

**Universidad de las Ciencias Informáticas  
Facultad 9**



**Implementación de los Subsistemas de  
Transmisión y Administración de la Transmisión  
en la Plataforma de Transmisión Abierta para  
Radio y Televisión.**

---

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

**Autores:**

**Yumar García Duarte  
Jorge Rodríguez Valdés**

**Tutor:**

**Ing. Yandy León Nuñez**

**Ciudad de La Habana, 26 de junio de 2010.**

**“Año 52 de la Revolución”**



# DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

***Implementación de los Subsistemas Transmisión y Administración de la Transmisión en la Plataforma de Transmisión Abierta para Radio y Televisión.***

Autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Yumar García Duarte  
[Autor]

\_\_\_\_\_  
Jorge Rodríguez Valdés  
[Autor]

\_\_\_\_\_  
Yandy León Núñez  
[Tutor]

## DEDICATORIA

*A mis abuelos Chiche, Cuñin y Haidee que se sentirían orgullosos de verme ingeniero.*

*A mis padres Laura y Orestes por ser mis primeros maestros y hacer de mí la persona que soy. Por el amor infinito y la confianza que siempre me han dado.*

*Yumar.*

*A Dios.*

*Al amor de mi vida, Yamy.*

*A mis padres, abuelos y familia.*

*A todos los que estuvieron y están a mi lado.*

*Jorge*

# AGRADECIMIENTOS

*Agradecerle a la Revolución y a Fidel por esta oportunidad de poder formarme como profesional.*

*A los muchachos de UCITeVe por ser como una gran familia.*

*A los miembros del proyecto PTAR TV que nos apoyaron siempre.*

*Al tribunal por todas sus sugerencias y recomendaciones.*

*A los tutores Yandy y Dunier que confiaron siempre en nosotros.*

*A mis tíos Alex, Candito, Orelvis (Pillo) por ser especiales conmigo.*

*A mi hermano Doimer por todo su cariño.*

*A mi abuela Esther y mi abuelo Orestes por tenerme siempre presente y quererme tanto.*

*A mi novia Mercedes, una mujer extraordinaria que llegó a mi vida en el tiempo indicado y que con su amor me ha ayudado a esforzarme y a seguir adelante. Gracias por todo tu apoyo mi niña.*

*Al Mello por ser más que amigo, hermano.*

*A mis amigos de toda la vida Joise y Jorge Osmani.*

*A Anay mi única amiga.*

*A mis amigos de la universidad Pepe, Yosiel el Puma, Falero el timbalero, Gustavo el Faly, Pacheco el enamorado de la luna, Olvier, Yanoisis mi hijo del tenis, Dennis el salvaje, a Yidian, los hermanos que me dio la vida.*

*A Ari (la luna), Kare, lili, Yuya, Dayi, Margelís, Suyen, Yake, Isa, Lisa, Zule, Dayi, Maggei, la china, Yaillet por pertenecer a ese grupo que nunca voy a olvidar.*

*A los que no pudieron terminar junto a nosotros Elizabeth, Yusniel, Adrian, Yunior y Fabián nunca los voy a olvidar esta tesis también es de ustedes.*

*A todos los profesores que pusieron su granito de arena en mi formación.*

*A todas las personas de mi pequeño pueblo Arimao que siempre estuvieron al tanto de mis resultados.*

*A todos Muchas gracias de todo corazón... Yumar*

*A Dios por darme las fuerzas, el apoyo y estar siempre a mi lado en todo en la vida.*

*Al amor de mi vida Yamila, por estar siempre a mi lado en los momentos difíciles y darme la fuerza para seguir adelante. Gracias mi niña linda. Te amo y esta Tesis también es tuya.*

*A mis padres Mayra y Jorge Luís, mis abuelos Panchy y mami Norma y mi bisabuela mama alla por educarme, apoyarme y siempre guiarme por el camino correcto. Gracias por su esfuerzo y apoyo, los quiero muchísimo.*

*A mi segunda mamá Belkis por ser mi médico, mi amiga y mi madre. Gracias por tu esfuerzo, sacrificio y tenerme siempre presente como un hijo más. Gracias a Héctor por todo su apoyo al igual que Hectín y Humbe por ser tan buenos hermanos.*

*A mis suegros, Adys y Armando, por todo el sacrificio que hicieron por mí durante estos años, a ustedes también les debo gran parte de lo que hoy soy.*

*A mis cuñados Liudmila y Michael por todo su apoyo para seguir adelante.*

*A tía Margarita por siempre tenerme presente en todo momento.*

*A mi tutor el Ingeniero Yandy León Núñez por su dedicación y esfuerzo para la terminación de este trabajo. Gracias mi hermano por tu ayuda y sacrificio.*

*A mi Jefe de Proyecto Dunier Domínguez por su dedicación, esfuerzo y comprensión para darle solución a cada tarea del presente trabajo. Gracias mi hermano por tu colaboración y abnegación.*

*A mi compañero de tesis Yumar García Duarte por siempre estar a mi lado y junto lograr este trabajo.*

*A mi compañero y amigo Renier Martínez por ayudarme en la confección de este trabajo. Gracias mi hermano por tu ayuda incondicional.*

*A mis grandes amigos Rubisel, Nelson y Machín por su apoyo durante estos años. Gracias mis hermanos.*

*A mis compañeros y amigos que durante estos cinco años, me brindaron su apoyo incondicional en cada obstáculo que se presentara. A todos ustedes muchas gracias.*

*A los profesores que me ayudaron durante estos años a formarme como profesional y brindarme su mano amiga. Gracias a todos.*

---

## **RESUMEN**

Los medios de comunicación son de extrema importancia para el desarrollo, entretenimiento, y educación de las sociedades, en mayor o menor grado según el nivel de desarrollo que se tenga. Es así como la televisión en los últimos años del siglo XX se ha convertido en un importante medio de difusión masiva. En la actualidad es normal ser un televidente habitual, sin embargo es difícil imaginar el volumen de trabajo que se halla detrás de esa pequeña pantalla. Un gran número de personas desempeña cada día los más diversos roles para posibilitar que ese fenómeno audiovisual llegue a distintos hogares. Disímiles son las actividades realizadas manualmente para facilitar las transmisiones televisivas, esto conlleva un gran cúmulo de trabajo humano.

El Centro de Geoinformática y Señales Digitales de la Universidad de las Ciencias Informáticas ha detectado que en Cuba el proceso de transmisión de contenidos multimedia se realiza de forma tal que es necesario un elevado número de especialistas, con el inconveniente de realizar este proceso de forma prácticamente manual, lo que trae consigo pérdida de tiempo productivo, a lo que se puede adicionar la posibilidad de errores humanos que pueden hacer colapsar todo el proceso.

El objetivo del presente es desarrollar un subsistema que permita la transmisión y administración de contenidos multimedia automatizando una gran parte del proceso e integrándose con el resto de los subsistemas que conforman la Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV).

Este documento recoge los resultados de la investigación realizada en el estudio y análisis de tecnologías, técnicas de programación y herramientas necesarias para el desarrollo de los subsistemas. Se describen las funcionalidades de las aplicaciones mostrándose a través del modelo de implementación y de los Diagramas de Componentes. Luego de la programación de las aplicaciones se prueban sus principales funcionalidades

## **Palabra claves**

Televisión, Plataforma, Transmisión, Administración, Streaming, Audio, Video.

## **ABSTRACT**

The media are extremely important for development, entertainment, and education societies, to a greater or lesser degree depending on the level of development you have. This is how television last years of the twentieth century has become an important means of dissemination. It is now normal to be a regular viewer, but it is difficult to imagine the amount of work that lies behind that little screen. A large number of people playing every day the most diverse roles to enable audiovisual phenomenon reaches across households. Dissimilar activities are performed manually to facilitate television broadcasts; this involves a great deal of human labor.

The Center for Geoinformatics and Digital Signals University of Information Sciences in Cuba has detected that the process of transmission of multimedia content is done in a way that requires a large number of specialists, with the disadvantage of making this process so virtually manual, which entails loss of productive time, to which you can add the possibility of human error that can bring down the whole process.

The purpose of this is to develop a subsystem that allows the transmission and multimedia content management by automating a large part of the process and integrating with the rest of the subsystems that make up the Open Platform for Radio Broadcasting and Television (PTARTV).

This document contains the results of research in the study and analysis of technologies, programming techniques and tools for the development of the subsystems. We describe the application functionality of the model showing through deployment and Component Diagrams. After the programming of applications tested its main features

## **Keyword**

Television, Platform, Broadcasting, Management, Streaming, Audio, Video.

## ÍNDICE

Introducción .....	1
CAPÍTULO 1: Fundamentación Teórica .....	6
1.1. Introducción.....	6
1.2. Metodología de Desarrollo. ....	6
1.2.1. PROCESO UNIFICADO DE RATIONAL (RATIONAL UNIFIED PROCESS RUP). ....	6
1.3. Lenguaje Unificado de Modelado (Unified Model Language UML). ....	9
1.4. Herramienta CASE.....	10
1.4.1. Visual Paradigm for UML.....	10
1.5. Sistema Gestor de Base de Datos. ....	11
1.5.1. POSTGRESQL 8.4.....	12
1.6. Lenguaje de programación.....	13
1.6.1. JAVA. ....	13
1.7. Entorno Integrado de Desarrollo (IDE). ....	14
1.7.1. NETBEANS 6.8 MILESTONE 2. ....	14
1.8. Framework. ....	15
1.8.1. HIBERNATE. ....	16
1.9. Streaming de Video.....	16
1.9.1. SERVIDOR DE STREAMING. ....	19
1.9.1.1. VLC MEDIA PLAYER 0.9.4. ....	19
1.9.1.2. DARWIN STREAMING SERVER 5.5.....	19
1.10. Conclusiones.....	20
CAPÍTULO 2: Descripción de la Solución Propuesta.....	22
2.1. Introducción.....	22
2.2. Descripción de los subsistemas. ....	22
2.2.1. SUBSISTEMA DE TRANSMISIÓN. ....	22
2.2.2. SUBSISTEMA DE ADMINISTRACIÓN DE LA TRANSMISIÓN.....	22
2.3. Modelo de Implementación. ....	23
2.4. Estándares de Codificación. ....	27
2.5. Conclusiones.....	31
CAPÍTULO 3: Validación de la Solución Propuesta.....	33
3.1. Introducción.....	33

---

3.2. Pruebas unitarias y funcionales.....	33
3.3. Resultados obtenidos de aplicar las pruebas funcionales.....	34
3.3.1. MÓDULO CANALES.....	36
3.3.2. MÓDULO MONITOREO.....	37
3.3.3. MÓDULO ENCADENAR.....	37
3.4. Conclusiones.....	38
Conclusiones Generales.....	39
Recomendaciones.....	40
Bibliografía Citada.....	41
Bibliografía.....	43
Glosario de Términos.....	46

# ÍNDICE DE TABLAS

Tabla 1: Límites generales de PostgreSQL.....	12
Tabla 2: Nomenclatura de los elementos.....	28
Tabla 3: Juegos de datos utilizados para las pruebas.....	36
Tabla 4: Resultados de las pruebas al CU Gestionar Canal.....	37
Tabla 5: Resultados de las pruebas al CU Monitorear Canal .....	37

## ÍNDICE DE FIGURA

Figura 1: Diagrama flujos de trabajos de RUP.....	7
Figura 2: Visual Paradigm en acción.....	11
Figura 3 : Transferencia estándar de ficheros.....	17
Figura 4: Transferencia por <i>Streaming</i> . ....	17
Figura 5: Interfaz Darwing Streaming Server .....	20
Figura 6: Modelo de Implementación.....	24
Figura 7: Diagrama de Componentes: Caso de Uso Autenticar.....	25
Figura 8: Diagrama de Componentes: Caso de Uso Encadenar Canales. ....	25
Figura 9: Diagrama de Componentes: Caso de Uso Transmitir.....	26
Figura 10: Diagrama de Componentes: Caso de Uso Monitorear Canales. ....	26
Figura 11: Diagrama de Componentes: Caso de Uso Gestionar Canal. ....	27

# INTRODUCCIÓN

Desde sus inicios, el hombre ha tenido la necesidad de comunicarse con los demás, de expresar pensamientos, ideas y emociones. Así también se reconoce en el ser humano la necesidad de buscar, de saber, de obtener información creada, expresada o transmitida por los medios de comunicación.

