

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6



Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas.

**Título: SIGMPTARTV: Sistema para la Gestión de la
Configuración del Servidor de Media para la Plataforma
de Transmisión Abierta para Radio y Televisión.**



Autor: Eduardo Cepero Utra

Tutor: Ing. Yandy León Núñez

Co-tutor: Ing. Dunier Domínguez Mora

Ciudad de La Habana, mayo de 2011.

Año 53 de la Revolución.



Considero más valiente al que conquista sus deseos, que al que conquista a sus enemigos, ya que la victoria más dura es la victoria sobre uno mismo.

ARISTÓTELES

Dedicatoria

A Dios por darme la oportunidad de conocerle y de vivir esta vida maravillosa.

A mis queridos padres por haberme apoyado en todo momento, por confiar en mí y darme confianza cuando más lo necesitaba, por tanto cariño y afecto que me han brindado. Por darme todo y ser todo para mí.

A mi mamá por devolverme la serenidad con sus consejos cuando asoma la desesperanza.

A mi papá por forjar mi carácter y ser guía en mi camino.

A mi hermana, por ser mi otra mitad, por ser el mejor regalo que me han dado mis padres.

A mis abuelos Anibal y Lelo por quererme tanto, en especial a mi Mima del alma, que me adora con locura.

A mi abuela Norma que está en el cielo.

A mi familia que adoro y está a mi lado en las malas cuando todos se van.

Agradecimientos

Agradecimientos

A la Revolución y a Fidel por haber creado esta Universidad de Excelencia y hacer posible mi formación como ingeniero informático.

A los profesores miembros del tribunal de tesis: Aliosmi, Neysis, Romanuel y Ekaterina, a los profesores: Dayami, Abel y Dunier por su apoyo, y especialmente a mi tutor Yandy, por todos los señalamientos y recomendaciones que me condujeron a un mejor trabajo.

A mis padres y mi hermana por darme todo el apoyo y las ganas de seguir adelante y lograr mis metas; porque a ellos les debo en gran medida quien soy y porque para ellos son mis triunfos.

A mis tíos Dayami, Odalis, Lázaro, Yolanda, Rebeca por su apoyo.

A mi primas Bety, Rebequita, Yolaine y Ana Laura por todo su cariño.

A Sara y Orlando porque son como padres para mí.

A mis compañeros inseparables a lo largo de estos 5 años: Leisys, Leidys, Osbel Ana Belkis, Roxana, Vanessa, Mavis, Lisbet, Liset, Bety, Ivellín, Lisandra, Merlin, Yisel, Kryna, Cecilia, Reynel, Leiber, Aruid, Jeans, Denis, Rafael Sánchez, Pedro, Reinaldo, Anyer, Leosmel, Iuda, Alejandro, Lugo, Alain, Simón, el Gato, Siomelis, Dorgis y a todos aquellos que estuvieron a mi lado y me apoyaron en el transcurso de este largo camino.

A todos los profesores que me han tenido que soportar desde primer año: Yoenis, Elianis, Yunciry, Armando, la Negra, Yunet, Alexander, Héctor, Aníbal Edwin, Sandy, no puede faltar el Primo que siempre ha estado ahí como un hermano y a todos los que aportaron de una forma u otra su granito de arena.

Declaración de autoría

Declaro que soy el único autor del trabajo titulado:

SIGMPTARTV: Sistema para la Gestión de la Configuración del Servidor de Media para la Plataforma de Transmisión Abierta para Radio y Televisión.

Y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2011.

Eduardo Cepero Utra

Ing. Yandy León Núñez.

Firma del Autor

Firma del Tutor

Resumen

La información juega en la actualidad un papel primordial, ya que es la base en la toma de decisiones y la que sustenta toda investigación científica. El desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha posibilitado que su proceso de gestión sea más eficiente.

La Universidad de la Ciencias Informáticas (UCI), fue creada 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 antigua facultad 9, actualmente facultad 6 ha venido desarrollando una serie de productos orientados al trabajo con ficheros multimedia. La Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV) no cuenta con una herramienta que le permita gestionar el almacenamiento de estos archivos.

El objetivo de este trabajo de diploma es desarrollar un sistema que permita automatizar el proceso de gestión de los archivos que se manipulan en la PTARTV. Para ello se realizaron dos aplicaciones, una servidora que se encarga de crear las reglas de configuración necesarias para restringir y dar acceso a los usuarios al servidor, así como controlar todas las operaciones que se realizan sobre estos ficheros. La aplicación cliente está enfocada en los procesos de copia de las medias hacia dicho servidor y la inserción de los metadatos asociados a cada una de ellas. Este sistema mejoró sensiblemente la gestión de las medias que llegan a la plataforma y centralizó el acceso a las mismas por los demás subsistemas; tributando a la experiencia y siendo un avance en el proceso de gestión de archivos de la PTARTV.

Palabras claves

Archivo, media, almacenamiento, transferencia.

Abstract

Information has a leading role due to its great power of making decisions nowadays, and supports all scientific investigations. The development of Information and Communications Technology (TIC) has made the information management process more efficient.

The University of the Informatics Sciences (UCI) arose with the purpose of giving software solutions to automatize those processes and become one of the main economical sources on the country. Digital Signal Department on ancient 9th school, at present 6th school, has developed several products related to the work with multimedia files. The Open Broadcasting Platform for Radio and Television (PTARTV) project doesn't have a subsystem that allows the storage management of these files.

The objective of this dissertation is to develop a system to automate the files management process of the PTARTV. To reach this it was developed two applications: a server, responsible of creating the configuration rules to restrict and give users access to it and control all the operations performed on these files; and the client, that is focused on the process of copying medias to that server and the inclusion of its associated metadata. This system improved significantly the management of the medias that arrive to the platform; it also centralized the access to them by the other subsystems, paying tribute to the experience and being a breakthrough in the files management process of the PTARTV.

KEY WORDS

File, media, storage, and transfer.

Índice

Introducción	12
Capítulo 1. Fundamentación teórica	18
1.1 Introducción.....	18
1.2. Conceptos asociados a la gestión y almacenamiento de archivos.....	18
1.2.1 Archivo.....	18
1.2.2 Almacenamiento de archivos	19
1.2.3 Sistemas de gestión de archivos.....	19
1.2.4 Procesos de gestión de archivos	19
1.2.4.1 Descripción general.....	20
1.2.5 Almacenamiento virtual de archivos.....	20
1.2.6 Sistemas de almacenamiento	21
1.3 Concepción del almacenamiento y gestión de archivos en el producto PTARTV 22	
1.4 Situación problemática	23
1.5 Análisis de soluciones existentes	24
1.6 Conclusiones.....	28
CAPÍTULO 2: Tendencias y tecnologías actuales a desarrollar	29
2.1 Introducción.....	29
2.2 Tecnologías que soportan la solución	29
2.2.1 Protocolo SSH	29
2.2.2 Socket.....	30
2.2.3 Capa de conexión segura (SSL)	31
2.2.4 ICE (Internet Communication Engine).....	32
2.3 Metodología de desarrollo	33
2.3.1 Metodologías tradicionales	34
2.3.1.1 Proceso Unificado de Desarrollo (RUP).....	34
2.3.2 Fundamentación de la metodología propuesta: RUP	36
2.4 Sistema Gestor de Base de Datos (SGBD)	36
2.4.1 PostgreSQL	37
2.4.2 Fundamentación del SGBD propuesto: PostgreSQL.....	38
2.6 Herramientas case	39
2.6.1 Visual Paradigm.....	40
2.6.2 Fundamentación de la herramienta propuesta: Visual Paradigm	41

2.7 Lenguaje de programación.....	41
2.7.1 C++.....	42
2.7.2 Fundamentación del lenguaje propuesto: C++.....	42
2.8 Entornos de Desarrollo Integrado (IDE).....	43
2.8.1 QT Creator.....	43
2.8.2 Fundamentación del entorno de desarrollo integrado: QT Creator.....	44
2.9 Conclusiones.....	44
CAPÍTULO 3: Construcción y validación de la solución propuesta.....	46
3.1 Introducción.....	46
3.2 Requisitos funcionales del software.....	46
3.3 Principios del Diseño.....	47
3.3.1 Estándares de la interfaz de la aplicación.....	47
3.4 Modelo de despliegue.....	48
3.4.1 Descripción de los nodos físicos del diagrama de despliegue.....	49
3.4.2 Descripción de los protocolos de comunicación.....	49
3.5 Modelo de implementación.....	49
3.5.1 Diagrama de implementación de la PTARTV.....	50
3.5.2 Diagrama de componentes.....	50
3.6 Estándar de codificación.....	51
3.6.1 Comentarios.....	51
3.6.2 Nombres de identificadores.....	52
3.6.2.1 Identificadores de variables.....	52
3.6.2.2 Identificadores de punteros (apuntadores).....	53
3.6.2.3 Identificadores de variables dimensionadas (arreglos, matrices).....	53
3.6.2.4 Identificadores de datos constantes.....	53
3.6.2.5 Identificadores de funciones.....	53
3.6.2.6 Identificadores de tipos definidos por el usuario.....	53
3.6.3 Organización visual del programa.....	54
3.6.3.1 Generales.....	54
3.6.3.2 Sangrías.....	54
3.6.3.3 Líneas y espacios en blanco.....	54

3.6.3.4 Paréntesis	55
3.7 Validación de la solución propuesta	55
3.7.1 Elementos del proceso de pruebas	55
3.7.1.1 Niveles de pruebas	56
3.7.1.2 Tipos de prueba	57
3.7.1.2.1 Funcionalidad	57
3.7.1.2.2 Fiabilidad	58
3.7.1.2.3 Rendimiento.....	59
3.7.1.2.4 Capacidad de soporte.....	59
3.7.2 Pruebas funcionales.	68
3.7.2.1 Diseño de pruebas funcionales	69
3.7.3 Pruebas de rendimiento	74
3.7.3.1 Diseño de pruebas de rendimiento.....	75
3.7.4 Resultado de las pruebas	76
3.8 Conclusiones.....	77
Conclusiones generales.....	78
Recomendaciones	79
Bibliografía.....	80
Bibliografía citada.....	80
Glosario de términos.....	83

Introducción

La información es la base de toda entidad y por tanto uno de los recursos principales, ya que sin ella no se podrían desarrollar muchas de las actividades que se encuentran concebidas en su seno, y sobre las cuales se toman muchas decisiones. Con el creciente desarrollo de las Tecnologías de la Información 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 aparición de la informática en casi todos los sectores de la sociedad ha provocado que grandes volúmenes de información se hayan digitalizado, siendo así más fácil su almacenamiento y que muchos procesos se ejecuten de forma automática. Gran cantidad de datos son almacenados en archivos informáticos, alcanzando estos cada vez más proporción, por lo que se necesita de grandes dispositivos para su procesamiento, control y registro en la memoria de un ordenador; acarreado consigo que su gestión sea cada vez más difícil.

Ante el cúmulo y heterogeneidad de datos que manejan las empresas y entidades en general, los profesionales de la tecnología enfrentan el desafío de administrar dicha información, reunida en diferentes lugares de las redes corporativas y disímiles dispositivos de almacenamiento. Debido a esto los especialistas necesitan potentes herramientas que les permita administrar y gestionar de forma segura, rápida y eficiente los datos almacenados. Esta creciente demanda ha sido aprovechada por potentes empresas a nivel mundial, dedicándose a la fabricación de tecnología para servidores, destacándose y convirtiéndose en líderes del mercado por la calidad de sus productos empresas como Apple, HP, Dell, Toshiba, Sony, entre otras.

La producción de hardware con este fin no es la única que se ha desarrollado, sino que existen muchas entidades y comunidades que se dedican a desarrollar software con el fin de gestionar y almacenar la información; llegando a convertirse en un jugoso negocio para determinadas compañías, ya que acarrea una gran demanda la adquisición de esta tecnología. La venta de estos productos a otras empresas genera cuantiosas ganancias, y a su vez éstas con fines de lucro ponen a disposición de la comunidad internacional, gracias a las ventajas de la Internet, vistosos y atractivos

portales que brindan servicios de descarga de archivos de todo tipo a un precio determinado.

