

Universidad de las Ciencias Informáticas

Facultad 6



**Plataforma Web Interactiva para el Sistema de Transmisión
de Canales Virtuales**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

Yenisel Tirado Gutiérrez

Juan Pablo Quevedo Rodríguez

Tutor:

M.Sc. Eddy Dangel Quesada Rodríguez

Ing. Guillermo Luzua Farias

Co-Tutor(es):

Ing. Asiel Hernández Valdés

La Habana, junio del 2015

“Año 57 de la Revolución”



“Todos creen que tener talento es cuestión de suerte; nadie piensa que la suerte puede ser cuestión de talento”.

Jacinto Benavente.

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores del trabajo **Plataforma Web Interactiva para el Sistema de Transmisión de Canales Virtuales** y autorizamos a la Universidad de las Ciencias Informáticas hacer el uso que estimen pertinente con el mismo.

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

Firma del autor

Yenisel Tirado Gutiérrez

Firma del autor

Juan Pablo Quevedo Rodríguez

Firma del Tutor

Ing. Guillermo Luzua Farias

Firma del Tutor

M.Sc. Eddy Dangel Quesada Rodríguez

Nombre y Apellidos: Ing. Guillermo Luzua Farias

Correo electrónico: gluzua@uci.cu

Año de graduado: 2012.

Nombre y Apellidos: M.Sc Eddy Dangel Quesada Rodríguez.

Correo electrónico: edquezada@uci.cu

Año de graduado: 2008.

Nombre y Apellidos: Yenisel Tirado Gutiérrez

Correo electrónico: ytgutierrez@estudiantes.uci.cu

Nombre y Apellidos: Juan Pablo Quevedo Rodríguez

Correo electrónico: jpquevedo@estudiantes.uci.cu

Yenisel:

Durante estos cinco años en la universidad he vivido experiencias únicas, unas muy buenas, otras no tanto. Hoy quiero agradecer aquellas personas que me han apoyado y ayudado en mi formación.

Primeramente a mi abuela Consuelo, por ser la luz que siempre guía mi vida y mi mayor motivo de inspiración.

A mi mamá, por su apoyo incondicional y por guiarme siempre por el camino correcto, Te Amo.

A mi papá y a Fabi, por confiar en mí y apoyarme tanto, y sobre todo, por darme ese regalo tan grande, mi hermanita Valentina.

A mi hermana Yadira, por darme tantos dolores de cabeza, eso me enseñó que sobre todas las cosas el amor de familia es único, te quiero mucho.

A mis princesas, Daynelis y Daniela, por ser mi mayor motivo de perseverancia y de lucha, ojalá y algún día lleguen a ser como su tía, jajaj

A mis primos, Betty, Eduardito, Osvaldo, Brayan, Darién, Melany, en especial a Will, gracias por existir en mi vida.

A mi tía Liset, gracias por tu apoyo incondicional y tus consejos, me han ayudado mucho, sobre todo en mi carrera.

A mis abuelos Hirma y Félix, gracias por su amor en todo momento y por siempre estar cuando los necesito.

A Papichurri, mi suegro Luis, por brindarme ese gran amor de padre.

A mis tí@s Félix, Lola, Angélica, Berto, Rubén, Lidia, en fin, todos, gracias por su comprensión.

A mi príncipe, Marcos, gracias por tu extraño amor y ojalá llegues a ser ese gran campeón que tanto deseas.

A mi madrina Julia, Magaly, Sara, Dania e Ileana, por brindarme ese amor de madres que solo ustedes saben ofrecer.

A mis niños, Gaby y Danielito, gracias por recordarme esa etapa tan bella que era mi infancia, y sobre todo, gracias a Lary y Dargel por aceptarme en su familia.

A mis amistades, Lianed, Pollita, Lizy, Taten, Yane, Wendy, Eli, Maikel, Frank, Tito, Yosiel, Paneque, Rafa, Vidal, Javier, Iván, Adrián, en fin, a todos, gracias por estar hoy y por aceptarme en su vida.

A las personas más tranquilas que he conocido, los revoltosos de ACAP, gracias por esas noches tan polifrásticas.

Al tribunal, en especial a Lidisy y a Vladimir, gracias por su dedicación.

