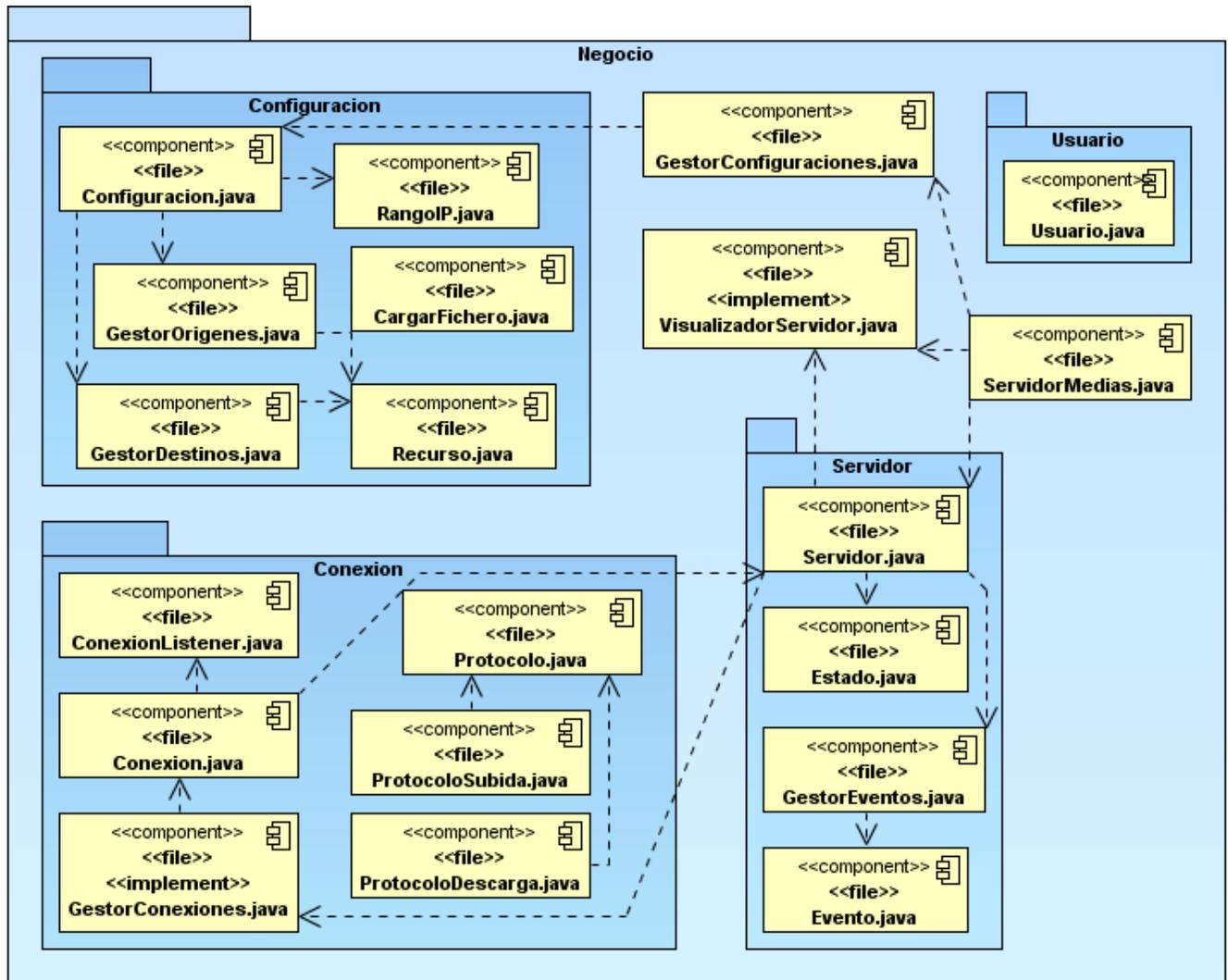


Figura 14: Diagrama de Componentes. Paquete Negocio.