Es ejemplo y líder en este negocio la tienda virtual “iTunes Store” de la multimillonaria y reconocida empresa Apple, que no solo desarrolla productos para la red de redes, sino que ha formado un imperio que controla en cuanto a tecnología y dispositivos se refiere, el mundo del entretenimiento. Esta compañía cuenta con potentes aplicaciones para la monitorización de valiosos dispositivos de multimedia y comunicación como iPods y iPhones.

No solo las grandes empresas de software propietario como Microsoft, Intel y la propia Apple se dedican a la producción de este tipo de tecnología; también existen comunidades que desarrollan una gama de productos, bajo tecnología no privativa. La Comunidad de Software Libre ha dado grandes avances en el campo con productos de gran calidad puestos a disposición de la comunidad de forma gratuita.

El desarrollo de software ha demostrado ser un factor importante para la economía de cualquier país o institución. Cuba ha logrado un avance considerable en este sector en los últimos años, con instituciones como DESOFT con sucursales en todo el país y SOFTEL, empresa cubana de software para la salud, que ha desarrollado diversos software para determinadas empresas. Sobresalen entre ellos el Registro Informatizado de Salud y el GalenLab, utilizado en los Centros de Diagnóstico Integral (CDI) de la República Bolivariana de Venezuela.

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 el análisis realizado sobre el mundo del audiovisual, apreciando las posibilidades de mercado que tienen los productos de este tipo, y bajo las políticas del software libre, en el Centro de Desarrollo de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 se está desarrollando una plataforma de transmisión para radio y televisión.

El producto Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV), tiene como objeto automatizar todo el proceso relacionado con la transmisión de

archivos multimedia, a través de un canal correctamente programado, así como la gestión, almacenamiento y codificación de estas medias¹ para brindar una mejor calidad tanto en las transmisiones televisivas como radiales. PTARTV se encuentra estructurado por 11 subsistemas, los cuales se mencionan a continuación:

- ✚ Subsistema Web.
- ✚ Subsistema Radial.
- ✚ Subsistema de Monitoreo.
- ✚ Subsistema de Programación.
- ✚ Subsistema de la Transmisión y Administración de la Transmisión.
- ✚ Subsistema de Seguridad.
- ✚ Subsistema de Transferencia (Codificador).
- ✚ Subsistema de Gestión de Medias.
- ✚ Subsistema de Reporte.
- ✚ Subsistema de Equipamiento.
- ✚ Subsistema de Producción.

En el desarrollo de las actividades tales como el proceso de programación, transferencia o conversión y transmisión de estos subsistemas, se hace imprescindible el uso de las medias como material principal, por lo que es necesario contar con una buena gestión las medias hacia un servidor. Esto permitirá almacenar la información y garantizar la disponibilidad y centralización de estas medias, así como mantener una correspondencia entre las medias en el servidor y las insertadas en la base de datos de la PTARTV.

El personal que se encuentra al frente del desarrollo del producto ha hecho un análisis de los precios de productos software que gestionan el contenido de un servidor de medias capacitado para centralizar los archivos y distribuirlos hasta donde se requieran, 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

¹ **Media:** Archivo de audio o video.

representa. Una inversión de este tipo disminuiría las posibilidades de mercado una vez que se vaya a comercializar, debido al incremento de los costos.

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, que brinde a sus usuarios una interfaz simple y agradable que facilite la gestión de archivos. Ante la presente situación se ha determinado como **problema a resolver** ¿cómo automatizar la gestión de la configuración y la transferencia de las medias al servidor de archivos multimedia de la plataforma PTARTV?

Para dar solución al problema detectado se ha planteado como **objetivo general** desarrollar una aplicación que gestione la transferencia de medias al servidor de archivos multimedia, así como los elementos de configuración del mismo en la Plataforma de Transmisión de Abierta de Radio y Televisión, teniendo como **objeto de estudio** a los procesos de gestión de archivos. Se ha definido como **campo de acción** los procesos para la gestión de ficheros en el servidor de medias de la PTARTV.

Se **defiende la idea** de que la implementación de una aplicación capaz de gestionar las medias hacia el servidor, así como la configuración del mismo garantizará una mejora de los procesos implicados y la unificación en la forma de acceso a las medias por los subsistemas de la PTARTV de la facultad 6.

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

1. Caracterizar las soluciones o sistemas existentes en la UCI y en el ámbito nacional e internacional para la gestión y transferencia de archivos multimedia.
2. Caracterizar los procedimientos y técnicas asociados a la transferencia y gestión de la configuración en servidores de archivos multimedia.
3. Determinar las tendencias y tecnologías de desarrollo más factibles para la construcción del sistema.
4. Desarrollar el modelo de Despliegue.
5. Construir el modelo de Implementación.
6. Implementar las funcionalidades críticas del sistema.
7. Validar el sistema desarrollado.

Para estudiar más a fondo las características del proceso de gestión de medias se han empleado diferentes métodos científicos de investigación, tanto teóricos como empíricos. Los métodos teóricos utilizados fueron el analítico - sintético, el análisis histórico - lógico, el inductivo - deductivo y la modelación. El método empírico utilizado es el del experimento.

El **análisis histórico - lógico**: Se empleó para realizar un estudio analítico de la trayectoria histórica del objeto de estudio de la presente investigación, 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.

El **método analítico - sintético**: Permitió concretar los elementos más importantes relacionados con el proceso de gestión y almacenamiento de archivos, que necesita el producto Plataforma de Transmisión Abierta para Radio y Televisión. 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.

La **modelación** es un método que permite la creación de modelos, que no son más que una reproducción simplificada de la realidad y que proporciona una mejor comprensión de las características del objeto de estudio y el descubrimiento de nuevas relaciones. Con el objetivo de realizar el análisis y diseño del software y posteriormente la implementación del prototipo funcional.

El **método inductivo - deductivo**: Posibilitó arribar a conocimientos generalizadores del proceso de gestión de la configuración de los servidores de medias, a partir de los aspectos particulares y generales encontrados en los documentos revisados, analizándolos tanto de lo particular a lo general como viceversa.

El método empírico **experimento**, ha sido útil para demostrar, mediante pruebas en las que se modifica de forma controlada las condiciones en las que funciona el sistema, la eficacia de las funcionalidades del mismo.

El contenido del presente trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1. Fundamentación teórica: se expone el estado del arte del objeto de estudio de la presente investigación y se definen los elementos teóricos que lo sustentan. Se enuncian conceptos que posibilitan un mejor entendimiento de lo planteado en la situación problemática y el marco del problema en sentido general. Se enuncian y argumentan otras aplicaciones ya existentes que aportan de alguna manera a la solución del problema del presente trabajo.

Capítulo 2. Tendencias y tecnologías actuales a desarrollar: se describe el lenguaje, la metodología y las tecnologías a considerar para su posterior utilización en el desarrollo de la aplicación, analizando sus características, ventajas y desventajas; con el objetivo de dar cumplimiento con la mayor eficiencia y calidad posible al objetivo general de la presente investigación.

Capítulo 3. Construcción y validación de la solución propuesta: se exponen algunos de los principales artefactos generados en los Flujos de Trabajo Implementación, Prueba y Despliegue necesarios en el desarrollo del software que dará solución al problema científico de la presente investigación. Las características generales de la implementación, los principios para el diseño gráfico, el plan de pruebas y los casos de pruebas diseñados para aplicar al sistema desarrollado.

Capítulo 1. Fundamentación teórica

1.1 Introducción

Para comprender correctamente en qué consiste la presente investigación, se hace necesario dominar los aspectos más significativos relacionados con el objeto de estudio de la misma: el proceso de gestión de archivos en el servidor de medias. Con el fin de hacer posible lo anterior ha sido redactado este capítulo, en el cual se exponen los conceptos y elementos que constituyen la base teórica para la realización del presente trabajo de diploma y lógicamente para dar solución al problema científico planteado.

1.2. Conceptos asociados a la 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 sujeta 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 PTARTV, que provee almacenamiento básico mediante la interconexión con otros sistemas existentes.

Para un mayor acercamiento a las características del proceso de gestión de la configuración en servidor de archivos multimedia y un mejor entendimiento de lo planteado en la situación problemática, es imprescindible enunciar varios conceptos, así como abordar detalles de estos.

1.2.1 Archivo

Espacio que se reserva en el dispositivo de memoria de un computador para almacenar porciones de información que tienen la misma estructura y que pueden manejarse mediante una instrucción única **(1)**.



Figura 1. Archivos

Un archivo es un contenedor de información. La mayoría de los archivos que se utilizan contienen información (datos) en un formato determinado: un documento, una hoja de cálculo, un gráfico. El formato es la disposición de los datos dentro del archivo **(2)**.

1.2.2 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, RAM²) y las tarjetas de almacenamiento portátil con interfaz USB³. 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 (3)**.

1.2.3 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 **(4)**.

1.2.4 Procesos de gestión de archivos

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

³ **USB:** Siglas en inglés de *Universal Serial Bus*.

1.2.4.1 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.
- ✚ 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. **(7)**

1.2.5 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. **(3)**

1.2.6 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 SAN⁴ y NAS⁵ han demostrado su excelente fiabilidad.

Teniendo en cuenta que los procesos concebidos en el producto PTARTV necesitan de 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:

- ✚ Protección de la inversión actual y futura / Conectividad modular.
- ✚ Red de almacenamiento de altas prestaciones.
- ✚ Gran ancho de banda.
- ✚ Centralización del backup/ Copia de seguridad independiente de la LAN.
- ✚ Tolerancia a fallos.
- ✚ Compartición de ficheros entre servidores en entornos heterogéneos.
- ✚ Alta escalabilidad y larga distancia entre nodos de la red.
- ✚ Alta disponibilidad.
- ✚ Fácil administración/ Gestión centralizada.
- ✚ Fácil integración.
- ✚ Bajo coste de mantenimiento. **(8)**

⁴ **SAN**: Por sus siglas en inglés (**Storage Area Network**) o Red de Área de Almacenamiento.

⁵ **NAS**: Del inglés (**Network Attached Storage**), tecnología de almacenamiento que comparte la capacidad de almacenamiento de un servidor a través de la red.

1.3 Concepción del almacenamiento y gestión de archivos en el producto PTARTV

El producto PTARTV está previsto para que en su composición cuente con 11 subsistemas, de los cuales 9 son desarrollados por el proyecto PTARTV:

- 1- Subsistema Web:** Encargado de la gestión de noticias, encuestas publicaciones de medias y de las secciones en las cuales son publicadas.
- 2- Subsistema de Programación:** Encargado de realizar las programaciones de las medias que serán transmitidas por canales o por la radio.
- 3- Subsistema de la Transmisión y Administración de la Transmisión:** Encargado de transmitir las programaciones realizadas previamente por el subsistema anteriormente descrito, así como de gestionar los datos de los canales de la plataforma.
- 4- Subsistema de Gestión de la Configuración y Gestión de Medias:** Encargado de gestionar las medias hacia el servidor, así como la configuración del mismo garantizando una mejora de los procesos implicados y la unificación de la forma de acceso a las medias por los subsistemas de la PTARTV.
- 5- Subsistema de Transferencia (Codificador):** Encargado de convertir las medias gestionadas por el Subsistema de Gestión de la Configuración y Gestión de Medias al formato establecido para la transmisión.
- 6- Subsistema de Reporte:** Encargado de controlar y gestionar todos los reportes que generen los subsistemas.
- 7- Subsistema de Seguridad:** Encargado de garantizar en todos los aspectos relacionados con la gestión de roles, usuarios y permisos que garantizan la seguridad del producto PTARTV.
- 8- Subsistema de Monitoreo:** Encargado de monitorear las transmisiones realizadas a través del subsistema de transmisión de la plataforma.
- 9- Subsistema Radial:** Encargado de la transmisión radial.

Para el funcionamiento de los subsistemas 2, 3, 5 y 9 es necesaria la existencia de un servidor de medias, con el fin de mantener centralizada la información y permitir su gestión. En los procesos de captura tanto de audio como de video, se debe efectuar el copiado del material capturado, a un lugar donde esté disponible para su posterior transmisión y codificación. Actualmente la información es almacenada localmente en

Capítulo 7. Fundamentación Teórica

los mismos puestos de trabajo (PCs del equipo de trabajo), ya que no existe un sistema para la gestión de la configuración del servidor de medias de la Plataforma de Transmisión Abierta para Radio y Televisión.

