



Universidad de las Ciencias Informáticas

Ciudad de La Habana

Facultad 6

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

*Título: Herramienta para la gestión y acceso a información en
Servidores de Streaming.*

Autor: Anaivis Alemán Misas.

Tutor: Yurisnel Corrales Valdés.

Co-Tutor: Geovanys Bermúdez Valdés.

27/6/2011

“Año 53 de la Revolución”



"La única lucha que se pierde es la que se abandona."

Ché

DEDICATORIA

Dedico este trabajo especialmente a toda mi familia.

A mi mamá Maritza que es la persona que más amo en la vida y mi mejor amiga.

A mi papá que ha sido mi ejemplo a seguir.

A mi segunda mamá, mi abuelita Ibis que siempre se preocupa por mí y mis estudios y con ella una dedicatoria bien especial a mi abuelito Pepe Camerro que sé que desde el cielo está orgulloso de mí.

A mi tía Maribel que siempre ha estado ahí para mí, aconsejándome en todo.

A mi abuelo Pedro y mi abuela Miriam que estando viva siempre luchó con todos sus nietos para que estudiaran como mismo hizo con sus hijas.

A mi novio Ernesto Luis que ha cambiado mi vida desde que lo conocí.

A mis hermanos Christian, Aida maria y Enmanuel esperando que sea un ejemplo para ellos, y mi hermanita Lisandra que siempre me ha acompañado desde que llegué a esta escuela.

A mi padrasto Jorge que ha convivido conmigo desde que soy una niña.

A mis primos Maribel, José Manuel, Ana María, Amanda, Ernesto, Carlos, y todos los demás que son unos cuantos.

En general les dedico esta tesis a todas las personas que son importantes en mi vida y que de una forma u otra siempre han estado ahí para mí.

AGRADECIMIENTOS

A mi mamita linda por ser la mejor mamá del mundo, por sacrificarse por mí todos estos años brindándome siempre su apoyo incondicional y porque a ella le debo todo lo que soy.

A mi papá que siempre ha confiado en mí, gracias por aconsejarme, por todo tu amor y dedicación.

A mi abuelita Ibis gracias por ser la luz de mi vida y por quererme tanto.

A mi tía Maribel y a Doris que siempre me han acompañado contando con ellas para todo y por ser mis doctorcitas cuando algo me aqueja.

A mi novio Ernesto Luis por dedicarme su tiempo para ayudarme en los estudios brindándome siempre su apoyo. Gracias por ser mi Oliverio y por darme siempre lo mejor de ti. Te amo.

A mi hermanito kiki por quererme tanto y a mi padrastro Jorge.

A mi hermana Lisandra que no tengo palabras para agradecerle todo lo que ha hecho por mí acompañándome siempre en las buenas y en las malas desde que entré a esta universidad.

A mis amigas Rocío, Alianis, Mailín, Yasnarys, Yanelis, Gelsys, Leydis y todas las que han convivido conmigo estos cinco años compartiendo grandes momentos que nunca olvidaré.

A mis primas por ser mis hermanas también, escribirme y preocuparse por mí.

A mi abuelo Pedro por preocuparse por mí y por ser su pelota.

A mi tío Luisi por cargar conmigo cuando me hace falta transporte.

A mis suegritos bellos que siempre están preocupándose por mí y por recibirme en su casa con mucho amor.

A mi primo Jose Manuel que siempre me está criticando para que sea una mejor persona, te quiero mucho!

A mi tía Mayra y tío Tony por acogerme en su casa y darme todo su cariño.

A mi co-tutor Geovanys por ayudarme en el desarrollo de mi tesis y por aguantar las preguntas cada cinco minutos.

A mi tutor Yurisnel por apoyarme y por ser mi amigo.

A los amigos de Ernesto en especial a Osmar gracias por ser mis amigos también.

A mis compañeros de proyecto que siempre los estoy molestando cuando me hace falta algo.

A todos mis compañeros de aula por ser los mejores y brindarme todo su cariño.

En general a todos los que de una forma u otra contribuyeron a la realización de mi sueño.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 27 días del mes de junio del año 2011.

"[Insertar nombre(s) de autor(es)]"

Anaivis Alemán Misas

"[Insertar nombre(s) de tutor(es)]"

Yurisnel Corrales Valdés

RESUMEN

En La Universidad de Ciencias Informáticas (UCI) existen actualmente varios servidores de streaming los cuales soportan gran contenido audiovisual. En la actualidad no existe un sistema que sea capaz de agilizar la creciente demanda de solicitudes a los objetos que se almacenan en ellos, provocando ineficiencias en el control y búsqueda de información en estos servidores.

Por lo que el presente trabajo de diploma surge con el objetivo de desarrollar una herramienta que permita la gestión y acceso a información en Servidores de Streaming. Todo el proceso de desarrollo fue guiado por la metodología tradicional RUP. Incluye un estudio de algunas de las tecnologías, herramientas y lenguajes de programación más usados en la actualidad para el desarrollo de aplicaciones de escritorio. Se efectuó la presentación de la solución propuesta donde se definieron los requisitos funcionales y no funcionales del sistema para formar posteriormente los casos de uso. La creación del sistema final permitirá agilizar el proceso de búsqueda de medias en servidores de streaming lo cual constituye la esencia de este trabajo.

PALABRAS CLAVES

Servidor, Streaming, acceso, solicitud, búsqueda, medias.

ÍNDICE

Contenido

1. INTRODUCCIÓN	1
1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1. INTRODUCCIÓN	4
1.2. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	4
1.2.1. La tecnología streaming.	4
1.2.2. Motores de Búsqueda.	8
1.3. MÉTODOS DE BÚSQUEDAS EN BASE DATOS.	10
1.4. ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES	12
1.5. CONCLUSIONES	19
2. CAPÍTULO 2: FUNDAMENTACIÓN DE LA TECNOLOGÍA A UTILIZAR.	21
2.1. INTRODUCCIÓN	21
2.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	21
2.2.1. El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.....	21
2.2.2. Microsoft Solution Framework (MSF)	23
2.2.3. Ingeniería de Software Basada en Componentes (ISBC).....	25
2.3. HERRAMIENTAS CASE.....	27
2.3.1. Visual Paradigm	28
2.3.2. Rational Rose Enterprise.....	29
2.4. LENGUAJE DE PROGRAMACIÓN.....	32
2.4.1. Lenguaje de Programación en C++	32
2.5. FRAMEWORK DE DESARROLLO.	33
2.5.1. Qt Creator	33
2.6. SISTEMA GESTOR DE BASE DATOS.	35
2.6.1. PostgreSQL.....	36
2.7. CONCLUSIONES	37

ÍNDICE

3. CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.	39
3.1. INTRODUCCIÓN.	39
3.2. MODELO DE DOMINIO.	39
3.2.1. Glosario de Términos del Dominio.....	40
3.3. LEVANTAMIENTO DE REQUISITOS.....	41
3.3.1. Requisitos Funcionales del sistema.....	41
3.3.2. Requisitos no Funcionales del Sistema.	42
3.4. ACTORES DEL SISTEMA.	43
3.5. DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.	44
3.6. CONCLUSIONES	55
4. CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA.	56
4.1. INTRODUCCIÓN.	56
4.2. PATRONES DE DISEÑO.	56
4.3. MODELO DE DISEÑO.	58
4.4. DISEÑO DE LA BASE DATOS.	59
4.4.1. Diagrama de clases persistentes.....	59
4.4.2. Diagrama entidad relación.....	60
4.5. DIAGRAMA DE SECUENCIA.....	60
4.6. MODELO DE DESPLIEGUE.....	61
4.6.1. Descripción de los Nodos.....	62
4.7. MODELO DE IMPLEMENTACIÓN.	62
4.8. PRUEBAS DE FUNCIONALIDAD.	63
4.8.1. Prueba de Caja Negra.....	64
4.8.2. Lista de chequeo tolerancia a fallo.	67
5. BIBLIOGRAFÍA	72

INTRODUCCIÓN

1. INTRODUCCIÓN

Desde hace varias décadas el impacto de la informática en la vida del hombre es cada vez más evidente. Para todos los países es una necesidad imperiosa adoptar las Tecnologías de la Informática y las Comunicaciones (TIC) para alcanzar sus perspectivas de desarrollo. La producción y consumo de aplicaciones informáticas está marcando pautas en las relaciones económicas a nivel internacional, uno de los recursos más utilizados actualmente en el mundo son las aplicaciones informáticas como la multimedia, pues es el medio que facilita la información digitalizada que combina texto, gráficos, video y audio. Multimedia es un término que se aplica a cualquier objeto, combinando diferentes formas de contenido informativo como texto, sonido, imágenes, animación y video.

En Cuba, en la Universidad de Ciencias Informáticas (UCI) existen diferentes centros de desarrollo, los cuales se dedican a investigar y a producir conocimiento, además de formar el personal necesario para brindar productos y servicios informáticos de alta calidad, uno de estos centros es el Centro de Geoinformática y Señales Digitales (GeySed). El departamento de Señales Digitales se especializa en los servicios informáticos en el área de video y sonido digital, donde su objetivo principal se enfoca en el desarrollo de aplicaciones audiovisuales, todo sobre la base de informatización de la sociedad cubana como primera prioridad y luego la comercialización como parte de la experiencia adquirida. Para cada uno de los proyectos del departamento del Señales Digitales el uso de la multimedia es primordial, y el medio más utilizado para la reproducción de las mismas en la red es con la ayuda del streaming, el cual permite reproducir la media a medida que va siendo transmitida.

La tecnología streaming se utiliza para aligerar la descarga y ejecución de audio y vídeo en la web, ya que permite escuchar y visualizar los archivos mientras se están descargando, permitiendo que esta tarea se ejecute de la manera más rápida y eficiente posible y que se pueda ver el contenido del archivo, sin la necesidad de descargar previamente el archivo en el ordenador. **(Alvarez, 2001)**

El Streaming de audio y vídeo apoya una amplia gama de aplicaciones multimedia de información, incluyendo bibliotecas digitales, telemedicina, educación a distancia, y las comunicaciones militares. Estos sistemas de información permiten el acceso en tiempo real y compartir los documentos almacenados en los servidores distribuidos. La implementación de servidores proxy como parte de la infraestructura de red, ayuda a satisfacer la creciente demanda de acceso a los documentos multimedia, ya que por lo general reducen la carga del servidor y el tiempo necesario para responder las peticiones del documento. Estos pueden almacenar temporalmente la solicitud de los documentos para que los usuarios puedan acceder a

INTRODUCCIÓN

estos archivos sin conexión en repetidas ocasiones a los servidores de ellos y agilizar el proceso de búsqueda. Apoyan además la interacción de las aplicaciones multimedia de streaming, permitiendo a los usuarios escuchar un fragmento de puntos antes de ver todo el contenido del video. **(Boldrito, 2004)**

Actualmente en la universidad existen varios servidores de streaming los cuales soportan grandes cantidades de contenido audiovisual que es utilizado tanto para la formación de los estudiantes como para la recreación y entretenimiento de la comunidad universitaria. Dichos servidores tienen dificultades al manejar la creciente demanda de solicitudes a los objetos que se almacenan en ellos, provocando ineficiencias en el control y búsqueda de información de los servidores de streaming de ahí que se identifique el **Problema a resolver** ¿Cómo gestionar el acceso a la información de los servidores de streaming?

En vista a resolver el problema descrito se hace necesario, Desarrollar una herramienta que permita centralizar la información multimedia de los servidores de streaming lo cual figura como **Objetivo general** de esta investigación.

La investigación se enmarca en el **Objeto de estudio**: Proceso de desarrollo de servidores de streaming y el **Campo de acción**: Gestión y búsqueda de información multimedia en servidores de streaming.

Como **idea a defender** se tiene que con el desarrollo de una herramienta que permita centralizar la información multimedia de los servidores de streaming se podrá agilizar la búsqueda de información multimedia en redes con muchos servidores de streaming.

Para lograr los objetivos trazados se acometen las siguientes tareas:

- Identificar los logros y limitaciones en los enfoques existentes sobre el desarrollo de Servidores de Streaming.
- Caracterizar los métodos de acceso a datos para la búsqueda de información.
- Valorar las herramientas a utilizar en la construcción de la solución.
- Seleccionar y argumentar la Metodología de Desarrollo de software a usar en el proceso.
- Identificar las funcionalidades que debe brindar el sistema.
- Diseñar el sistema.

INTRODUCCIÓN

- Implementar el sistema.
- Realizar las pruebas al sistema.

Para la realización de la investigación se hizo necesario aplicar algunos métodos científicos dentro de los que incluyen métodos teóricos y empíricos, que tienen su sustento en la concepción materialista dialéctica y facilitan la recopilación de la información necesaria para la creación de una herramienta que permita centralizar la información multimedia de los servidores de streaming. De los teóricos se utilizaron el Análisis histórico – Lógico, Analítico – Sintético y la Modelación

- El método histórico – Lógico permitió determinar las características esenciales de las tecnologías de servidores de streaming existentes y realizar el estudio de la trayectoria real de determinados elementos que servirán de guía para la construcción de una herramienta que permita centralizar la información multimedia de los servidores de streaming.
- El método Analítico – Sintético permitió realizar un análisis de la documentación de cada una de las herramientas que se utilizaron en el sistema a diseñar.
- El método Modelación se empleó para modelar el sistema de la herramienta que permita centralizar la información multimedia de los servidores de streaming.

De los métodos empíricos se utilizó:

- Experimento permitiendo realizar pruebas al sistema una vez que fue desarrollado.

Como posibles resultados de la investigación se espera obtener un prototipo funcional de búsqueda de información en Servidores de Streaming que sirva como herramienta para agilizar el proceso de búsqueda de medias en dichos servidores y como reservorio para futuros desarrollos en la entidad.

1. CAPÍTULO 1: Fundamentación Teórica.

1.1. Introducción

Este capítulo abarca temas de la investigación relacionados con algunos conceptos básicos y elementos teóricos a tener en cuenta para el desarrollo del sistema que se necesita. Se realizará un estudio de algunos sistemas similares que gestionan el proceso de búsqueda de información para lograr una idea de cómo resolver la situación problemática que se presenta en este trabajo de diploma, teniendo en cuenta las características de los servidores de streaming.

1.2. Conceptos asociados al dominio del problema

1.2.1. La tecnología streaming.

El Internet ha permitido desde hace tiempo bajar y reproducir archivos de audio y video de buena calidad. Sin embargo, la transferencia de archivos completos se traduce en tiempos de transferencia muy largos y la imposibilidad de ver y escuchar en tiempo real. El método ideal para enviar medios sería generar un flujo de video (video stream) a través de Internet desde el servidor al cliente en respuesta a una solicitud del mismo. El cliente reproduce el flujo entrante en tiempo real, a medida que va recibiendo los datos. (Sciara, 2004)

La evolución en el terreno de las comunicaciones, parece sufrir una gran demora evolutiva en cuanto a las velocidades de transmisión de la información. La solución a este problema de espera pasa por el streaming, consistente en la reproducción del archivo de vídeo o música a medida que va llegando al ordenador y sin necesidad de tener el archivo en su totalidad para comenzar su escucha o visionado. (Proveda, 2008)

Por lo planteado anteriormente se considera que el streaming es uno de los métodos de reproducción de contenidos multimedia más utilizados en la actualidad ya que esta tecnología permite ver u oír grandes archivos a través de la red, sin la necesidad de descargarlo antes al computador, permitiendo escuchar o visualizar el contenido en tiempo real o previamente editado.

CAPÍTULO 1: Fundamentación teórica

El servicio streaming puede ser:

Streaming en directo:

Servicio para la transmisión de audio y/o video a través de internet en el instante en que es emitido o producido.

Streaming Bajo demanda:

Almacenamiento de archivos, previamente producidos de audio y/o videos en servidores que luego son solicitados por los visitantes del sitio web.

Usos de Streaming:

- Transmisión de eventos.
- Transmisión de estaciones de radio y TV en línea.
- Sistemas de audio y video privado para empresas.
- Transmisión de conferencias, charlas y alocuciones en vivo.
- Permite cumplir con normativas legales (licitaciones).