4.5.3 Diagrama de Componentes. Paquete de Acceso a Datos.

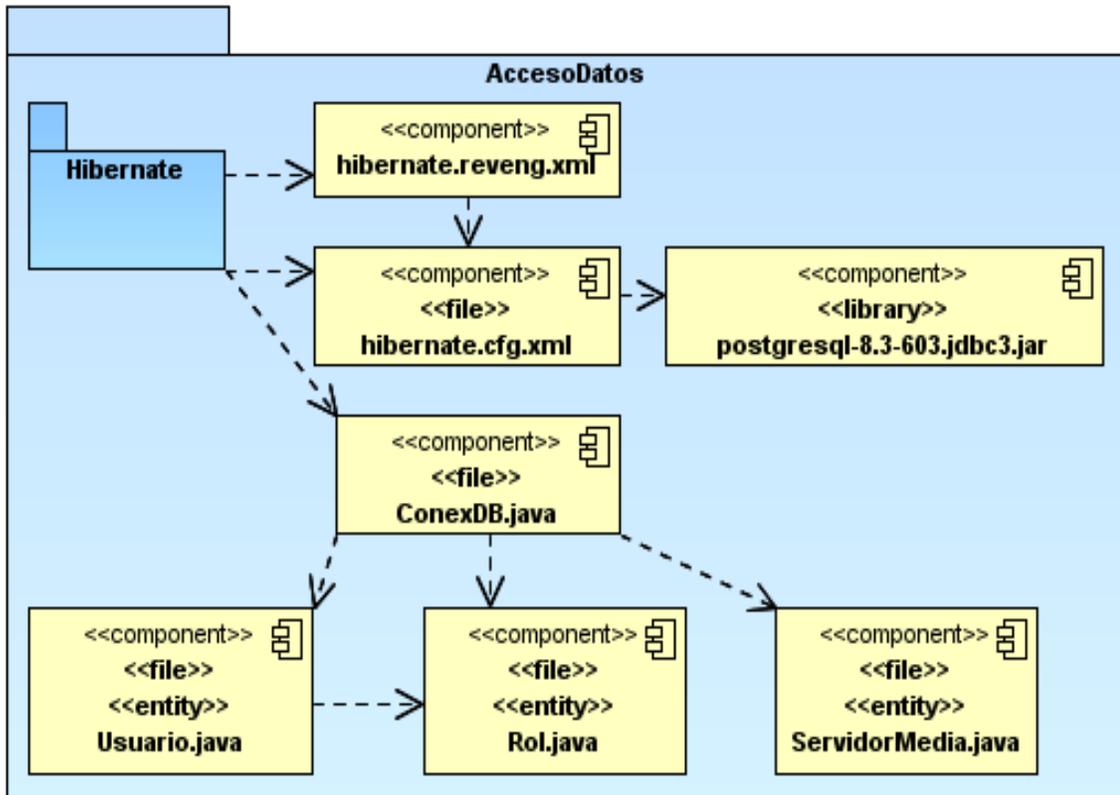


Figura 15: Diagrama de Componentes. Paquete de Acceso a Datos.