En el producto PTARTV es imprescindible 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 programación, transferencia, transmisión y administración de la transmisión, así como el radial. Como mecanismo de seguridad y para mantener una comunicación eficiente se proponen lineamientos o políticas que tienen que seguir los módulos que interactuarán con el sistema de gestión. A continuación se describen los requisitos necesarios para poder establecer el intercambio con el servidor:

- 1- Los diferentes módulos que precisen de 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 registrarse en el sistema donde se encuentra la aplicación cliente para poder efectuar operaciones de copia de archivos.

La administración del servidor se efectuará de forma local, permitiendo eliminar físicamente los ficheros. Además permitirá la búsqueda especializada, entre otras operaciones.

1.4 Situación problemática

Con el aumento del uso de los materiales audiovisuales en diferentes sectores se hace imperiosa y cada vez más difícil su distribución y gestión. Esto ha provocado un aumento en el desarrollo de aplicaciones que permitan realizar este tipo de operaciones de forma eficiente.

PTARTV está concebido para trabajar con ficheros multimedia, definiendo operaciones sobre estas medias almacenadas, como la copia, transferencia, publicación, programación y transmisión. Los ficheros relacionados con audio y video tienen la

Capítulo 7. Fundamentación Teórica

característica de poseer grandes tamaños, es por eso que para su almacenamiento y gestión se requieren capacidades superiores a las que se utilizan con ficheros de textos. Tomando en cuenta esto fue ideado como solución, el uso de un servidor de archivos 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 inevitable 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 para dar solución al problema detectado. Esto permitirá reducir los costos del producto final y tendrá la característica de ser una solución propia de la Plataforma de Transmisión Abierta para Radio y Televisión.

La nueva herramienta brindará un alto nivel de compatibilidad entre las aplicaciones encontradas en los diferentes subsistemas, y gestionará la información de dispositivos de almacenamiento externos, permitiendo así el incremento de capacidad. Permitirá a su vez que los archivos estén centralizados y que no existan fallas en la disponibilidad de las medias.

1.5 Análisis de soluciones existentes

Existen diversos software que brindan una solución a la problemática planteada, pues permiten el intercambio de ficheros entre un cliente y un servidor. 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. 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.

Capítulo 7. Fundamentación Teórica

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. Sirve especialmente para el envío de imágenes de gran tamaño, planos tanto de piezas como de estructuras o de viviendas con autoCAD⁶, fichas técnicas y toda clase de ficheros pesados, que de otra manera pueden tener un envío o recepción problemáticos. **(9)**.

Requisitos 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 IP⁷.

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 el ordenador.

Requisitos de uso:

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

Pese a presentar todas estas ventajas no es posible su integración a los subsistemas de la PTARTV y es una herramienta de software propietario.

ProFTPD

⁶ **AutoCAD**: Plano, utilizado para imágenes grandes, por ejemplo el plano de una vivienda.

⁷ **IP**: Internet Protocol.

Capítulo 7. Fundamentación Teórica

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. 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 SSL⁸/TLS⁹, RADIUS¹⁰, LDAP¹¹ o SQL¹² como módulos. ProFTPD no se adapta a las necesidades de la PTARTV, puesto que la misma necesita registrar de las medias los metadatos asociados, lo cual no es una funcionalidad de este software.

WS_FTP Server

El Ipswitch WS_FTP Server con SSH (Secure Shell) 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 AES¹³ sobre conexiones SSL y SSH, sondeo de integridad de archivos de hasta SHA¹⁴-512, 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. **(6)**.

Servidor de archivos de Windows Server

⁸ **TSL:** Transport Layer Security

⁹ **SSL:** Secure Socket Layer

¹⁰ **RADIUS:** Remote Authentication Dial-In User Server

¹¹ **LDAP:** Lightweight Directory Access Protocol

¹² **QSL:** Structured Query Language

¹³ **AES:** Advanced Encryption Standard

¹⁴ **SHA:** Secure Hash Algorithm

Capítulo 7. Fundamentación Teórica

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 PTARTV, se dependería del uso del sistema operativo Windows en algunos casos, lo que implicaría tener que pagar por su uso. En caso de utilizar 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, ya que el sistema se tendría que adaptar al reducido número de servicios que proporcionan estas soluciones FTP de forma muy general.

Un servidor de almacenamiento y gestión de archivos permitirá cumplir con todas las funcionalidades que requieren los subsistemas de la plataforma que hacen uso de las medias. 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, basado en flujos de mensajes.

Teniendo en cuenta las desventajas considerables que implicarían el uso de las soluciones anteriormente analizadas para el producto PTARTV, 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.

1.6 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. También se pudo apreciar la implicación que trae el uso de soluciones externas al producto PTARTV. 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 PTARTV, teniendo en cuenta algunos requisitos como, la seguridad, portabilidad, soporte, flexibilidad a cambios, incorporación de funcionalidades 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. El propósito del estudio que se brinda a continuación está basado en dar a conocer que tecnologías son más factibles y el por qué.

2.2 Tecnologías que soportan la solución

La selección de las herramientas adecuadas para el desarrollo de un producto de software, requiere de un detallado análisis a la hora de elegir las más adecuadas, con el fin de aprovechar más la potencialidades que brindan y facilitar el proceso. Para lograr la comunicación entre las estaciones de trabajo y el servidor de archivos se necesita de un proceso de comunicación que sea sobre todas las cosas seguro, por lo que se realiza un estudio de las tecnologías y protocolos de comunicación más usados actualmente.

2.2.1 Protocolo SSH

SSH es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. El protocolo SSH proporciona los

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

siguientes tipos de protección: Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.

- ✚ El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- ✚ Todos los datos enviados y recibidos durante la sesión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremadamente difícil de descifrar y leer.
- ✚ El cliente tiene la posibilidad de reenviar aplicaciones X11¹⁵ desde el servidor. Esta técnica, llamada reenvío por X11, proporciona un medio seguro para usar aplicaciones gráficas sobre una red. **(24)**

2.2.2 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.

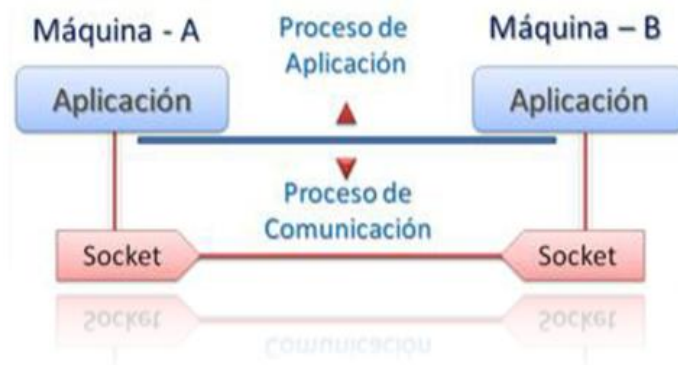


Figura 2: Representación del funcionamiento de un Socket.

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 **(10)**.

¹⁵ **X11**: se refiere al sistema de visión por ventanas X11R6.7, tradicionalmente llamado sistema de ventanas X o simplemente X. Red Hat Enterprise Linux contiene XFree86, un sistema de ventanas X de código abierto.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

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, máximo de conexiones, y su integración con una capa de conexión segura (SSL), entre otras.

2.2.3 Capa de conexión segura (SSL)

Secure Sockets Layer (Protocolo de Capa de Conexión Segura), proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (eavesdropping), la falsificación de la identidad del remitente (phishing) y alterar la integridad del mensaje.

SSL implica una serie de fases básicas:

- ✚ Negociar entre las partes el algoritmo que se usará en la comunicación.
- ✚ Intercambio de claves públicas y autenticación basada en certificados digitales.
- ✚ Cifrado del tráfico basado en cifrado simétrico.

Durante la primera fase, el cliente y el servidor negocian qué algoritmos criptográficos se van a usar.

Protección de la Integridad. SSL protege contra la modificación de los mensajes.

Autenticación. En la mayoría de los modos, SSL proporciona autenticación. Los servidores son por lo general autenticados y los clientes pueden ser autenticados al solicitar comunicación con el servidor.

Confidencialidad. En la mayoría de los modos, SSL cifra los datos enviados entre el cliente y el servidor. Esto protege la confidencialidad de los datos, de modo que antes una situación de un ataque pasivo por parte de un individuo, no verá los datos sensibles, como información personal de muchos tipos. **(11)**.

2.2.4 ICE (Internet Communication Engine)

Es un middleware¹⁶ útil para programadores, es libre, con licencia GPL, desarrollado por la empresa ZeroC. ICE no es más que una plataforma de comunicación de alto rendimiento a través de Internet.

ICE está orientado a objetos, disponible para plataformas heterogéneas, posibilitando que la plataforma sea eficiente en cuanto al uso del CPU, memoria y ancho de banda e incluye una gran variedad de servicios. Está formado por los siguientes paquetes:

- ✚ **Slice:** El lenguaje de especificación para Ice. Establece el contacto entre servidores y clientes y además es utilizado para describir datos persistentes.
- ✚ **Compiladores Slice:** Las especificaciones en Slice se pueden compilar en varios lenguajes de programación: C++, Java, Pitón, PHP, C# y Visual Basic. Los servidores y clientes Ice trabajan juntos, a pesar del lenguaje de programación.
- ✚ **Ice:** El núcleo de librerías Ice, entre otras características este paquete gestiona todas las tareas de comunicación utilizando un protocolo de altamente eficiente, proporciona soporte de hilos para servidores multihilos y funcionalidades adicionales para soportar escalabilidad extrema con a priori millones de objetos Ice.
- ✚ **IceUtil:** una colección utilidades como manejo de unicode y programación con hilos (C++ solo).
- ✚ **IceBox:** Un servidor de aplicación específico para aplicaciones Ice. IceBox puede ejecutar y administrar servicios Ice que son cargados dinámicamente como una DLL, librería compartida o clase Java.
- ✚ **IceGrid:** Un sofisticado servidor de activación y herramienta de despliegue para computación grid avanzada.
- ✚ **Freeze:** proporciona persistencia automática para servants Ice. Con unas pocas líneas de código, una aplicación puede incorporar un vector altamente escalable que gestiona eficientemente objetos persistentes.
- ✚ **FreezeScript:** es necesario para cambiar tipos de datos persistentes, especialmente en proyectos de software grandes. Para minimizar el impacto de estos cambios, FreezeScript proporciona herramientas de inspección y

¹⁶ **middleware:** Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

migración para bases de datos Freeze. La herramienta soporta scripts basados en XML ya que es potente y fácil de usar.

- ✚ **IceSSL:** Un plug-in de transporte SSL dinámico para el núcleo Ice. Proporciona autenticación, encriptación e integridad de mensaje, utilizando un protocolo SSL estándar.
- ✚ **Glaciar:** Algunos retos de los sistemas middleware son la seguridad y los firewalls. Glaciar es una solución firewall de Ice, que simplifica el despliegue de aplicaciones seguras. Autentica y filtra las solicitudes de los clientes y permite callbacks al cliente de una manera segura. Utilizado en combinación con IceSSL, proporciona una solución segura que no permite intrusos y es fácil de configurar.
- ✚ **IceStorm:** un servicio de mensajería que soporta la federación. En comparación con otros servicios de mensajería o eventos, IceStorm soporta tipado de eventos, entendiendo que la difusión de un mensaje en una federación es tan fácil como invocar un método sobre un interface.
- ✚ **IcePath:** es un servicio de parcheado para distribuciones software. Mantener el software actualizado es una tarea tediosa. IcePath automatiza las actualizaciones tanto de archivos individuales y de jerarquías de directorios. Solo los archivos modificados son descargados en las máquinas cliente, utilizando eficientemente algoritmos de compresión. **(30)**

Todo proceso de desarrollo de software debe estar guiado por alguna metodología de desarrollo, que permita planificarlo y controlarlo.

2.3 Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software. Una metodología está compuesta por:

- ✚ Cómo dividir un proyecto en etapas.
- ✚ Qué tareas se llevan a cabo en cada etapa.
- ✚ Qué restricciones deben aplicarse.
- ✚ Qué técnicas y herramientas se emplean.
- ✚ Cómo se controla y gestiona un proyecto. **(13)**

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

2.3.1 Metodologías tradicionales

Las metodologías tradicionales son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

2.3.1.1 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 4P¹⁷, sustentan el proceso de desarrollo de software. RUP define para cada fase: 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 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 **(12)**.