Los medios de comunicación tradicionales no pueden hacerse a un lado cuando el mundo en su conjunto se abre paso a experimentar un fenómeno que apoyado en internet recibe diversos nombres, pero a pesar de las diferentes denominaciones este trae consigo interminables cambios. Este fenómeno constituye la evolución de la comunicación humana desde la perspectiva tecnológica. La rapidez y constancia con la que han evolucionado las Tecnologías de la Información y Comunicación (TIC) han provocado grandes cambios en el mundo de hoy.

Cuba no ha estado ajena a estas transformaciones en el mundo de las TIC. Debido a esto se creó en septiembre del 2002 la Universidad de las Ciencias Informáticas (UCI), organizada en diferentes facultades con sus propias líneas de investigación y desarrollo, la cual estaría encaminada a promover la producción del software en el país.

La Facultad 9 de la UCI desde su creación desarrolló proyectos orientados a servicios televisivos. Con el transcurso de los años se crea el Centro de Geoinformática y Señales Digitales el cual lleva a cabo la realización de la Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV). Esta solución cuenta con varios subsistemas: Transferencia, Gestión de Medias, Web, Programación, Seguridad, Reporte, Transmisión y Administración de la Transmisión. De los subsistemas anteriores no se encuentran desarrollados los de Transmisión y Administración de la Transmisión, aunque se cuenta con un análisis y diseño de los mismos. Estos subsistemas de transmisión son los encargados de transmitir los recursos multimedia programados para que sean presentados a los usuarios.

Entre las funcionalidades que pertenecen a estos subsistemas se encuentran: *realizar transmisión, encadenar canal y monitorear canal*. En su conjunto estas deben permitir realizar las transmisiones programadas para cada canal, intervenir una emisión para insertar un nuevo material y reanudar la transmisión a partir del momento en que se interrumpió. Debe posibilitar la visualización de los canales que se transmiten en busca de posibles errores así como la vinculación de una transmisión a diversos canales.

Debido a que no se encuentran implementados los subsistemas Transmisión y Administración de la Transmisión para el control de todos los servicios de audio y video, persiste en la Dirección de Televisión Universitaria (DTU) un gran número de problemas en las transmisiones de contenidos multimedia.

Las transmisiones se ven con frecuencia interrumpidas con errores inesperados. Al ocurrir este suceso en los receptores se aprecia una pantalla de color negro que muestra el reproductor con el que se realiza la transmisión. Los eventos asociados a la reanudación de la transmisión los realiza una persona (transmisor) de forma manual siendo visible para los usuarios.

Para el control de los canales de televisión que se transmiten a la comunidad universitaria, el transmisor cuenta con un televisor de control remoto que le permite visualizar todos los canales de forma individual. Esto imposibilita que pueda monitorear más de un canal en el mismo instante, trayendo consigo que mientras se está monitoreando una emisión televisiva determinada, pueda existir otra que esté presentando fallas en la transmisión sin que consiga notarlo.

El departamento de transmisión no realiza los procesos de interrupción y encadenamiento de canales al no contar con herramientas que posibiliten este suceso. El proceso de interrupción tendría como objetivo detener una transmisión en un momento determinado y reemplazarla por cualquier otra en caso de ser necesario, así como restituirla una vez concluido el material que la sustituyó. Otro valor agregado sería el encadenamiento de canales logrando que una misma transmisión o material pueda ser visualizado simultáneamente por varios canales.

De lo anteriormente expuesto se puede plantear el siguiente **problema a resolver**: Inexistencia de los subsistemas Transmisión y Administración de la Transmisión para gestionar y transmitir el contenido multimedia en la Plataforma de Transmisión Abierta para Radio y Televisión para la DTU.

Teniendo en cuenta el problema planteado se define como **objeto de estudio**: Los procesos de transmisión de contenidos multimedia y se especifica el siguiente **campo de acción**: Automatización de los procesos de transmisión de contenidos multimedia en la Plataforma de Transmisión Abierta para Radio y Televisión.

Para darle solución al problema planteado se trazó como **objetivo general** de la investigación: Implementar los subsistemas Transmisión y Administración de la Transmisión en la Plataforma de Transmisión Abierta para Radio y Televisión.

Se planteó como **Idea a defender**: La implementación de los subsistemas para la transmisión de contenido multimedia en el sistema Plataforma de Transmisión Abierta para Radio y Televisión producirá una mejora de los procesos implicados y un aumento de la productividad.

Para lograr este objetivo se desarrollan las siguientes **tareas de investigación**:

1. Identificar aplicaciones o soluciones existentes que puedan contribuir a la implementación de los subsistemas.
2. Valorar tendencias y tecnologías actuales.
3. Caracterizar los entornos integrados de desarrollo y los lenguajes de programación.
4. Describir la metodología de desarrollo.
5. Caracterizar el lenguaje de modelado UML y la herramienta case.
6. Caracterizar el sistema de streaming de video en software libre.
7. Caracterizar el lenguaje de programación a utilizar.
8. Construir el modelo de implementación.
9. Implementar los subsistemas Transmisión y Administración de la Transmisión.
10. Desarrollar los casos de prueba que certifiquen la veracidad de los algoritmos empleados.
11. Documentar los resultados de las pruebas unitarias y funcionales de los subsistemas.

Para la realización de estas tareas se emplearon diferentes **métodos de la investigación**.

### **Métodos Teóricos:**

- *Histórico-Lógico*: ayuda a caracterizar las soluciones existentes de transmisión de señales en vivo. Lo que hace posible que se entienda la evolución de las mismas, para aprovechar los puntos en común de las que existen actualmente con la que se pretende construir. Utilizando en el desarrollo de la que se realiza, conceptos teóricos existentes que sean de relevancia.

- *Analítico-Sintético:* permite sintetizar la información necesaria para los subsistemas de transmisión, o sea, de los diferentes sistemas que permitan la transmisión de radio y televisión. Esto posibilita la comprensión de la misma, con ideas claras y concisas.
- *Modelación:* en el desarrollo de la investigación se utiliza este método para hacer modelos que ofrecen la posibilidad de crear abstracciones para explicar la realidad. Se hace visible en el trabajo al crear modelos como el de implementación.

### Métodos Empíricos

- *Observación:* con la utilización de este método se determinaron los rasgos imprescindibles en el desarrollo de los subsistemas de transmisión de la Plataforma de Transmisión Abierta para Radio y Televisión.



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

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción.

En el presente capítulo se aborda sobre el análisis del estado del arte sustentado en el diseño de dicha plataforma, se definen una serie de conceptos necesarios para comprender el problema. Son expuestas las tendencias tecnológicas, técnicas y lenguajes de programación, metodología de desarrollo, frameworks, así como las plataformas que la soportan y las herramientas más usadas en la actualidad. Además se precisan las principales características que debe tener el sistema para a partir de estas desarrollar los subsistemas Transmisión y Administración de la Transmisión.

### 1.2. Metodología de Desarrollo.

En el mundo de la informática uno de los temas más comunes hoy día es el de las metodologías de desarrollo de software. Estas en su mayoría establecen cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos. Una metodología tiene como objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo, por medio de una mayor transparencia y control sobre el proceso; “producir lo esperado en el tiempo esperado y con el coste esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo”. (1)

Para el desarrollo de esta investigación se determinó con anterioridad que la metodología idónea para realizar estos subsistemas es el Proceso Unificado de Rational, por sus siglas en inglés RUP. A continuación se describe detalladamente las características que las distingue de otras metodologías existentes.

#### 1.2.1. Proceso Unificado de Rational (Rational Unified Process RUP).

Es una metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Rational Unified Process (RUP) que apareció en 1998, fue creado por James Rumbaugh, Grady Booch e Ivar Jacobson, considerados padres de la Ingeniería del Software. Según sus autores el proceso de desarrollo de software lo conforman el conjunto de actividades necesarias para transformar los requisitos funcionales de un usuario en un sistema de software.

Lo que sustenta el proceso de desarrollo de software son: el proyecto, las personas, el producto y el proceso, existe una estrecha relación entre ellas. Es conocido como las cuatro P en el desarrollo del software. Los aspectos más importantes que se definen en este Proceso Unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. (2)

RUP está enfocado a cualquier tipo de proyecto, es válido decir que RUP posee cuatro fases que representan el ciclo de desarrollo del software: inicio, elaboración, construcción y transición, con hitos definidos en cada una de ellas; estas deben satisfacer la construcción de un grupo de artefactos (productos tangibles) para cumplir sus objetivos o hitos. A continuación se muestra la imagen de los flujos de trabajo que define RUP para cada una de las fases.

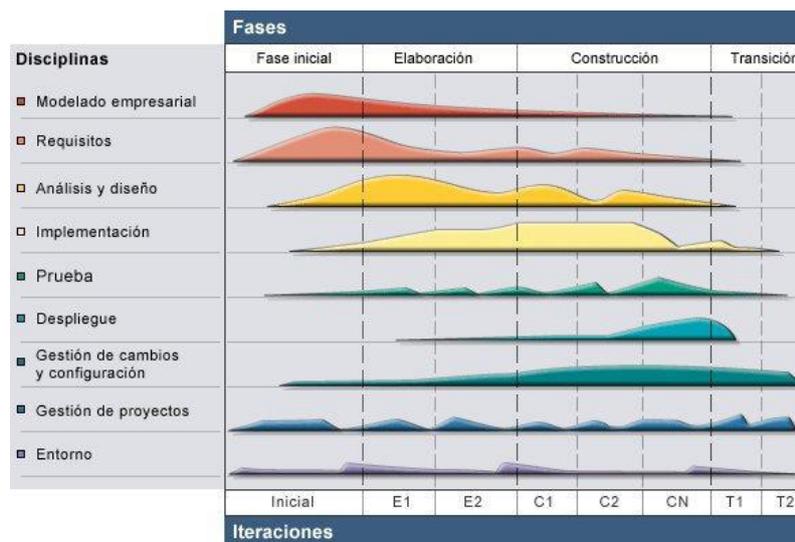


Figura 1: Diagrama flujos de trabajos de RUP.

RUP en el Flujo de Trabajo de Implementación tiene los siguientes objetivos: (3)

- Definir la organización del código en términos de Subsistemas de Implementación organizados en capas.
- Implementar los elementos de diseño en términos de elementos de implementación (ficheros Fuentes, binarios, ejecutables y otros).
- Probar los componentes desarrollados independientemente como unidades.
- Integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable.

RUP define cuales son las actividades y los artefactos que se generan para los diferentes roles que ocupan las personas durante cada una de las fases en un proceso de desarrollo software. Se hará énfasis en las actividades que son responsabilidad del rol de implementador en flujo trabajo implementación, ellas son:

- Implementar elementos de diseño.
- Implementar los subsistemas de implementación.
- Ejecutar pruebas de desarrollador.
- Implementar las prueba de desarrollador.
- Corregir la Implementación. (4)

Proceso Unificado de Desarrollo de software plantea los siguientes artefactos para el rol de implementador, los cuales se mencionan a continuación.

- Subsistema de implementación.
- Modelo de implementación.
- Prueba de desarrollador.

A continuación se describe cada uno de los artefactos mencionados.