4.5.4 Modelo de implementación

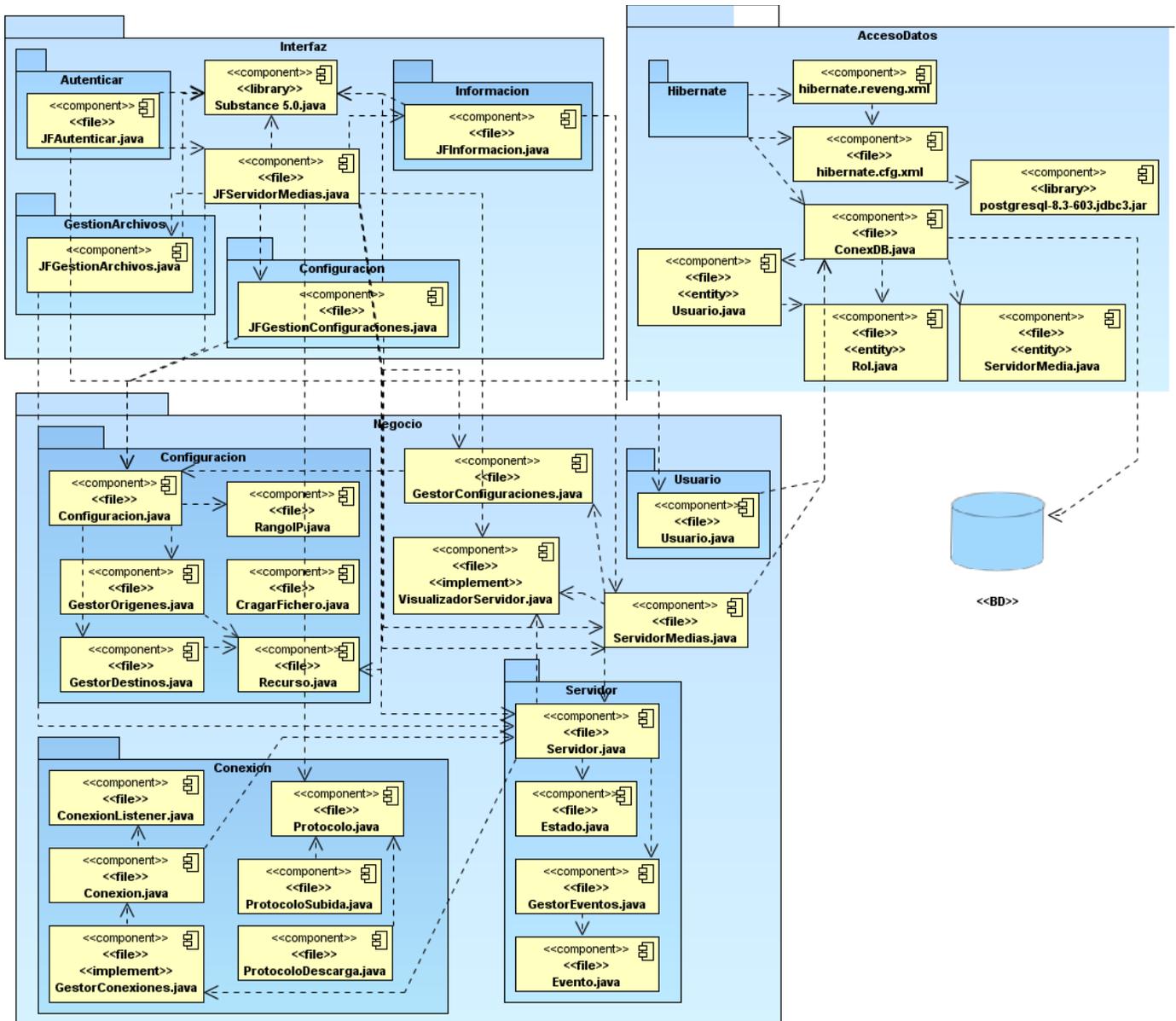


Figura 16: Modelo de implementación.

## 4.6 Conclusiones

Durante el desarrollo del capítulo fueron expuestos algunos temas de gran importancia para la construcción del sistema. Dentro de estos se encuentran los diferentes diagramas de clases correspondientes al diseño, que contribuyeron a su adecuada implementación. Además se abordaron las generalidades para el desarrollo de la aplicación teniendo en cuenta los componentes utilizados, así como las relaciones entre ellos a través del modelo de implementación. También se pudo obtener una vista del sistema en términos de nodos físicos, representado en el diagrama de despliegue. Se han tratado además aspectos relacionados con los estándares seguidos en el diseño de la interfaz de la aplicación y su codificación.

## CONCLUSIONES

Los resultados obtenidos con la realización del presente trabajo han dado cumplimiento a los objetivos trazados en esta investigación:

- ✚ Se establecieron las bases teóricas y tecnológicas de la solución propuesta.
- ✚ Fueron modelado los procesos de almacenamiento y gestión de archivos.
- ✚ Se realizó el análisis y diseño del Sistema de Almacenamiento y Gestión de Archivo.
- ✚ Se concluyó la implementación del prototipo funcional del Sistema de Almacenamiento y Gestión de Archivos.

Los objetivos alcanzados a través del desarrollo del presente trabajo, así como la utilización del sistema logrado serán de gran utilidad para los productos de requieran un sistema de almacenamiento de archivos, especialmente para el producto de CCM el objetivo por el cual se puso en marcha el desarrollo del sistema obtenido. El Departamento de Señales Digitales contará además con un nuevo producto multiplataforma e integrable a otras soluciones en las cuales se necesite gestionar el almacenamiento de archivos.