Distribución del streaming:

Unicast: *Utiliza una conexión punto a punto en la que se establece comunicación entre dos elementos de forma bidireccional. Esta es la forma en la que funcionan la mayor parte de las conexiones en Internet y presenta la ventaja de ser soportada por todos los tipos de equipos. (Agüera, 2004)*

Multicast: *No emite para todos los destinos de una red, se trata de una conexión uno-a-varios, no uno-a-todos. El proceso de conexión implica el envío de un datagrama IP a un grupo de equipos identificados por una sola dirección IP. Esto es lo que se denomina un grupo multicast. (Agüera, 2004)*

Broadcast: *Define un tipo de conexión uno-a-todos, es decir, que la comunicación se realiza entre un nodo emisor y los puntos que le rodean. Los datagramas IP enviados durante este tipo de conexiones incluyen una dirección de destino correspondiente a todos los equipos conectados a la misma red local.*

CAPÍTULO 1: Fundamentación teórica

Es un tipo de conexión utilizada en intranets fundamentalmente para la difusión programada de contenidos, aunque no es especialmente eficiente en la administración de recursos. (Agüera, 2004)

Logros del uso de la tecnología streaming.

El Streaming es una tecnología que permite publicar video y mensajes audiovisuales en la red sin que el usuario pueda esperar que se cargue completamente, esto significa la posibilidad de ver el contenido antes que termine la transmisión con la facilidad de reproducir sin tener que grabar la información en disco, por ello puede elegir quedarse o abandonar, según su elección. (IguanaHosting, 2009)

Los mayores beneficios del streaming en cuanto a mejoras en la calidad y capacidad de comunicación, se engloban en los siguientes ámbitos:

- Con el streaming se alcanza una audiencia global y dispersa, de manera fácil y económica.
- Mejora la calidad de la comunicación interna entre las personas que lo utilizan.
- Diferentes usuarios conectados que pueden solicitar un mismo video y al mismo tiempo.
- Alarga la vida de los contenidos de comunicación o formación (y por lo tanto su rentabilidad).

- Ahorro de costes

El nivel de comprensión de un usuario que atiende a un streaming en directo o bajo demanda es superior al de una persona que atiende al evento presencialmente ya que se puede parar, rebobinar, acceder a un punto de interés cuantas veces necesite. Con este procedimiento se funden el concepto de directo y en diferido en una sola pantalla anulando los costes de producción de videos para su difusión en internet, ya que se hace de forma automática. (Mnemmix, 2008)

Aplicaciones Generales:

- ✓ Difusión de eventos:
 - Retransmisión de eventos
 - Difusión de coloquios, noticias, conferencias.
 - Demostraciones de producto.
- ✓ Radio y TV en internet. En directo y bajo de manda en una misma pantalla.

Limitaciones del Streaming.

Si bueno es destacar las ventajas de esta tecnología, también es necesario conocer algunas desventajas, como la falta de un estándar consolidado en archivos y arquitectura de administración del Streaming. También la necesidad de tener un ancho de banda superior a 28,8 Kbps, para tener una reproducción de calidad alta o media, pues cuando se realiza el streaming este consume gran cantidad de ancho de banda. Disponer de un software servidor de ficheros y otro cliente de la misma naturaleza, para obtener una buena calidad en las transmisiones y en la reproducción. Difícil acceso debido a la descentralización de la información provocando problemas en la búsqueda de información.

Diferentes tipos de servidores de Streaming

Servidores de Streaming basados en proxies

Los servidores proxy apoyan la interacción de las aplicaciones multimedia de streaming, permitiendo a los usuarios escuchar un fragmento de puntos antes de ver todo el contenido del video. El almacenamiento en caché de los segmentos de punto de acceso mejora la función de este ya que estos segmentos requieren mucho menos espacio que un archivo de video completo. (Boldrito, 2004)

Aunque los servidores proxy pueden almacenar en caché, texto, documentos e imagen, utilizados para la transmisión de documentos multimedia no ha sido eficaz por su gran tamaño. Para estos problemas, se puede utilizar el prefijo Caching para almacenar las primeras imágenes de los clips populares. El servidor transmite imágenes posteriores al proxy, mientras que el proxy transmite los datos al cliente. (Boldrito, 2004)

Servidores de streaming Distribuidos basados en Clúster

Los servidores de streaming basados en clúster se caracterizan por trabajar con el sistema operativo de Linux. La idea básica de un clúster es trabajar con un conjunto de computadoras conectadas a través de una red, trabajando un gran problema de cómputo y dividirlo en varios problemas pequeños para aligerar la carga de los servidores y aprovechando al máximo los recursos disponibles, ya que cuando varios

usuarios se encuentran solicitando un mismo recurso al servidor, este experimenta una sobre carga y por eso el servicio no alcanza la calidad requerida.

Estos sistemas constituyen una solución flexible, de bajo costo y gran escalabilidad para aplicaciones que requieren una elevada capacidad de cómputo y memoria. Cuando cualquier nodo del clúster se daña, todos los otros siguen funcionando normalmente absorbiendo la carga total del sistema y de esta forma se obtiene una tolerancia a fallas superior. En cuanto a la escalabilidad, es muy simple incorporar nuevos nodos en un clúster y expandirlo progresivamente con una baja inversión. Todos estos paradigmas de resolución de problemas tienen sus bases en el “cómputo paralelo”. El paralelismo consiste en poder dividir una tarea en partes que trabajen independientemente en lugar de poseer una única tarea en la que todos sus procesos se encadenan uno tras otro, necesitando de los resultados del anterior para poder comenzar.

Los servidores de streaming Distribuidos se caracterizan por tener:

- Colector de Datos.
- Manejador de Datos.
- Distribuidor de Datos.
- Manejador de Clúster.

1.2.2. Motores de Búsqueda.

Tipos de buscadores:

- ✓ Directorios, Índices, Catálogos que agrupan sus enlaces por categorías.
- ✓ Spiders de búsqueda, o buscadores propiamente dichos.
- ✓ Sistemas Mixtos spiders que tienen un directorio, además de su motor de búsqueda.

Los Directorios:

Muchos buscadores en realidad son directorios como Yahoo. La diferencia entre un buscador y un directorio es que el buscador usa sistemas automáticos (robots) para indexar las páginas mientras los

CAPÍTULO 1: Fundamentación teórica

directorios usan personas para calificar cada página. Aunque la calificación de los directorios es de mejor calidad que las de los buscadores, la cantidad de páginas indexadas es menor. (Taboexa, 2007)

Los directorios más conocidos son los de Yahoo!, Open Directory Project y Looksmart.

Características:

- Los algoritmos son mucho más sencillos.
- No recorren las webs ni almacenan sus contenidos.
- Solo registran algunos de los datos. Como el título y la descripción de la web que se introduzcan a la hora de registrar las webs.
- Los resultados de la búsqueda, estarán determinados por la información que se haya suministrado al directorio cuando se registra la web.
- Más que buscar información sobre contenidos de la página, los resultados serán presentados haciendo referencia a los contenidos y temática de la web.
- Su tecnología, es muy barata y sencilla. Imagínese que es una base de datos como la que usted utiliza, sobre la que se realizan búsquedas.
- Tiene un coste de operación relativamente alto, pues tiene que ser operado por humanos práctica y exclusivamente.
- Son apropiados para buscar categorías, más que informaciones específicas.

Los Spiders:

La mayoría de grandes sitios internacionales que todos usan y conocen son de este tipo. Requieren muchos recursos para su funcionamiento, no están al alcance de cualquiera. Recorren las páginas recopilando información sobre los contenidos de las páginas. Los spiders, tienen una colección de programas simples y potentes con diferentes cometidos. Se suelen dividir en tres partes. Los programas que exploran la red spiders, los que construyen la base de datos y los que utiliza el usuario. Los spiders, son mejores para encontrar informaciones concretas en páginas de una web. (Pereda, 2008)

Ejemplos de Spiders: Google, AltaVista, Hotbot, Lycos

Los sistemas mixtos Buscador - Directorio:

Son una mezcla entre spider y directorio. Además de tener características de arañas, presentan las webs registradas en catálogos sobre contenidos de Informática, cultura y sociedad.

Ejemplo de sistema mixto Excite, Voila, Infoseek.

Metabuscadores

Utilidades que buscan varios motores de búsqueda y/o directorios temáticos de manera simultánea, luego recopilan los resultados y los muestran en forma fácil de entender, algunas veces consolidan todos los resultados en un formato uniforme y los listan. Algunos ofrecen características con valor agregado como la habilidad para refinar las búsquedas, personalizar los motores de búsqueda o directorios que son solicitados, el tiempo gastado en cada uno, entre otros. (Barker, 1995)

Limitación:

La implementación tiene como consecuencia que se limita la habilidad para controlar donde se quiere ir a “comprar”. Desde mi punto de vista, NINGUNO de los Metabuscadores es tan bueno como puede serlo USTED si aprende a buscar efectivamente. Son una magnífica idea pero producen desilusión al ponerlos en práctica. (Barker, 1995)

Recomendación:

A pesar de continuar monitoreando el desarrollo de las tecnologías de los Metabuscadores, no se recomienda que hagan parte de la estrategia de búsqueda. Son rápidos y burdos, poco minuciosos, con frecuencia omiten a Google (el mejor Motor de Búsqueda en este momento), siempre omiten Northern Light (útil para investigaciones académicas) y no permiten utilizar las ventajas que ofrecen las características avanzadas de cualquier motor de búsqueda. (Barker, 1995)

1.3. Métodos de búsquedas en base datos.

Las bases de datos son fuentes muy valiosas para almacenar y recuperar la información. Están pensadas para poder interrogarlas de diferentes maneras y recuperar la información de forma rápida y ordenada.

CAPÍTULO 1: Fundamentación teórica

Son herramientas imprescindibles para la investigación, el trabajo académico o el mundo profesional. Son pues el conjunto de información que es almacenada siguiendo la misma estructura para cada una de las unidades de información. Permiten la búsqueda y la recuperación ordenada, normalmente aparecen en formato electrónico. (Inforesco, 2002)

Tipos de bases de datos existentes de modo general.

En líneas generales pueden ser de varios tipos:

- ✓ Bases de Datos Bibliográficas: contienen referencias de documentos.
- ✓ Bases de Datos Textuales: contienen el texto íntegro de los documentos.
- ✓ Bases de Datos Mixtas: contienen ambos tipos de datos.

Sistema Gestor de base datos. (SGBD)

En realidad, casi todas las bases de datos actuales, son SGBD. El término hace referencia a que una base de datos no es simplemente un conjunto de ficheros, sino que incorpora una serie de herramientas que permiten manipular la información introducida. Es esta característica la que permite, entre otras cosas, hacer las búsquedas.

Existen diferentes tipos de bases de datos (depende del criterio que escojamos para clasificarlas):

Según su contenido:

- ✚ **Bases de datos referenciales:** Remiten a otra fuente en la que se puede encontrar la información completa, o sea, sólo proporcionan una descripción de la información. Fundamentalmente, se encuentran en este apartado bases de datos bibliográficas (dan referencias sobre libros, revistas, tesis, documentos de trabajo) y directorios (proporcionan información sobre personas y/o instituciones).
- ✚ **Bases de datos fuente:** Estas contienen los datos completos de la fuente original de información. Pueden ser numéricas, textuales y textual-numéricas (presentan tanto textos como datos numéricos).

Para la realización de la herramienta que se pretende lograr con el presente trabajo de diploma, es necesario una base de datos pues es donde se van a encontrar almacenada toda la información de las medias de los servidores de streaming, para luego realizar una mejor búsqueda ya que va a estar toda la información centralizada en un catálogo.

Bases de datos multimedia

Proporcionan servicios de almacenamiento y consulta que hacen más eficiente el acceso a los datos.

Requisitos específicos de los SGBD Multimedia:

- ✓ Mecanismos para descripción de datos multimedia.
- ✓ Almacenamiento de grandes volúmenes de información.
- ✓ Almacenamiento en diferentes formatos y dispositivos.
- ✓ Transferencia de datos en tiempo real.
- ✓ Métodos de búsqueda específicos e independientes del formato.
- ✓ Acceso simultáneo a datos multimedia desde varias aplicaciones.
- ✓ Mecanismos sincronización de diferentes datos multimedia.

1.4. Análisis de otras soluciones existentes

A escala mundial se han desarrollado soluciones a la problemática que da origen a esta investigación. En el presente epígrafe se hace un análisis de dichas soluciones para intentar buscar apoyo en alguna de ellas que sirva de base para proponer la solución de esta investigación, o comprobar si existe alguna semejante que permita enfrentar al problema planteado. Ejemplo de estas soluciones son:

MEDIABOX es el producto de **XTREAM** que facilita la catalogación y archivo multimedia (vídeo, audio, documentos electrónicos) desde diferentes tipos de fuentes y formatos. El Sistema permite asociar documentos según criterio del usuario (por autor, materia, fechas, etc.) digitalizando y capturando la información para guardarlos con la calidad deseada. Trabaja en red y almacena los datos en un Servidor Central. Está indicado para edificios con múltiples puntos de captura y que pueden requerir interconexión con otros centros.

CAPÍTULO 1: Fundamentación teórica

Aplicaciones

El Sistema está orientado a todos aquellos sectores (industria, corporaciones e instituciones públicas) con necesidades de documentación y archivo de contenido de audiovisuales.

Universidades

- Grabación de conferencias, ponencias, simposios etc. marcando intervinientes y almacenamiento en Servidor Central asociando a documentos relacionados.
- Grabación en proyectos de investigación (biología, medicina, sociología, etc.) y archivo junto a documentos relacionados: de texto, en formato electrónico, PDF, etc.
- Digitalización de vídeos en formato analógico existentes en archivo previo.

Centros de investigación

- Grabación de conferencias, ponencias, simposios, etc. marcando intervinientes y almacenamiento en Servidor Central con asociación a documentos en otros formatos y datos de catalogación.
- Grabación de desarrollo de proyectos de investigación y archivo catalogado.

Corporaciones

- Grabación de juntas, actos en salas de conferencias, sesiones de formación y archivo con asociación de otros documentos relacionados y datos de catalogación.
- Grabación de cámaras en entornos de seguridad y operaciones.

Seguridad

Acceso por usuario y contraseña con diferentes tipos de usuarios y permisos posibles:

- Administración
- Consulta

Flexible

- Captura de cinta o señal en vivo.
- Admite diferentes formatos de compresión de entrada: MPEG1, MPEG2, WM8, WM9 MP3, DV25, DV50.

CAPÍTULO 1: Fundamentación teórica

- Almacenamiento en alta calidad y media: Alto bit Rate: MPEG2, DV25 y DV50; Bajo bit Rate: MPEG1, MPEG4, WM, RV.
- Posibilidad de captura continua o planificada.
- Búsqueda por palabra clave o varias unidas por booleanas (and, or, not).

Abierto y Escalable

- Permite trabajar en local o almacenando en el Servidor Central.
- El sistema está preparado para conectar distintos centros y permitir el intercambio de información entre ellos.
- Permite la recuperación de vídeo desde sistemas robotizados de almacenamiento en cinta.

Fácil Operación

- Generación automática de índice visual, para facilitar la catalogación.
- El sistema muestra un árbol de directorios que representa un espacio virtual de trabajo.
- Sistema de planificación de captura basada en parrilla.

Tolerante a Fallos

- Ante caída de red durante la captura, se continúa la grabación y se almacena localmente. Al recuperarse, MEDIABOX transfiere los contenidos al Servidor Central.

Funcionalidad

El sistema está formado por dos módulos cuyas funciones se complementan. Algunos de los módulos que la autora considera más importantes por su aporte para el estudio del sistema que se desea realizar en dicho trabajo de diploma son a los que se harán referencia:

Consulta

- Búsqueda por palabra clave o varias unidas por booleanas.
- Búsqueda por similitud.