Quiero hacer 4 agradecimientos especiales:

A mis dolores de cabeza más grandes en esta universidad, Gretel, Nury, Ary y Sheyla, gracias por estar a mi lado todo este tiempo, y gracias por enseñarme a tener tanta paciencia, si no las quisiera tanto diría que es odio lo que siento, las quiero mucho!!!

A mis tutores, Eddy, Asiel y Guille, gracias por confiar en este dúo, por dedicarnos tantas horas, brindarnos tantos consejos y por ser sobre todo buenos amigos. Les estaré eternamente agradecida.

A mi compañero de tesis, gracias por no matarme cada vez que te levantaba, lo siento!!!, siempre supe que éramos el dúo perfecto.

Y por último y no menos especial, a mi casi esposo, Angel Luis, gracias por estos 3 años que has estado a mi lado, por apoyarme tanto y brindarme tanto amor, sin ti, no sé si lo hubiese logrado, Te Amo.

Juan Pablo:

A mi mamá que es mi luz e inspiración, siempre preocupada por mí dándome su apoyo en los momentos difíciles.

A mi tía Tata, que siempre me apoyó y me guió por el bien camino.

A mi abuela, esto es para ti.

A mi hermano Pablo y mi prima Naryara, que siempre han estado ahí.

AGRADECIMIENTOS

A toda mi familia, los quiero.

A toda la gente nueva que conocí en mi reingreso que ahora son mis amigos.

A mis amigos de todos los años, gracias.

A mi compañera de tesis, que esta escapa haciendo documentos, casi la mataba cada vez que me levantaba, gracias.

A mis tutores Eddy, Guillermo y Asiel, gracias por guiarnos y orientarnos durante todo este tiempo, muchas cosas hemos aprendido de ustedes, gracias.

Yenisel:

A mi abuela Consuelo, por ser la luz que guía mi vida y ser mi mayor motivo de inspiración.

A mis padres, por su apoyo y amor incondicional.

A Vale, por ser mi regalo más grande.

A mis sobrinas, por ser mi motivo de perseverancia y de lucha.

Juan Pablo:

A mi mamá que es mi principal motivo de inspiración

A mi tía Tata, por estar siempre preocupada por mi

A mi abuela querida le dedico esta tesis.

A mi hermano Pablo E. y mi prima Naryara.

Resumen

Con el creciente desarrollo tecnológico, la televisión se ha convertido en uno de los medios de comunicación social de masas con mayor audiencia a nivel mundial. A raíz de esta evolución, se crea la televisión sobre IP, la misma se identifica por la difusión de servicios multimedia como televisión, video y audio a través de redes de datos. El avance más importante que ofrece este tipo de televisión es la interactividad, la cual abre la posibilidad de nuevos tipos de servicios que anteriormente no estaban disponibles en otras redes. Las plataformas interactivas forman parte de este tipo de sistemas. En el centro de Geoinformática y Señales Digitales se desarrolla el proyecto Sistema de Transmisión de Canales Virtuales, conformado por subsistemas, uno de ellos el Interactivo. El mismo solo puede ser visualizado correctamente a través de computadoras y televisores analógicos, lo que representa una desventaja para los usuarios que posean dispositivos móviles, ya que no posee un diseño adaptable a dichos dispositivos. El presente trabajo de diploma tiene como objetivo desarrollar una plataforma web interactiva que les permita a los usuarios del sistema acceder desde distintos dispositivos móviles a los servicios interactivos. Con el fin de obtener los mejores resultados, la plataforma se realizó bajo el lenguaje de programación JavaScript, utilizando como tecnología asociada NodeJS. Se realiza un estudio de los conceptos asociados a las plataformas y servicios interactivos. Se emplean los patrones de diseño en la confección e implementación del sistema. Con el fin de obtener una aplicación de calidad se realizaron pruebas al sistema que contribuyeron a su validación.

Palabras Claves: aplicaciones interactivas, interactividad, plataforma, servicios interactivos, televisión sobre IP.