## RECOMENDACIONES

Al concluir el desarrollo de este trabajo se tienen las siguientes recomendaciones:

- ✚ Continuar el desarrollo de la investigación con el objetivo de perfeccionar y aumentar las funcionalidades de la aplicación.
- ✚ Definir un protocolo de comunicación robusto e incluir técnicas y algoritmos de cifrados avanzados para aumentar la seguridad del sistema, así como la implementación del servidor mediante el uso de la clase SSLSocket.
- ✚ Considerando que la comunicación entre procesos en diferentes lenguajes de programación resulta un poco complicada. Se recomienda la implementación del proceso de comunicación y copia del sistema mediante el uso de tecnologías como ICE o CORBA, esto permitirá al sistema contar con clientes de diferentes lenguajes de programación como C++, Python, C#, PHP, entre otros.
- ✚ Refinar la implementación de las funcionalidades según las descripciones realizadas y manteniendo los estándares establecidos.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA CIUDAD

1. **Gobierno Mexicano.** ciberHábitat. [En línea] [Citado el: 11 de diciembre de 2009.]  
<http://www.ciberhabitat.gob.mx/biblioteca/extensiones/>.
2. **EDUARDO ARAGÓN MONTES, JULIÁN CAMILO PÉREZ PARRA.**  
*SISTEMA PARA EL ALMACENAMIENTO VIRTUAL Y GESTION DE ARCHIVOS*. Cali, Colombia : s.n., 2009.
3. **David Luis la Red Martínez.** Sistemas Operativos. [En línea] [Citado el: 2009 de diciembre de 11.]  
<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/GESTAR02.htm>.
4. **Ipswitch.** Ipswitch File Transfer. [En línea] [Citado el: 20 de marzo de 2009.]  
<http://www.ipswitchft.com/Business/Products/WsFtpServer/Index.aspx?is>.
5. **Rebaza, Jorge Carlos Valverde.** *Sockets y su programación en Java*. Trujillo, Perú : s.n., 2007 .
6. **Ibar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educación.S.A, 2000. 84-7829-036-2.
7. **Extreme Programming.** Extreme Programming. [En línea] [Citado el: 23 de enero de 2010.]  
<http://www.extremeprogramming.org/>.
8. **José H. Canós, Patricio Letelier y M<sup>a</sup> Carmen Penadés.** [En línea] [Citado el: 25 de enero de 2010.]  
[www.willydev.net/descargas/prev/TodoAgil.pdf](http://www.willydev.net/descargas/prev/TodoAgil.pdf).
9. **Schmuller, Joseph.** *Aprendiendo UML en 24 Horas*. México : PEARSON EDUCACION, 2000 . 968-444-463-X .
10. **Masip, David.** Desarrolloweb.com. [En línea] 19 de julio de 2002. [Citado el: 5 de febrero de 2010.]  
<http://www.desarrolloweb.com/articulos/840.php>.
11. **MySQL Company.** MySQL.com. [En línea] [Citado el: 5 de febrero de 2010.]  
<http://dev.mysql.com/doc/refman/5.0/es/introduction.html>.
12. **PostgreSQL.org.** postgresql-es.org. [En línea] 22 de marzo de 2009. [Citado el: 5 de febrero de 2010.]  
[http://www.postgresql-es.org/sobre\\_postgresql](http://www.postgresql-es.org/sobre_postgresql).
13. **Aguilar, Luis Joyanes.** *Fundamentos de programación. Algoritmos, estructuras*. España : McGraw Hill, 2003. ISBN: 8448139860.

14. **Javier Gracia de Jalón, José Ignacio Rodríguez.** *Aprenda Java como si estuviera en primero.* San Sebastián : s.n., 2002.
15. —. *Aprendiendo C++ como si estuviera en primero.* San Sebastián : s.n., 1998.
16. **Microsoft Corporation.** Visual Studio Developer Center. [En línea] [Citado el: 30 de enero de 2010.] <http://msdn.microsoft.com/en-us/library/aa287558.aspx>.
17. **Python Software Foundation.** Python Programming Language. [En línea] [Citado el: 30 de Enero de 2010.] <http://www.python.org/>.
18. **Álvarez Marañón, Gonzalo.** Características de Java. [En línea] Instituto de Física Aplicada del CSIC, 1999. [Citado el: 31 de enero de 2010.] <http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html>.
19. **Eclipse Foundation.** eclipsecon. [En línea] [Citado el: 2 de febrero de 2010.] <http://www.eclipse.org/>.
20. **NetBenas.org.** NetBeans. [En línea] [Citado el: 2 de enero de 2010.] [http://www.netbeans.com/index\\_es.html](http://www.netbeans.com/index_es.html).
21. **SlideShare Inc.** Borland Together - Administración Bases de Datos. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.slideshare.net/unimauro/borland-together-administracion-bases-de-datos>.
22. **Grupo Soluciones Innova.** Rational Rose Enterprise. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
23. **Visual Paradigm International.** Visual Paradigm. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.visual-paradigm.com/product/vpuml/communityedition.jsp>.
24. **Garlan, David y Shaw, Mary.** *An Introduction to Software Architecture.* New Jersey : School of Computer Science, 1994. CMU-CS-94-166.
25. **Internet Engineering Task Force .** ietf.org. [En línea] [Citado el: 1 de mayo de 2010.] <http://tools.ietf.org/html/rfc3720>.