Archivo

- Gestión caché de contenidos y espacio en disco.
- Las publicaciones pueden realizarse de forma masiva con procesado por lotes.
- Los metadatos se almacenan en Base de Datos.

Principales Ventajas:

- Gestión de la aplicación desde interfaz gráfico. Captura manual o programada. Digitalización desde cinta de video o señal en vivo.
- Posibilidad de asociar archivos de texto, PDFs y metadatos de catalogación a vídeo formando una "unidad de catalogación".
- Se pueden realizar marcas manuales durante la grabación.
- Se puede catalogar mientras se graba y una vez almacenado en el Servidor permite una gestión de almacenamiento robotizado.
- Admite diferentes formatos de compresión en entrada (MPEG1, MPEG2, WM8, WM9, MP3, DV25, DV50) y calidad media o alta de almacenamiento.
- Búsqueda flexible empleando caracteres booleanos (and, or, not) para consulta.

Después de analizar las características y los métodos de trabajo de MediaBox la autora considera que este sistema puede ser de gran ayuda para el estudio y desarrollo de la herramienta que se necesita implementar pues MediaBox facilita el trabajo en red y almacena los datos en un Servidor Central que en el caso de la herramienta de este trabajo no sería un servidor sino en una base datos, donde se encontrarán almacenadas los datos de las medias de los servidores streaming. MediaBox está orientado a todos aquellos sectores con necesidades de documentación y archivo de contenido de audiovisuales, este podría ser sin lugar a dudas el departamento de Señales Digitales, además de que realiza consultas de búsqueda al igual que la herramienta que se desea desarrollar en dicho trabajo de diploma.

Sistema de Captura y Catalogación de Medias (SCCM)

Luego de una investigación en el centro de GeySed de la Universidad de las Ciencias Informáticas para buscar soluciones que pudieran servir de ayuda a la problemática que se presenta en este trabajo, se encontró el proyecto de SCCM: Sistema de Captura y Catalogación de Medias en el cual se desarrolla una solución de software que se adapte a las características de los clientes para la obtención, procesamiento, descripción y catalogación de materiales audiovisuales. Teniendo además como objetivos específicos:

- Desarrollar una herramienta en software libre, que permita realizar el análisis automatizado de los contenidos almacenados.
- Desarrollar una herramienta que permita mediante streaming en acceso a medias almacenadas.

CAPÍTULO 1: Fundamentación teórica

- Desarrollar una herramienta que permita la inserción de datos asociados a materiales audiovisuales.
- Desarrollar una herramienta que permita emitir reportes configurables.

El desarrollo de este proyecto será llevado a cabo siguiendo el Proceso Unificado de Desarrollo de Software (RUP), cuyo ciclo de vida se caracteriza por ser dirigido por casos de uso, centrado en la Arquitectura e iterativo e incremental.

El Sistema de Captura y Catalogación de Medias estará estructurado en cuatro módulos Administración, Control de Ingestas, Catalogación y Recuperación. Se desarrollarán además componentes de software que se integrarán a la solución haciendo que la misma sea lo más genérica y adaptable posible al entorno donde se implante.

Sus características y funcionalidades principales son:

- Sistema multiplataforma.
- Captura de señales de audio y video.
- Almacenamiento de señales de audio y video.
- Codificación de audio y video.
- Streaming de audio y video.
- Reproducción de audio y video.
- Catalogación de medias.
- Recuperación de materiales.
- Gestión de préstamos y servicios sobre materiales.
- Gestión de roles y usuarios.
- Configuración de servicios y recursos de trabajo.
- Monitoreo de procesos sobre medias.
- Planificación de procesos sobre medias.

Como se puede apreciar este sistema cuenta con algunas características similares a la herramienta que se pretende desarrollar, ya que permite la catalogación de medias para luego realizar una mejor búsqueda

CAPÍTULO 1: Fundamentación teórica

de las mismas. En el marco mundial existen varios sistemas similares como la empresa española Tedral con su software Tarsys, la empresa Videoma con sus módulos para empresas en el mundo del Broadcast, MediaBox, empresa francesa para la gestión de mediatecas a la que se hace referencia en este trabajo y por último Video Ingest, todas ellas con gran potencial en el mundo audiovisual, pero desarrollados en software propietario y con elevados precios de licenciamiento. Por lo que el éxito radica en el desarrollo del producto con plataforma completamente libre.

Helix Proxy

Otra de las soluciones encontradas es Helix Proxy que es un software que se instala en una red o puerta de enlace ISP que agrega y controla las solicitudes de cliente para los medios de comunicación a la carta en vivo y en directo desde los servidores alojados externamente. Helix proxy reduce el tráfico de red mediante la eliminación de las solicitudes redundantes para el streaming de medios de comunicación.

Se destina para el despliegue de una intranet corporativa, actuando como puerta de entrada que las solicitudes de servicios internos de RealPlayer para los medios de transmisión que se encuentra alojado en los servidores Helix en el interior o fuera de la red. También puede ser utilizado por los proveedores de servicios en busca de una solución que pueda gestionar las solicitudes de contenido que se originan fuera de su red. (Network, 2010)

Proceso de ejecución de Helix Proxy

- 1- El primer paso en el proceso de Helix Proxy sucede cuando un RealPlayer realiza las solicitudes de un archivo multimedia digital alojado en el servidor Helix.
- 2- Esta solicitud se dirige a través del Helix Proxy, que remite la solicitud al servidor de la Helix en los archivos de medios digitales que se almacenan.
- 3- El Servidor de Helix verifica la existencia del archivo, y que el RealPlayer está autorizado a través de las direcciones IP o la autenticación de contenido para acceder al contenido. Si el Servidor de Helix niega la solicitud, no se transmite el archivo solicitado, y tampoco lo hace el Helix Proxy.

Si está autorizado Helix Proxy tiene tres diferentes formas de envío de datos a los clientes y

CAPÍTULO 1: Fundamentación teórica

automáticamente elige el método más eficiente posible, en función del tipo de contenido solicitado y la configuración de la red. (Network, 2010)

Los tres métodos de envío de datos a los clientes:

- Pass Through: No existe conservación de ancho de banda, pero todas las peticiones de los medios de transmisión (tanto en la demanda y en vivo) pasan por Helix Proxy.
- Pull Splitting: Para las solicitudes de medios en vivo, Helix Proxy "comparte" el streaming entre los clientes que lo soliciten.
- Content Caching: Para las peticiones de la demanda, Helix Proxy almacena los datos de streaming de medios para su posterior visualización por otros clientes.

Características de Helix Proxy:

- Reduce los costos de ancho de banda asociado con el contenido que entra a la red mediante la eliminación de peticiones que sean repetidas para el streaming de medias.
- Reduce los problemas de transmisión que degradan la calidad de la experiencia de reproducción.
- Oculta la dirección IP de los usuarios internos y autentica cada solicitud de cliente en el origen, manteniendo el control de contenidos y aumentando la seguridad de los mismos.
- Es el componente de banda ancha de proxy-caché de la entrega de medios Helix- Plataforma, el estándar de oro para multi-formato, multi-plataforma de video digital.

Beneficios clave

- Reducir los costos de ancho de banda al eliminar la transmisión de datos redundantes entre el servidor y proxy, requieren menos ancho de banda para recibir la misma cantidad de contenido. Bajar el uso de ancho de banda se traduce en un ahorro en el balance final. Esto permite que si existen varias solicitudes que necesitan la misma información se agrupen dichas solicitudes en una sola y así se libera el ancho de banda en la red.
- Gestionar la Red de entrada y salida de ancho de banda, asegurando que las aplicaciones de misión crítica no se vean afectados por la transmisión de las actividades.
- Contenido de Control Helix Proxy autentica cada solicitud de cliente en el origen, por lo que el control del contenido siempre permanece en el servidor de origen.
- Seguridad Helix Proxy enmascara la dirección IP de los usuarios internos.

CAPÍTULO 1: Fundamentación teórica

- Mejorar la reproducción de Usuario Final Por el contenido de caché y la división más cerca del usuario final, Helix proxy reduce los problemas de transmisión que degradan la calidad de la experiencia de reproducción.

Luego de estudiar las posibilidades que brinda Helix Proxy se puede llegar a la conclusión que este sistema brinda mejoras para reducir los costos de ancho de banda al eliminar la transmisión de datos redundantes entre el servidor y proxy, pues requieren menos ancho de banda para recibir la misma cantidad de contenido. Este sistema verifica si existe el archivo y si el usuario posee la autorización requerida para realizar la solicitud. Además de que elige el método más eficiente posible de envío de datos en respuesta a la solicitud que se le confirió, en función del tipo de contenido solicitado y la configuración de la red. Aunque la herramienta que se pretende desarrollar no necesita necesariamente ninguna de las funciones que brinda Helix Proxy porque no se realiza el streaming fue de gran ayuda para la comprensión y el estudio de aplicaciones que se relacionen con dicho proceso.

Conclusiones generales de las soluciones encontradas.

Luego de haber realizado la investigación sobre algunas herramientas que proporcionen soluciones a la problemática que da origen a esta investigación, se puede concluir que la herramienta que se desarrolla en el proyecto de SCCM del departamento de señales digitales de la facultad 6 de la Universidad de las Ciencias Informáticas es la que más se asemeja para dar soluciones a la problemática de este trabajo pues posee algunas de las principales características como ser creada para un ambiente de software libre y poseer una base datos para guardar la información y así agilizar el proceso de búsqueda posterior. Por otro lado Helix proxy y MediaBox fue de gran apoyo al estudio y comprensión de temas para realizar búsquedas de información multimedia sirviendo de apoyo al proceso de desarrollo de la aplicación final, a pesar de que no se halló ninguna herramienta que verdaderamente cumpliera con todas las necesidades que requería la problemática final.

1.5. Conclusiones

En la elaboración de este capítulo se abordaron conceptos generales que sirvieron de guía para lograr un correcto desarrollo de esta investigación. Realizando una caracterización de los servidores de streaming en cuanto a su uso a nivel internacional, así como las ventajas y desventajas que estos poseen, se logró

CAPÍTULO 1: Fundamentación teórica

demostrar la importancia de su utilización para la reproducción de medias en internet. Se realizó un estudio de los métodos de búsquedas en base datos, los cuales aportaron un mayor conocimiento sobre su uso y sus características generales. Buscar y estudiar soluciones existentes de herramientas que permitan mejorar la gestión y el acceso a la información multimedia en servidores, sirvió de gran apoyo para la búsqueda de la solución a nuestro problema científico.

2. CAPÍTULO 2: Fundamentación de la tecnología a utilizar.

2.1. Introducción

En el presente capítulo se abordarán los temas relacionados con las tecnologías más actuales que se utilizarán en el desarrollo de la herramienta para la gestión y acceso a la información de los servidores de streaming.

2.2. Metodología de desarrollo de software.

El desarrollo de software y la programación es uno de los pilares fundamentales de la informática. Para obtener un resultado exitoso y conseguir la satisfacción de los clientes es necesario aplicar una metodología de desarrollo adecuada que pueda proporcionar seguridad y calidad al sistema que deseamos desarrollar. Se puede decir además que lo más importante antes de elegir la metodología que se debe usar para implementar el software, es determinar el alcance que tendrá y luego de allí ver cuál es la que más se acomoda a la aplicación.

Algunas de las Metodologías más conocidas de desarrollo de software son:

- La metodología RUP es la más adaptable para proyectos de largo plazo.
- La metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.
- La metodología XP en cambio, se recomienda para proyectos de corto plazo.
- Metodología de desarrollo de software basada en componentes.

2.2.1. El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- Transmisión: El objetivo es llegar a obtener la liberación del proyecto.

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: Las necesidades del negocio.
- Requerimientos: Trasladando las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Trasladando los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto.

Disciplina de Soporte

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto.

Características de RUP

1. Guiado por casos de uso: Los casos de uso capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.
2. Centrado en la Arquitectura: Comprende los aspectos estáticos y dinámicos más importantes del sistema.
3. Iterativo e Incremental: El trabajo se divide en piezas pequeñas o mini proyectos; cada uno proveyendo un subproducto incremental.

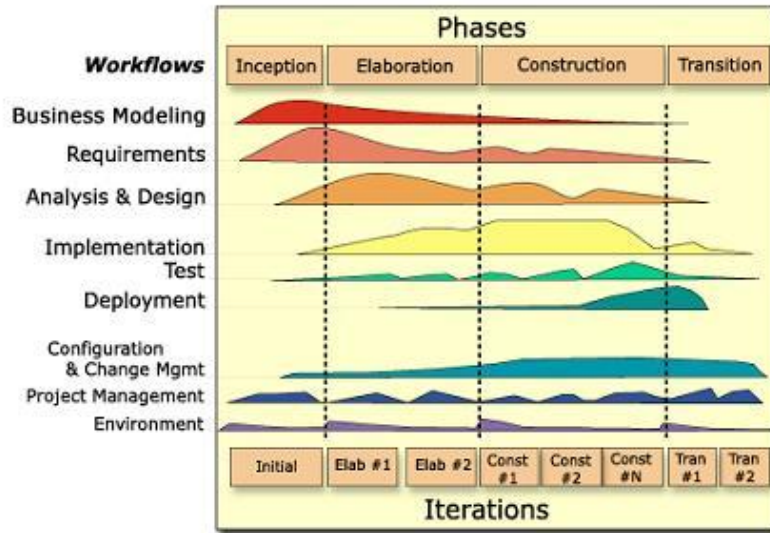


Ilustración 1: Fases e Iteraciones de la Metodología RUP (Jacobson, 2000)

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores:** Vienen siendo las personas o gentes involucradas en cada proceso.
- **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

2.2.2. Microsoft Solution Framework (MSF)

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



Ilustración 2: Metodología MSF (Prat, 2004)

MSF tiene las siguientes características:

- Adaptable: Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- Escalable: Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- Flexible: Es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Agnóstica: Puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

Modelo de Arquitectura del Proyecto: Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.

Modelo de Equipo: Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

Modelo de Proceso: Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

Modelo de Gestión del Riesgo: Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

Modelo de Diseño del Proceso: Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

Modelo de Aplicación: Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

2.2.3. Ingeniería de Software Basada en Componentes (ISBC)

La metodología de software basada en Componentes surgió a finales de los 90's como una aproximación basada en la reutilización al desarrollo de sistemas de software. Esta metodología fue motivada por la frustración de los desarrolladores de que el modelo orientados a objetos no aplicaba una reutilización extensiva, tal como ésta sugería originalmente, debido a que la utilización de clases implica el conocimiento detallado de ellas, lo cual significa que se debía tener acceso al código fuente lo que imposibilitaba el marketing de clases para su reutilización.

Tradicionalmente los ingenieros del software han seguido un enfoque de desarrollo descendente (top-Down) basado en el ciclo de vida en cascada (análisis-diseño-implementación) para la construcción de sus sistemas, donde se establecen los requisitos y la arquitectura de la aplicación y luego se va desarrollando, diseñando e implementando cada parte software hasta obtener la aplicación completa implementada. (Martínez., 2003)

La ISBC parte de la idea de la integración de componentes software ya existente (desarrollo ascendente o bottom-up).

Las tecnologías de objetos proporcionan el marco de trabajo técnico para la ingeniería de software, para un modelo de proceso basado en componentes. El paradigma orientado a objetos enfatiza la creación de

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

clases que encapsulan tanto los datos como los algoritmos para manejar esos datos. Si se diseñan y se implementan adecuadamente, las clases orientadas a objetos son reutilizables por diferentes aplicaciones. Es la reutilización la que permite a los desarrolladores construir aplicaciones sin partir desde cero, sino acercándose más al modelo de construcción de otras ingenierías, donde los productos se construyen en base al ensamblaje y adaptación de distintos componentes desarrollados por terceros (como lo es la industria del hardware para computadoras). Según Pressman, la metodología que propone entonces la “Ingeniería de Software Basada en Componentes” (ISBC) (Ilustración 3) incorpora muchas de las características del Modelo en Espiral. Es evolutivo por naturaleza, y por ello exige también un enfoque iterativo para la creación del software.