Abstract

With the growing technological development, television has become one of the social media of masses with largest audience worldwide. As a result of this evolution, television over IP is created, it is identified by the diffusion of multimedia services such as TV, video and audio through data networks. The most important development that offers this type of television is interactivity, which opens the possibility of new types of services that were not previously available on other networks. Interactive platforms are part of this type of systems. In the Center for geoinformatics and signals digital develops the project transmission system of virtual channels, consisting of subsystems, one of them the interactive. It only can be viewed correctly through computers and analog televisions, which represents a disadvantage for users who have mobile devices, since it lacks a design adaptable to such devices. This diploma work aims to develop an interactive web platform that allows the users of the system from different mobile devices to access interactive services. In order to obtain the best results, the platform was carried out under the language of programming JavaScript, using associated technology NodeJS. He is a study of the concepts associated with platforms and interactive services. Design patterns are used in the preparation and implementation of the system. In order to obtain a quality application were tested to the system that contributed to their validation.

Keywords: interactivity, interactive services, interactive applications, IPTV, platform.

Índice General

Introducción	1
Capítulo 1. Fundamentación teórica	6
1.1 Interactividad	6
1.2 Elementos fundamentales de las plataformas de televisión sobre IP	8
1.3 Servicios Interactivos	10
1.4 Servicios interactivos que ofrecen los sistemas IPTV	11
1.5 Soluciones existentes.....	14
1.5.1 <i>Análisis de las soluciones existentes.</i>	16
1.6 Conclusiones del capítulo	17
Capítulo 2. Características de la solución propuesta	19
2.1 Lenguajes, tecnologías y herramientas	19
2.1.1 <i>Metodología de desarrollo</i>	19
2.1.2 <i>Framework</i>	20
2.1.3 <i>Node JS v0.10.24</i>	20
2.1.4 <i>Lenguaje de Marcado de Hipertexto</i>	21
2.1.5 <i>Lenguaje Unificado de Modelado v2.0</i>	21
2.1.6 <i>Hoja de estilos en cascadas</i>	21
2.1.7 <i>JavaScript</i>	22
2.1.8 <i>Entorno de Desarrollo Integrado</i>	22
2.1.9 <i>Visual Paradigm 8.0 (Visual Paradigm for UML v8.0 Enterprise Edition)</i>	23
2.1.10 <i>Protocolo de comunicación</i>	23
2.2 Modelo de Dominio	24
2.2.1 <i>Descripción de los conceptos que intervienen en el dominio del problema</i>	25
2.3 Requisitos.....	26
2.3.1 <i>Requisitos funcionales.</i>	26

2.3.2	<i>Requisitos no funcionales.</i>	29
2.4	Casos de Uso del Sistema	30
2.4.1	<i>Descripción de los actores del sistema.</i>	31
2.4.2	<i>Descripción de los casos de usos del sistema.</i>	32
2.5	Conclusiones del capítulo	35
Capítulo 3: Diseño e implementación de la solución propuesta		36
3.1	Propuesta de solución	36
3.2	Patrones	37
3.2.1	<i>Patrones de arquitectura</i>	37
3.2.2	<i>Patrones de diseño</i>	38
3.3	Diseño de la interfaz	39
3.4	Modelo de Diseño	40
3.4.1	<i>Diagrama de Clases del Diseño</i>	40
3.5	Modelo de Implementación	41
3.6	Diagrama de Despliegue	42
3.7	Código fuente	43
3.7.1	<i>Estándares de codificación</i>	44
3.8	Interfaces de la plataforma	45
3.9	Conclusiones del capítulo	48
Capítulo 4: Pruebas del sistema		49
4.1	Pruebas realizadas	49
4.1.1	<i>Resultados finales de las pruebas realizadas a la plataforma.</i>	49
4.2	Conclusiones del capítulo	55
Conclusiones Generales		56
Recomendaciones		57
Bibliografía		58

Índice de Tablas

Tabla 1. Características principales de las plataformas analizadas	16
Tabla 2. Descripción de los actores del sistema.	31
Tabla 3. Descripción de caso de uso: Autenticar Usuario.	32
Tabla 4. Descripción de caso de uso: Registrar Usuario.	33
Tabla 5. Caso de prueba para el caso de uso: Registrar usuario.....	49
Tabla 6. Caminos básicos obtenidos del grafo.	54
Tabla 7. Caso de prueba para el camino básico No.1	54