## BIBLIOGRAFÍA CONSULTADA

1. **Gobierno Mexicano.** ciberHábitat. [En línea] [Citado el: 11 de diciembre de 2009.] <http://www.ciberhabitat.gob.mx/biblioteca/extensiones/>.
2. **EDUARDO ARAGÓN MONTES, JULIÁN CAMILO PÉREZ PARRA.** *SISTEMA PARA EL ALMACENAMIENTO VIRTUAL Y GESTION DE ARCHIVOS .* Cali, Colombia : s.n., 2009.

3. **David Luis la Red Martínez.** Sistemas Operativos. [En línea] [Citado el: 2009 de diciembre de 11.] <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/GESTAR02.htm>.
4. **Ipswitch.** Ipswitch File Transfer. [En línea] [Citado el: 20 de marzo de 2009.] <http://www.ipswitchft.com/Business/Products/WsFtpServer/Index.aspx?is>.
5. **Rebaza, Jorge Carlos Valverde.** *Sockets y su programación en Java*. Trujillo, Perú : s.n., 2007 .
6. **Ibar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educación.S.A, 2000. 84-7829-036-2.
7. **Extreme Programming.** Extreme Programming. [En línea] [Citado el: 23 de enero de 2010.] <http://www.extremeprogramming.org/>.
8. **José H. Canós, Patricio Letelier y M<sup>a</sup> Carmen Penadés.** [En línea] [Citado el: 25 de enero de 2010.] [www.willydev.net/descargas/prev/TodoAgil.pdf](http://www.willydev.net/descargas/prev/TodoAgil.pdf).
9. **Schmuller, Joseph.** *Aprendiendo UML en 24 Horas*. México : PEARSON EDUCACION, 2000 . 968-444-463-X .
10. **Masip, David.** Desarrolloweb.com. [En línea] 19 de julio de 2002. [Citado el: 5 de febrero de 2010.] <http://www.desarrolloweb.com/articulos/840.php>.
11. **MySQL Company.** MySQL.com. [En línea] [Citado el: 5 de febrero de 2010.] <http://dev.mysql.com/doc/refman/5.0/es/introduction.html>.
12. **PostgreSQL.org.** postgresql-es.org. [En línea] 22 de marzo de 2009. [Citado el: 5 de febrero de 2010.] [http://www.postgresql-es.org/sobre\\_postgresql](http://www.postgresql-es.org/sobre_postgresql).
13. **Aguilar, Luis Joyanes.** *Fundamentos de programación. Algoritmos, estructuras*. España : McGraw Hill, 2003. ISBN: 8448139860.
14. **Javier Gracia de Jalón, José Ignacio Rodríguez.** *Aprenda Java como si estuviera en primero*. San Sebastián : s.n., 2002.
15. —. *Aprendiendo C++ como si estuviera en primero*. San Sebastián : s.n., 1998.
16. **Microsoft Corporation.** Visual Studio Developer Center. [En línea] [Citado el: 30 de enero de 2010.] <http://msdn.microsoft.com/en-us/library/aa287558.aspx>.
17. **Python Software Foundation.** Python Programming Language. [En línea] [Citado el: 30 de Enero de 2010.] <http://www.python.org/>.