Pero reemplaza las fases de Ingeniería, Construcción y Acción de éste modelo por una sola fase de Construcción y adaptación de la Ingeniería:

Comunicación con el cliente: Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.

Planificación: Las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.

Análisis de riesgos: Las tareas requeridas para evaluar riesgos técnicos y de gestión.

Construcción y adaptación de la Ingeniería

Evaluación del cliente: Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

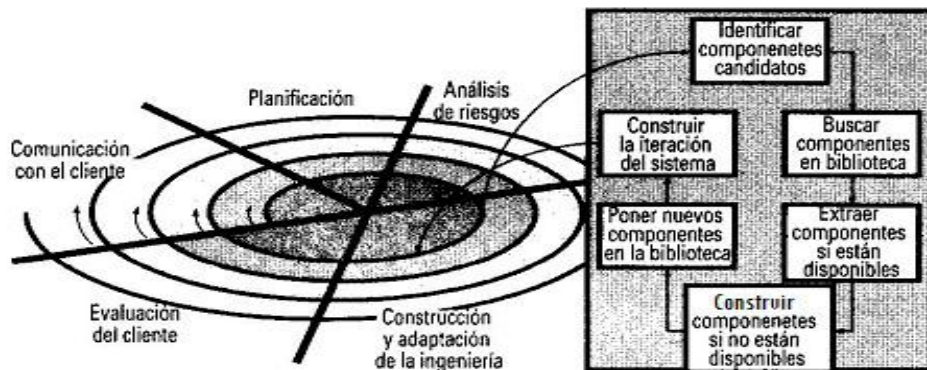


Ilustración 3: Ingeniería basada en componentes (Pressman, 2002)

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

El modelo de desarrollo basado en componentes conduce a la reutilización del software, y la reutilización proporciona beneficios a los ingenieros de software. Reduce en gran medida el tiempo del ciclo de desarrollo y el coste del proyecto, y aumenta la productividad. (Pressman, 2002)

Opinión de selección de la metodología de desarrollo a utilizar.

Después de analizar las anteriores metodologías de desarrollo de software es seleccionado RUP, ya que es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable, además es uno de los procesos más generales de los existentes actualmente ya que está pensado para adaptarse a cualquier proyecto.

Al contrario de otras las metodologías, la plataforma RUP hace que el proceso sea práctico con bases de conocimiento y guías para ayudar en el despegue de la planificación del proyecto, integrar rápidamente a los miembros del equipo y poner en acción el proceso personalizado.

Se adapta a las necesidades de los proyectos, solo la plataforma RUP proporciona un framework de proceso configurable que permite seleccionar e implantar los componentes específicos de proceso necesarios para proporcionar un proceso consistente para cada equipo y proyecto.

Es considerada una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software, donde el cliente no forma parte del equipo de trabajo al contrario de las otras metodologías expuestas anteriormente.

2.3. Herramientas case

Las Herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Actualmente existen una gran variedad de herramientas CASE para el proceso de desarrollo de software, por lo que seleccionar alguna se convierte en una difícil tarea. Actualmente existe un gran desarrollo y una gran cantidad de este tipo de herramientas, por lo que se hace difícil la elección de una de ellas para el trabajo, tanto personal como corporativo.

2.3.1. Visual Paradigm

El Visual Paradigm es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto, constituye una herramienta profesional para el modelado UML que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Posibilita el modelado de base de datos, requerimientos, proceso de negocio, permite realizar todo tipo de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Características que posee Visual Paradigm:

- ✚ Producto de calidad.
- ✚ Soporta aplicaciones web.
- ✚ Generación de código para Java y exportación como HTML.
- ✚ Fácil de instalar y actualizar.
- ✚ Licencia: Gratuita y Comercial.
- ✚ Compatibilidad entre ediciones.

Algunas ventajas que ofrece:

- Multiplataforma (Incluyendo GNU/Linux)
- Diseño centrado en casos de uso y enfocado al negocio, permitiendo generar un diseño de software de gran calidad.
- Permite realizar ingeniería directa e inversa.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo facilitando la comunicación.
- Permite el control de versiones.
- Modelo y código permanecen sincronizados en todo el ciclo de desarrollo.
- Permite integrarse con diferentes IDE.

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, entre otros. Además, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

excesivos, también permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código. (**Paradigm, 2004**)

Entre los lenguajes conocidos para la ingeniería inversa, se encuentran los siguientes:

Java, C++, PHP 5.0, XML.

2.3.2. Rational Rose Enterprise

Rational Rose Enterprise el producto más completo de la familia Rational Rose. Todos los productos incluyen soporte UML, para el modelado. Es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java, Visual C++ y Visual Basic. Como todos los demás productos **Rational Rose**, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Principales Características de Rational Rose Enterprise

Tabla 1: Características de Rational Rose Enterprise (NNova, 2007)

Características	
Incluye	Rose Enterprise
Integraciones IDE	
Borland Builder versiones 7.0 a 10.0	X
Sun Forte for Java Community y Enterprise Editions 3.0	X
Microsoft Visual Studio 6	VB6
Microsoft Visual Studio 2003	MSVC++
Microsoft Visual Studio 2005	MSVC++
Wind River Tornado	
Green Hills MULTI	
Diagramas	
Clases, Componentes, Deployment, Secuencia, Caso de Uso	X

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

Colaboración	X
Physical Storage / Deployment	X
Physical Data / Tables	X
Dominios - Windows	
UML 1.x	X
Data Modeling	X
Visual C++ 6.0	X
ANSI C++	X
J2SE, J2EE	Java SE 1.4 con algunas características 1.5 características EJB 1.2
J2ME	
JSP	X
Ada, CORBA IDL y MIDL, XML DTD	X
Desarrollo Orientado al Modelado	
Patterns	X
Build y Deployment	
J2EE deployment	X
Team Support	
Merge Changes	X
Repository / CM	X
Documentación / Reportes	
SoDA Integration	X
Print Diagrams	X
Web Publishing and Report Generation	X
Soportado por Procesos	
Soporte RUP for Systems Engineering	X

Características adicionales incluidas:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++.

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java.
- 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.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational Clear Case.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

Opinión de selección de la herramienta case para el modelado de diagramas.

Después de conocer algunas de las características de las herramientas case referidas anteriormente es el Visual Paradigm la herramienta seleccionada pues emplea UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar.

Además otra de las ventajas que posee el Visual Paradigm es que se adecua al entorno de desarrollo permitiéndoles a los diseñadores, analistas y a todo aquel que trabaje con el mismo, una mayor organización y claridad en el trabajo. Esta herramienta es bastante común y muy utilizada a nivel mundial por la variedad de funciones que permite realizar.

2.4. Lenguaje de programación.

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Los lenguajes de programación facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar.

2.4.1. Lenguaje de Programación en C++

El lenguaje de programación C++ es uno de los más empleados en la actualidad. Se puede decir que es un lenguaje híbrido, ya que permite programar tanto en el estilo procedimental, como en estilo orientado a objetos, como en ambos a la vez. Además también se puede emplear mediante programación en eventos para crear programas que usen interfaz gráfica de usuario. (Mora, 2006)

Es un lenguaje imperativo orientado a objetos derivado del C. En realidad un súper conjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción.

Características de C++

Una de las características más interesantes del lenguaje es la sobrecarga de operadores. Esto significa que a los operadores intrínsecos del lenguaje se les puede redefinir la semántica: se pueden escribir funciones que en vez de tener un nombre, se asocian a un operador, que debe tener por lo menos un parámetro de tipo clase. La libre utilización de punteros por parte de los programadores, por un lado aporta eficiencia, pero por otro es una fuente de errores de lógica. Por este motivo, lenguajes derivados de C++, como C# y Java, quitaron este recurso y solo permiten referencias a objetos.

Programación orientada a objetos: La posibilidad de la programación orientada a objetos permite al programador diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además, permite la reutilización del código de una manera más lógica y productiva.

Portabilidad: Un código escrito en C++ puede ser compilado en casi todo tipo de ordenadores y sistemas operativos sin hacer apenas cambios. Como el sistema operativo de GNU-Linux. Es utilizado para trabajar con el IDE de desarrollo de Qt-Creator.

Brevedad: El código escrito en C++ es muy corto en comparación con otros lenguajes, porque en este lenguaje es preferible el uso de caracteres especiales que las "palabras clave".

Programación modular: Un cuerpo de aplicación en C++ puede estar hecho con varios ficheros de código fuente que son compilados por separado y después unidos. Además, esta característica permite unir código en C++ con código producido en otros lenguajes de programación como Ensamblador o el propio C.

Velocidad: El código resultante de una compilación en C++ es muy eficiente, gracias a su capacidad de actuar como lenguaje de alto y bajo nivel.

2.5. Framework de desarrollo.

Es un conjunto estandarizado de conceptos, prácticas y criterios, con una estructura conceptual y tecnológica bien definida. Incluye soporte de programas, bibliotecas y lenguajes para ayudar a desarrollar un proyecto y facilita el desarrollo de software, permitiendo pasar más tiempo identificando requerimientos de software.

2.5.1. Qt Creator

¿Qué es Qt?

Qt es un framework de desarrollo de aplicaciones multiplataforma diseñadas principalmente para desarrollar interfaces gráficas de usuario, aunque es posible desarrollar una aplicación completa haciendo sólo uso de estas librerías. Qt utiliza el lenguaje de programación C++ de forma nativa y es utilizada principalmente en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros.
(Cambiaso, 2010)

Características de Qt

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

- Está escrito en C++.
- Es posible utilizar Qt con otros lenguajes.
- Viene acompañado de un conjunto de herramientas para facilitar su uso.

Licencias de Qt

- GPL: Código abierto. Los cambios realizados al código fuente de Qt deben ser compartidos con la comunidad.
- LGPL: Aplicaciones de código cerrado, los cambios realizados al código fuente de Qt deben ser compartidos con la comunidad.

Qt-Creator es un Entorno de Desarrollo Integrado (IDE) multiplataforma muy completo para el desarrollo de aplicaciones utilizando las bibliotecas de Qt, algunas de las principales características de Qt-Creator son:

- Posee un avanzado editor de código C++.
- Además soporta los lenguajes: C#/.NET, Python: PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby.
- Posee también una GUI integrada y diseñador de formularios.
- Herramienta para proyectos y administración.
- Ayuda sensible al contexto integrado.
- Depurador visual.
- Resaltado y auto-completado de código.
- Soporte para refactorización de código.

Qt-Creator es distribuido bajo tres tipos de licencias: Qt Comercial Developer License, Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y está disponible para las plataformas: Linux, Mac OSX; Windows, Windows CE, Symbian y Maemo. (Cambiaso, 2010)

Para la comunicación entre procesos (IPC) Qt-Creator proporciona varias maneras de implementarla, algunas de las que se pueden mencionar son:

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

QTDBus: Este módulo es una biblioteca de tipo Unix que se utiliza para poner en práctica la comunicación entre procesos mediante el protocolo D-Bus.

QSharedMemory: Memoria compartida, proporciona acceso a la aplicación del sistema operativo de memoria compartida. Permite un acceso seguro a los segmentos de memoria compartida por varios subprocesos y procesos.

Qt Protocolo de Comunicaciones (QCOP): La clase QCopChannel implementa un protocolo para la transferencia de mensajes entre procesos de cliente a través de canales con nombre.

TCP / IP: Este módulo Qt-Network proporciona clases que hacen que la programación de la red sea portátil y fácil. Ofrece clases de alto nivel como QHttp, que implementan protocolos específicos a nivel de aplicación, y las clases de menor nivel como, QTcpSocket, QTcpServer, para la aplicación de protocolos. Utilizando este módulo y estas clases que proporciona Qt se logrará la comunicación entre las aplicaciones que se realizarán en este trabajo de diploma ya que no es necesario utilizar un middleware específico para lograr esta comunicación.

Se ha seleccionado Qt-Creator porque es un proyecto de Código Abierto, gratuito y de libre distribución. Es multiplataforma, una aplicación desarrollada utilizando éstas librerías se podría compilar tanto en Linux como en Windows (entre otros) sin necesidad de hacer ningún cambio en el código. Permite construir interfaces de usuario complejas de una forma visual y rápida, lo que nos deja más tiempo para centrarnos en la parte más importante de todo programa. Al utilizar el lenguaje de programación C++, podemos hacer uso del potencial de la programación orientada a objetos. Se puede construir interfaces gráficas visualmente muy atractivas en poco tiempo. Además que provee clases para la comunicación entre procesos evitando tener que utilizar un middleware específico para lograr dicha comunicación.

2.6. Sistema Gestor de Base Datos.

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (Database Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (Cavsi, 2011)

2.6.1. PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

Características de PostgreSQL:

PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente lista contiene algunas de esas características, a partir de PostgreSQL.

➤ **DBMS Objeto-Relacional**

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia, y arreglos.

➤ **Altamente Extensible**

PostgreSQL soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.

➤ **Soporte SQL Comprensivo**

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

➤ **Integridad Referencial**

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

➤ **API Flexible**

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

➤ **Lenguajes Procedurales**

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

➤ **Cliente/Servidor**

PostgreSQL usa una arquitectura cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Se ha seleccionado como sistema gestor de base de datos PostgreSQL por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que funciona en la mayoría de los sistemas Unix, además tiene características que permiten extender fácilmente el sistema. PostgreSQL puede ser integrada al ambiente Windows permitiendo de esta manera a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes. Permite desarrollar o migrar aplicaciones desde Access, Visual Basic, FoxPro, Visual FoxPro, C/C++, Visual C/C++, Delphi, para que utilicen a PostgreSQL como servidor de BD. Con este SGBD se podrá mantener una base de datos actualizada, confiable y configurable. Por lo expuesto PostgreSQL se convierte en una gran alternativa al momento de decidirse por un sistema de bases de datos.

2.7. Conclusiones

En el desarrollo de este capítulo se definieron las tecnologías y herramientas fundamentales que se utilizarán para lograr que el sistema que se desea implementar cuente con las tecnologías adecuadas teniendo en cuenta las necesidades actuales de la Universidad de Ciencias Informáticas. Fue muy importante conocer e investigar sobre las tecnologías más usadas así como comparar algunas de estas, para seleccionar la más adecuada.

A continuación se mencionan las metodologías, herramientas y lenguajes que se utilizarán: Como metodología de desarrollo a utilizar se eligió a RUP, como IDE de desarrollo Qt-Creator, como gestor de

CAPÍTULO 2: Fundamentación de la tecnología a utilizar

base de datos PostgreSQL, como herramienta CASE para el modelado fue seleccionada Visual Paradigm y como lenguaje de programación C++.

3. Capítulo 3: Presentación de la solución propuesta.

3.1. Introducción.

En este capítulo se presenta todo lo referente al modelo de dominio en el cual se exponen todos los conceptos relacionados con el mismo y se analizan las entidades relacionadas con el sistema. Se especifica detalladamente los requisitos funcionales y no funcionales con los que debe contar el sistema que se desea desarrollar, se describen los actores y los casos de uso del sistema.

3.2. Modelo de Dominio.

El Modelo de Dominio es una representación visual estática del entorno real del objeto del proyecto. Es decir, un diagrama con los objetos que existen (reales) relacionados con el proyecto que se va a desarrollar y las relaciones que hay entre ellos, teniendo como ventaja el permitir ayudar a los usuarios, clientes y desarrolladores a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

Diagrama de clases del Modelo de Dominio.