**Subsistema de implementación:** este artefacto consta de un conjunto de elementos de implementación. Estructura el modelo de implementación dividiéndolo en componentes más pequeños que se pueden integrar y probar de manera independiente. Los subsistemas de implementación se definen inicialmente en la vista de implementación y, por lo tanto, son de vital importancia ya que posibilita desglosar todo el sistema en partes más pequeñas e independientes para que su desarrollo sea menos complejo.

**Modelo de implementación:** representa la composición física de la implementación en términos de subsistemas de implementación, y elementos de implementación (directorios y archivos, incluyendo código fuente, datos y archivos ejecutables). En él están representados ambos subsistemas en una vista global con todos los componentes, relaciones, librerías y código fuente.

**Prueba de desarrollador:** proporciona a la implementación un subconjunto de pruebas necesarias, efectivas y eficaces, pensadas bajo las categorías siguientes: pruebas de unidad, parte de las pruebas de integración, y algunos aspectos de lo que se denomina

pruebas del sistema. Esto posibilita a los desarrolladores comprobar que el componente funciona correctamente antes de que se realicen más pruebas formales en él. (5)

Para lograr un producto de óptima calidad y que además se acerque lo más posible a lo deseado por el cliente, es necesaria la realización de este artefacto. El cual le otorga al desarrollador los resultados necesarios para un correcto monitoreo del funcionamiento de cada componente en el sistema.

RUP utiliza Unified Model Language (UML) como el lenguaje de modelado para la representación de los diagramas. A continuación se describen las características de dicho lenguaje.

### **1.3. Lenguaje Unificado de Modelado (Unified Model Language UML).**

El Proceso Unificado de Desarrollo utiliza Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo. (6)

El lenguaje de modelado utilizado es UML para la modelación de la solución propuesta, ya que:

- UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. (7)
- El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue. (7)
- Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos). (7)

Para modelar con este lenguaje se usan herramientas que se han desarrollado específicamente para ello, y reciben el nombre de herramientas CASE.

### 1.4. Herramienta CASE.

Las herramientas CASE<sup>1</sup> son una gran variedad de aplicaciones informáticas que tienen como objetivo garantizar que se alcance la consistencia, completitud y cumplimiento de los estándares; mejoramiento de la productividad y la calidad, a través de un entorno interactivo; así como automatizar e integrar las tareas de las distintas etapas del ciclo de vida. Lo que constituyen un eslabón fundamental en el modelado de los procesos de negocios de las empresas. (8)

Para hacer la representación de los modelos necesarios en la investigación se decidió que se iba a utilizar la herramienta CASE Visual Paradigm for UML, por las potencialidades que esta brinda, descritas a continuación.

#### 1.4.1. Visual Paradigm for UML.

Con el objetivo de mejorar la productividad, disminuyendo el tiempo de producción, facilitando la generación automática de documentación y la reutilización de componentes durante el proceso de desarrollo de los subsistemas Transmisión y Administración de la Transmisión, se determinó utilizar Visual Paradigm for UML, influyendo en esto la posibilidad que ofrece la versión Community con su licencia gratuita. Es una herramienta colaborativa con CVS<sup>2</sup> y Subversion, varios usuarios pueden trabajar sobre el mismo proyecto, proporciona la integración con diversos entornos de desarrollo integrados, como por ejemplo NetBeans. (9)

Otras características que ratifican el uso de la herramienta son su capacidad de trabajo multiplataforma, diseñado tanto para el sistema operativo Windows como para las diferentes distribuciones de Linux, facilitando la importación y exportación de componentes, ya sea como imagen o como archivos XML. Ofrece soporte para el modelado y visualización de componentes, entre los modelos que se pueden generar se encuentran los de Implementación y los diagramas de componentes.

---

<sup>1</sup> Ingeniería de Software Asistida por Computadoras del inglés Computer Aided Software Engineering.

<sup>2</sup> Sistema de Control de Versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto.

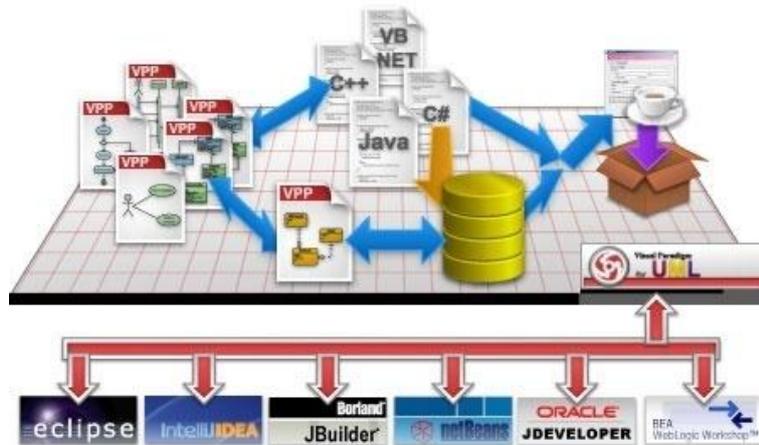


Figura 2: Descripción de Visual Paradigm for UML (Community Edition).

La utilización de bases de datos (BD) juega un papel fundamental en los sistemas donde es necesario gestionar información. Estas brindan seguridad y protección en los datos almacenados, además posibilitan eliminar la redundancia e inconsistencia de estos y certifican su integridad.

### 1.5. Sistema Gestor de Base de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (*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 (10)

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server entre otros.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes. (10)

Para la realización de estos subsistemas se necesita un sistema gestor de base de datos potente, por tal motivo se selecciona PostgreSQL. Se definen en el siguiente epígrafe las características que hacen factible utilizar el mismo en la construcción de los subsistemas.

### 1.5.1. PostgreSQL 8.4.

Es un potente sistema de gestión de base de datos objeto-relacional multiplataforma desarrollado por la Universidad de Berkeley en 1977 bajo la licencia BSD (Berkeley Software Distribution) que permite el uso del código fuente en software no libre, principal diferencia con la GPL, bajo la cual se encuentra el gestor MySQL. En la actualidad el proyecto está bajo el control de una comunidad de desarrolladores de software libre denominada *PostgreSQL Global Development Group* (PGDG). (9)

PostgreSQL es adaptable a las necesidades del cliente y posee una variada documentación en diferentes idiomas debido fundamentalmente a sus numerosas comunidades de desarrollo. Presenta entre sus características más significativas la atomicidad, consistencia, aislamiento y durabilidad de las operaciones tratadas.

En la siguiente tabla de PostgreSQL se especifican algunos límites generales del sistema gestor de Base de Datos.

Límites	Valor
Máximo de base de datos	ILIMITADO
<b>Máximo de tamaño de tabla</b>	32TB
Máximo de tamaño de registro	1.6TB
Máximo de tamaño de campo	1GB
Máximo de registros por tabla	ILIMITADO
Máximo de campos por tabla	250 a 1600 (depende de los tipos usados)
Máximo de índices por tabla	ILIMITADO
Lenguajes compatibles	java, perl, python, php, C, C++, Ruby

Tabla 1: Límites generales de PostgreSQL.

Determinando estas características la utilización de PostgreSQL como SGBD de los subsistemas de Transmisión y de forma general de la PTARTV obedece al propósito de la elaboración de un sistema con gran robustez y alto nivel de escalabilidad.

Se hace necesario además definir que lenguaje de programación se utilizará para el desarrollo de los subsistemas propuestos. En el siguiente epígrafe, se especifican que son los lenguajes de programación y más adelante se define cual se escogió para el desarrollo de los mismos.

### 1.6. Lenguaje de programación.

Las computadoras procesan la información en ceros y unos: el llamado *lenguaje de máquina (binario)*, por lo que la forma que tiene el hombre de comunicarse directamente con la máquina es mediante los lenguajes de programación.

Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático. Un lenguaje de programación permite a un programador especificar de manera precisa sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. (11) Con la evolución de las TIC se puede observar una “humanización” de los lenguajes de programación. Entre los principales criterios asumidos para seleccionar un lenguaje de programación están: rendimiento (velocidad de ejecución, estabilidad), flexibilidad (existencia de bibliotecas de funciones, clases), disposición de documentación viable, unido a la facilidad de uso y aprendizaje del mismo.

Teniendo en cuenta la definición del término lenguaje de programación antes realizada, para facilitar la comprensión del epígrafe, se continúa con la caracterización del lenguaje que se utilizará para la implementación de los subsistemas Transmisión y Administración de la Transmisión.

#### 1.6.1. Java.

A través de la investigación realizada se llegó a la conclusión de utilizar Java como lenguaje de programación para el desarrollo de los subsistemas Transmisión y Administración de la Transmisión, por la facilidades que brinda de interacción con archivos multimedia, cuenta con librerías como la *jvlc* que permite dicha interacción, además los elevados niveles de seguridad que proporciona, aspecto de vital importancia ya que cumple óptimamente con la necesidad de desarrollar una solución para administrar y configurar los diferentes canales de televisión.

Java reúne una serie de características oportunamente consideradas, por ejemplo es un lenguaje dinámico simple, orientado a objetos, distribuido, interpretado, robusto, de arquitectura neutra, portable y multihilo. De apariencia similar a C++, pero sin sobrecarga de operadores, herencia múltiple, y aritmética de punteros, tiene recolector de basura. (12)

Está diseñado para ser seguro, con técnicas de autenticación basada en cifrado de clave pública. Interpretado con una pila a base de máquina virtual. Una interfaz de red permite que el código compilado pueda ser enviado a través de Internet y ejecutado remotamente, lo que permite a los usuarios añadir sus programas a páginas web. (13) Está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos.

Como única desventaja se vislumbraba la necesidad de elevadas prestaciones del hardware existente, aspecto gestionado, logrando un rendimiento aceptable con los recursos técnicos actuales. Para realizar una correcta implementación en este lenguaje existen herramienta que proporcionan a los desarrolladores de ciertas facilidades en la confección del código fuente tal es el caso de los entornos integrados de desarrollo.

### **1.7. Entorno Integrado de Desarrollo (IDE).**

La evolución de las TIC ha traído consigo una complejidad cada vez mayor en el código de las aplicaciones existentes en la actualidad. Para generar este código los programadores necesitan algún software que además de facilitar el trabajo lo agilice, por este motivo se utilizan los Entornos de Desarrollo Integrado o IDEs (*Integrated Development Environment*).

Teniendo ya certeza del lenguaje de programación a utilizar (Java), se pasa a definir el IDE que facilite la implementación de los subsistemas: Transmisión y Administración de la Transmisión, realizándose una caracterización de NetBeans, una de las soluciones que provee herramientas para desarrollar aplicaciones en Java.

#### **1.7.1. NetBeans 6.8 Milestone 2.**

El IDE NetBeans es un entorno de desarrollo integrado, disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permite a los desarrolladores la creación de páginas Web,

aplicaciones de escritorio y aplicaciones móviles utilizando fundamentalmente la plataforma Java. (14)

El IDE que se va a utilizar posee algunas características que lo hacen destacado entre los más utilizados en el desarrollo de software:

- Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones con elevado contenido gráfico.
- Posee componentes visuales bastante familiares con entornos como *.NET*, permitiendo diseñar interfaces con solo arrastrar y soltar estos en el formulario de la aplicación deseada.
- Utiliza como soporte EJB 3.1.
- JPA 2.0, implementación, depuración y perfilado con servidor de aplicaciones GlassFish v3.
- Completamiento de código, pistas de error, ventanas emergentes de documentación.
- Paleta de componentes personalizable. (14)

Una vez definido que entorno de desarrollo se va a utilizar es conveniente definir el framework que facilite el desarrollo de los subsistemas de Transmisión de forma rápida y con buena calidad.

### 1.8. Framework.

Un framework es un conjunto de librerías, componentes y un lenguaje interpretado entre otros programas, junto con una documentación y metodología de uso, que permite diseñar, construir e implantar aplicaciones de forma rápida, y con mayor calidad. Los frameworks son desarrollados con POO lo que permite una mayor estabilidad, flexibilidad y reusabilidad del código empleado.