Índice de Figuras

Fig 1. Tipos de servicios interactivos.....	11
Fig 2. Modelo de dominio	25
Fig 3. Diagrama de caso de uso del sistema	31
Fig 4. Propuesta de solución.	36
Fig 5. Patrón arquitectónico Modelo-Vista-Controlador.....	38
Fig 6. Diagrama de clases del diseño: Registrar usuario.....	41
Fig 7. Diagrama de Componentes de la plataforma	42
Fig 8. Diagrama de Despliegue	43
Fig 9. Uso del estándar de codificación CamelCase.	45
Fig 10. Interfaz principal: autenticar usuario.	46
Fig 11. Interfaz correspondiente a la búsqueda de usuarios.....	46
Fig 12. Interfaz correspondiente a la gestión de Usuarios.....	47
Fig 13. Interfaz correspondiente a la administración de espacio de almacenamiento por usuarios.	47
Fig 14. Interfaz correspondiente a la visualización de las trazas por usuarios.....	48
Fig 15. Gráfica de No Conformidades de las pruebas funcionales.	52
Fig 16. Pruebas de Caja Blanca con la técnica del Camino Básico.....	53
Fig 17. Grafo obtenido del código seleccionado.....	53

Introducción

La televisión, desde sus orígenes se ha caracterizado por ser un medio de comunicación social de masas, puesto que ha sido y continúa siendo una referencia clave para el ciudadano en su conjunto de actividades cotidianas. Conocer su entorno social, formar grandes temas de la actualidad que definen la opinión pública, establecer nuevas formas de vínculo y de relaciones sociales, son unos de los tantos rasgos que caracterizan a este tipo de comunicación. Sin embargo, la televisión ha evolucionado de forma notable a lo largo del tiempo, ha mejorado la calidad de la imagen, sonido y ha aumentado la oferta de canales difundidos; surgiendo así la televisión digital. Conocida mundialmente como DTV (responde a las siglas del inglés *Digital Television*) la digitalización de la televisión no es más que la evolución de las emisiones tradicionales al formato digital **(Morris, y otros, 2005)**.

Desde el punto de vista técnico, la DTV posee mejor uso del espectro. Esta característica técnica se traduce en más canales, mejor calidad de imagen, un formato de pantalla más amplio, mejor calidad de sonido, televisión personalizada, además ofrece servicios de datos adicionales, tales como: guías electrónicas de programación, servicios móviles y aplicaciones interactivas, que no son posibles desde la tradicional televisión analógica. **(Morris, y otros, 2005)**

Una de las principales características de esta nueva televisión es la interactividad, más conocida como Televisión Interactiva o ITV (responde a las siglas del inglés *Interactive Television*), con la que se pretende convertir al usuario en un ente más activo mediante el enriquecimiento de su experiencia **(Morris, y otros, 2005)**. Debido a esta nueva propiedad de la televisión, la tendencia de los nuevos productores que están al corriente de las nuevas tecnologías es diseñar contenido interactivo y ofrecerlo en forma de aplicaciones acompañando sus transmisiones y/o programas de televisión **(Jiménez, 2009)**.

Como plantea **(Benoit, 2008)** la interactividad es un pilar importante para la DTV, como lo es para la Televisión sobre IP o IPTV (responde a las siglas del inglés *Internet Protocol Television*), la misma es un sistema que se emplea para distribuir señales de televisión y video a través de Internet con conexión de banda ancha sobre el protocolo TCP/IP¹, esto supone, la transmisión de información a través de una red segura. Las plataformas interactivas son parte de estos sistemas, las mismas garantizan el intercambio de

¹ TCP/IP: son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés *Transmission Control Protocol/Internet Protocol*). Permite a dos anfitriones establecer una conexión e intercambiar datos.

información entre la plataforma y los usuarios del sistema, además de incorporar servicios variados. Dichos servicios son accedidos por los usuarios, dígame cartelera, estado del tiempo, videoconferencias, mostrar sus canales favoritos en mosaico, establecer una comunicación entre varios usuarios mediante una aplicación, entre otros.

La IPTV estimula a los clientes a participar del contenido, aumentando el intercambio de información entre ellos. El uso del protocolo TCP/IP para la transmisión de video permite que los recursos dedicados a esta función sean utilizados tradicionalmente por estos operadores, lo que no supone una gran inversión en equipamiento de red para esta tarea. Estas transmisiones de flujos de datos permiten la integración de los clientes o usuarios con los proveedores de los contenidos, lo que le facilita la posibilidad de ofrecer varios servicios en un solo paquete integrado. A su vez, le proporciona al cliente la personalización de las mismas, la cual reciben en función de sus hábitos de consumo, permitiéndole decidir qué es lo que quiere ver y cuándo.