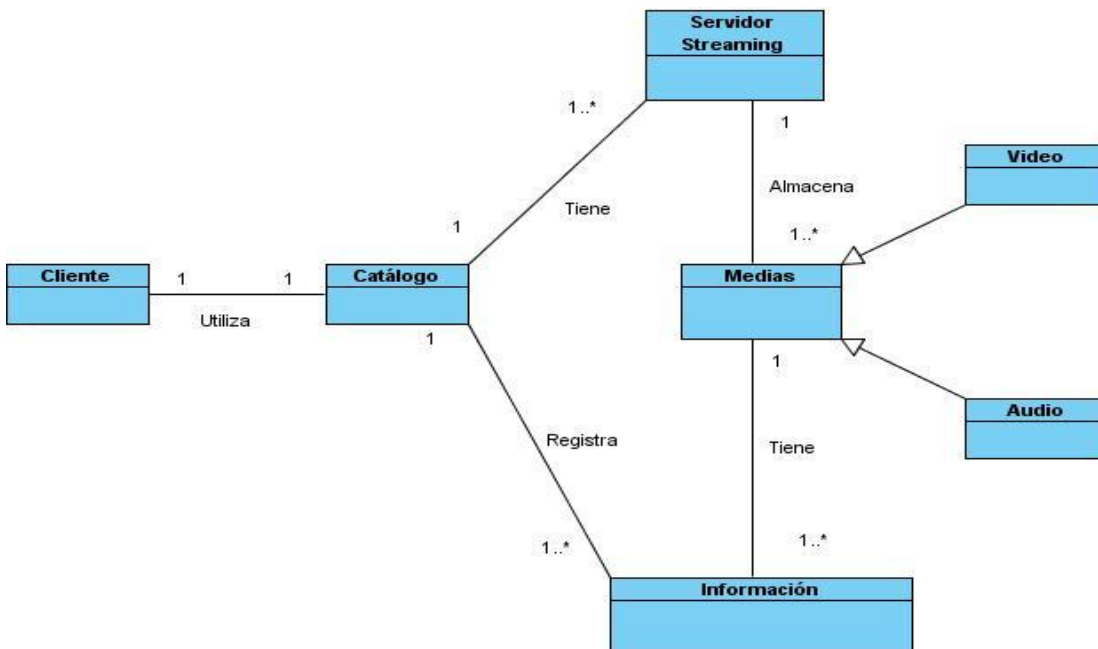


Ilustración 4: Diagrama de Modelo de dominio

CAPÍTULO 3: Presentación de la solución propuesta

3.2.1. Glosario de Términos del Dominio.

Catálogo: Es una publicación de información cuyo fin es la promoción de las medias que se encuentran en los servidores streaming que están registrados dentro del catálogo. El catálogo va a contener toda la información que se necesita de las medias para que el administrador pueda realizar la búsqueda.

Servidor de Streaming: Son ordenadores dedicados a ofrecer (servir) información a los usuarios que la demandan, este va a almacenar las medias pertinentes para realizar el streaming cuando los usuarios lo demanden, ya sea de video o audio.

Cliente: Persona que accede a un producto o servicio en este caso sería la persona que accede al servicio que proporciona el catálogo mediante la búsqueda de medias.

Media: Información digitalizada que combina texto, gráficos, video y audio. Multimedia es un término que se aplica a cualquier objeto, informático o no, que combina diferentes formas de contenido informativo como texto, sonido, imágenes, animación y video. Las medias no son más que los archivos de audio o video que van a estar almacenadas en los servidores para que los usuarios puedan acceder a ellos mediante el streaming.

Información: Conjunto de datos almacenados sobre las medias que se almacenan en los servidores de streaming.

Video: La tecnología de la captación, grabación, procesamiento, almacenamiento, transmisión y reconstrucción por medios electrónicos o analógicos de una secuencia de imágenes que representan escenas en movimiento. Estos videos son almacenados en los servidores para que el usuario pueda acceder a ellos cuando él lo solicite.

Audio: Es un contenedor multimedia que guarda una grabación de audio (música, voces, etc.). Lo que hace a un archivo distinto del otro son sus propiedades; cómo se almacenan los datos, sus capacidades de reproducción, y cómo puede utilizarse el archivo en un sistema de administración de archivos (etiquetado).

3.3. Levantamiento de requisitos.

3.3.1. Requisitos Funcionales del sistema.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

El sistema debe ser capaz de:

RF1-Determinar cantidad de conexiones de los servidores: Es necesario conocer la cantidad de conexiones que tiene un servidor en un momento determinado para cuando se realice una búsqueda de medias y esta se encuentre en dos o más servidores el usuario conozca cual tiene mayor disponibilidad y así evitar la carga del servidor.

RF2- Buscar una media en el catálogo: El módulo del Catálogo debe ser capaz de buscar una media que esté registrada en cualquiera de los servidores que se encuentren en el catálogo y una vez encontrada la media mostrar la URL de publicación, en caso de que dicha media buscada sea encontrada en dos o más servidores debe mostrar la dirección URL de publicación de la media y la cantidad de conexiones que posee cada servidor en ese momento para que el usuario pueda conocer cual está más libre de carga.

RF3- Registrar servidor en el catálogo: Se registra un nuevo servidor al catálogo para brindar los servicios de búsqueda. El administrador del servidor es el que registra un servidor al catálogo.

RF4-Modificar servidor en el catálogo: El administrador del servidor debe ser capaz de modificar los datos de su servidor una vez que este se encuentre inscrito en el catálogo.

RF5-Eliminar un servidor del catálogo: El administrador del servidor debe ser capaz de eliminar su servidor del Catálogo.

RF6-Insertar media en el catálogo: El administrador del Servidor es el encargado de insertar las medias de su servidor en el catálogo y luego se actualiza la media en la base datos.

RF7- Modificar media del catálogo: El administrador del Servidor debe ser capaz de modificar los datos de las medias que se encuentren publicadas en el catálogo, siempre y cuando estas se encuentren en su servidor.

RF8- Eliminar media del catálogo: El administrador del servidor debe ser capaz de eliminar una media de su servidor y debe notificar al sistema del catálogo para que este elimine la media de la base de datos.

RF9-Actualizar dirección IP del catálogo: Cuando el catálogo se mueve a otra dirección IP este debe notificar a todos los servidores registrados la nueva dirección.

3.3.2. Requisitos no Funcionales del Sistema.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son las características que hacen al producto atractivo, usable, rápido o confiable.

➤ Usabilidad

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Así como también que posean un servidor de streaming y que quieran utilizar el sistema para que la información del contenido de su servidor este centralizada para lograr una búsqueda más rápida.

➤ Rendimiento

El tiempo de respuesta para la búsqueda de una media debe ser de 2 segundos, para registrar o eliminar un servidor al catálogo y para agregar una media o eliminarla del catálogo 1 segundo, en dependencia de la cantidad de información que se desee agregar en el catálogo.

➤ Restricciones de diseño

El diseño debe ser sencillo, con pocas entradas, donde no es necesario mucho entrenamiento para utilizar el sistema. Se logrará un producto altamente configurable y extensible, teniendo en cuenta que se desarrolla sobre el framework Qt-Creator y que constituye una plataforma de desarrollo para ser utilizada en software libre.

➤ Apariencia o interfaz externa

CAPÍTULO 3: Presentación de la solución propuesta

Debe brindar una interfaz amigable, interactiva, intuitiva y de fácil comprensión para el usuario, facilitando en todo momento la interacción de este con el sistema.

➤ **Hardware**

Para la PC donde se va a encontrar el catálogo:

Se requiere tengan tarjeta de red.

Al menos 1 GB de memoria RAM.

Se requiere al menos 10GB de disco duro.

Procesador 512 MHz como mínimo.

Para las PC donde se encuentran los servidores de streaming:

Se requiere tarjeta de red.

El Servidor de streaming tenga como mínimo 1GB de RAM y 2GB de disco duro.

Procesador 3 GHz como mínimo.

➤ **Software.**

Sistema Operativo Ubuntu.

Framework Qt.

SGBD PostgreSQL.

3.4. Actores del Sistema.

Un actor del sistema no es parte del sistema en desarrollo, es un agente externo que intercambia información con el mismo en pos de obtener un resultado esperado.

Este sistema cuenta con los siguientes actores:

Tabla 2: Descripción de actores

Actor	Descripción
Administrador General	Es la persona que utiliza y manipula el catálogo, donde va estar toda la información de todos los servidores que se encuentren registrados en el mismo para realizar la búsqueda de medias.

CAPÍTULO 3: Presentación de la solución propuesta

Administrador del Servidor de Streaming	Es la persona que administra los servicios del servidor, este va a ser el que agregue un servidor en el catálogo, y el responsable de agregar las medias que contiene el servidor.
-----------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Diagrama de Casos de Uso del Sistema.

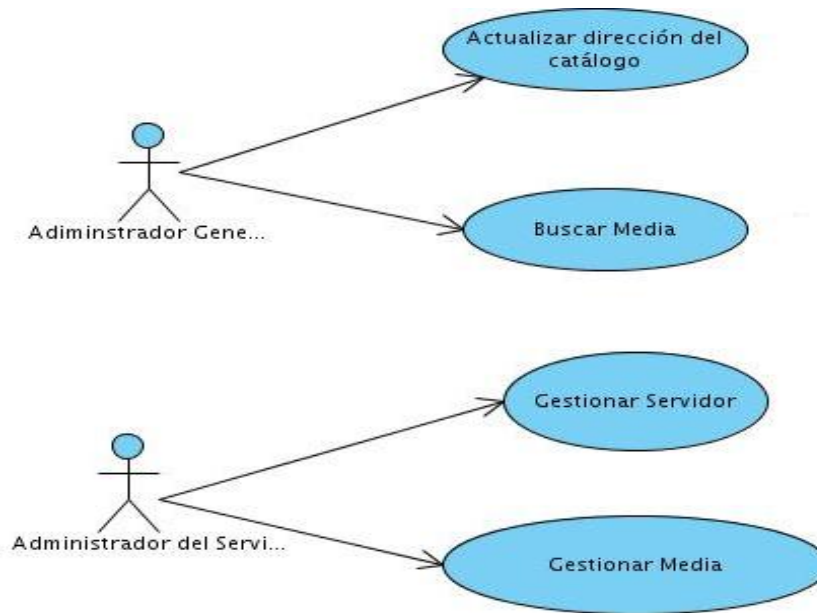


Ilustración 5: Diagrama de Casos de Uso del Sistema.

3.5. Descripción de los Casos de Uso del Sistema.

Tabla 3: Descripción textual del caso de uso Buscar Media.

Caso de Uso:	Buscar Media
Actor:	Administrador General.
Propósito:	Este caso de uso se lleva a cabo con el objetivo de buscar una media que se encuentre en cualquiera de los servidores que estén registrados en el catálogo.
Resumen:	El caso de uso inicia cuando el Administrador General accede al servicio de buscar una media. Luego introduce el nombre de la media que desea buscar y el

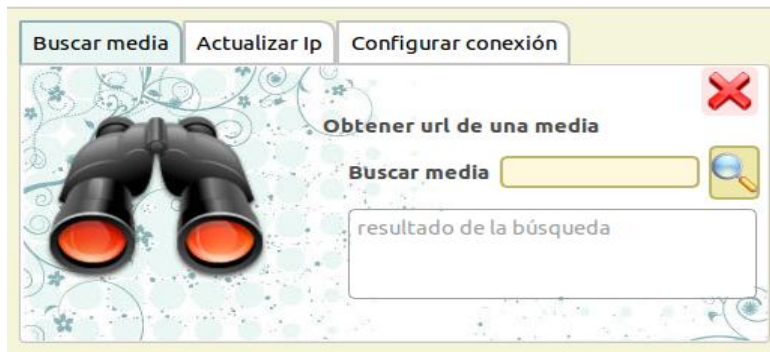
CAPÍTULO 3: Presentación de la solución propuesta

	sistema le devuelve la dirección de publicación de la media que buscaba.
Precondiciones:	Debe insertar los datos para la conexión con la base datos antes de realizar la búsqueda.
Referencias	RF2
Prioridad	Alta

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el administrador general selecciona la opción, "Buscar Media".	2. El sistema muestra el campo para introducir el nombre de la media que desea buscar.
3. El administrador general introduce el nombre de la media y presiona el botón "Buscar".	4. El sistema como resultado de la búsqueda muestra la dirección de publicación de la media.

Prototipo de Interfaz



Post-Condiciones

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	<p>4. En caso de que la media que se deseaba buscar no se encuentre en ninguno de los servidores registrados en el catálogo se mostrará como resultado un mensaje informando que no se ha encontrado ninguna coincidencia.</p> <p style="text-align: center;">Retorna a 2 del flujo normal de eventos.</p>

CAPÍTULO 3: Presentación de la solución propuesta

	4. Si la media es encontrada en dos o más de los servidores que están en el catálogo se mostrará como resultado la dirección IP de los servidores, la cantidad de conexiones que tiene cada uno de los servidores y la dirección de publicación de la media.
3. El actor deja el campo para introducir el nombre vacío o sea que no introduzca ningún nombre.	4. Se mostrará como resultado un mensaje informando que debe introducir el nombre de la media que desea buscar.

Tabla 4: Descripción textual del caso de uso Gestionar Servidor.

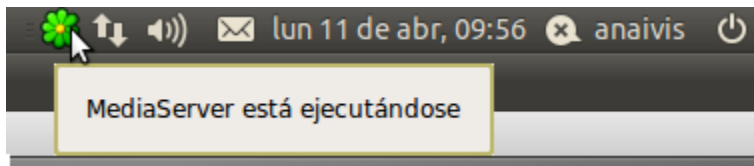
Caso de Uso:	Gestionar Servidor	
Actor:	Administrador del Servidor.	
Propósito:	Este caso de uso se lleva a cabo con el objetivo de insertar, modificar o eliminar un servidor en el catálogo.	
Resumen:	El caso de uso inicia cuando el Administrador del Servidor desea gestionar los datos de su servidor. Una vez instalada la aplicación el administrador del servidor puede seleccionar la opción Configurar Datos del Servidor, donde puede insertar, modificar o eliminar el servidor en el catálogo.	
Precondiciones:	El administrador debe instalar previamente el programa y estar en estado de ejecución para poder insertar, modificar o eliminar el servidor, además debe insertar los datos para la conexión con la base datos.	
Referencias	RF3, RF4, RF5	
Prioridad	Alta	
Flujo Normal de Eventos		
Sección "Insertar Servidor"		
Acción del Actor	Respuesta del Sistema	

CAPÍTULO 3: Presentación de la solución propuesta

1. El caso de uso inicia cuando el Administrador del Servidor selecciona la opción "Configurar Datos del Servidor".	2. El sistema muestra un formulario con los datos necesarios que se requieren para insertar el servidor, como: -Nombre -Dirección IP -URL de Publicación -Ubicación de las medias
3. El Administrador del Servidor llena los datos solicitados por el sistema y presiona el botón Guardar.	4. El sistema comprueba que estén correctos todos los datos que se han introducido, y finaliza guardando el nuevo ingreso.

Prototipo de Interfaz

Ejecutando la Aplicación



Datos para insertar un servidor



Datos del servidor

Nombre:

Ip:

Url:

Medias:

Datos de la Conexión

Base de Datos:

Host:

Puerto:

Usuario:

Contraseña:

Post-condiciones

- El administrador del servidor podrá insertar el

CAPÍTULO 3: Presentación de la solución propuesta

	<p>servidor en el catálogo solo una vez, luego de haber instalado el programa.</p> <ul style="list-style-type: none"> - Cuando se insertan los datos para ingresar un servidor en el catálogo, la dirección IP se inserta automáticamente.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4. En caso que se dejen campos vacíos el sistema muestra como resultado un mensaje informando el error.</p>
Flujo Normal de Eventos	
Sección “Modificar Servidor”	
Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso inicia cuando el Administrador del Servidor selecciona la opción de “Configurar Datos del Servidor”.</p>	<p>2. El sistema muestra un formulario donde se encuentran todos los datos guardados del servidor como:</p> <ul style="list-style-type: none"> -Nombre -Dirección IP -URL de Publicación -Ubicación de las medias
<p>3. El administrador del servidor introduce los nuevos datos que desea cambiar y presiona el botón Guardar.</p>	<p>4. El sistema comprueba que estén correctos todos los datos que se han modificado. Y finaliza guardando los datos de la nueva modificación.</p>
<i>Prototipo de Interfaz</i>	

CAPÍTULO 3: Presentación de la solución propuesta



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4. En caso que se dejen campos vacíos el sistema mostrará como resultado un mensaje informando el error.