18. **Álvarez Marañón, Gonzalo.** Características de Java. [En línea] Instituto de Física Aplicada del CSIC, 1999. [Citado el: 31 de enero de 2010.] <http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html>.
19. **Eclipse Foundation.** eclipsecon. [En línea] [Citado el: 2 de febrero de 2010.] <http://www.eclipse.org/>.
20. **NetBenas.org.** NetBeans. [En línea] [Citado el: 2 de enero de 2010.] [http://www.netbeans.com/index\\_es.html](http://www.netbeans.com/index_es.html).
21. **SlideShare Inc.** Borland Together - Administración Bases de Datos. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.slideshare.net/unimauro/borland-together-administracion-bases-de-datos>.
22. **Grupo Soluciones Innova.** Rational Rose Enterprise. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
23. **Visual Paradigm International.** Visual Paradigm. [En línea] [Citado el: 5 de Febrero de 2010.] <http://www.visual-paradigm.com/product/vpuml/communityedition.jsp>.
24. **Garlan, David y Shaw, Mary.** *An Introduction to Software Architecture*. New Jersey : School of Computer Science, 1994. CMU-CS-94-166.
25. **Internet Engineering Task Force .** ietf.org. [En línea] [Citado el: 1 de mayo de 2010.] <http://tools.ietf.org/html/rfc3720>.
26. **Zorilin Alonso Guerrero, Genry Leyva González.** *Sistema de Captura y Transcripción de Audio*. Ciudad de La Habana : s.n., 2009 .
27. **Zayas, Dr. Cs. Carlos Alvarez de.** *Metodología de la Investigación Científica*. Santiago de Cuba : Universidad de Oriente, 1995.
28. **Microsoft Corporation.** Windows Server TechCenter. [En línea] [Citado el: 18 de marzo de 2010.] <http://technet.microsoft.com/es-es/default.aspx>.
29. **MIGUEL GARCIA FEAL, JOSE LUIS CHOUCIÑO FERREIRO.** Biblioteca. [En línea] [Citado el: 23 de enero de 2010.] <http://biblioteca.uci.cu/tit.computacion.htm>.
30. **Alvarez, Sara.** Desarrolloweb.com. [En línea] 31 de julio de 2007. [Citado el: 5 de febrero de 2010.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
31. **ProFTPD.org.** ProFTPD. [En línea] [Citado el: 10 de marzo de 2010.] <http://proftpd.org/>.
32. PergaminoVirtual. *Buscador Hispano*. [En línea] [Citado el: 13 de diciembre de 2009.] <http://www.pergaminovirtual.com.ar/definicion/C.html>.

33. **OIT.** *Organización Internacional del Trabajo.* [En línea] [Citado el: 12 de diciembre de 2009.] <http://www.oit.or.cr/bidiped/Glosario.html>.
34. **Hernández Orallo, Enrique.** El Lenguaje Unificado de Modelado (UML). [En línea] [Citado el: 28 de Enero de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=11361>.
35. **Fujitsu Corporation.** *Un paseo por fibre channel.* España : s.n., 2000.
36. **Abel Díaz Berenguer, Alberto Ramón Román Vieito.** *Sistema de Administración y Configuración de la solución de Captura y .* Ciudad de La Habana : s.n., 2009.
37. [En línea] [Citado el: 12 de diciembre de 2009.] <http://www.e-abaco.net/2009/05/11/tecnologias-de-almacenamiento-de-red-san-nas-das/>.
38. **Kroll Ontrack ®.** Ontrack Data Recovery. [En línea] [Citado el: 13 de diciembre de 2009.] [http://www.ontrackdatarecovery.es/sistAlmacen\\_0904/](http://www.ontrackdatarecovery.es/sistAlmacen_0904/).

## GLOSARIO

**CCM:** Siglas referentes al producto Captura y Catalogación de Medias.

**Hibernate:** Es una herramienta de Mapeo Objeto-Relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación.

**RAM:** La memoria de acceso aleatorio (en inglés: Random-Access Memory) es la memoria desde donde el procesador recibe las instrucciones y guarda los resultados.

**Socket:** Es una interfaz para la comunicación entre dos procesos cualquiera que pertenezcan a un mismo sistema o a dos sistemas diferentes.

**TCP:** Transmission Control Protocol (en español Protocolo de Control de Transmisión), es uno de los protocolos principales de Internet. Es utilizado para crear conexiones entre ordenadores para realizar el envío de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

**iSCSI:** Internet SCSI (iSCSI) es un estándar oficial ratificado el 11 de febrero de 2003 por la Internet Engineering Task Force que permite el uso del protocolo SCSI sobre redes TCP/IP. iSCSI es un protocolo de la capa de transporte definido en las especificaciones SCSI-3 (26).

**Fibre Channel:** El Canal de fibra, del inglés Fibre Channel, es una tecnología de red utilizada principalmente para redes de almacenamiento, disponible primero a la velocidad de 1 Gbps y posteriormente a 2, 4 y 8 Gbps.

**SAN:** Una red de área de almacenamiento, en inglés SAN (storage area network), está concebida para conectar servidores, matrices de discos y librerías de soporte. Principalmente, está basada en tecnología fibre channel y más recientemente en iSCSI. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.