Uno de estos frameworks es Hibernate, utilizado por el lenguaje de programación JAVA para realizar la abstracción a la BD.

Teniendo en cuenta la definición de framework realizada, se continúa con la caracterización del framework Hibernate que se utilizará para el mapeo de la base de datos.

### 1.8.1. Hibernate.

El desarrollo de software orientado a objetos y el uso de bases de datos relacionales pueden invertir mucho tiempo en los entornos actuales. Hibernate es una herramienta que realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM<sup>3</sup> (object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Entre las principales ventajas de Hibernate se puede observar el trabajo con objetos, atributos y relaciones, evitando así los conjuntos y tuplas de las bases de datos relacionales. Potente lenguaje de consulta HQL, lo que le facilita al desarrollador realizar consultas sin la necesidad de utilizar sentencias SQL. Posee además una comunidad activa, con un gran número de usuarios y muy buena documentación.

Uno de los aspectos que hay que tener en cuenta para la realización de esta investigación es como se transmiten los flujos de videos. Por lo que es preciso definir que tecnología se va a usar para lograr este fin.

### 1.9. Streaming de Video.

La tecnología de 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. (15)

Si no se emplea esta tecnología, para mostrar un contenido multimedia en la red, hay que descargar el archivo completamente en el ordenador y después reproducirlo, para poder escuchar y visualizar su contenido. Sin embargo, el streaming permite reproducir un archivo multimedia sin necesidad de descargarlo en el ordenador, lo que recibe el ordenador es un flujo de paquetes ensamblados en el buffer del reproductor permitiendo la visualización de una manera más rápida.

El streaming funciona como se explica a continuación: Primero el ordenador (el cliente) conecta con el servidor y éste le empieza a enviar el fichero. El cliente comienza a recibir

---

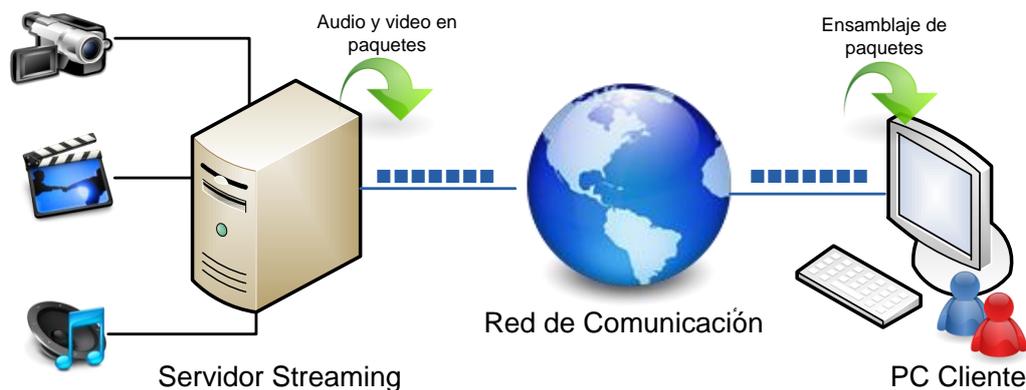
<sup>3</sup>ORM: Object-Relational Mapping es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

el fichero y construye un buffer donde empieza a guardar la información. Cuando se ha llenado el buffer con una pequeña parte del archivo, el cliente lo empieza a mostrar y a la vez continúa con la descarga. El sistema está sincronizado para que el archivo se pueda ver mientras que el archivo se descarga, de modo que cuando el archivo acaba de descargarse el fichero también ha acabado de visualizarse. Si en algún momento la conexión sufre descensos de velocidad se utiliza la información que hay en el buffer, de modo que se puede aguantar un poco ese descenso. Si la comunicación se corta demasiado tiempo, el buffer se vacía y la ejecución el archivo se cortaría también hasta que se restaurase la señal. (15)

En la figura 3 y 4 se muestra el estándar para la transferencia de ficheros y *Streaming*.



**Figura 3 : Transferencia estándar de ficheros.**



**Figura 4: Transferencia por *Streaming*.**

Características de la Tecnología de Streaming:

- Se utiliza para sistemas multimedia distribuidos.

- Se fracciona la información para transmitirla.
- El envío de la información se realiza de forma temporizada.
- No es necesario que el cliente almacene toda la información que recibe.
- La reproducción puede comenzar instantes después del comienzo de la transmisión. (16)

En tecnología *Streaming* existen una serie de elementos no que se pueden dejar de mencionar tales como:

- **Servidor Streaming:** Es un tipo de software que realiza ciertas tareas en nombre de los usuarios. Es decir un servidor streaming se utiliza con el propósito de proveer contenidos multimedia a otras máquinas que soliciten este servicio.

Los servidores pueden brindar dos tipos de servicios:

- **Bajo demanda:** más conocido por la siglas VoD, en Inglés *Video on Demand* se basa en la idea de que los usuarios clientes desde un ordenador o televisor o cualquier otro dispositivo multimedia decide cuales de los archivos multimedia disponibles quiere visualizar además lo puede detener, avanzar y rebobinar. El VoD permite realizar las mismas operaciones que un reproductor de video convencional (VCR<sup>4</sup>). La diferencia radica en que en bajo demanda todos los archivos están almacenados en un servidor.
- **Difusión (broadcast):** es el acto más frecuente de transmitir señales de audio y video en diversos formatos donde el emisor envía información a una multitud de receptores de manera simultánea, como involucra a una gran audiencia esta puede ser local, regional e internacional. Los clientes que reciben la señal no pueden detener, avanzar o rebobinar el archivo multimedia. (16)

Como se ha podido apreciar para realizar las transmisiones de archivos multimedia se necesita un servidor streaming potente. En el siguiente epígrafe se continúa con la caracterización de los servidores streaming que se utilizarán en la implementación de los subsistemas Transmisión y Administración de la Transmisión.

---

<sup>4</sup> VCR: (*Video Cassette Recorder*) conocido como grabador de video.

### 1.9.1. Servidor de Streaming.

#### 1.9.1.1. VLC Media Player 0.9.4.

*VLC media player* es una solución de software para *streaming* que funciona bajo los sistemas operativos *GNU/Linux*, *Windows* y *Mac OS X*. Es distribuido gratuitamente bajo licencia GNU. Puede transmitir una gran cantidad de archivos, entre los que se encuentran MPEG-1, MPEG-2 y MPEG-4, DVDs, vídeo en vivo sobre la red en *unicast* o *multicast* y soporta IPV4 e IPV6. (17)

Con esta solución se puede reproducir y enviar un flujo de *streaming* por la red. Cuenta con varias interfaces gráficas mediante las cuales se pueden realizar casi todas las operaciones, pero algunas solo están disponibles mediante la línea de comandos.

El *VLC* es una excelente propuesta para ser usado en PTARTV. Para ello se necesita hacer uso de la línea de comandos, de tal forma que a través del módulo correspondiente se pueda hacer *streaming* con las medias que se encuentren en una carpeta determinada.

#### 1.9.1.2. Darwin Streaming Server 5.5.

El *Darwin Streaming Server* es la versión de código abierto para los servidores de *Apple Inc.*<sup>5</sup>. Permite hacer *streaming* de forma increíblemente sencilla, simplemente situar los vídeos en el directorio especificado y conectar remotamente.

Esta herramienta puede ser utilizada para transmitir *Unicast*<sup>6</sup>, *Multicast*<sup>7</sup> y *Relay*<sup>8</sup>, esta última es muy interesante pues el servidor escucha la información entrante y la reenvía a uno o más destinos, estos pueden estar ubicados en distintos puntos geográficos sintonizando la misma señal. Los usuarios también pueden sintonizar transmisiones en directo o pregrabadas, o pueden acceder al contenido multimedia bajo demanda.

---

<sup>5</sup> Empresa multinacional estadounidense que diseña y produce equipos electrónicos y software.

<sup>6</sup> Envío de información a un sólo receptor.

<sup>7</sup> Envío de información a varios receptores.

<sup>8</sup> Técnica de comunicación mediante retransmisión de tramas.

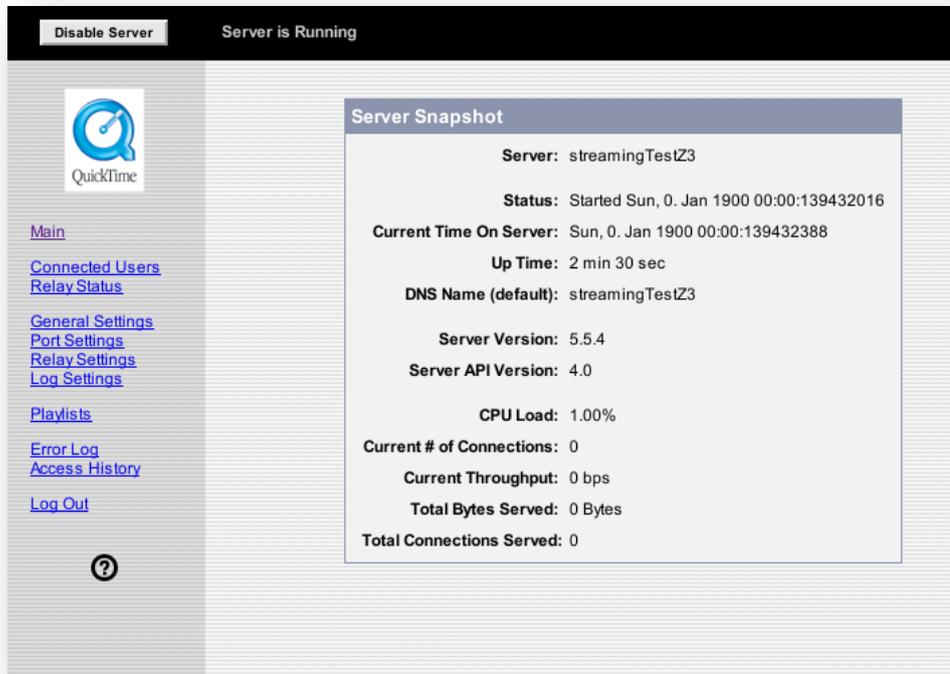


Figura 5: Interfaz Darwing Streaming Server

### 1.10. Conclusiones.

En el presente capítulo se realizó la fundamentación teórica de todas las tecnologías, herramientas, metodologías de desarrollo y lenguajes de programación que serán usados para la implementación de los subsistemas Transmisión y Administración de la Transmisión, lo permitió realizar una adecuada selección de las mismas de acuerdo a los requerimientos establecidos para el sistema aportando elementos necesarios para la implementación.



## **CAPÍTULO 2:** *Descripción de la Solución Propuesta*

# CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

### 2.1. Introducción.

Este capítulo presenta una descripción de las funcionalidades de los subsistemas que se van a implementar, así como también los módulos contenidos en cada uno de ellos. Se referencia el modelo de implementación y los diagramas de componentes correspondientes a los subsistemas: Transmisión y Administración de la Transmisión, con el objetivo de comprender la estructura de ambos subsistemas. Se analiza el estilo de programación empleado en el desarrollo de los mismos, así como también los estándares de codificación utilizados.

### 2.2. Descripción de los subsistemas.

La Plataforma de Transmisión Abierta para Radio y Televisión está dividida en varios subsistemas. A continuación se describen los subsistemas: Transmisión y Administración de la Transmisión.

#### 2.2.1. Subsistema de Transmisión.

El subsistema de Transmisión es el encargado de transmitir los recursos multimedia programados, para que sean presentados a los televidentes.

##### ➤ **Módulo Transmisión**

Una de las funcionalidades más importantes de la Plataforma de Transmisión Abierta para Radio y Televisión es que los archivos multimedia puedan ser visualizados por los usuarios. Este módulo es el encargado de iniciar la transmisión de los archivos multimedia que se hayan programado para transmitirse.