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.

Además de estar compuesto por las fases anteriormente mencionadas, también cuenta con 9 flujos de trabajo:

¹⁷ **4 P:** (proyecto, personas, producto y proceso).

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

- ✚ **Modelado del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ✚ **Requisitos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✚ **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ✚ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✚ **Prueba:** Busca los defectos a lo largo del ciclo de vida.
- ✚ **Despliegue:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios) para entregar el software a los usuarios finales.
- ✚ **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✚ **Configuración y control de cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ✚ **Entorno:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. **(31)**

Los primeros seis son conocidos como flujos de ingeniería y los tres últimos como de apoyo. En la Figura 2 se muestra como varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

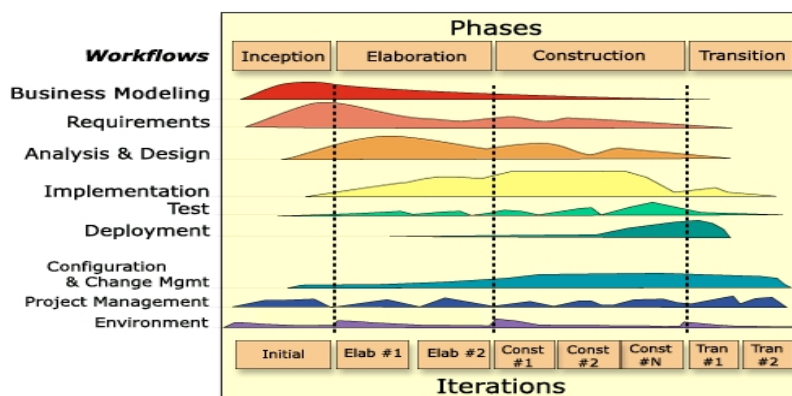


Figura 2: Imagen representativa de la estructura de RUP

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

2.3.2 Fundamentación de la metodología propuesta: RUP

La metodología RUP fue seleccionada pues es muy utilizada para la documentación y realización de software orientados a objetos. El equipo de desarrollo cuenta con experiencia en el uso de esta metodología, aumentando la rapidez del desarrollo del producto. RUP propone varias iteraciones en el proceso de desarrollo de software posibilitando la corrección de errores y mejoras del software a medida que avanza el proyecto, lo que permite a los desarrolladores poder realizar un producto con gran calidad.

Brinda una fuerte y abundante documentación, permitiendo un futuro entendimiento de cada componente del producto. Esto es fundamental para la PTARTV pues con esta metodología el sistema se va desarrollando y documentando al mismo tiempo, por si algún miembro del equipo no puede seguir con el trabajo el que ocupe su lugar tenga por donde guiarse, y conozca lo que se realizó hasta la fecha. La Plataforma de Transmisión Abierta para Radio y Televisión necesita de una herramienta gestora que le permita mantener y actualizar la base de datos con que cuenta.

2.4 Sistema Gestor de Base de Datos (SGBD)

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

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

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

- ✚ Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

- ✚ Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- ✚ Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad. **(23)**

2.4.1 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD¹⁸ 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 multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

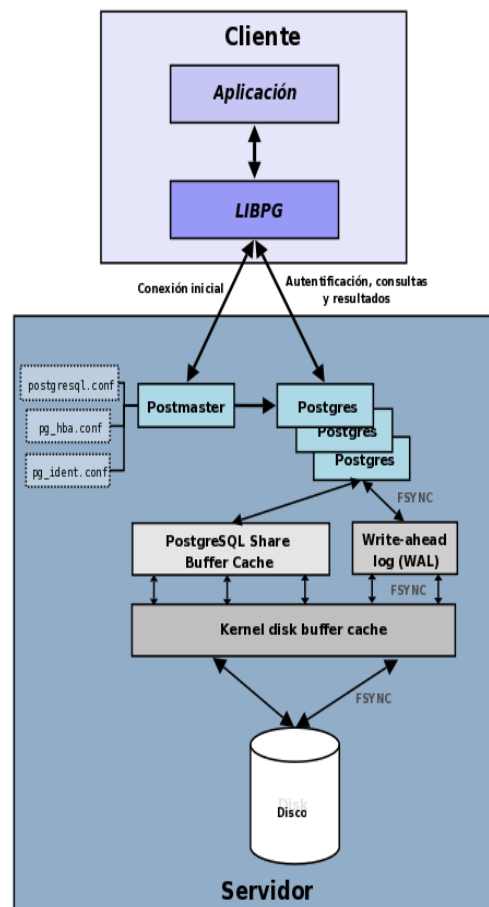


Figura 3: Gráfico que ilustra de manera general los componentes más importantes en un sistema PostgreSQL

¹⁸ **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.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

- ✚ **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales.
- ✚ **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes
- ✚ **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- ✚ **Procesos hijos postgres:** Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- ✚ **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- ✚ **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)
- ✚ **Kernel disk buffer cache:** Caché de disco del sistema operativo
- ✚ **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione. **(14)**

2.4.2 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 la PTARTV, ya que éstas utilizan este SGBD. Para una

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

mejor comprensión de los procesos implicados en el desarrollo de la solución propuesta se necesita como paso fundamental la modelación de muchos de ellos.

2.5 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 **(16)**.

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).

2.6 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 **(28)**. Las herramientas CASE son

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

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.

2.6.1 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 **(22)**. 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 objetos.
- ✚ 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.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

- ✚ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requisitos 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. **(29)**

2.6.2 Fundamentación de la herramienta propuesta: Visual Paradigm

Se selecciona esta herramienta para la modelación del sistema por las facilidades que brinda para el diseño UML del ciclo de vida de un producto de software. 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.

Además esta herramienta 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 y brinda la posibilidad de integrarse con los servidores de control de versiones (Subversion), siendo una de las características fundamentales por las cuales se propone su uso, especialmente la versión 6.4 Enterprise Edition.

Para la implementación de cada una de las funcionalidades y el sistema como tal se requiere de un lenguaje de programación, robusto y que satisfaga los requisitos de la plataforma.

2.7 Lenguaje 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. **(17)**. Permite al programador especificar de manera precisa sobre qué datos debe

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. **(18)**

2.7.1 C++

El C++ es un lenguaje de programación híbrido, que se puede compilar. Una de las ventajas que ofrece este lenguaje es que es mucho más sencillo de aprender para los programadores que ya conocen el C. El C++ mantiene una enorme compatibilidad con el C principalmente por dos razones: Por la gran cantidad de código C que existe, y para facilitar el paso de los programadores de C al nuevo lenguaje C++.

El lenguaje C++ es un lenguaje de programación orientado a objetos, pero no es un lenguaje orientado a objetos puro. El C++ nació como evolución del C, y desde su creación fue un lenguaje de programación hecho por programadores con un diseño muy práctico al que se le fueron añadiendo todos los elementos que se comprobaron eran necesarios sin tener en cuenta aspectos como su imagen, diseño, etc. **(19)**

2.7.2 Fundamentación del lenguaje propuesto: C++

C++ es uno de lenguajes de programación más robustos de la actualidad y fue el seleccionado para la implementación de los subsistemas de escritorio de la PTARTV, por las disímiles ventajas que presenta.

Las principales ventajas que presenta el lenguaje C++ son:

- ✚ **Difusión:** al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos, páginas web, etc. dedicado a él.
- ✚ **Versatilidad:** C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- ✚ **Portabilidad:** el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- ✚ **Eficiencia:** C++ es uno de los lenguajes más rápidos en cuanto ejecución.
- ✚ **Herramientas:** existe una gran cantidad de compiladores, depuradores, librerías, entre otros. **(25)**

2.8 Entornos de Desarrollo Integrado (IDE)

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 único 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 QT Creator

Qt es un marco de trabajo multiplataforma para el desarrollo de aplicaciones e interfaces de usuario. Presenta un gran número de características entre las que se destacan las siguientes: es una intuitiva librería de C++, garantiza la portabilidad entre sistemas operativos embebidos y de escritorio, presenta herramientas de desarrollo integrado con IDE multiplataforma y un alto desempeño en dispositivos embebidos.

Entre las posibilidades que brinda este marco de trabajo está la del trabajo con hilos independientemente del sistema operativo, contiene un módulo para el trabajo con protocolos de red, posee soporte para aplicaciones orientadas a componentes, trabajo con los gestores de bases de datos más conocidos. Hasta el momento ha sido liberado bajo dos licencias, la LGPL y la comercial **(20)**.

Qt Creator es un entorno multiplataforma de desarrollo integrado adaptado a las necesidades de los desarrolladores. Se puede ejecutar en Windows, Linux/X11 y MacOS X permitiendo a los desarrolladores crear aplicaciones de escritorio y plataformas de dispositivos móviles. A continuación se detallan sus características más relevantes:

- ✚ Editor de código sofisticado. Avanzado editor de código de Qt Creador provee soporte para edición de C++, sensible al contexto de ayuda y completamiento de código.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

- ✚ El control de versiones. Qt Creator se integra con la mayoría de los sistemas de control de versiones populares, incluyendo Git, Subversion, Perforce, CVS y Mercurial.
- ✚ Los diseñadores de interfaz de usuario. Qt Creator ofrece el editor visual integrado: Qt Diseñador, muy intuitivo para el desarrollo de interfaces.
- ✚ Proyecto y Construcción. Si se importa un proyecto existente o se crea uno desde cero, Qt Creator genera todos los archivos necesarios.
- ✚ Escritorio y objetivos móviles. Qt Creator ofrece soporte para crear y ejecutar aplicaciones Qt para equipos de escritorio y dispositivos móviles (21).

2.8.2 Fundamentación del entorno de desarrollo integrado: QT Creator

Se selecciona QT Creator como IDE en su versión 4.7 ya que ofrece innumerables ventajas para realizar un proyecto de software, ya que el programador encontrará en él ayuda sensible al contexto, así como resaltado de sintaxis y completamiento de código. Este IDE fue definido por el equipo de desarrollo para todos los subsistemas de escritorio de la PTARTV. Además de las características antes mencionadas se debe resaltar que las bibliotecas Qt son multiplataforma y las aplicaciones que se apoyan en ellas tienen buena respuesta y un consumo de recursos aceptable.

2.9 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. Concluyendo que el PostgreSQL será el sistema gestor de bases de datos a utilizar.

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.

Capítulo 2: Tendencias y tecnologías actuales a desarrollar.

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.

C++ con el IDE QtCreator 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, así como fue definido para la implementación de los subsistemas en la PTARTV por lo cual se convierte en la propuesta para el desarrollo de la aplicación.

CAPÍTULO 3: Construcción y validación de la solución propuesta

3.1 Introducción

En este capítulo se abordan temas fundamentales para la construcción y validación del sistema propuesto como los diagramas de Implementación, Componentes y Despliegue 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.

Se describirán las pruebas funcionales y de rendimiento realizadas al Subsistema de Gestión de Medias.

3.2 Requisitos funcionales del software

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. Listar archivos

El sistema debe permitir listar los archivos que se encuentran en el servidor de medias.

R3. Gestionar archivo

El sistema debe ser capaz de permitir adicionar, buscar, y eliminar archivos del servidor, consecuentemente con lo que está almacenado en la base de datos, en un trabajo paralelo con la misma.

R4. Visualizar reporte

R4.1 Generar reporte de actividades

Se deberá realizar un reporte de las actividades del servidor, donde se deberán apreciar el número de ficheros subidos y eliminados tanto físicos como en Base de Datos.

Capítulo 3: Construcción y validación de la solución propuesta.

R4.3. Visualizar el estado de capacidad

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

R5. Gestionar configuración del servidor

El sistema debe ser capaz de gestionar toda la configuración para acceder al servidor y llevar a cabo las actividades requeridas.

3.3 Principios del Diseño

3.3.1 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 este 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 el 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 y el azul 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.

Capítulo 3: Construcción y validación de la solución propuesta.

3.4 Generalidades de la implementación

La aplicación es un sistema distribuido, que contará con un software que realiza la función de cliente y otro como servidor. El cliente, en cada una de las estaciones de trabajo que necesitan comunicarse con el servidor, realizará operaciones de añadidura y eliminación de archivos. No posee base de datos propia pero si realiza operaciones de autenticación, consulta e inserción en las diferentes tablas de la base de datos correspondiente al producto de PTARTV del cual forma parte. Esta base de datos está soportada sobre PostgreSQL y para efectuar la conexión a la misma se utiliza el driver QPSQL.