Flujo Normal de Eventos

Sección "Eliminar Servidor"

Acción del Actor	Respuesta del Sistema
1. El Administrador del Servidor selecciona la opción de "Configurar Datos del Servidor".	2. El sistema muestra el formulario donde se encuentran los datos del servidor y además el botón para eliminar el servidor.
3. El Administrador del Servidor presiona el botón de "Eliminar Servidor"	4. El sistema muestra un mensaje informando para comprobar si realmente desea continuar con la eliminación del Servidor.
5. El administrador del Servidor presiona el botón "Si".	6. El sistema como resultado elimina los datos asociados al servidor.

CAPÍTULO 3: Presentación de la solución propuesta

Prototipo de Interfaz

Para eliminar el servidor del Catálogo

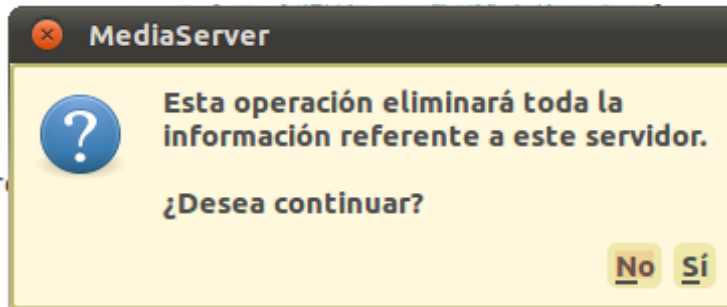
Datos del servidor

Nombre: video-web
Ip: 10.34.16.128
Url: http://inter-nos/publicaciones
Medias: Escritorio/watcher/Medias ...

eliminar servidor

Datos de la Conexión

Base de Datos: MediaServer
Host: localhost
Puerto: 5432
Usuario: postgres
Contraseña: [oculto]



Flujos Alternos

Acción del Actor	Respuesta del Sistema
4. El administrador presiona el botón "No"	5. El sistema como respuesta cancela la eliminación del servidor y finaliza retornando a la ventana de configurar los datos del servidor.

CAPÍTULO 3: Presentación de la solución propuesta

Tabla 5: Descripción textual del caso de uso Gestionar Media.

Caso de Uso:	Gestionar Media	
Actor:	Administrador del Servidor.	
Propósito:	Este caso de uso se lleva a cabo con el objetivo de insertar, modificar o eliminar una media en el catálogo.	
Resumen:	El caso de uso inicia cuando el Administrador del Servidor inserta, modifica o elimina una media en la carpeta de publicación, el sistema captura automáticamente la información de la media y la guarda en el catálogo.	
Precondiciones:	<p>Debe existir una carpeta de publicación donde estarán todas las medias asociadas al servidor.</p> <p>Solo se agregarán a esta carpeta archivos de audio o video.</p> <p>Solo se modificarán datos como el nombre de la media.</p>	
Referencias	RF6, RF7, RF8	
Prioridad	Media	
Flujo Normal de Eventos		
Sección “Insertar Media”		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el administrador del servidor agrega una media a la carpeta de publicación.	2. El sistema detecta la nueva media y guarda la información referente a: <ul style="list-style-type: none"> -Nombre. -Extensión. -Ubicación. -Dirección de Publicación. -ID Media. -ID Servidor. 	
<i>Prototipo de Interfaz</i>		
<p><i>Cuando se inserta, modifica o elimina una media de la carpeta de publicación la información de las medias se guarda en la base datos.</i></p>		

CAPÍTULO 3: Presentación de la solución propuesta

Ejemplo:

	id_media [PK] serial	nombre character vary	extension character varyi	ubicacion character \	url_publica character \	servidor_id integer
1	3173	03 - Tu la pagar	mp3	/home/geo	h03 - Tu la p	62
2	3174	03 Yo Vengo a C	wma	/home/geo	h03 Yo Veng	62
3	3177	ana	wma	/home/geo	hana.wma	62
4	3178	Ay, amor	wma	/home/geo	hAy, amor.v	62
5	3179	03 Yo Vengo a C	wma	/home/geo	hd/03 Yo Ve	62
6	3180	05	mp3	/home/geo	hd/05.Te Ar	62
7	3181	Ana Belén & Jo	Mp3	/home/geo	hDuos ocas	62
8	3182	Ana Belén & Ju	Mp3	/home/geo	hDuos ocas	62
9	3183	Ana Belén & Lec	Mp3	/home/geo	hDuos ocas	62
10	3184	Ana Belén & Vic	Mp3	/home/geo	hDuos ocas	62
11	3185	Ana Belén & Vic	Mp3	/home/geo	hDuos ocas	62
12	3227	Amor quédate (DAT	/media/d/L	http://serv	63
13	3228	Amor quédate	DAT	/media/d/L	http://serv	63
14	3229	Estes donde est	DAT	/media/d/L	http://serv	63

Flujos Alternos

Acción del Actor	Respuesta del Sistema
1. En caso que el actor agregue un archivo que no sea ni video o audio.	2. El sistema como resultado no detectará el archivo para cumplir que solo se agreguen formatos de audio o video.

Flujo Normal de Eventos

Sección "Modificar Media"

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Administrador del Servidor modifica una media que se encuentra en la carpeta de publicación.	2. El sistema detecta la nueva modificación y guarda los cambios.

Prototipo de Interfaz

La interfaz sería la carpeta de publicación donde estarán las medias del servidor de streaming y se modificarían los datos de las medias que se encuentren allí.

CAPÍTULO 3: Presentación de la solución propuesta

Post-Condiciones	Solo se podrán modificar datos como el nombre.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Flujo Normal de Eventos	
Sección “Eliminar Media”	
Acción del Actor	Respuesta del Sistema
1. El Administrador del Servidor Elimina una media de las que se encuentra en la carpeta de publicación.	2. El sistema detecta la media eliminada y como resultado la elimina de la base datos.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema

Tabla 6: Descripción textual del Caso de Uso Actualizar dirección del catálogo.

Caso de Uso:	Actualizar dirección del catálogo.
Actor:	Administrador General.
Propósito:	Este caso de uso actualiza la dirección del catálogo en caso de que cambie de lugar o de dirección IP la PC del administrador General que es donde va a estar el catálogo con el objetivo que los servidores que se encuentren inscritos conozcan la nueva dirección IP.
Resumen:	El caso de uso inicia cuando el Administrador General decide cambiar su dirección IP y accede al servicio actualizar dirección del catálogo. Luego introduce su nueva dirección IP para que los usuarios de los servidores que tiene inscrito en el catálogo conozcan la nueva dirección.
Precondiciones:	El usuario debe tener activado los permisos de administración para poder realizar la acción de cambiar la dirección IP del ordenador.
Referencias	RF9

CAPÍTULO 3: Presentación de la solución propuesta

Prioridad	Baja		
Flujo Normal de Eventos			
Acción del Actor		Respuesta del Sistema	
1. El actor selecciona la opción de Actualizar Dirección IP.		2. El sistema muestra un formulario donde se encuentran los datos necesarios para cambiar la dirección IP como: <ul style="list-style-type: none"> - Dirección IP - Máscara de red - Puerta de enlace 	
3. El actor llena los datos solicitados por el sistema y presiona el botón Guardar.		4. El sistema como respuesta actualiza la nueva dirección en el sistema y la guarda, además notifica a los servidores que estén inscritos en el catálogo su nueva dirección.	

Prototipo de Interfaz



Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4. En caso que falten campos por llenar el sistema mostrará como respuesta un mensaje informando el error.

3.6. Conclusiones

En este capítulo se presentó toda la información perteneciente al modelo de dominio, utilizando un lenguaje común para que los usuarios consigan entender el contexto en el que se desarrolla el sistema. Se describieron a los actores que participan y las actividades que estos ejecutan para brindar una idea general de las acciones que podrán realizar cada uno de ellos. Identificar los requisitos funcionales y no funcionales con los que debe contar el sistema que se va a desarrollar garantizó la solución del problema planteado y permitió posteriormente modelar el sistema basado en sus casos de uso. Y para finalizar se realizó la descripción de cada uno de los casos de uso, haciendo evidente la forma en la que queda estructurado el sistema a construir.

4. CAPÍTULO 4: Implementación y Prueba del sistema.

4.1. Introducción.

En el presente capítulo se expone todo lo referente a la construcción y prueba del sistema definiendo el modelo de diseño y los patrones utilizados para la implementación de la aplicación. El diseño de la base de datos para identificar las tablas que se necesitan para el funcionamiento de la solución propuesta, además de los diagramas de secuencia correspondientes a cada caso de uso descritos anteriormente. Se presenta también el Diagrama de despliegue y de componentes en la fase de implementación y para terminar se efectúan las pruebas de caja negra para comprobar el buen funcionamiento de la aplicación.

La aplicación consta de dos módulos:

- ✓ Catálogo: En esta se encuentra la base de datos y es donde se realiza la búsqueda de las medias que se encuentran en los servidores que estén inscritos en el catálogo.
- ✓ Notificador: Se instala en el servidor de streaming que se desea registrar en el catálogo. Esta aplicación notifica al Catálogo de los cambios que ocurren en el servidor (agregar o eliminar media, cambio de IP del servidor, eliminar registro del servidor, enviar disponibilidad del servidor).

4.2. Patrones de Diseño.

Los patrones de diseño son principios generales de soluciones que aplican ciertos estilos que ayudan a la creación del software. Es una descripción de un problema y la solución a la que le da el nombre y que se puede aplicar en nuevos contextos. Muchos patrones ayudan a asignar responsabilidades a los objetos.

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.”

En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Debemos tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (Tedeschi, 2011)

Tipos de patrones:

CAPÍTULO 4: Implementación y Prueba del sistema

- ✓ *Patrones arquitectónicos: estos patrones definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes del software y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, componentes, subsistemas) de la arquitectura.*
- ✓ *Patrones de diseño: estos patrones se aplican a un elemento específico del diseño como un agregado de componentes para resolver algún problema de diseño relaciones entre los componentes o los mecanismos para efectuar la comunicación de componente a componente.*
- ✓ *Idiomas: a veces llamados patrones de código, estos patrones específicos de lenguaje por lo general implementan un elemento algorítmico o un componente, un protocolo de interfaz específico o un mecanismo de comunicación entre los componentes. (Barrera, 2006)*

Clasificación de los patrones de diseño:

- ✓ **Patrones Creacionales:** Inicialización y configuración de objetos.
- ✓ **Patrones Estructurales:** Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.
- ✓ **Patrones de Comportamiento:** Más que describir objetos o clases, describen la comunicación entre ellos.

Patrones GRASP (General Responsibility Assignment Software Patterns)

- **Experto:** Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele ser útil en el diseño orientado a objetos. El patrón experto es utilizado en casi todas las clases de la aplicación, porque se hacen clases diferentes de acuerdo a las funcionalidades que van a tener y a la información que van a manejar, estas clases son expertas en esa información y por lo tanto las más indicadas para manejarla.
- **Creador:** Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, da soporte al bajo acoplamiento. Este patrón es utilizado al crear objetos en la aplicación.

CAPÍTULO 4: Implementación y Prueba del sistema

- **Bajo Acoplamiento:** El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con qué las conoce y con qué recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases.
- **Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas, la alta cohesión simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización.

4.3. Modelo de diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales tienen impacto en el sistema. Sirve como abstracción de la implementación del sistema.

Diagrama de clases del diseño.

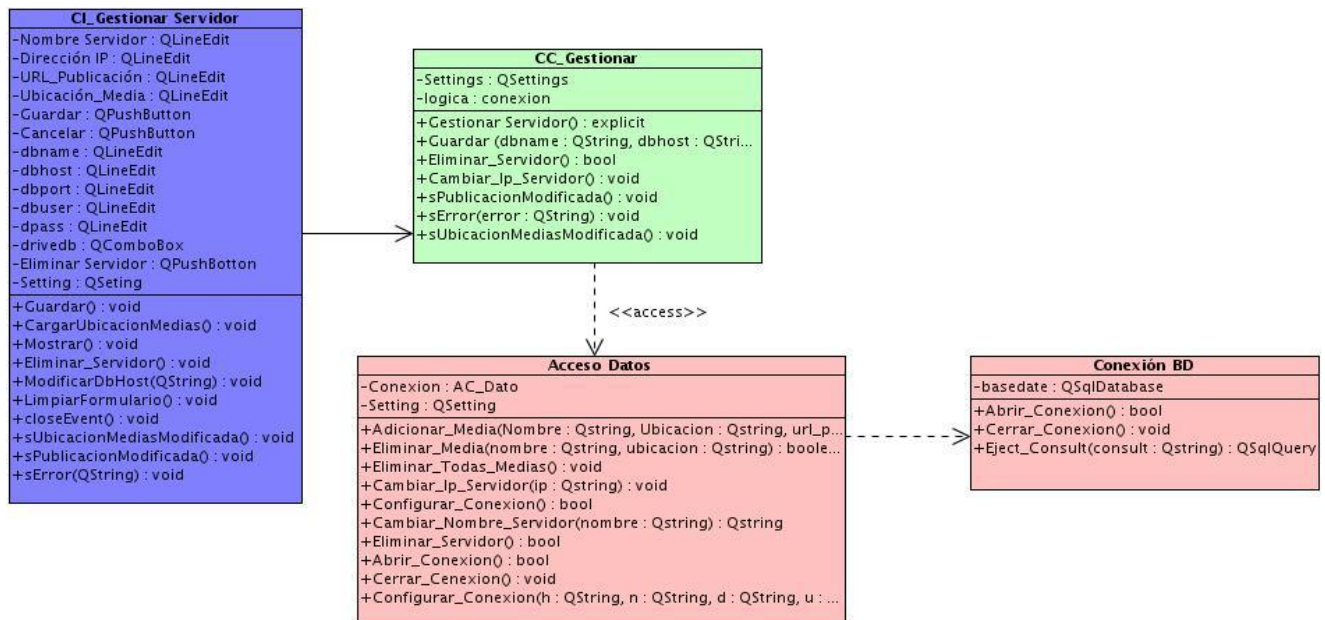


Ilustración 6 : Diagrama de clases del diseño para el CU Gestionar Servidor.

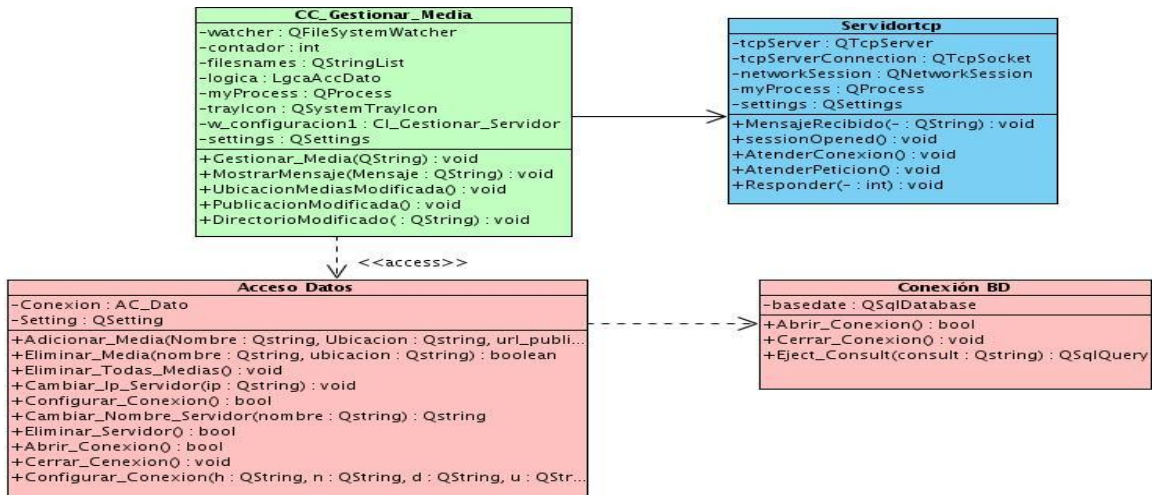


Ilustración 7: Diagrama de clases del diseño para el CU Gestionar Media.