#### 2.2.2. Subsistema de Administración de la Transmisión.

El subsistema Administración de la Transmisión es el encargado de la gestión de los canales de la Plataforma de Transmisión Abierta para Radio y Televisión. Permite el monitoreo de los mismos una vez que se están transmitiendo, así como el encadenamiento de los canales.

##### ➤ **Módulo Canales**

El módulo Canales es el encargado de la gestión de los canales de PTARTV. Esto implica la modificación, inserción, eliminación y búsqueda de los canales por los cuales se emitirán los archivos multimedia programados para cada uno de ellos. Cada canal podrá transmitir un contenido diferente, lo que permitirá que se puedan satisfacer los gustos de numerosos usuarios al permitir una programación variada.

➤ **Módulo Monitoreo**

Es muy importante que cada canal sea supervisado para detectar alguna anomalía en la transmisión. Este proceso se realizará mediante la visualización en diferentes interfaces de cada una de las transmisiones en curso. Una vez que el transmisor detecta un error, procede a realizar un reporte tomando los datos necesarios del mismo con el objetivo de descubrir la causa y poder solventarlo para futuras transmisiones.

➤ **Módulo Encadenar Canales**

El objetivo principal de este módulo consiste en lograr que una misma programación pueda ser transmitida simultáneamente por varios canales. La importancia de esta funcionalidad reside fundamentalmente en las emisiones en vivo de noticieros, así como en los cortes informativos para dar a conocer alguna información de última hora.

Una vez descritos los subsistemas Transmisión y Administración de la Transmisión conjuntamente con sus respectivos módulos, se hace necesario construir el modelo de implementación, que representa una organización más detallada de ambos subsistemas. A continuación se muestra dicho modelo y sus respectivos diagramas de componentes.

### **2.3. Modelo de Implementación.**

El modelo de implementación denota la programación de la aplicación en términos de componentes y subsistemas de implementación. Describe la organización de los datos, archivos, ejecutables, código fuente y directorios de acuerdo a los mecanismos (18) de estructuración disponibles en el entorno de implementación. Fundamentalmente, se describen las relaciones que existen entre los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. En este artefacto se describe cómo se implementan los componentes que lo conforman, acoplándolos en subsistemas organizados en capas y jerarquías, señalando las dependencias entre estos.

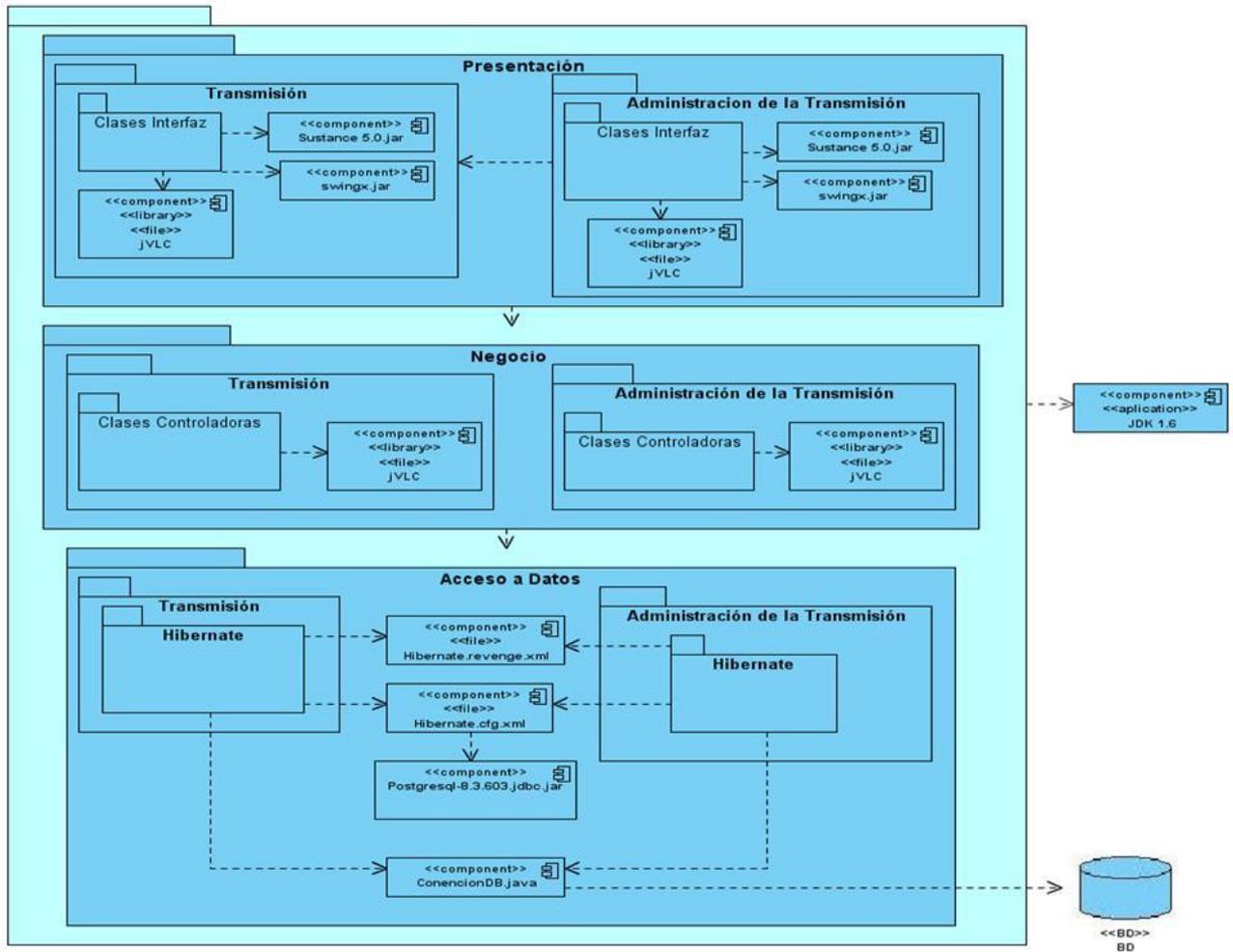


Figura 6: Modelo de Implementación.

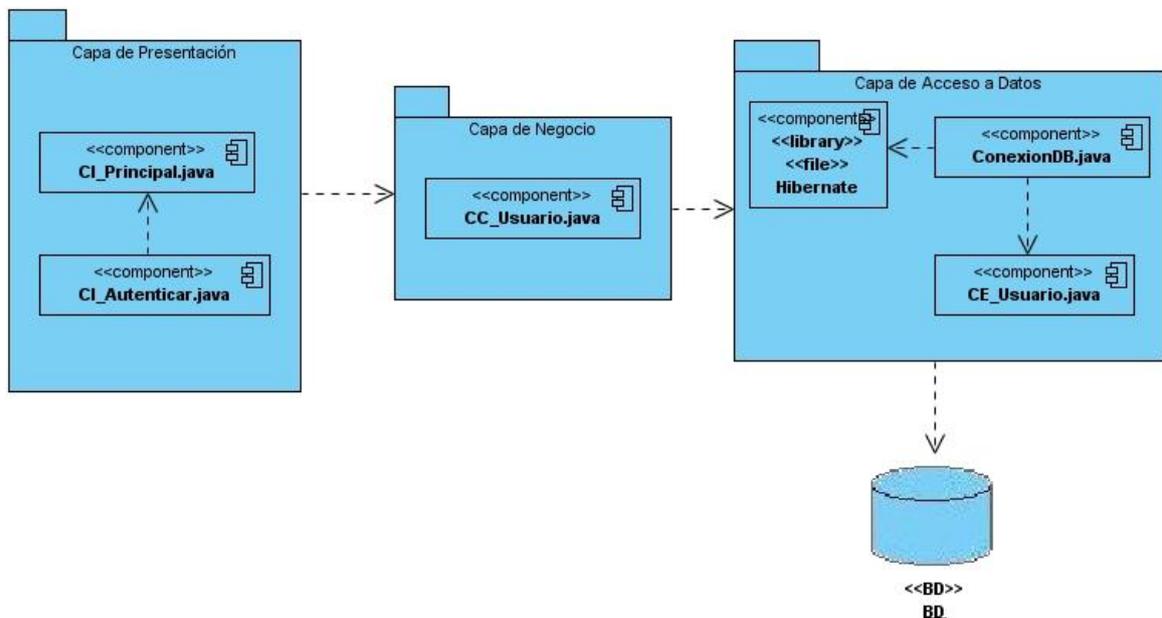


Figura 7: Diagrama de Componentes: Caso de Uso Autenticar.

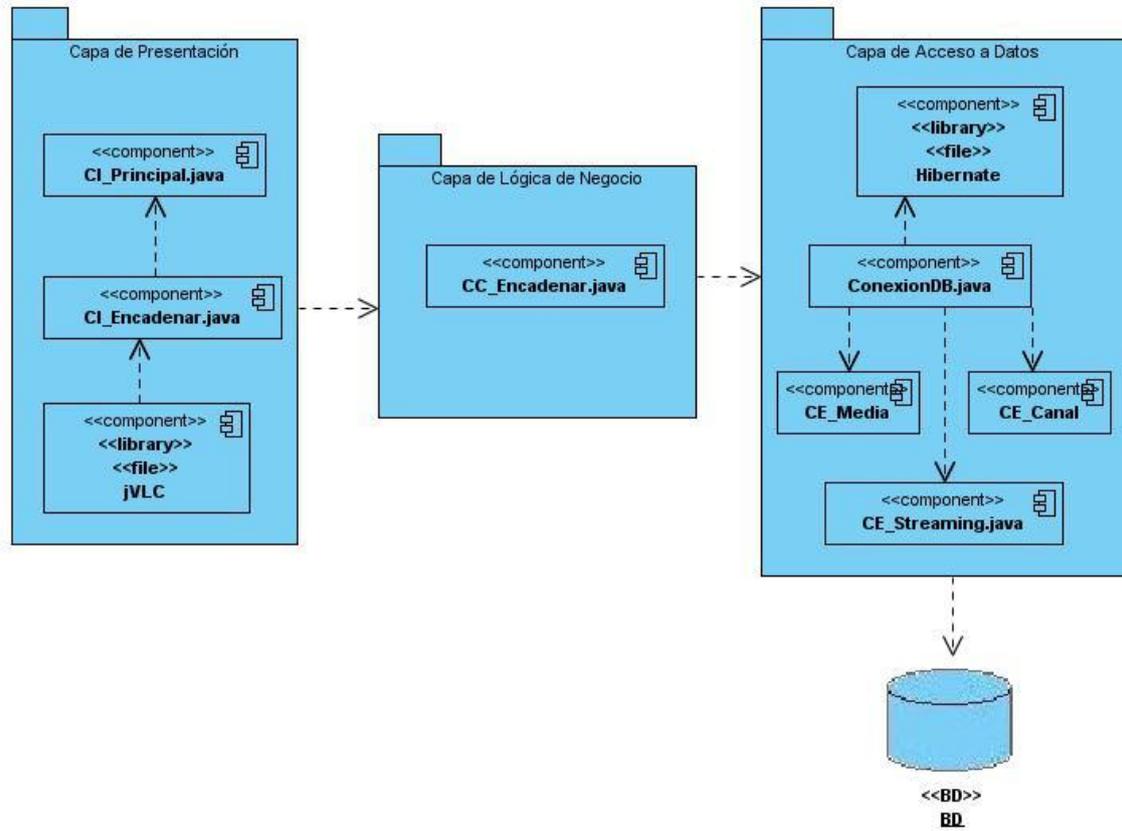


Figura 8: Diagrama de Componentes: Caso de Uso Encadenar Canales.