Para la transferencia de ficheros a través de la red desde las estaciones clientes se hace uso de algunas de las clases pertenecientes al paquete de clases predefinidas en el IDE QT Creator como QFile. Las clases y métodos han sido nombrados de manera sencilla representando de manera intuitiva las funcionalidades que contienen.

La confección de modelos tales como, el modelo de despliegue, implementación y de componentes es de gran utilidad y guía a la hora de construir el subsistema.

3.4 Modelo de despliegue

El Modelo de Despliegue muestra las relaciones físicas entre los componentes de hardware y software en el sistema, entrelazadas por conexiones de comunicación y los protocolos empleados en 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.

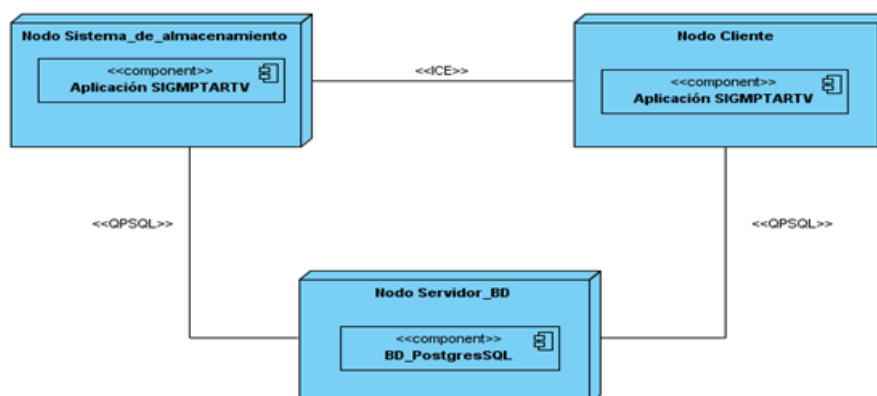


Figura 5: Diagrama de Despliegue.

Capítulo 3: Construcción y validación de la solución propuesta.

3.4.1 Descripción de los nodos físicos del diagrama de despliegue

Servidor de Archivos: Representa el nodo que contiene todos los servicios del subsistema para la Gestión de la Configuración del Servidor de Media para la Plataforma de Transmisión Abierta para Radio y Televisión. Estos servicios son: gestión de la configuración, copia de archivos, así como su gestión.

Aplicación SIGPTARTV: Representa el nodo que contiene los servicios subida de archivos, así como su gestión desde el lado del cliente.

Servidor de base de datos: Es el nodo en el que se montará el sistema gestor de base de datos, en este caso PostgreSQL.

3.4.2 Descripción de los protocolos de comunicación

QPSQL: Protocolo de comunicación entre las aplicaciones con la base de datos.

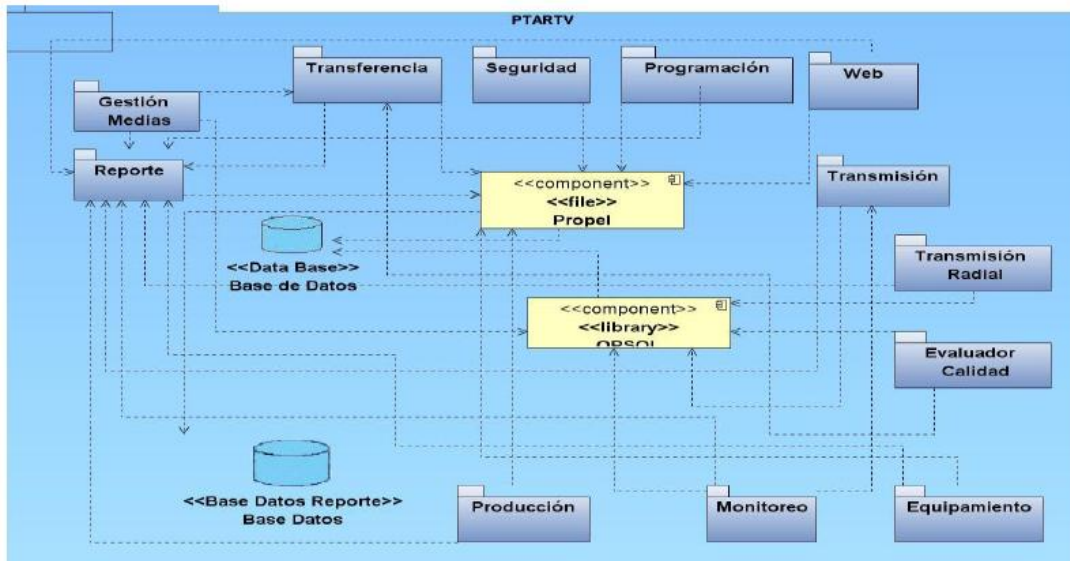
ICE: Protocolo de comunicación entre las PC clientes y el servidor de archivos.

3.5 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 que se utilizarán en la construcción del subsistema.

Capítulo 3: Construcción y validación de la solución propuesta.

3.5.1 Diagrama de implementación de la PTARTV



3.5.2 Diagrama de componentes

Partiendo de los diagramas de componentes se estructura el modelo de implementación, mostrando el funcionamiento de todos los subsistemas de un software funcionando como un todo. Al utilizar la metodología RUP debe conocerse que la misma hace énfasis en el uso de diagramas de componentes para modelar las dependencias entre los componentes de los subsistemas de implementación.

“Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, entre otros. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente (26).”

Capítulo 3: Construcción y validación de la solución propuesta.

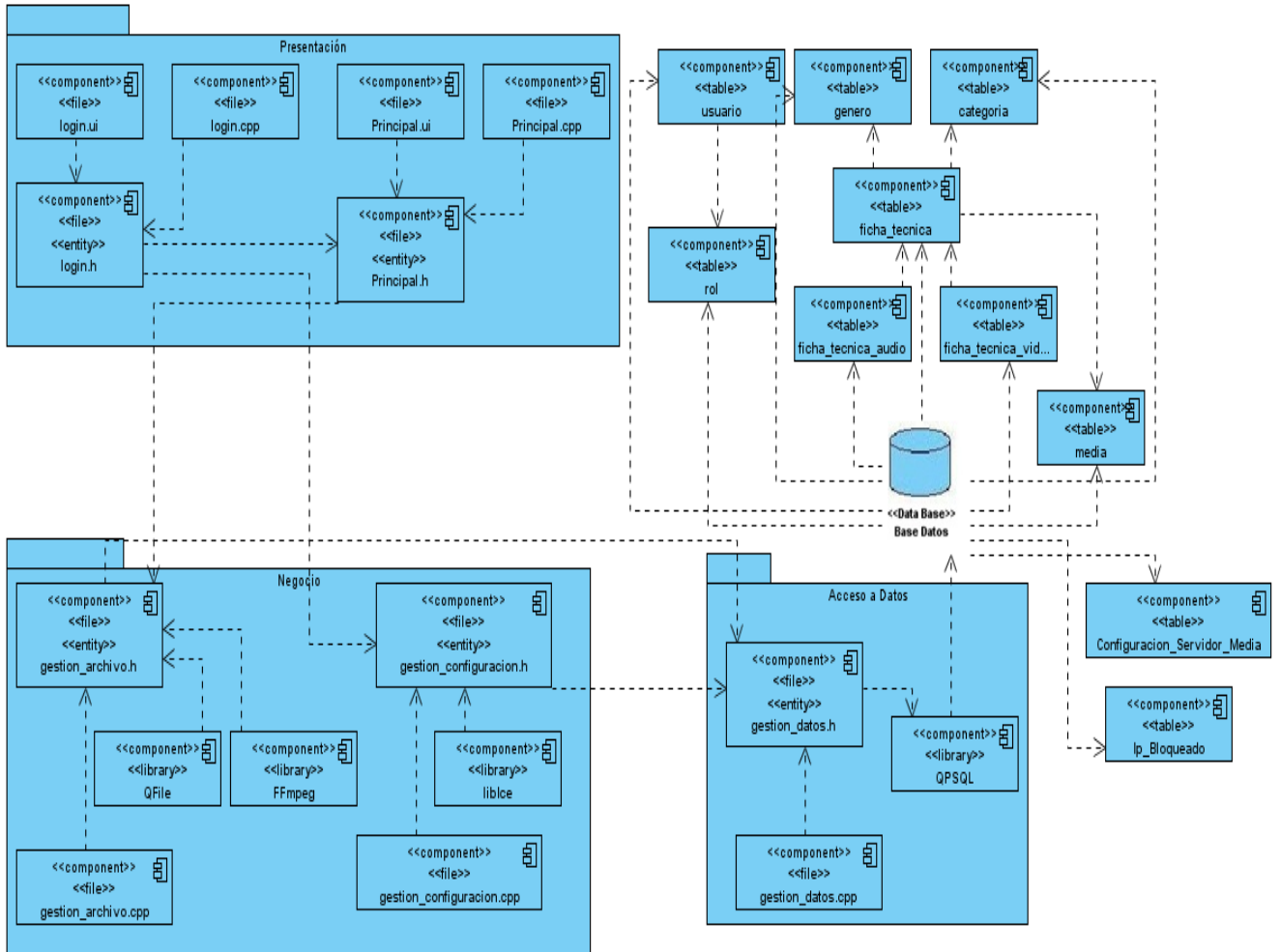


Figura 6: Diagrama de Componentes

3.6 Estándar de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. A continuación se presentan las convenciones establecidas por el centro GEYSED para la programación en lenguaje C++.

3.6.1 Comentarios

Cada programa deberá comenzar con un comentario que incluya:

Capítulo 3: Construcción y validación de la solución propuesta.

- ✚ Autor
- ✚ Fecha
- ✚ Objetivo, o problema que resuelve el programa
- ✚ Fecha de creación y bitácora de versiones con las dos últimas fechas de modificación.
- ✚ Algoritmo.

Cada función debe tener un encabezado que contenga:

- ✚ Objetivo de la función y no descripción del procedimiento.
- ✚ Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- ✚ Explicación de uso de argumentos (parámetros) no obvios.
- ✚ Explicación de uso de valores devueltos (de retorno).

3.6.2 Nombres de identificadores

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- ✚ Deberán tener un nombre significativo para que por su simple lectura, pueda conocerse su función, sin tener que consultar manuales o hacer demasiados comentarios.
- ✚ Para nombres que se usen con frecuencia o para términos largos, se recomienda usar abreviaturas estándar para que éstos tengan una longitud razonable. Si usa abreviaturas deben manejar la misma lógica en todo el programa.
- ✚ Evitar identificadores que comiencen con uno o dos caracteres de subrayado para evitar que se confundan con los que el compilador selecciona.
- ✚ Cada identificador de función, variable o procedimiento deberá ser precedido por la abreviación del tipo de dato de que es la variable, o si se trata de una función o procedimiento del tipo de dato que regresa.

3.6.2.1 Identificadores de variables

Capítulo 3: Construcción y validación de la solución propuesta.

Comenzarán siempre con la primera letra minúscula correspondiente a su tipo de dato.

Para distinguir palabras dentro del nombre se empleará una letra un guión bajo (_), sin mezclar ambas formas en un mismo programa.

Ejemplo: temperatura_de_vapor

3.6.2.2 Identificadores de punteros (apuntadores)

Su nombre deberá comenzar con la letra p.

Ejemplo: p_alumno donde *p_alumno* es un puntero que podrá tener la dirección del lugar donde se almacena información de un alumno.

3.6.2.3 Identificadores de variables dimensionadas (arreglos, matrices)

Su nombre deberá comenzar con las letras ar.

Ejemplo: ar_alumnos

Donde *ar_alumnos* es un arreglo de datos para guardar información de alumnos.

3.6.2.4 Identificadores de datos constantes.

Serán declaradas en letras mayúsculas.

Ejemplo: const IVA=0.15;

3.6.2.5 Identificadores de funciones

Separadas por un guión bajo.

Ejemplo: void vEliminar_media();

3.6.2.6 Identificadores de tipos definidos por el usuario

La primera letra será mayúscula.

Ejemplo: class Clase o struct Estructura

Capítulo 3: Construcción y validación de la solución propuesta.