Este tipo de televisión puede ser accedida por múltiples dispositivos, la cual no está limitada al uso del televisor. Los consumidores, pueden acceder al servicio a través de la computadora, desde cualquier dispositivo móvil, incluso desde los televisores avanzados que no tienen la necesidad de contar con una caja decodificadora para poder visualizar la señal digital. **(Gallego, 2008)**

El avance más importante que ofrece este tipo de televisión es la interactividad, la cual abre la posibilidad de nuevos tipos de servicios que anteriormente no estaban disponibles en otras redes. Además, la capacidad que tiene de comunicación bidireccional, la cual permite a los proveedores desarrollar una serie de aplicaciones interactivas en las que no es necesario un desarrollo adicional para obtener el canal de retorno, ya que el usuario dispone de un módem o router con capacidad para comunicarse desde su casa con el proveedor. Han surgido así las plataformas de distribución de televisión sobre IP tanto para visualización en televisor, como para computadoras y dispositivos móviles. **(García, 2010)**

Actualmente los proveedores de servicios de televisión digital no solo se limitan a la transmisión de canales, sino que invierten grandes cantidades de dinero en lograr que sus usuarios disfruten de la interactividad. Cuba, cuenta con una infraestructura televisiva basada totalmente en tecnologías analógicas, y comenzó a dar sus primeros pasos hacia las tecnologías digitales. Con la colaboración de entidades, ya sean nacionales o extranjeras, el Instituto Cubano de Radio y Televisión (ICRT) ha posibilitado la incorporación de sistemas, tecnologías, profesionales y consultores que ayudan al desarrollo, implementación y capacitación en los temas de la televisión digital en la isla.

Una de estas entidades es la Universidad de las Ciencias informáticas (UCI), que cuenta con un centro de desarrollo que trabaja las líneas de procesamiento de imagen y video, incluyendo los elementos relacionados con la televisión, lleva por nombre Geoinformática y Señales Digitales (GEYSED), perteneciente a la Facultad 6. En el mismo se desarrolla el proyecto Sistema de Transmisión de Canales Virtuales (STCV), el cual aporta mejoras en la búsqueda de la calidad y la automatización de los procesos de planificación, monitorización y transmisión de contenidos audiovisuales, incorporando la interactividad entre los usuarios y los proveedores de contenidos, por solo mencionar uno de los servicios que ofrece. Dicho sistema cuenta actualmente con una plataforma web para la televisión digital interactiva que no se adapta a las nuevas especificaciones que necesita el proyecto.

En el análisis realizado a la plataforma existente, se detectó que para agregarle funcionalidades es necesario modificarle el código, pues la misma no se diseñó para que se adaptara y reconociera aplicaciones nuevas sin que perdiera la calidad de sus servicios. Además los perfiles de usuarios deben ser creados previamente por el administrador del sistema, lo que limita el nivel de interacción de los usuarios con la plataforma.

Dicha plataforma solo puede ser visualizada correctamente a través de computadoras y televisores analógicos, este último conectado a una caja decodificadora. Esta característica de la plataforma evidencia una limitante en correspondencia con las nuevas tecnologías móviles y con el desarrollo que presentan los sistemas de este tipo. Además los usuarios finales del sistema ven afectada su experiencia de navegación mediante sus dispositivos móviles, ya que no son capaces de acceder a los servicios interactivos a través de los mismos. Esto encarece el nivel de aceptabilidad de los clientes, lo que influye en el acceso de estos al sistema para visualizar los archivos multimedia que son transmitidos.

Ante la problemática planteada se deriva el siguiente **problema de la investigación**: ¿Cómo facilitar la accesibilidad desde distintos dispositivos móviles a los servicios interactivos del Sistema de Transmisión de Canales Virtuales?

Se define como **objetivo general**, desarrollar una plataforma web interactiva que permita a los usuarios del Sistema de Transmisión de Canales Virtuales acceder desde distintos dispositivos móviles a los servicios interactivos.

Se centra el estudio de la investigación en la interactividad en las plataformas web, lo cual constituye el **objeto de estudio**. Acotando el **campo de acción** a la automatización de la interactividad en la plataforma web interactiva del Sistema de Transmisión de Canales Virtuales.