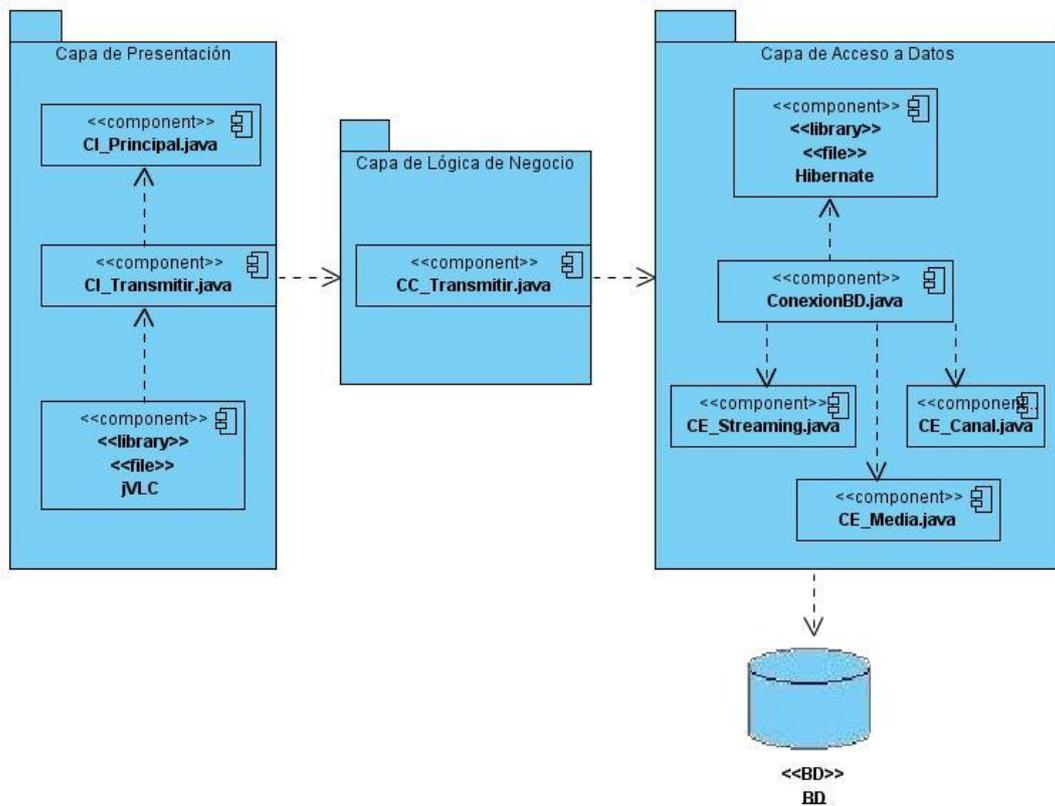


Figura 9: Diagrama de Componentes: Caso de Uso Transmitir.

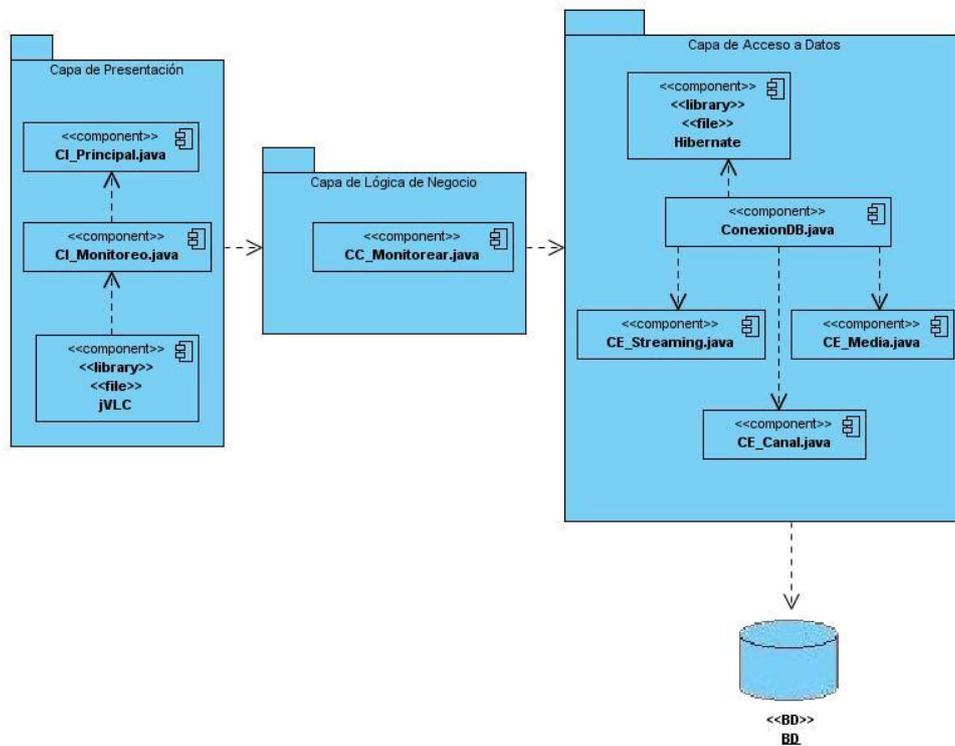


Figura 10: Diagrama de Componentes: Caso de Uso Monitorear Canales.

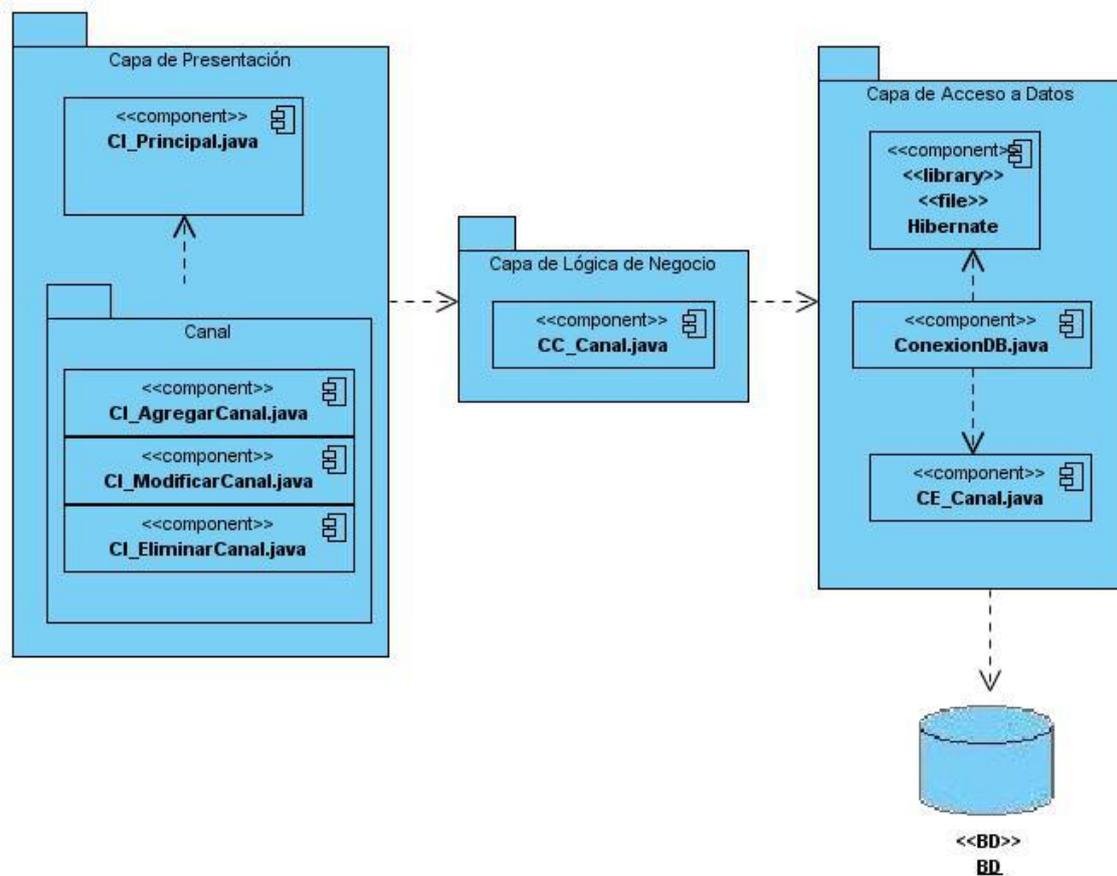


Figura 11: Diagrama de Componentes: Caso de Uso Gestionar Canal.

Para implementar de manera correcta y legible los subsistemas Transmisión y Administración de la Transmisión, es necesario definir un estándar de codificación. A continuación se especifica el utilizado para desarrollar ambos subsistemas.

#### 2.4. Estándares de Codificación.

Se define estilo de programación a la manera en que se le da un formato al código fuente, esto incluye la forma en que se colocan las llaves, cómo se organiza el código y dónde se emplean los paréntesis. Es una realidad que cada programador tiene un estilo personal para escribir su código fuente pero al definirse un estándar para utilizar se debe acoplar dicho estilo personal con el de todo el equipo.

Al conformar un estándar de codificación deben tenerse en cuenta varios **principios generales** de los cuales se puede mencionar: (19)

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.

- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas y los métodos deben comenzar con letra mayúsculas.

### Nomenclatura de los elementos.

Elemento	Reglas de nombramiento
Clases, interfaces y archivos fuente	<p>Nombre sustantivo singular, con la primera letra en mayúscula y las demás en minúsculas.</p> <p>Si el nombre de la clase está comprimido en más de una palabra la primera letra debe de ser mayúscula. Letras mayúsculas seguidas no son permitidas.</p> <p>Ejemplo: “Zend_PDF” no es permitido mientras que “Zend_Pdf” si lo es.</p>
Constantes	Nombre sustantivo en mayúsculas. Para separar palabras se usará el guión bajo: _
Paquetes y directorios	Nombre sustantivo singular en minúsculas

**Tabla 2: Nomenclatura de los elementos.**

### Comentarios

Cada programa deberá comenzar con un comentario que incluya:

- Autor
- Fecha
- Objetivo, o problema que resuelve el programa
- Algoritmo.

Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.

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

Se utilizará dos tipos de comentarios.

```
/* Se puede utilizar en el caso de un atributo, de un método o algún segmento de código */ (comentario de implementación)
```

### Otros Comentarios Internos

Los otros comentarios necesarios deben figurar a partir de la columna 35. (Comentario de documentación)

```
// .....  
//.....
```

### Nombres de identificadores.

Se considera como identificador a los nombres de variables (arreglos, matrices, apuntadores), funciones, así como cualquier tipo de dato definido por el usuario (estructura, clase). Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje. (19)

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

### Identificadores de variables dimensionadas (arreglos, matrices)

Su nombre deberá comenzar con las letras ar.

**Ejemplo:** arRol

Donde arRol un arreglo de datos para guardar información de un rol.

### Identificadores de listas y estructuras

Para el caso de las listas sería listMedia y para las estructuras stEstructura.

### Identificadores de datos constantes

Serán declaradas en letras mayúsculas.

**Ejemplo:** const IVA = 0.15;

### Identificadores de funciones

La primera letra deberá ser mayúscula.

**Ejemplo:** void Funcion ( );

### Identificadores de tipos definidos por el usuario

La primera letra será mayúscula.

**Ejemplo:** class Clase o struct Estructura

### Sangrías

- Las sangrías tendrán una longitud de cuatro espacios.
- Las llaves abren y cierran con la misma sangría.

**Ejemplo:** void Funcion ( )

```
{  
    //Instrucciones de la función  
}
```

### Líneas y espacios en blanco

- No insertar líneas en blanco entre instrucciones mientras no sean complejas.

```
a = b + c;  
int f;           //declaración entre instrucciones  
f = a;
```

- Las declaraciones de datos dentro de una función deberán ir al inicio.
- Deben incluirse espacios en ambos lados de los operadores binarios.

**Ejemplo:** y = 50 + 15 - x;

- Para claridad del programa es recomendable dejar dos líneas en blanco para las definiciones de clases e interfaces y una entre métodos.