3.6.3 Organización visual del programa

3.6.3.1 Generales

- ✚ No manejar en los programas más de una instrucción por línea.
- ✚ Declarar las variables en líneas separadas
- ✚ Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.

3.6.3.2 Sangrías

- ✚ Las sangrías tendrán una longitud de tres espacios.
- ✚ Para las llaves que definen el cuerpo de una función, sangre un nivel.

Ejemplo: void vEliminar_media()
 {//Instrucciones de la función}

- ✚ Sangre las instrucciones del cuerpo de cada estructura de control.

Ejemplo: for (int x = 0; x < 5; x++)
 {//Instrucciones a ejecutar}

- ✚ Trate de evitar codificar más de tres niveles de sangrado.

3.6.3.3 Líneas y espacios en blanco

- ✚ Insertar una línea en blanco antes y después de una declaración de datos que aparezca entre instrucciones ejecutables.

```
a=b+c;  
// Línea en blanco  
int f; // Declaración entre instrucciones  
// Línea en blanco  
f = a;
```

- ✚ Las declaraciones de datos dentro de una función, deberán ir al inicio y separadas de las instrucciones ejecutables de la función por medio de una línea en blanco.
- ✚ Deben incluirse espacios en ambos lados de los operadores binarios.

Ejemplo:
 $y = 50 + 15 - x$

Capítulo 3: Construcción y validación de la solución propuesta.

- ✚ Es posible distribuir una instrucción grande sobre varias líneas. Si lo hace, seleccione puntos de ruptura que tengan sentido, como después de una coma en el caso de una lista, o después de un operador en el caso de una expresión larga.
- ✚ Sangre todas las líneas subsecuentes.

Ejemplo:

```
cout << "Ejemplo de ruptura de una instrucción en más de una"  
<< " Línea de comandos";
```

- ✚ Los operadores unarios (++ , -- , etc.) deben ponerse junto a sus operandos, sin espacios intermedios.
- ✚ Antes y después de cada estructura de control deberá poner una línea en blanco.

3.6.3.4 Paréntesis

- ✚ Para hacer más clara una expresión, es aceptable agregarle paréntesis innecesarios. Dichos paréntesis se llaman paréntesis redundantes.

3.7 Validación de la solución propuesta

Uno de los elementos principales para certificar la calidad de una aplicación informática, lo constituye el resultado de las pruebas que se le practican. A un software se le pueden realizar disímiles pruebas debido a la gran variedad de tipologías de pruebas que existen, cada una de ellas con metas y estrategias bien definidas. En el presente epígrafe se detallarán las pruebas funcionales y de fiabilidad cometidas al subsistema de gestión de medias.

3.7.1 Elementos del proceso de pruebas

Los niveles y tipos de pruebas son elementos a tener en cuenta en el proceso de validación de cualquier solución informática. A continuación se detallan características de dichos elementos.

Capítulo 3: Construcción y validación de la solución propuesta.

3.7.1.1 Niveles de pruebas

Prueba de desarrollador: Indica los aspectos de diseño e implementación de las pruebas más adecuadas que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes (27).

Prueba independiente: Indica el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerar esta distinción un súper conjunto, que incluye validación y verificación independientes. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador (27).

Prueba de unidad: Se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. El implementador realiza la prueba de unidad mientras se desarrolla la unidad. Los detalles de la prueba de unidad se describen en la disciplina de implementación (27).

Prueba de integración: Las pruebas de integración se realizan para garantizar que los componentes del modelo de implementación funcionan correctamente cuando se combinan para ejecutar un guión de uso. El destino de la prueba es un paquete o un conjunto de paquetes del modelo de implementación. A menudo, los paquetes que se combinan proceden de diferentes empresas de desarrollo. Las pruebas de integración exponen el estado incompleto o los errores de las especificaciones de la interfaz del paquete (27).

Capítulo 3: Construcción y validación de la solución propuesta.

Prueba del sistema: Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema (27).

Prueba de aceptación: La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó (27).

Después de un análisis de los niveles de prueba por los que puede pasar una aplicación informática, se debe tener conocimiento de cuáles son los tipos de pruebas que se aplican en cada uno de estos niveles.

3.7.1.2 Tipos de prueba (27)

3.7.1.2.1 Funcionalidad

Prueba de función: pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas.

Prueba de seguridad: pruebas que se centran en garantizar que los datos del destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba.

Capítulo 3: Construcción y validación de la solución propuesta.

Prueba de volumen: pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad máxima de datos para cada campo.

3.7.1.2.2 Fiabilidad

Prueba de integridad: pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas.

Prueba de estructura: pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano.

Prueba de tensión: se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación.

Prueba de puntos de referencia: se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema.

Capítulo 3: Construcción y validación de la solución propuesta.

Prueba de contienda: pruebas que se centran en la validación de la capacidad del destino de la prueba para manejar de forma aceptable varias demandas del actor en el mismo recurso (registros de datos, memoria, etc.).

3.7.1.2.3 Rendimiento

Prueba de carga: se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal.

Perfil de rendimiento: se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces.

Prueba de configuración: pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema.

3.7.1.2.4 Capacidad de soporte

Prueba de instalación: pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes (como, por ejemplo, espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones y sistemas. **(27)**

Capítulo 3: Construcción y validación de la solución propuesta.

Después de analizar los niveles y tipos de pruebas a los que se puede someter un producto de software, se decide aplicar pruebas a nivel de unidad (pruebas de caja blanca), pruebas funcionales (pruebas de caja negra) y pruebas de rendimiento (prueba de carga).

Pruebas de caja blanca

Las pruebas de caja blanca ejercitan que se cumplan todos los caminos lógicos del programa.

Los pasos para la realización de esta técnica son los siguientes:

Paso 1. Generar el grafo de flujo de datos a partir del código/diseño procedimental (Nodos, Aristas, Regiones)

Paso 2. Calcular la complejidad ciclomática $V(G)$: métrica del software que da una medición de la complejidad de un programa.

$$V(G) = NA \text{ (Número de Aristas)} - NN \text{ (Número de Nodos)} + 2$$

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = \text{Número de regiones}$$

Paso 3. Definir un conjunto básicos de caminos de ejecución a partir de la complejidad ciclomática.

Paso 4. Generar un caso de prueba para cada camino de ejecución.

Diseños de casos de prueba de caja blanca para las funciones críticas implementadas.

Función 1. Eliminar Archivo

```
void gestion_archivo::vEliminar_archivo(int id_media)
{
    1 p_obj_gestion_datos->vEliminar_media(id_media);
}
```

Paso 1: Generar grafo de flujo de datos.

1

Paso 2: Cálculo de complejidad ciclomática.

$$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$$

Capítulo 3: Construcción y validación de la solución propuesta.

V (G)= 0 - 1 + 2

V (G)= 1

Paso 3: Caminos básicos.

CB1: 1

Paso 4: Caso de prueba para el camino básico.

CB1: 1

Caso de prueba: Eliminar archivo.

Entrada: id_media =5671

Resultado de la prueba: Satisfactoria.

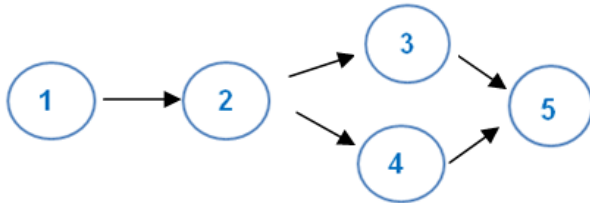
Función 2: Eliminar Media

```
void gestion_datos::vEliminar_media(int id_media)
{
    1  QString consulta1= "SELECT ficha_tecnica_video.id_media FROM
ficha_tecnica_video WHERE ficha_tecnica_video.id_media=
"+QString::number(id_media)+"";
    QSqlQuery result = ejecutar_consulta(consulta1);
    int cant=result.record().count();
    2  if(cant !=0)
    {
        3  QString consulta2="delete from ficha_tecnica_video where
ficha_tecnica_video.id_media= "+QString::number(id_media)+"";
        ejecutar_consulta(consulta2);
    }
    else
    {
        4  QString consulta3="delete from ficha_tecnica_audio where
ficha_tecnica_audio.id_media= "+QString::number(id_media)+"";
        ejecutar_consulta(consulta3);
    }
    5  QString consulta4="delete from ficha_tecnica where ficha_tecnica.id_media=
"+QString::number(id_media)+"";
    ejecutar_consulta(consulta4);
}
```

Capítulo 3: Construcción y validación de la solución propuesta.

```
QString consulta5="delete from media where media.id_media=
"+QString::number(id_media)+"";
ejecutar_consulta(consulta5);
}
```

Paso 1: Generar grafo de flujo de datos.



Paso 2: Cálculo de complejidad ciclomática.

$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$

$V(G) = 5 - 5 + 2$

$V(G) = 2$

Paso 3: Caminos básicos.

CB1: 1 - 2 - 3 - 5

CB2: 1 - 2 - 4 - 5

Paso 4: Caso de prueba para el camino básico.

CB1: 1 - 2 - 3 - 5

Caso de prueba: Eliminar una media de tipo video.

Entrada: id_media = 5671

Resultado de la prueba: Satisfactoria.

✚ Función 3. Ejecutar Consulta.

```
QSqlQuery gestion_datos::ejecutar_consulta(QString consulta)
{
    QSqlQuery q;
    q.prepare(consulta);
    q.exec();
    return q;
}
```

Paso 1: Generar grafo de flujo de datos.

Capítulo 3: Construcción y validación de la solución propuesta.

1

Paso 2: Cálculo de complejidad ciclomática.

$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$

$V(G) = 0 - 1 + 2$

$V(G) = 1$

Paso 3: Caminos básicos.

CB1: 1

Paso 4: Caso de prueba para el camino básico.

CB1: 1

Caso de prueba: Ejecutar consulta.

Entrada: delete from ficha_tecnica_audio where ficha_tecnica_audio.id_media= "+QString::number(id_media)+"";

Resultado de la prueba: Satisfactoria.

✚ Función 4. Buscar Archivo

```
QString gestion_archivo::buscar_archivo(QString nombre_media, QString
formato,int tamanno,bool estado_menor, bool estado_mayor)
{
    1 QString lista= p_obj_gestion_datos->buscar_media
(nombre_media,formato,tamanno,estado_menor,estado_mayor);
    return lista;
}
```

Paso 1: Generar grafo de flujo de datos.

1

Paso 2: Cálculo de complejidad ciclomática.

$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$

$V(G) = 0 - 1 + 2$

$V(G) = 1$

Paso 3: Caminos básicos.

CB1: 1

Paso 4: Caso de prueba para el camino básico.

Capítulo 3: Construcción y validación de la solución propuesta.

CB1: 1

Caso de prueba: Buscar Archivo.

Entrada: E.t, mpeg2, 703kb, false, false

Resultado de la prueba: Satisfactoria.

Función 5. Buscar media.

```
QString gestion_datos::buscar_media(QString nombre_media, QString formato,
double tamanno, bool estado_menor, bool estado_mayor)
```

```
{
```

```
// Buscar por nombre
```

```
1 if(nombre_media!="" && formato=="")
```

```
{
```

```
2 return "SELECT Distinct media.nombre_media, media.id_media, media.tamanno,
categoria.nombre, genero.tipo, media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.nombre_media
='"+nombre_media+"'and media.id_media = ficha_tecnica.id_media and
ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria";
```

```
}
```

```
// Buscar por Nombre y Formato
```

```
3 else if(nombre_media!="" && formato!="")
```

```
{
```

```
4 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.nombre_media
='"+nombre_media+"'and media.id_media = ficha_tecnica.id_media and
ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.id_formato = formato.id_formato and
formato.formato='"+formato+"'";
```

```
}
```

```
// Buscar por Formato
```

```
5 else if(formato!="")
```

```
{
```

Capítulo 3: Construcción y validación de la solución propuesta.