Para cumplir el objetivo general de la investigación se trazaron las siguientes **preguntas científicas**:

1. ¿Cuáles son los elementos fundamentales de las plataformas web interactivas de televisión sobre IP?
2. ¿Cuáles son los diferentes servicios interactivos que existen?
3. ¿Cómo implementar una plataforma web interactiva que permita la gestión de servicios que sean accesibles desde distintos dispositivos móviles?
4. ¿Qué resultados se obtendrán al aplicar técnicas y/o métodos ingenieriles a la plataforma web interactiva?

Para dar cumplimiento a las preguntas formuladas anteriormente se hace necesario desarrollar las siguientes **tareas de investigación**:

1. Caracterización de los principales elementos que componen las plataformas web interactivas de los sistemas de televisión sobre IP y de los servicios que ofrecen.
2. Caracterización de los niveles de interactividad y de los diferentes servicios interactivos.
3. Caracterización de soluciones similares existentes a nivel nacional e internacional a fin de identificar funcionalidades que brindan las plataformas web interactivas sobre redes IP.
4. Selección de las herramientas y tecnologías a utilizar para el desarrollo de la plataforma.
5. Realización del diseño del software de la plataforma web interactiva.
6. Realización de la implementación de la plataforma web interactiva para el Sistema de Transmisión de Canales Virtuales.
7. Realización de las pruebas de software para validar el correcto funcionamiento de la misma.

El desarrollo de la investigación está dirigido por **métodos científicos** que se aplicaron a lo largo de la misma.

Métodos Teóricos:

- **Analítico - Sintético:** utilizado con el objetivo de analizar la bibliografía disponible relacionada con el proceso de funcionamiento de las plataformas web que posean servicios interactivos y las metodologías

existentes para la construcción de los artefactos necesarios para elaborar la solución, así como para realizar una síntesis de la bibliografía.

El presente trabajo de diploma está estructurado en 4 capítulos:

Capítulo 1: “Fundamentación Teórica”: En este capítulo se realiza la elaboración del marco teórico de las plataformas de televisión sobre IP y los servicios interactivos. Además, de efectuar una evaluación de los resultados obtenidos en el trabajo precedente.

Capítulo 2: “Características de la solución propuesta”: En este capítulo se definen las herramientas y lenguaje de programación a utilizar, así como la metodología de desarrollo más recomendable. Se especifican los requisitos funcionales y no funcionales que cumple la solución. Se presenta el diagrama de Casos de Uso del Sistema, se describen los actores y se especifican los Casos de Uso del mismo.

Capítulo 3: “Diseño e implementación de la solución propuesta”: En este capítulo se exponen los artefactos del diseño y de la implementación de la solución. Se define la estructura del diseño de la aplicación desarrollada, y para ello se representa el diagrama de clases del diseño, el diagrama de despliegue y el diagrama de componentes del sistema. Por último se presentan los estándares de codificación utilizados en la implementación de la solución.

Capítulo 4: “Pruebas del sistema”: En este capítulo se exponen los resultados de las principales pruebas realizadas a la aplicación para verificar su correcto funcionamiento de acuerdo a los requisitos establecidos.

Capítulo 1. Fundamentación teórica

En el presente capítulo se expondrán las principales definiciones relacionadas con la investigación. Además, se realiza un estudio de las principales tendencias existentes sobre las plataformas que utilizan televisión sobre IP a nivel nacional e internacional, presentando una propuesta de solución del problema planteado.

1.1 Interactividad

En la presente investigación, se consultaron diferentes conceptos de interactividad de los autores **(Rost, 2014)** y **(García, 2010)**. Los mismos plantean que: *“La capacidad gradual y variable que tiene un medio de comunicación para darle a los usuarios o lectores un mayor poder tanto en la selección de contenidos como en las posibilidades de expresión y comunicación.”* Además que: *“La interactividad es la forma en la que los mensajes de una secuencia se relacionan entre ellos, y especialmente la forma en la que mensajes posteriores influyen en mensajes anteriores.”*

Estos conceptos se consideran fundamentales para la presente investigación, por lo que se toman como base de conocimientos en el desarrollo investigativo. Además, se adicionan a los mismos que, la interactividad es el intercambio directo de mensajes entre medios de comunicación o personas, en la que cada cual adapta su comportamiento y acciones, dándole la posibilidad de expresarse y comunicarse.