### Tamaño de Líneas

El largo máximo ideal de una línea de código es de 80 caracteres, esto permite una cómoda lectura del mismo. Si es necesario partir la línea, la siguiente línea debe alinearse dejando doble sangría. (19)

**Ejemplo:** `if ((Key>=97&&Key<=122)||((Key>=48&&Key<=57)||((Key>=65)`

`|| (Key == 8) || (Key == 13))`

### 2.5. Conclusiones.

En este capítulo se puede observar el resultado del análisis de los diferentes diagramas y principios ilustrados durante el período de construcción de la solución que se propone. Se expusieron las distintas funcionalidades que presentan los subsistemas Transmisión y Administración de la Transmisión, haciendo énfasis en el empaquetamiento de los módulos por subsistemas y se mostró el funcionamiento interno de cada uno de los módulos a través de los Diagramas de Componentes. Se definió y caracterizó el estilo de programación puesto en práctica en la implementación de los subsistemas, para que otros desarrolladores puedan entender el código fuente.



# **CAPÍTULO 3:**

## *Validación de la Solución Propuesta*

### CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

#### 3.1. Introducción.

Las pruebas son de vital importancia, ya que aseguran la calidad del software, para que el mismo llegue a manos de los clientes cumpliendo todas sus funcionalidades y quedando lo más libre posible de fallas. Cada prueba tiene su estrategia y propósito. La ausencia de defectos no puede ser asegurada a través de las pruebas, sino que solo se puede demostrar que existen errores en el software. El proceso de pruebas cuenta con actividades bien definidas, comienza con la planificación de las pruebas, luego la ejecución, el control y por último la evaluación de las mismas. La creación de extensos y monótonos casos de pruebas ha sido remplazada al transcurrir el tiempo por procesos automatizados.

En el presente capítulo se aborda todo lo referente a las herramientas y elementos empleados en la validación de los subsistemas Transmisión y Administración de la Transmisión con el objetivo de garantizar el correcto funcionamiento de los mismos.

#### 3.2. Pruebas unitarias y funcionales.

En el desarrollo de *software* la ausencia de errores es una autentica quimera. Tantos son los factores que finalmente pueden alterar el flujo que consideramos "normal" en el programa como los errores propios a cualquier actividad humana. (20)

El uso de las pruebas unitarias permitirá que:

*Los errores sean más fáciles de localizar:* bastará con ejecutar la batería de "Colecciones de pruebas", y ver qué módulos no las pasan.

*Los errores están más acotados:* cuando un programa falla, muchas veces no se sabe de dónde pueden venir los problemas. Con las pruebas unitarias se consigue acotar los errores, sabiendo qué módulos no están pasando las pruebas. (21)

*Se reducen los "efectos secundarios":* muchas veces, cuando se quiere arreglar algo bajo presión, se cometen otros errores, o no se tiene en cuenta ciertos aspectos, que hacen que el programa deje de funcionar por otro lado. Incluso a veces, es más peligroso arreglar un error que dejarlo como está, ya que se puede subsanar el error,

pero generar otros distintos. Al aplicar las pruebas unitarias es más fácil controlar el programa pues se asegura que todo funciona tal y como se esperaba. (21)

*Se da más seguridad al programador:* normalmente, la persona que ha programado un módulo no es la misma que la que debe corregir sus errores. Esto crea una sensación de inseguridad al programador pues a la hora de corregir un error, no tiene la certeza de que su corrección no va a afectar a otros módulos que desconoce. Las pruebas unitarias aseguran que una corrección no repercuta en otros módulos, y permite al programador centrarse en su trabajo.

*Las pruebas funcionales se hacen más sencillas:* la mayoría de los aspectos individuales de cada unidad ya están probados a través de las pruebas unitarias. De este modo, las pruebas funcionales deben centrarse sólo en verificar la correcta cooperación de las distintas unidades, y en los funcionamientos generales del programa. (22)

*Las pruebas funcionales se hacen más sencillas:* la mayoría de los aspectos individuales de cada unidad ya están probados a través de las pruebas unitarias. De este modo, las pruebas funcionales deben centrarse sólo en verificar la correcta cooperación de las distintas unidades, y en los funcionamientos generales del programa. (22)

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con el menor tiempo y esfuerzo posible. Desde etapas iniciales de un producto informático siempre se tiene en cuenta garantizar la calidad del mismo a máxima escala. Es por esto es que se hace necesario analizar los resultados de las pruebas de software, proceso que se describe detalladamente a continuación.

### **3.3. Resultados obtenidos de aplicar las pruebas funcionales.**

El alcance de las pruebas indica qué funcionalidades se van a probar y cuáles no. Para poder definir el alcance, se divide el sistema en módulos, componentes o subsistemas, no todos los componentes serán probados con la misma importancia y pueden existir componentes que queden fuera del alcance de las pruebas. Cada componente agrupa varias funcionalidades, se dividen las funcionalidades hasta un nivel en el que sea

posible definir el alcance. Luego de esto, se analizan las funcionalidades, dando como resultado un Inventario de Pruebas. (23)

Juegos de datos utilizados en los Subsistemas de Transmisión para las pruebas Funcionales		
Campo de la interfaz	Datos Válidos	Datos Inválidos
Nombre de canal (Campo de texto)	Cualquier combinación de Letras y número (Señal 3, Canal 1) hasta 30 caracteres.	Dejar el campo vacío
Dirección IP	Cualquier dirección IP que pertenezca a IPv4 (10.7.2.66, 193.128.45.135)	Dejar campo vacío Dirección IP inválida (01.02.03.4, 256.10.2.35) Combinación de caracteres y números (10v.b12.vb4,5)
Descripción	Cualquier combinación de Letras y número (Señal 3, Canal 1) hasta 30 caracteres.	Dejar campo vacío
Puerto	Cualquier combinación de números menor que 65000 ejemplo (8080, 554, 1024 )	Dejar campo vacío o escribir combinación de números mayor que 6500 y caracteres.
Canal Fuente (lista desplegable)	Seleccionar uno de los canales que están transmitiendo	No seleccionar canal fuente
Los JCheckBox	Se selecciona	No se selecciona ninguno
Menú	Se seleccionan todas las opciones, Gestionar Canal, Encadenar Canal, etc.	No se selecciona ninguna opción.

Tabla 3: Juegos de datos utilizados para las pruebas

### 3.3.1. Módulo Canales.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Adicionar Canal de TV exitosamente		Se adiciona el canal de TV de forma correcta.  El sistema mantiene abierta la interfaz para agregar un nuevo canal TV.	Satisfactoria.	Cuando se adiciona el Canal de TV no se muestra un mensaje para informar al usuario el resultado de la operación
	Adicionar Canal de TV falla	Muestra mensaje advirtiendo el error en la información que se desea adicionar.	Satisfactoria.	
Modificar Canal TV correctamente		Se actualiza el Canal de TV en la Base de Datos	Satisfactoria.	Cuando se modifica el Canal de TV no se muestra un mensaje para informar al usuario el resultado de la operación.
	Modificar Canal de TV falla	Se emite mensaje de error en la información del Canal de TV que se desea modificar.	Satisfactoria.	
Eliminar Canal de TV correctamente		Se actualiza el Canal de TV en la Base de Datos.	Satisfactoria.	
	Eliminar Canal de TV	Se emite mensaje de error en la información del Canal de TV que se desea Eliminar.	Satisfactoria.	

**Tabla 4: Resultados de las pruebas al CU Gestionar Canal**

### 3.3.2. Módulo Monitoreo.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Monitorear Canales de TV exitosamente		El sistema reproduce las medias que se están transmitiendo por cada Canal de TV.	Satisfactoria.	
	Fallo en la conexión al servidor de base de datos.	El sistema emite un mensaje de error informado que existe algún problema de conexión con la base de datos.	Satisfactoria.	
	Fallo en la conexión al servidor de streaming.	El sistema emite un mensaje de error informado que existe algún problema de conexión con el servidor de streaming.	No Satisfactoria.	Si no existe conexión con el servidor streaming no muestra un mensaje para informar al usuario el resultado de la operación.

**Tabla 5: Resultados de las pruebas al CU Monitorear Canal**

### 3.3.3. Módulo Encadenar.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Encadenar todos los Canales de TV exitosamente		Se encadenan los canales de TV de forma correcta.  El sistema Cierra la interfaz de encadenar	Satisfactoria.	Cuando se encadenan todos los Canales de TV no se muestra un mensaje para informar al usuario

	Fallo el encadenamiento de todos los canales de TV.	Muestra mensaje advirtiendo el error en el encadenamiento.	Satisfactoria.	
Encadenar los Canales TV seleccionados correctamente		Se encadenan los canales de TV de forma correcta.  El sistema mantiene la interfaz abierta para realizar un nuevo encadenamiento.	Satisfactoria.	Cuando se encadenan los Canales de TV seleccionados no se muestra un mensaje para informar al usuario
	Fallo el encadenamiento de los canales de TV seleccionados.	Muestra mensaje advirtiendo el error en el encadenamiento de los canales de TV seleccionados.	Satisfactoria.	

**Tabla 6: Resultados de las pruebas al CU Encadenar Canal**

### 3.4. Conclusiones.

En este capítulo se prestó atención a las características relacionadas con las pruebas para la fase de implementación. Para lograr que los subsistemas Transmisión y Administración de la Transmisión sean lo más robustos posible es vital que sean sometidos a las pruebas pertinentes que confirmen las funcionalidades previstas. Resulta importante explicar que los casos de pruebas presentados no son los únicos diseñados para las aplicaciones sino que el objetivo es plasmar la forma de la realización de los mismos.

### CONCLUSIONES GENERALES

Una vez culminado el desarrollo de la investigación es posible arribar a las siguientes conclusiones:

- Durante la investigación se identificaron los diferentes tecnologías utilizadas actualmente para la administración y configuración de aplicaciones informáticas, priorizando los elementos y herramientas relacionados con el desarrollo de los Subsistemas Transmisión y Administración de la Transmisión, lo que aportó elementos necesarios para la implementación.
- Con el desarrollo los subsistemas Transmisión y Administración de la Transmisión se cumplieron con los requerimientos establecidos para el rol de implementador.
- Para certificar la veracidad de los algoritmos empleados en la implementación de los subsistemas Transmisión y Administración de la Transmisión se desarrollaron Casos de Prueba que aseguran la validación de los mismos.
- Se desarrolló un manual de usuario para la instalación de los subsistemas, lo cual facilita el trabajo de los clientes con esta aplicación.

Como resultado de la investigación se logró solucionar el problema científico que da origen a la misma pues se cumplió el objetivo general trazado, al desarrollarse los subsistemas Transmisión y Administración de la Transmisión de contenido multimedia en la PTARTV del Centro GEYSED.

### RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la implementación de los subsistemas Transmisión y Administración Transmisión, se proponen las siguientes recomendaciones:

- Realizar estudios más profundos con el fin de incorporar funcionalidades que permitan su uso en televisoras de mayor envergadura que rebase los marcos de la Dirección de Televisión Universitaria (DTU).
- Dar continuidad al desarrollo y soporte de los subsistemas desarrollados adicionándole nuevas funcionalidades y refinando las existentes para versiones posteriores.
- Poner a prueba los subsistemas Transmisión y Administración de la Transmisión en la DTU durante un periodo de tiempo, para comprobar si los resultados son los esperados.