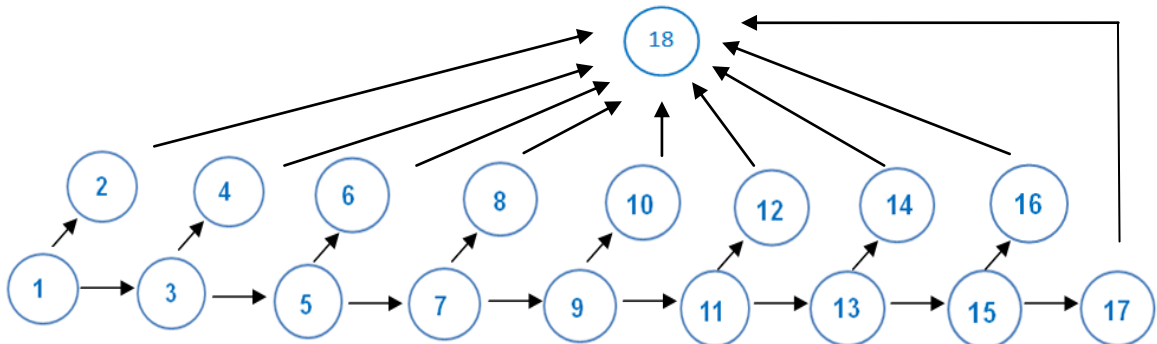
```
6 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.id_media = ficha_tecnica.id_media
and ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.id_formato = formato.id_formato and
formato.formato='"+formato+"'ORDER By(id_media)";
}
// Buscar por selección de mayor tamaño
7 else if(estadoMayor)
{
8 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.id_media = ficha_tecnica.id_media
and ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.tamanno > "+QString::number(tamanno)+"
ORDER By(id_media)";
}
// Buscar por selección de menor tamaño
9 else if(estadoMenor)
{
10 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.id_media = ficha_tecnica.id_media
and ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.tamanno < "+QString::number(tamanno)+"
ORDER By(id_media)";
}
// Buscar por nombre y selección de mayor tamaño
11 else if(nombre_media!="" &&formato=="" && estadoMayor )
{
12 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.nombre_media
='"+nombre_media+"'and media.id_media = ficha_tecnica.id_media and
ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
```


Capítulo 3: Construcción y validación de la solución propuesta.

```
categoria.id_categoria and media.tamanno < "+QString::number(tamanno)+"
ORDER By(id_media)";
}
// Buscar por nombre y selección de menor tamaño
13 else if(nombre_media!="" &&formato=="" && estadoMenor )
{
14 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.nombre_media
="+nombre_media+"and media.id_media = ficha_tecnica.id_media and
ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.tamanno > "+QString::number(tamanno)+"
ORDER By(id_media)";
}
// Buscar por nombre, formato y selección de mayor tamaño
15 else if(nombre_media=="" &&formato!="" && estadoMayor )
{
16 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.id_media = ficha_tecnica.id_media
and ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.id_formato = formato.id_formato and
formato.formato="+formato+" and media.tamanno < "+QString::number(tamanno)+"
ORDER By(id_media)";
}
// Buscar por nombre, formato y selección de menor tamaño
17 return "SELECT Distinct media.nombre_media,media.id_media, media.tamanno,
categoria.nombre, genero.tipo,media.direccion FROM media, categoria,
genero,ficha_tecnica,formato WHERE media.id_media = ficha_tecnica.id_media
and ficha_tecnica.id_genero = genero.id_genero and ficha_tecnica.id_categoria =
categoria.id_categoria and media.id_formato = formato.id_formato and
formato.formato="+formato+" and media.tamanno > "+QString::number(tamanno)+"
ORDER By(id_media)";
18 }
```

Capítulo 3: Construcción y validación de la solución propuesta.

Paso 1: Generar grafo de flujo de datos.



Paso 2: Cálculo de complejidad ciclomática.

$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$

$V(G) = 25 - 18 + 2$

$V(G) = 9$

Paso 3: Caminos básicos.

CB1: 1 - 2 - 18

CB2: 1 - 3 - 4 - 18

CB3: 1 - 3 - 5 - 6 - 18

CB4: 1 - 3 - 5 - 7 - 9 - 11 - 13 - 15 - 17 - 18

CB5: 1 - 3 - 5 - 7 - 8 - 18

CB6: 1 - 3 - 5 - 7 - 9 - 10 - 18

CB7: 1 - 3 - 5 - 7 - 9 - 11 - 12 - 18

CB8: 1 - 3 - 5 - 7 - 9 - 11 - 13 - 14 - 18

CB9: 1 - 3 - 5 - 7 - 9 - 11 - 13 - 15 - 16 - 18

Paso 4: Caso de prueba para el camino básico.

CB1: 1-2-18

Caso de prueba: Buscar Media.

Entrada: E.t

Salida: E.t - 15- 703 - Película - Infantil.

Resultado de la prueba: Satisfactoria.

Capítulo 3: Construcción y validación de la solución propuesta.

Resumen Pruebas de Caja Blanca

No	Funcionalidad	Complejidad ciclomática	Camino Básico	Resultado
1	Eliminar Archivo.	1	1	Satisfactorio
2	Eliminar Media	2	1 - 2 - 3 - 5	Satisfactorio
3	Ejecutar Consulta	1	1	Satisfactorio
4	Buscar Archivo.	1	1	Satisfactorio
5	Buscar media.	1	1 - 2- 18	Satisfactorio

3.7.2 Pruebas funcionales.

Se denominan pruebas funcionales a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre ésta el paso siguiente es el pase a producción **(28)**.

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas **(28)**.

Las pruebas de caja negra se centran en los requisitos funcionales de software, permitiendo obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Este tipo de prueba intenta encontrar errores de las siguientes categorías:

- ✚ Función incorrecta o ausente
- ✚ Errores de interfaz
- ✚ Errores en estructura de datos

Capítulo 3: Construcción y validación de la solución propuesta.

- ✚ Errores de rendimiento
- ✚ Errores de inicialización y terminación.

3.7.2.1 Diseño de pruebas funcionales

Con el objetivo de validar el correcto funcionamiento del Subsistema de Gestión de medias se realizaron pruebas de caja negra a los casos de uso Gestionar media y Autenticar usuario.

A continuación se relacionan los casos de pruebas para los casos de uso seleccionados:

Secciones a probar en el caso de uso Gestionar Media

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Buscar media	EC 1.1: Buscar media satisfactoriamente	El sistema verifica la validez de los datos, que los campos requeridos no se encuentren vacíos y de acuerdo al criterio de búsqueda muestra los datos asociados a la media.
	EC 1.2: Buscar media falla.	El sistema detecta ausencia de datos requeridos, o incorrectos en el formulario y no se muestra ninguna media. Muestra un mensaje informando el error que se ha producido.
SC 2: Eliminar media	EC 2.1: Eliminar media satisfactoriamente.	El sistema verifica que la media no esté en ninguna planificación venidera y de ser así permite la eliminación de la media en todas las tablas asociadas en la base de datos, como el archivo físico en el servidor de medias.
	EC 2.2: Eliminar media falla.	Si la media está programada muestra un mensaje de información de que no puede ser eliminada.

Capítulo 3: Construcción y validación de la solución propuesta.

SC 3: Insertar media	EC 2.1: Insertar media satisfactoriamente.	El sistema verifica la validez de los datos asociados a la media, que los campos requeridos no se encuentren vacíos. Realiza la copia del fichero hacia el servidor y posteriormente se inserta en la base de datos.
	EC 2.2: Insertar media falla.	El sistema verifica los datos asociados a la media y encuentra errores en los datos de entrada y muestra un mensaje de alerta.

Tabla 1. Secciones a probar en el caso de uso Gestionar media.

Descripción de variables para el caso de uso Insertar media

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto (alfanumérico)	No	El usuario introducirá en el campo el nombre de la media.
2	Fuente	Campo de texto (alfanumérico)	No	El usuario introducirá en el campo la fuente de donde proviene la media.
3	Código	Campo de texto (alfanumérico)	No	El usuario introducirá en el campo el código de la fuente.
4	Tipología	Campo de texto	No	El usuario selecciona la tipología
5	Género	Campo de texto	No	El usuario selecciona el género
6	Álbum	Campo de texto (alfanumérico)	No	El usuario introducirá en el campo el álbum de la media de tipo audio.
7	Casa disquera	Campo de texto (alfanumérico)	No	El usuario introducirá en el campo la casa disquera de la media de tipo audio.
8	País	Campo de texto (alfanumérico)	Sí	El usuario introducirá en el campo el país de la media.
9	Título en Español	Campo de texto (alfanumérico)	Sí	El usuario introducirá en el campo el título en español de la media.

Capítulo 3: Construcción y validación de la solución propuesta.

10	Título Inglés	Campo de texto (alfanumérico)	Sí	El usuario introducirá en el campo el título en inglés de la media.
11	Fecha de producción	Fecha	Sí	El usuario introduce la fecha de producción.
12	Nuevo	Campo de texto	Sí	El usuario introduce un nuevo tipo de género.

Tabla 2. Descripción de variables para el caso de uso Insertar media.

Matriz de datos para la sección Insertar nueva media

No	Escenario insertar media satisfactoriamente	Descripción	Respuesta del sistema	Resultado de la Prueba
1	Nombre	3_metros	El sistema valida que los datos introducidos por el usuario sean correctos, inserta los mismos en la base de datos y muestra un mensaje indicando que la acción fue ejecutada satisfactoriamente.	Satisfactorio.
2	Fuente	USB		
3	Código	123_abc		
4	Tipología	Película		
5	Género	Drama		
6	Álbum	-		
7	Casa disquera	-		
8	País	España		
9	Título en Español	3_metros		
10	Título Inglés	-		

Capítulo 3: Construcción y validación de la solución propuesta.

	Escenario insertar media falla	Descripción	Respuesta del sistema	Resultado de la Prueba
11	Fecha de producción	-		
1	Nombre	-	<p>El sistema valida los datos introducidos por el usuario y:</p> <p>Si los datos están incorrectos el sistema muestra un mensaje indicando el error ocurrido.</p> <p>Si existen campos obligatorios vacíos el sistema muestra un mensaje indicando que no pueden existir campos vacíos.</p>	Satisfactorio.
2	Fuente	USB		
3	Código	123_abc		
4	Tipología	Película		
5	Género	Drama		
6	Álbum	-		
7	Casa disquera	-		
8	País	España		
9	Título en Español	3_metros		
10	Título Inglés	-		
11	Fecha de producción	-		

Tabla 3. Matriz de datos para la sección Insertar media.

Capítulo 3: Construcción y validación de la solución propuesta.

Secciones a probar en el caso de uso Autenticar usuario

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Autenticar usuario	EC 1.1: Autenticar usuario satisfactoriamente	El usuario escribe su nombre de usuario y contraseña y presiona el botón "Aceptar" para entrar en el sistema. El sistema verifica la validez del usuario y la contraseña introducida dando acceso a la interfaz principal.
	EC 1.2: Autenticar usuario falla.	Los datos introducidos por el usuario no son válidos, el sistema muestra el siguiente mensaje: "Datos incorrectos". El sistema da la posibilidad de volver a introducir los datos.

Tabla 4. Secciones a probar en el caso de uso Autenticar usuario.

Descripción de variables para el caso de uso Autenticar usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	usuario	Campo de texto	No	El usuario introduce en este campo su usuario para autenticarse.
2	contraseña	Campo de texto	No	El usuario introduce en este campo su contraseña para autenticarse en el sistema.

Tabla 5. Descripción de variables para el caso de uso Autenticar usuario.

Matriz de datos para la sección Autenticar usuario

Escenario	usuario	contraseña	Respuesta del sistema	Resultado de la prueba
Autenticar usuario satisfactoriamente.	Eduardo	gestión	El sistema verifica que el usuario y la contraseña existan en la Base de Datos	Satisfactorio.

Capítulo 3: Construcción y validación de la solución propuesta.

			y se correspondan. Verifica los permisos que tiene asignado ese usuario y dependiendo de ellos le da acceso a la aplicación.	
Autenticar usuario falla.	Pepe	gestión	Si los datos de autenticación introducidos por el usuario no son válidos, el sistema muestra el siguiente mensaje: "Datos incorrectos". El sistema da la posibilidad de volver a introducir los datos.	Satisfactorio.

Tabla 6. Matriz de datos para la sección Autenticar usuario.

Las pruebas funcionales que se le aplicaron al subsistema gestión de medias de la plataforma PTARTV dieron al traste con resultados que permitieron saber hasta qué punto la aplicación satisface los requisitos funcionales establecidos en un comienzo.

3.7.3 Pruebas de rendimiento

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar que tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación (29).

Capítulo 3: Construcción y validación de la solución propuesta.