4.4. Diseño de la base de datos.

Los sistemas de base de datos se diseñan para manejar grandes cantidades de información, la manipulación de los datos involucra tanto la definición de estructuras para el almacenamiento de la información como la provisión de mecanismos para la manipulación de la información, además un sistema de base de datos debe de tener implementados mecanismos de seguridad que garanticen la integridad de la información, a pesar de caídas del sistema o intentos de accesos no autorizados.

Para diseñar la base de datos es necesario tener en cuenta un conjunto de pasos que comienzan con definir las clases persistentes, luego refinarlas y clasificarlas junto con sus atributos, para más tarde realizar el diagrama de clases persistentes. Realizar el diagrama de transición de estado es el siguiente paso para diseñar la base de datos, y el último es la conversión de las clases al medio de almacenamiento.

4.4.1. Diagrama de clases persistentes.

Mediante la persistencia se puede lograr que un objeto pueda mantener su valor en el espacio y en el tiempo. Es responsabilidad del diseñador definir cuáles son las clases persistentes entre aquellas que se han definido.

CAPÍTULO 4: Implementación y Prueba del sistema



Ilustración 8: Diagrama de Clases Persistentes.

4.4.2. Diagrama entidad relación.

A continuación se muestra el modelo lógico de la base datos en el diagrama Entidad-Relación.

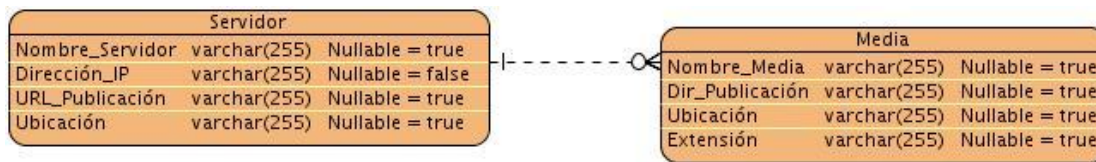


Ilustración 98: Diagrama Entidad Relación.

4.5. Diagrama de secuencia.

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia muestran gráficamente las interacciones del actor y de las operaciones a que dan origen.

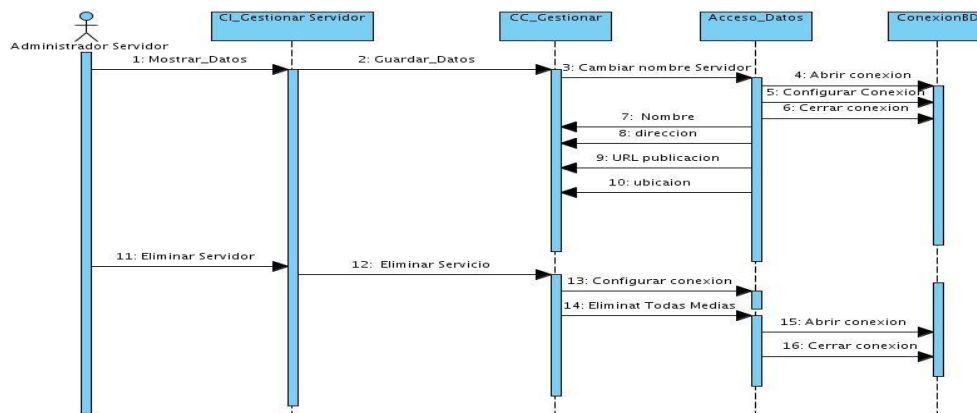


Ilustración 90: Diagrama de secuencia del CU Gestionar Servidor.

CAPÍTULO 4: Implementación y Prueba del sistema

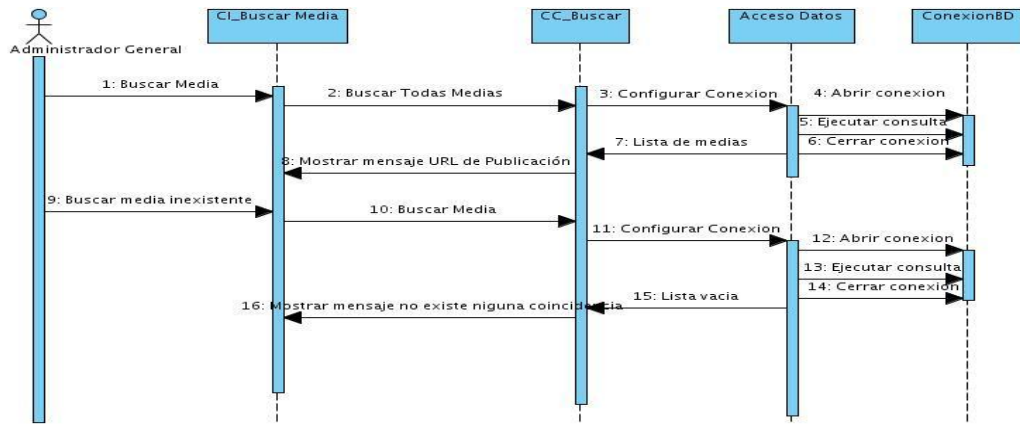


Ilustración 101: Diagrama de secuencia del CU Buscar Media.

4.6. Modelo de despliegue.

El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. El modelo consiste en uno o más nodos, dispositivos, y conectores entre estos.

Los nodos son los elementos donde se ejecutan los componentes. Los componentes son los elementos que participan en la ejecución de un sistema. Los componentes representan el empaquetamiento físico de los elementos lógicos. Los nodos representan el despliegue físico de los componentes. La relación entre un nodo y el componente que despliega puede mostrarse con una relación de dependencia, o listando los nodos desplegados en un compartimiento adicional dentro del nodo.

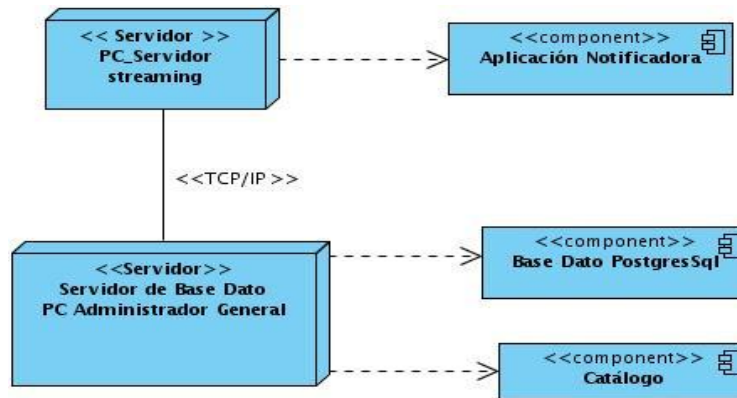


Ilustración 12 : Diagrama de Despliegue.

4.6.1. Descripción de los Nodos.

A continuación se describen los nodos físicos con sus componentes específicos representados en el diagrama de despliegue.



Ilustración 13: Nodo Servidor de la PC Servidor Streaming.

Este nodo representa una PC donde se encuentra el servidor de streaming que va a proporcionar los datos de las medias que se encuentran en su servidor.

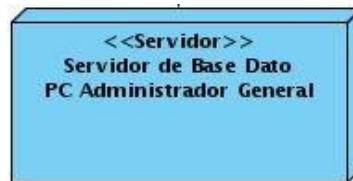


Ilustración 1411: Nodo Servidor de la PC Administrador General.

Este nodo representa una PC donde se encuentra la aplicación del catálogo que va a estar manejada por un administrador general y además donde va a estar la base datos de la aplicación.

4.7. Modelo de implementación.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de módulos y mostrar las relaciones entre los elementos de implementación. Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Un componente puede comprender una gran porción de un sistema.

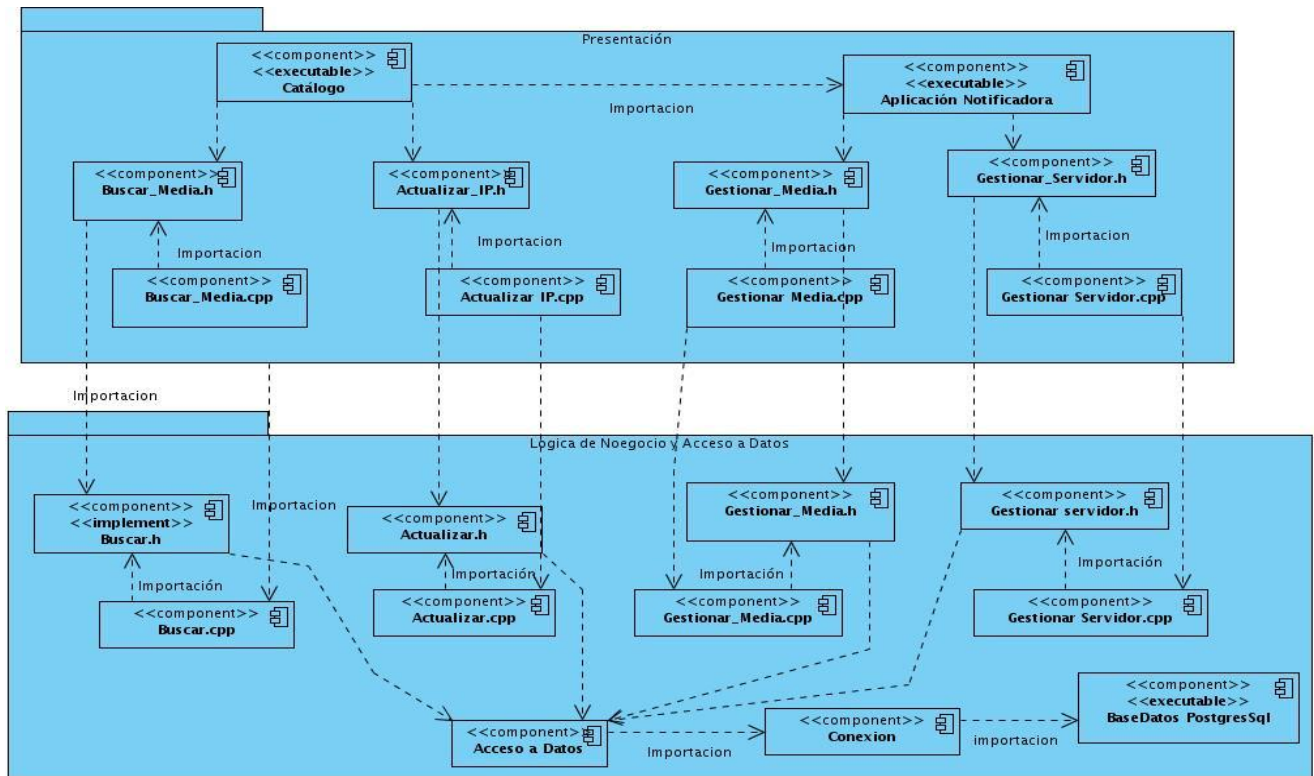


Ilustración 1512: Diagrama de Componentes.

4.8. Pruebas de funcionalidad.

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Básicamente es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas. Estas pruebas se integran dentro de las diferentes fases del ciclo de desarrollo del software, donde se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene, para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento de las especificaciones iniciales del sistema.

Existen diferentes tipos de pruebas: unitarias, de integración, del sistema y de aceptación. La prueba del sistema forma parte de las pruebas de Caja Negra y permite comprobar que se cumplan los requerimientos funcionales.

4.8.1. Prueba de Caja Negra.

La técnica de prueba de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

Algunos de los métodos empleados en las pruebas de caja negra son los siguientes:

- ✓ Métodos de prueba basados en grafos.
- ✓ Partición equivalente.
- ✓ Análisis de valores límite.
- ✓ Prueba de la tabla ortogonal.
- ✓ Adivinando el error.

Para realizar dichas pruebas al sistema se utilizó el método de partición equivalente el cual presenta las siguientes características:

La partición equivalente se presenta como un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. (Pressman, 2002)

El objetivo de partición equivalente es reducir el posible conjunto de casos de prueba en uno más pequeño, un conjunto manejable que evalúe bien el software. Se toma un riesgo porque se escoge no probar todo. Así que se necesita tener mucho cuidado al escoger las clases. (Patton, 2005)

En el diseño de casos de prueba para partición equivalente se procede en dos pasos (Myers, 2004):

1. Se identifican las clases de equivalencia. Las clases de equivalencia son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos.

CAPÍTULO 4: Implementación y Prueba del sistema

Es de notar que dos tipos de clases de equivalencia están identificados: las clases de equivalencia válidas representan entradas válidas al programa, y las clases de equivalencia inválidas que representan el resto de los estados posibles de la condición (es decir, valores erróneos de la entrada).

2. Se define los casos de prueba. El segundo paso es el uso de las clases de equivalencia para identificar los casos de prueba.

En la siguiente tabla se muestran las descripciones de los casos de prueba empleados durante la aplicación del método de partición equivalente:

Tabla 7: Descripciones de los casos de prueba.

Nombre del caso de uso.	Escenarios por secciones	Respuesta del sistema	Resultado de la prueba
C1: Buscar Media	EC 1.1: Buscar Media Satisfactoriamente.	Devuelve la Url de publicación de la media que deseaba.	Satisfactorio
	EC 1.2: Nombre de la media no encontrado.	Muestra un mensaje informando que la media no fue encontrada.	Satisfactorio
	EC1.3: Media encontrada en varios servidores.	Se muestra la dirección IP de cada uno de los servidores donde fue encontrada la media, la cantidad de conexiones que tiene cada uno de los servidores y la Url de publicación.	Satisfactorio
	EC 1.4: Buscar Media con campo vacío.	Muestra un mensaje de error porque no puede estar el campo vacío.	Satisfactorio

CAPÍTULO 4: Implementación y Prueba del sistema

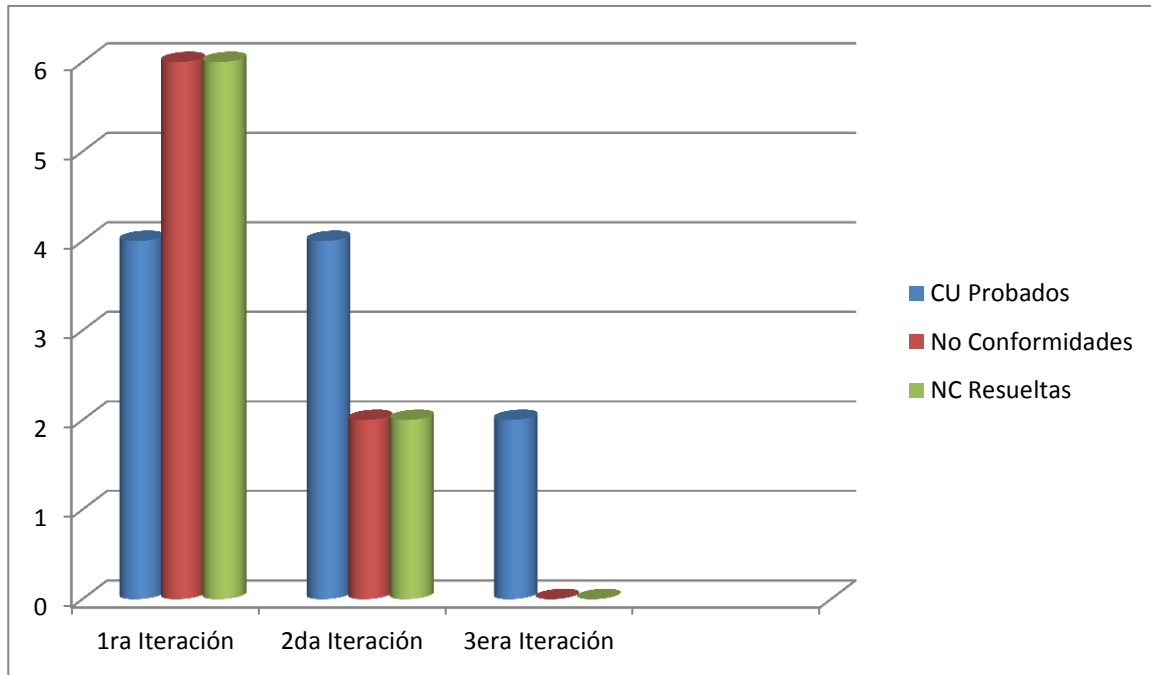
C2: Gestionar Media	EC 2.1: Insertar Media.	2.1.1: Insertar Media con éxito.	Se inserta correctamente la media en la base datos.	Satisfactorio
		2.1.2: Insertar media sin formato adecuado.	No inserta la media en la base datos.	Satisfactorio con recomendaciones
	EC 2.2: Modificar Media.		Se modifican correctamente los datos de la media.	Satisfactorio
	EC 2.3: Eliminar Media.		Se elimina la media de la base datos.	Satisfactorio
C 3: Gestionar Servidor	EC 3.1: Insertar Servidor.	3.1.1: Insertar Servidor con éxito.	Se inserta correctamente el servidor.	Satisfactorio
		3.1.2: Insertar con campos vacíos.	Error, no puede dejar campos vacíos.	Satisfactorio
	EC 3.2: Modificar Servidor.	3.2.1: Modificar datos con éxito.	Se modifican correctamente los datos.	Satisfactorio
		3.2.2: Modificar datos dejando campos vacíos.	Mostrar mensaje de error, no puede dejar campos vacíos.	Satisfactorio
	EC 3.3: Eliminar Servidor.	3.3.1: Eliminar Servidor con éxito.	Se eliminan todos los datos referentes al servidor.	Satisfactorio
		3.3.2: Eliminar servidor cancelando la eliminación.	Regresar a la aplicación sin borrar ningún dato del servidor.	Satisfactorio

Resultados de Prueba de Caja Negra.