### BIBLIOGRAFÍA CITADA

1. **Molpeceres, Alberto.** Procesos de desarrollo: RUP, XP y FDD. [En línea] 2002. [Citado el: 11 de 12 de 2009.] <http://www.javahispano.org/articles.article.action?id=76>.
2. **Orallo, Enrique Hernández.** <http://www.acta.es>. [En línea] [Citado el: 15 de 2 de 2010.] [http://www.acta.es/articulos\\_mf/26067.pdf](http://www.acta.es/articulos_mf/26067.pdf).
3. <http://eva.uci.cu>. <http://eva.uci.cu>. [En línea] 2005. [Citado el: 15 de 2 de 2010.] [http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias\\_IS2\\_05-06](http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS2_05-06).
4. <http://www.fing.edu.uy>. [En línea] [Citado el: 15 de 4 de 2010.] <http://www.fing.edu.uy/inco/cursos/ingsoft/pis/proceso/MUM/roles/imp.htm>.
5. **Lic. Carlos Galindo González, y Dr. Ramiro Pérez Vázquez.** Rational Unified Process® . *Rational Unified Process®* . 2009.
6. **Schmuller, Joseph.** *APRENDIENDO UML EN 24 HORAS*. México : PEARSON EDUCACION, 2000. 968-444-463-X.
7. **Enrique, Hernández Orallo.** El Lenguaje Unificado de Modelado (UML). <http://teleformacion.uci.cu>. [En línea] 2006. [Citado el: 12 de 01 de 2010.] <http://teleformacion.uci.cu/mod/resource/view.php?id=11361>.
8. **Gómez, Ruth Priscila Landeros.** *Herramientas Case*. México : Universidad Veracruzana, 2007.
9. **Díaz Berenguer, Abel y Román Vieito, Alberto Ramón.** *Sistema de Administración y Configuración de la solución de Captura y Catalogación de Medias*. [Digital] Ciudad de la Habana : Universidad de la Ciencias Informáticas, 2009.
10. CAVSI. <http://www.cavsi.com>. [En línea] [Citado el: 2009 de 12 de 4.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
11. **Joyanes, Aguilat L.** *Fundamentos de programación. Algoritmos, estructuras*. s.l. : McGraw Hill, 2003.

12. **Gracia de Jalón, Javier y Rodríguez, José Ignacio.** *Aprenda Java como si estuviera en primero.* s.l. : San Sebastián, 2002.
13. **Aguilar, Luis Joyanes.** *Fundamentos de programación. Algoritmos, estructuras.* España : McGraw Hill, 2003. 8448139860.
14. Welcome to NetBeans. *NetBeans IDE 6.8 Release Information.* [En línea] 5 de 1 de 2010. [Citado el: 12 de 2 de 2010.] <http://netbeans.org/community/releases/68>.
15. <http://www.desarrolloweb.com>. [En línea] 7 de 9 de 2001. [Citado el: 15 de 2 de 2010.] <http://www.desarrolloweb.com/articulos/482.php>.
16. **Austerberry, David.** *The Technology of Video and Audio Streaming Second Edition.* s.l. : Focal Press, 2005. ISBN 0-240-80580-1.
17. [www.videolan.org](http://www.videolan.org). [En línea] [Citado el: 26 de 1 de 2010.] <http://www.videolan.org/doc/vls-user-guide/es/ch01.html>.
18. **Mayedo, Rolando Morejón.** *Plataforma de Transmisión Abierta para Radio y Televisión. Subsistemas Programación y Gestión de Medias.* Ciudad de la Habana : UCI, 2009.
19. **Rondón, Yoandry Quintana.** *Estándar de codificación del Centro de Geoinformática y Señales Digitales (GEYSED).* Ciudad de la Habana : s.n., 2010.
20. **Abraham Covelo.** <http://www.novanebula.net>. [En línea] 2007. [Citado el: 28 de 04 de 2010.] <http://www.novanebula.net/blog/archives/99-Unit-testing-pruebas-unitarias.html>.
21. **Domínguez Mora, Dunier.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión .* La Habana : s.n., 2009.
22. Comunidades de wikis libres para aprender. Ventajas de las pruebas unitarias. . <http://www.wikilearning.com>. [En línea] 2009. [Citado el: 30 de 04 de 2010.] <http://www.wikilearning.com>.
23. <http://www.sistedes.es>. [En línea] [Citado el: 20 de 5 de 2010.] <http://www.sistedes.es/TJISBD/Vol-1/No-4/articulos/pris-07-perez-gpf.pdf>.

### BIBLIOGRAFÍA

1. **Gómez Ballester, Eva.** Bases de Datos 1. [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.alu.ua.es/j/jmr36/Conectate/Base%20Datos/Apuntes2006.pdf>.
2. **Domínguez Mora, Dunier.** *Implementación de los Subsistemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión.* La Habana : s.n., 2009.
3. **IBM Corporation.** Rational Unified Process. [En línea] 2007. [Citado el: 28 de Febrero de 2010.] [ayuda.del.RUP.2007. Español/core.base\\_rup/guidances/guidelines/important\\_decisions\\_in\\_implementation\\_BC39CD13.html](http://ayuda.del.RUP.2007.Español/core.base_rup/guidances/guidelines/important_decisions_in_implementation_BC39CD13.html).
4. **Letelier Torres, Patricio.** Desarrollo de Software Orientado a Objeto usando UML. [En línea] [Citado el: 10 de Marzo de 2010.]
5. *Investigación sobre estado del arte en diseño y aplicación de pruebas de software.* **Johanna Rijas Rojas, Emilio José Barrios.** 2007.
6. **Wiley, John.** *Software Engineering Standards.* s.l. : IEEE Press, 2001. ANSI/IEEE Standard 829-1983 .
7. Universidad Distrital Francisco Jose de Caldas. [En línea] [Citado el: 6 de 12 de 2009.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node28.html>.
8. Especialista Universitario Java Enterprise. [En línea] 2009-2010. [Citado el: 11 de diciembre de 2009.] <http://www.jtech.ua.es/tutoriales/apuntes/sesion-junit-apuntes.htm> .
9. **calisoft.** Curso calidad. *Introducción a la Pruebas de software.* C. Habana : s.n., 2009.
10. Departemanto de lenguajes y sistemas Informáticos:Universidad de Granadas. [En línea] [Citado el: 2 de Diciembre de 2009.] [http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/prod\\_des/documento/plan\\_pruebas\\_d.php](http://lsi.ugr.es/~arroyo/inndoc/doc/pruebas/prod_des/documento/plan_pruebas_d.php).
11. **Oberto Canales Mora.** Adictos al Trabajo. [En línea] 2007. [Citado el: 10 de 1 de 2100.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=cmmi>.
12. **Collazo, Manuel.** *Técnicas de prueba del software. Estrategias de prueba del software.* s.l. : IEEE Standard Glossary of Software Engineering Terminology 1990, 2003.

13. **Larman, Craig.** *UML y Patrones.* 1999.
14. **Contreras, Eridniel Suárez.** Diseño de una Plataforma de Transmisión Abierta para la Radio y la Televisión. Habana : s.n., 2008.
15. **Austerberry, David.** *The Technology of Video and Audio Streaming Second Edition.* 2. s.l. : Focal Press, 2005. pág. 360. 0-240-80580-1.
16. **Larman, Craig.** *UML y Patrones, Introducción al análisis y diseño orientado a objetos.* La Habana : Félix Varela, 2004.
17. **Ivar Jacobson, Grady Booch y James Rumbaugh.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* 2000. 78-84-7829-087-1.
18. **Mauricio Venegas M., Aquiles Yáñez C., Agustín J. González.** *TRANSMISIÓN DE VIDEO DE ALTA CALIDAD A TRAVÉS DE REDES IP UTILIZANDO HERRAMIENTAS DE CÓDIGO ABIERTO.* Departamento de Electrónica, Universidad Técnica Federico Santa María, Casilla 110-V, Valparaíso : s.n.
19. **Ibarra Naranjo, Hyldeé y Mañas Argemí, José Antonio.** RBAC: Alternativa actual para la realización de Control de Accesos a gran escala. [Documento PDF] Mexico DF : Universidad Politécnica de Madrid (España), 2003. ISBN: 970-36-010409.
20. **Hernández Callao, Jorge Alberto, Martínez Mendoza, Ezequiel y Gamboa Laina, Yordan.** *Concepción y Desarrollo del Módulo de Administración del Proyecto TeleBanca.* [Documento PDF] Ciudad de la Habana: Universidad de las Ciencias Informáticas, 2007.
21. **Córcoles, Jose Eduardo y Lozano, Maria Dolores.** *Modelando Aplicaciones con UML.* [Cuaderno de Practicas. Documento PDF] España: Universidad de Castilla-La Mancha.
22. **SUN Microsystems.** Java. [En línea] SUN Microsystems. [Citado el: 25 de Febrero de 2009.] <http://www.java.com/es/about/>.
23. **Valdés Gómez, Abel.** *Servicio de media streaming para la web. Portal Inter-nos. Módulos: Teleclases y TV.* [Documento Tesis] Ciudad de la Habana: Universidad de las Ciencias Informáticas, 2006.

24. **Hernández Sampier, R y otros.** *Metodología de la Investigación. Segunda reimpresión.* Ciudad de la Habana, Cuba : Editorial Félix Valera., 2003.

### GLOSARIO DE TÉRMINOS

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

**Cliente:** Aplicación informática cuya función es acceder a los servicios que ofrece un servidor, haciendo uso generalmente de una red de telecomunicaciones.

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

**Framework:** Estructura de soporte definida, que está constituido por programas, bibliotecas y un lenguaje interpretado que facilitarán el desarrollo de un proyecto informático.

**GNU:** Licencia Pública General de GNU o más conocida por su nombre en inglés General Public License (GNU).

**GPL:** La Licencia Pública General de GNU o más conocida por su nombre en inglés General Public License (GPL).

**Herramientas CASE:** Herramientas utilizadas para el desarrollo de proyectos de Ingeniería.

**IDE:** Entorno de Desarrollo Integrado. Es un programa compuesto por un conjunto de herramientas para facilitar la interacción del programador con el lenguaje en cuestión.

**Internet:** Conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

**Interfaz:** Zona de contacto o conexión entre dos componentes de "hardware"; entre dos aplicaciones, o entre un usuario y una aplicación. Apariencia externa de una aplicación informática.

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

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

**Open Source:** Literalmente, código abierto. Hace referencia al software libre, escrito bajo la Licencia GPL que permite ver, modificar y distribuir el código fuente.

**ORM (Object Relational Mapping):** en español, Mapeo Relacional de Objetos, técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

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

**Radio:** Tecnología que permite la transmisión de señales sonoras mediante la modulación de ondas electromagnéticas.

**RTP:** *Real Time Transport Protocol* (Protocolo de Transporte de Tiempo real), es un protocolo de nivel de sesión que se utiliza para transmitir información en tiempo real la cual puede ser audio y vídeo.

**RTSP:** *Real Time Streaming Protocol* (Protocolo de Flujo de Datos en Tiempo Real), es un protocolo no orientado a conexión y que en lugar de esto el servidor mantiene una sesión asociada a un identificador. RTSP establece y controla uno o muchos flujos sincronizados de datos pudiendo ser de audio o video.

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

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

**Streaming:** Transmisión de audio y video online de manera continua, se caracteriza por la visualización de los contenidos en el cliente sin la necesidad de esperar la descarga completa del fichero, lo que constituye una gran ventaja para el trabajo con medias debido al gran tamaño que pueden llegar a tener estas.

**Televisión:** Medio de comunicación que combina los mensajes de imágenes fijas y en movimiento con voz, música, efectos sonoros y especiales.

**TIC:** Tecnologías de la información y las comunicaciones.

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

**Transmisión:** Emitir la señal de radio-televisión desde una estación emisora a otra receptora.

**UCI:** Universidad de las Ciencias Informáticas.

**UML** (*Unified Modeling Language*): Notación estándar para modelar objetos del mundo.

**VLC media player:** Reproductor multimedia del proyecto VideoLAN. *Software* libre distribuido bajo la licencia GPL. Soporta muchos *códecs* de audio y video, así como diferentes tipos de archivos, además de *DVD*, *VCD*.

**Video:** Es una tecnología de captura electrónica, tratamiento, almacenamiento, transmisión, y reconstrucción de una sucesión de imágenes y sonidos que representan escenas en movimiento.