3.7.3.1 Diseño de pruebas de rendimiento

Con el objetivo de evaluar el desempeño del subsistema gestión de medias de la plataforma PTARTV se ejecutó el proceso de subida de medias al servidor de archivos desde varias estaciones de trabajo. Las características del servidor de archivos utilizado para la prueba son las siguientes:

- ✓ Sistema Operativo:
 - Ubuntu Release 10.4 (lucid).
 - Kernel Linux 2.6.32-21 generic.
 - GNOME 2.30.0.
- ✓ Hardware:
 - Memoria: 0.99 GB.
 - Procesador 0: Intel(R) Core(TM)2 Duo CPU T5800 @ 2.20GHz.
 - Procesador 1: Intel(R) Core(TM)2 Duo CPU T5800 @ 2.20GHz.
- ✓ Estado del sistema:
 - Espacio libre en disco: 22.4 GB.

Niveles de consumo de memoria y de la CPU en el servidor

Cantidad de PCs subiendo medias.	Porcentaje consumido de la CPU.	Memoria consumida (MB).
0	2%	200MB
1	15%	220MB
3	35%	240MB
5	45%	320MB

Tabla 7. Niveles de consumo de memoria y de la CPU

Flujo desde las estaciones: 2.0 - 2.5 Mbps

Flujo de recepción de datos del servidor: 11.5 Mbps

Capítulo 3: Construcción y validación de la solución propuesta.

3.7.4 Resultado de las pruebas

A continuación se especifican las no conformidades detectadas durante en proceso de pruebas de la aplicación.

A través de las pruebas de unidad pudieron obtenerse resultados satisfactorios lo que corroboró el buen funcionamiento de los elementos que conforman el software. En el transcurso de la fase de pruebas, específicamente funcionales, se pudieron detectar situaciones que afectaban el desenvolvimiento de la aplicación. A continuación éstos se relacionan:

- ✚ El sistema permitía dejar campos vacíos al insertar una nueva media.
- ✚ De no existir conexión con la base de datos el sistema no lanzaba ninguna alerta.
- ✚ El sistema insertaba la media en la base de datos, pero no la copiaba en el servidor de archivos.

Las deficiencias encontradas fueron solucionadas rápidamente.

Luego de realizar las pruebas de rendimiento se llegó a la conclusión de que el procesamiento correspondiente a la aplicación se sustenta sobre la capacidad operativa del microprocesador, definiendo las características técnicas mínimas como:

- ✓ Hardware:
 - Memoria: 1.99 MB.
 - Procesador 0: Intel(R) Core(TM)2 Duo CPU T5800 @ 2.20GHz.
 - Procesador 1: Intel(R) Core(TM)2 Duo CPU T5800 @ 2.20GHz.
 - Capacidad en disco duro: 2 TB.

Para un mejor rendimiento y aumento de las conexiones simultáneas al servidor de archivos:

- ✓ Hardware:
 - Memoria: 3.99 MB.
 - Procesador 0: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
 - Procesador 1: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
 - Procesador 2: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
 - Procesador 3: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
 - Capacidad en disco duro: Uso de una SAN.

Capítulo 3: Construcción y validación de la solución propuesta.

3.8 Conclusiones

La industria del software cada vez es más competitiva y sus usuarios más exigentes en cuanto a la calidad de los productos. El conocer que tipos de pruebas pueden aplicársele a un sistema informático permitió seleccionar las que se le aplicarían a la solución propuesta.

A través de las pruebas seleccionadas se pudo validar el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del producto, constatando el correcto funcionamiento del subsistema gestión de medias de la plataforma PTARTV.

Los resultados obtenidos durante la fase de pruebas del subsistema representaron un pilar fundamental para corregir las deficiencias detectadas y así dar por concluida la construcción de una nueva versión de los procedimientos y técnicas de almacenamiento y gestión de archivos de la plataforma PTARTV.

Conclusiones generales

Durante el desarrollo del trabajo de diploma y el tiempo dedicado a la investigación, implementación y prueba de la solución propuesta se adquirieron conocimientos necesarios para el desarrollo del sistema y fundamentales para la formación profesional del autor.

Luego de la realización de las actividades anteriores se logró:

- ✚ Facilitar el proceso de gestión de las medias que llegan a la plataforma, así como centralizar el acceso a las mismas por los demás subsistemas.
- ✚ Controlar el acceso al servidor de archivos de la PTARTV.
- ✚ Que el trabajo realizado tribute a la plataforma PTARTV como experiencia en el proceso de gestión de archivos.
- ✚ Que el presente trabajo constituya un avance en la gestión de archivos para la PTARTV.

Se confirma haber dado cumplimiento al objetivo general trazado al inicio del trabajo. La afirmación anterior se sustenta en el logro del desarrollo de un sistema automatizado para la gestión de contenido multimedia en la PTARTV, lo cual garantiza mayor rapidez y mejor funcionamiento en los procesos relacionados con el recurso media.

Recomendaciones

Al finalizar el presente trabajo se recomienda:

- ✚ Dar continuidad al desarrollo y soporte del subsistema desarrollado.
- ✚ Agregar funcionalidades que permitan mejorar la gestión y configuración tanto de las medias como del servidor de archivos, como es el caso de un módulo de salvas automáticas y de salvas temporales a las medias eliminadas.

Bibliografía

Bibliografía citada

1. buscon.rae. *buscon.rae*. [En línea] [Citado el: 16 de diciembre de 2010.] http://buscon.rae.es/drael/SrvltConsulta?TIPO_BUS=3&LEMA=archivo.
2. Oracle Corporation. [En línea] [Citado el: 16 de diciembre de 2010.] <http://docs.sun.com/app/docs/doc/816-3938/6ma6eh79q?l=es&a=view>.
3. **Eduardo Aragón Montes, Julián Camilo Pérez Parra.** *SISTEMA PARA EL ALMACENAMIENTO VIRTUAL Y GESTION DE ARCHIVOS*. Cali, Colombia: s.n., 2010.
4. **Martínez, David Luis la Red.** *exa. exa*. [En línea] [Citado el: 16 de diciembre de 2010.] <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/GESTAR02.htm>.
5. Microsoft Corporation. *Microsoft Corporation*. [En línea] [Citado el: 2010 de noviembre de 18.] <http://msdn.microsoft.com/en-us/library/aa287558.aspx>.
6. Ipswitch. *Ipswitch*. [En línea] [Citado el: 24 de noviembre de 2010.] <http://www.ipswitchft.com/Business/Products/WsFtpServer/Index.aspx?is>.
7. readsoft. *readsoft*. [En línea] [Citado el: 16 de diciembre de 2010.] http://www.readsoft.cl/digitalizacion-de-facturas_terminologia/digitalizacion-de-documentos/almacenamiento-digital.aspx.
8. SMDATA. *SNDATA*. [En línea] [Citado el: 14 de diciembre de 2010.] www.smdata.com/ventajasSAN.htm.
9. CLETU. *CLETU*. [En línea] [Citado el: 14 de diciembre de 2010.] http://www.cletu.com/disenio-web/gestor_de_contenidos/ftp-imprentas-ingenierias_176_35.
10. Valverde, Jorge Carlos. *SCRIBD. SCRIBD*. [En línea] [Citado el: 13 de enero de 2011.] <http://www.scribd.com/doc/19795778/sockets-y-su-Programacion-en-Java>.
11. Inicia Tu Web. *Inicia Tu Web*. [En línea] [Citado el: 13 de enero de 2011.] <http://www.iniciatuweb.com/ssl-secure-sockets-layer-tls-transport-layer-security/>.
12. Ibar Jacobson, Grady Booch, James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson Educación.S.A, 2000. 84-7829-036-2.
13. Rhernando. *Rhernando*. [En línea] [Citado el: 2011 de Febrero de 2011.] http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.

14. PostgreSQL.org. postgresql-es.org. [En línea] [Citado el: 15 de febrero de 2011.] http://www.postgresql-es.org/sobre_postgresql.
16. Schuller, Joseph. *Aprendiendo UML en 24 Horas*. México: PEARSON EDUCACION, 2000. 968-444-463-X.
17. Melodysoft . *Melodysoft*. [En línea] [Citado el: 2011 de Febrero de 2011.] <http://boards5.melodysoft.com/2007AISC0107/los-lenguajes-de-programacion-15.html>.
18. Slideshare. *Slideshare*. [En línea] [Citado el: 15 de febrero de 2011.] <http://www.slideshare.net/mandre55/conceptos-basicos-programacion>.
19. La Revista Informática. *La Revista Informática*. [En línea] [Citado el: 15 de Febrero de 2011.] <http://www.larevistainformatica.com/C++.htm>.
20. **Pérez, Reynier Martínez**. *Subsistema de Almacenamiento y Gestión de Archivos para el producto Captura y Catalogación de Medias*. Ciudad de La Habana : Universidad de Las Ciencias Informáticas, 2010.
21. Nokia. *Nokia*. [En línea] [Citado el: 16 de Febrero de 2011.] <http://qt.nokia.com/products/developer-tools> .
22. Visual Paradigm International. *Visual Paradigm International*. [En línea] [Citado el: 16 de Febrero de 2011.] <http://www.visual-paradigm.com/product/vpuml/communityedition.jsp>.
23. Error500. *Error500*. [En línea] [Citado el: 17 de Febrero de 2011.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php .
24. web.mit. *web.mit*. [En línea] [Citado el: 28 de Febrero de 2011.] <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rq-es-4/ch-ssh.html>.
25. Publicaciones. *Publicaciones*. [En línea] [Citado el: 01 de Marzo de 2011.] <http://publicaciones.ua.es/publica/Detalles.aspx?fndCod=LI9788479088880&idet=958>.
26. **Alonso Guerrero, Zorlin y Leyva Gonzáles, Genry**. *Sistema de Captura y Transcripción de Audio*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2009.
27. **Torres, Patricio Letelier**. *Desarrollo de Software Orientado a Objeto usando UML*. Universidad Politécnica de Valencia.: Departamento Sistemas Informáticos y Computación.
28. **García Pañeda, Xabiel, Marín Prendes, Ignacio y Melendi Palacios, David**. *Universidad de Oviedo. Área de Ingeniería Telemática, Departamento de Informática*. [En línea] [Citado el: 28 de mayo de 2011.] http://www.it.uniovi.es/old/material/cursos/streaming_EU_122004/Intro.pdf.

29. Corporación Sybven-Integración Tecnológica. [En línea] 2011. [Citado el: 29 de mayo de 2011.]

http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246.

30. **Fernández, Raúl Castro.** Desarrollo de software de videovigilancia para sistemas embarcados distribuidos con ICE.

31. ecured. *ecured*. [En línea] [Citado el: 21 de junio de 2011.]

http://www.ecured.cu/index.php/Proceso_Unificado_de_Desarrollo#Flujo_de_Trabajo_de_RUP.

Glosario de términos

Ficheros: Directorios. Agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier otro criterio.

Aplicación: Es una clase de programa informático creado para facilitar al usuario un determinado tipo de trabajo. Esto lo caracteriza frente a otros programas como los sistemas operativos, las utilidades y los lenguajes de programación.

Multimedia: Se aplica a cualquier objeto que utilice simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes y video para informar o entretener al usuario.

Ciente: Es una aplicación informática cuya función es acceder a los servicios que ofrece un servidor, haciendo uso generalmente de una red de telecomunicaciones.

Ficheros: Directorios. Agrupación de archivos de datos, atendiendo a su contenido, a su propósito o a cualquier otro criterio.

Protocolo: En informática es un método estándar que permite la comunicación entre procesos. Comprende un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red.

Multiplataforma: Se utiliza el término para denominar a los programas, lenguajes de programación u otra clase de software que pueden brindar sus prestaciones funcionando sobre diversas combinaciones de hardware y software.

Permisos: Tareas o acciones que puede realizar un usuario dentro de un sistema en dependencia de su nivel de acceso.

SQL: *Structured Query Language* (lenguaje de consulta estructurado), es un lenguaje computacional de Bases de Datos diseñado para la recuperación y gestión en sistemas de bases de datos relacionales.

Glosario de Términos

Servidor: Software u ordenador que provee servicios a otros programas o equipos denominados clientes.

Transmisión Abierta: Señal de libre acceso por la cual no es necesario pagar para consumirla.