En la siguiente tabla se exponen los resultados obtenidos durante el proceso de realización de las pruebas de caja negra. Dicho proceso fue realizado en tres iteraciones de prueba donde se muestran los Casos de

CAPÍTULO 4: Implementación y Prueba del sistema

Uso Probados, las No conformidades que fueron arrojadas y las No Conformidades que fueron resueltas para lograr un exitoso funcionamiento de la aplicación.



Gráfica 1: Resultados de Pruebas de Caja Negra

4.8.2. Lista de chequeo tolerancia a fallo.

Para evaluar el comportamiento ante la posibilidad de fallos de la aplicación se realizó las pruebas del sistema utilizando la lista de chequeo de tolerancia a fallo propuesta por el Centro de Calidad para Soluciones Informáticas (**Calisoft**) de la facultad 6 en la Universidad de Ciencias Informáticas.

El objetivo general de la lista de chequeo es evaluar cómo actúan los sistemas ante la tolerancia de fallos en los proyectos de la universidad. Esta plantilla ha sido confeccionada para guiar a desarrolladores, especialistas o expertos técnicos en la verificación y evaluación de los sistemas ante la tolerancia de fallos en los proyectos de la universidad. Permitirá además recoger los puntos eficientes y los ineficientes que tienen los elementos chequeados. En la tabla 8 se exponen algunos de estos elementos:

CAPÍTULO 4: Implementación y Prueba del sistema

Tabla 8: Lista de chequeo tolerancia a fallo.

Elementos relacionados con el Sistema				
Peso	Indicadores a Evaluar	NP	Respuesta Esperada	Respuesta Final
crítico	1. ¿Qué respuesta brinda el sistema al reanudarse las operaciones luego de interrumpir la energía a la PC cliente?(Operaciones: Enviar o actualizar información al/del servidor)		El cliente debe reanudar la aplicación para poder utilizarla.	Una vez encendido el ordenador el cliente puede ejecutar la aplicación nuevamente.
crítico	2. ¿Qué respuesta brinda el sistema al reanudarse las operaciones luego de interrumpir la energía de la PC del servidor? (Operaciones: Enviar información a la PC cliente)		Deben quedar guardados los datos y los cambios que ocurrieron antes de ocurrir la interrupción.	Los datos del servidor quedan guardados en la base datos, pero si no activa nuevamente la aplicación el servidor no será identificado cuando se realice la búsqueda de medias.
crítico	3. ¿Qué respuesta brinda el sistema al reanudarse las operaciones luego de interrumpir las comunicaciones de red en la PC cliente?(Operaciones: Enviar o		Debe mostrar un mensaje que no hay conexión con la base dato y por tanto no se puede realizar la búsqueda.	Muestra un mensaje de error pues no hay conexión con la base datos.

CAPÍTULO 4: Implementación y Prueba del sistema

	actualizar información al/del servidor)			
--	-----------------------------------------	--	--	--

Resultados de pruebas al sistema.

Luego de realizar las pruebas de caja negra utilizando el método de partición equivalente y probar los posibles fallos que se proponen en la tabla anterior se puede concluir que el sistema reacciona de la forma que se esperaba garantizando que se cumplan los requerimientos propuestos. Se identificaron además mediante las pruebas algunas recomendaciones para un futuro perfeccionamiento de la aplicación. En general se obtuvo una aplicación con un alto valor práctico permitiendo realizar búsquedas de medias en los servidores de streaming.

4.9. Conclusiones

Durante el capítulo se expusieron los diferentes diagramas de clases correspondientes al diseño los cuales contribuyeron a una adecuada implementación del sistema. Se abordaron las generalidades para el desarrollo de la aplicación, teniendo en cuenta el diseño de la base de datos y los componentes utilizados, así como las relaciones entre ellos a través del modelo de implementación. Con la creación del modelo de despliegue se obtuvo una vista de la distribución física del sistema para lograr una mayor explotación de sus funcionalidades. Se realizó la descripción de los casos de prueba comprobando que la aplicación cumple satisfactoriamente con los requerimientos definidos en el flujo de trabajo análisis y diseño.

CONCLUSIONES GENERALES

Luego de culminar todo el proceso de construcción de la herramienta que permite centralizar la información multimedia de los servidores de streaming dando solución al ineficiente control y búsqueda de información en dichos servidores, se arriban a las siguientes conclusiones:

- ✓ La investigación ha pasado por los diferentes flujos de trabajo que propone la metodología de desarrollo RUP para llegar a la propuesta final de la solución y garantizar que cumpla con los requerimientos establecidos.
- ✓ Las tecnologías y herramientas utilizadas para la construcción del sistema contribuyeron a lograr el éxito de la aplicación garantizando que funcione según las necesidades actuales de la Universidad ya que fue creada para trabajar en un ambiente de software libre.
- ✓ Como resultado se obtuvieron dos aplicaciones que mediante ellas se logran realizar búsquedas de medias en servidores de streaming ofreciendo las funcionalidades necesarias para garantizar que el catálogo se mantenga actualizado y de esta manera asegurar las búsquedas.

La herramienta desarrollada como cumplimiento al objetivo general de la investigación:

- ✓ Permite insertar, modificar o eliminar un servidor al catálogo garantizando la entrada de medias a la base datos.
- ✓ Permite realizar búsqueda de medias en servidores de streaming insertando el nombre de la media que desea buscar.
- ✓ Garantiza que el proceso de búsqueda se realice teniendo en cuenta la cantidad de conexiones que tienen los servidores en ese momento para informar al usuario y evitar la sobrecarga de conexiones en el servidor.
- ✓ Permite cambiar la dirección IP del catálogo e informar el cambio a los servidores para asegurar el buen funcionamiento de las conexiones entre las dos aplicaciones.

RECOMENDACIONES

Partiendo de la experiencia acumulada durante todo el período de desarrollo de la investigación y de los resultados obtenidos, se proponen las siguientes recomendaciones:

- ✓ Para lograr una mejor comprensión y utilización del sistema se recomienda la realización de un manual o ayuda.
- ✓ Restringir dinámicamente la extensión de las medias que se van a publicar a través de la definición de estas en un XML para asegurar que solo existan archivos de audio y video.
- ✓ Sería conveniente implementar técnicas de autenticación para la seguridad de la aplicación.
- ✓ Debido a que esta es la primera versión de la aplicación se recomienda seguir estudiando sobre el tema y así perfeccionarla.

5. Bibliografía

- Agüera, José Ramón Pérez. 2004. *ADAPTACIÓN DE TECNOLOGÍAS STREAM Y XML A CENTROS DE DOCUMENTACIÓN Y TELEVISIÓN*. 2004.
- Alejandro Cáceres, Emmanuel Fontán, Agustín Pizzio, Leandro Tisocco, Hernán Vallejos. 2010. *Metodología de Desarrollo de Software basada en componentes*. 2010.
- Alvarez, Miguel Angel. 2001. *desarrolloweb.com*. [En línea] 9 de 7 de 2001. [Citado el: 10 de 11 de 2010.] <http://www.desarrolloweb.com/articulos/482.php>.
- Angelina, Cleto Martín. 2010. *IcePick: Compilador modular extensible para desarrollo de*. UNIVERSIDAD DE CASTILLA-LA MANCHA : s.n., 2010.
- Barker, Joe. 1995. *UNA APROXIMACIÓN A INTERNET Y A SUS HERRAMIENTAS DE BÚSQUEDA*. Universidad de California, Berkeley : s.n., 1995.
- Barrera, Pedro Francisco Godoy. 2006. *Universidad Técnica Federico Santa María. Ramos en Línea*. [En línea] 16 de 11 de 2006. [Citado el: 28 de 4 de 2011.] www.ramos.utfsm.cl/doc/1125/sc/Clase2.ppt.
- Bermúdez, Carlos. 2010. *dgtallika 30/8/2010*. [En línea] 30 de 8 de 2010. [Citado el: 5 de 12 de 2010.] <http://www.dgtallika.com/2010/08/middleware-definicion-de-hoy/>.
- Boldrito, Remo Suppi. 2004. *Admission Control and Media Delivery Subsystems for Video on Demand*. Barcelona, España : s.n., 2004.
- Cambiaso, Diego. 2010. *Pixelco blog*. [En línea] 10 de 06 de 2010. [Citado el: 16 de 1 de 2011.] <http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>.
- Cavsi. 2011. *Cavsi. Computer Audio Video Systems Integrator*. [En línea] 2011. [Citado el: 9 de 02 de 2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
- CORBA. Saldivia, César Guerrero. 1999. 1999.
- Córdova, Fernando García. 2004. *La tesis y el trabajo de tesis: Recomendaciones Metodológicas para la elaboración de los trabajos de tesis*. México : Limusa, 2004.
- Cruz, Maria del Carmen Gomez. 2010. *Estudio del Streaming de Audio y Video sobre redes heterogeneas*. Madrid- España : s.n., 2010.
- Fernández, David Vallejo. 2006. *Documentación de ZeroC ICE*. 2006.

BIBLIOGRAFÍA

- Garcia, Joaquin. 2003. *IngenieroSoftware*. [En línea] 2003. [Citado el: 5 de 5 de 2011.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
- 1997-1999 . Gonzalo Álvarez Marañón . *Java*. [En línea] CSIC. Todos los derechos reservados, 1997-1999 . [Citado el: 10 de 1 de 2011.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
- IguanaHosting. 2009. *IguanaHosting*. [En línea] 30 de 07 de 2009. [Citado el: 15 de 11 de 2010.] <http://news.iguanahosting.com/ventajas-y-desventajas-del-streaming/>.
- Inforeco. 2002. *Inforeco*. [En línea] 2002. [Citado el: 12 de 3 de 2010.] <http://www.ucm.es/BUCM/cee/inforeco/InforEcoFlash/1/1104.htm>.
- Jacobson, Ivan. 2000. *El proceso Unificado de desarrollo de software*. Madrid : s.n., 2000.
- Luis Giraldo, Yuliana Zapata. 2005. *Herramientas de Desarrollo de Ingeniería de SW para Linux*. 2005.
- Marañón, Gonzalo Álvarez. 1997-1999. *Criptomicón*. *Java*. [En línea] CSIC. Todos los derechos reservados, 1997-1999. [Citado el: 10 de 1 de 2011.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
- Martínez, Luis F. Iribarne. 2003. *Un Modelo de Mediación para el desarrollo de Software Basado en Componentes COTS*. España : s.n., 2003.
- Mnemmix, Grupo. 2008. *Mirame*. [En línea] Septiembre de 2008. [Citado el: 24 de Octubre de 2010.] www.mirame.tv.
- Monroy, José Arturo García. *Globus Toolkit*. Madrid : s.n.
- Mora, Sergio Luján. 2006. *C++ paso a paso*. 2006.
- —. *C++ paso a paso*.
- Myers, Glenford J. 2004. *The art of sotware Testing*. 2004.
- Network, Real. 2010. *Helix Proxy*. [En línea] 2010. [Citado el: 11 de Noviembre de 2010.] <http://www.realnworks.com/products-services/helix-server-proxy.aspx>.
- NNova, Grupo. 2007. *Grupo de soluciones GSI NNova*. [En línea] 2007. [Citado el: 10 de 12 de 2010.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
- Paradigm, Visual. 2004. *Increase productivity, communication, and collaboration using UML visual modeling plataform*. [En línea] 2004. [Citado el: 10 de 12 de 2010.] <http://www.visual-paradigm.com/product/vpuml>.
- Patton, Ron. 2005. *Software Testing*. 2005.

BIBLIOGRAFÍA

- Pereda, Héctor Fernández. 2008. *Portal Servicios Información Gratuitos Buscadores y Profesionales*. [En línea] 5 de 4 de 2008. [Citado el: 20 de 1 de 2011.] http://www.buscarportal.com/articulos/motores_busqueda.html.
- Pérez, José Carlos Cortizo. *Programación Concurrente y Distribuida*.
- Pixelco blog. [En línea] [Citado el: 16 de 1 de 2011.] <http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>.
- Prat, Andres. 2004. *Microsoft Solution Framework*. [En línea] 7 de 6 de 2004. [Citado el: 26 de 2 de 2011.] www.e-market.cl/dir/umayor/ingsw/06-01_vesp/msf.ppt.
- Pressman, Roger S. 2002. *Ingeniería de Software, Un enfoque práctico*. s.l.: 5ta Edición McGRAW-HILL, 2002.
- Proveda, Jordy. 2008. [En línea] 10 de Abril de 2008. <http://www.suespacio.net/blog/2008/04/10/que-es-streaming-concepto-historia-evolucion-y-futuro/>.
- Sanchez, Maria A. Mendoza. 2004. *Metodologías de desarrollo de Software*. 2004.
- Sciara, Daniel Rijo. 2004. *Fundamentos de Video Streaming*. Montevideo, Uruguay : s.n., 2004.
- Seco, Jose Antonio González. 2001. *Intorucción a C#*. [En línea] 2001. [Citado el: 18 de 12 de 2010.] http://programmatium.blogspot.com/2008/01/el-lenguaje-de-programacin-c_23.html.
- Slater, Silvina. *BÚSQUEDAS BIBLIOGRÁFICAS*. [En línea] 15-4188-5707. [Citado el: 20 de 1 de 2011.] <http://www.silvinaslater.com.ar/preguntasfrecuentes.htm>.
- *Streaming media proxy and caching solutions for Windows Media, Real Video/Audio and Quick Time streams*. [En línea] [Citado el: 13 de Noviembre de 2010.] <http://all-streaming-media.com/streaming-media-faq/>.
- StreamingMedia. 2000. *StreamingMedia.com*. [En línea] 25 de Enero de 2000. [Citado el: 15 de Noviembre de 2010.] <http://www.streamingmedia.com/Articles/News/Featured-News/Network-Appliance-Introduces-NetCache-Streaming-Media-Cache-61775.aspx>.
- *Supercomputadoras basadas en "cluster" de PCs*. Diego H. Milone, Adrian A. Azar, Leonardo H. Rufiner. 2002. 25, 2002, Vol. XIII.
- Taboexa, Grupo Empresas. 2007. *Buscadores-ABC*. [En línea] Grupo Empresas Taboexa, 3 de 5 de 2007. [Citado el: 13 de 6 de 2011.] <http://www.buscadores-abc.com/TiposBuscadores.htm>.
- Tedeschi, Nicolás. 2011. *Developers. msdn*. [En línea] 2011. [Citado el: 27 de 4 de 2011.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.