Grados de Interactividad

Desde un punto de vista de servicio y aplicación, actualmente se han definido tres perfiles, permitiendo a los fabricantes desarrollar una serie de productos que proporcionen diferentes funcionalidades. Cada perfil puede coexistir con los demás y se corresponde con un nivel de interactividad según **(Jiménez, 2009)**:

- **Enhanced broadcast profile**: Nivel 1 de interactividad. Interactividad local. Está presente en aplicaciones como el teletexto digital y grabador de video personal, que obtienen la información requerida para el funcionamiento de un flujo de datos.

- **Interactive broadcast profile**: Nivel 2 de interactividad. Interactividad sin respuesta por el canal de retorno. Presentes en aplicaciones como el voto *“on-line”* permiten al usuario proporcionar información de respuesta en una sola dirección, sin una contestación directa sobre el mismo enlace, ya que la información vuelve por el canal de radiodifusión.

- **Internet acces profile:** Nivel 3 de interactividad. Acceso total a Internet. Aplicaciones como el T-Mail, navegación Web o juegos on-line, permiten al usuario adquirir información de fuentes externas al *stream*² emitido por el canal de retorno. Suele necesitar conexión de banda ancha.

La plataforma propuesta contiene aplicaciones interactivas, las cuales presentan todos los niveles de interactividad descritos anteriormente. Por lo que se considera que la plataforma con las aplicaciones integradas podrá tener todos los niveles de interactividad.

Principales funcionalidades de los sistemas interactivos según su nivel de interactividad:

En este apartado se describe las principales aplicaciones que soportan los sistemas interactivos según los diferentes niveles de interactividad. Normalmente los más utilizados son los del nivel de interactividad uno ya que son los que principalmente prioriza el proveedor de servicios televisivos, según **(Jiménez, 2009)**:

- **Nivel 1 de interactividad.**

Guía Electrónica de Programación o EPG (*responde a las siglas del inglés, Electronic Program Guide*).

Es una agenda donde se presenta la programación prevista de los canales en vivo contratados. El alcance de la misma en general llega a ser hasta aproximadamente 14 días en adelante. Además de sustituir la revista mensual que se envía a los clientes con esta información, esta facilidad agiliza el cambio de canales. A su vez se puede acceder a detalles de los metadatos³ de los programas actuales, pasados o futuros: Duración, Género, Origen, Director, Actores entre otras cosas.

Grabación de Video Personal o PVR (*responde a las siglas del inglés, Personal Video Recorder*).

Permite a los suscriptores del servicio de IPTV la grabación de programas en vivo para luego ser reproducidos emulando las funcionalidades de un video grabador. Durante la reproducción se dispondrá las funciones de *Play* (comenzar), *Pause* (pausar), *Stop* (parar), *REW* (reversa), *FF* (avance rápido). Esta facilidad puede estar integrada al EPG, en ese caso se podrá programar la grabación marcando uno o más programas anunciados en la guía para ser almacenados en su momento de emisión. Por otra parte es útil que se disponga de un acceso rápido desde el control remoto para el comienzo de grabación en forma inmediata.

² *Stream*: flujo de datos emitidos por una red.

³ Metadatos: son datos que describen a otros datos.

- Nivel 2 de interactividad.

Pagar por Ver o PPV (*responde a las siglas del inglés, Pay Per View*). Mediante el PPV se brinda a los subscriptores la facilidad de contratar la posibilidad de ver un programa en particular. Es una característica que generalmente es utilizada para eventos deportivos o conciertos que están siendo emitidos en vivo. La modalidad conjuga características de transmisión de TV en vivo y exige una contratación del contenido como en los casos de video bajo demanda.

T-Encuestas. El servicio de tele-encuesta, permite elegir una de las opciones disponibles, brindando la posibilidad de elegir por mayoría qué programa será retransmitido o incluso qué quiere que suceda en una serie de televisión. Además el cliente, puede responder simplemente a una encuesta hecha por el proveedor de servicios y votar para alguna clase de evento local o privado.

- Nivel 3 de interactividad.

T-Comunicación. En el ámbito de la comunicación encontramos tanto los clientes de correo por TV como los espacios de mensajería instantánea. También son interesantes las aplicaciones de mensajería instantánea similar a las utilizadas para computadoras, que permitiera, mediante el uso de una cámara, poder hacer video-llamadas.

T-Juegos. Aquí cabe destacar que los juegos pueden pertenecer a cualquiera de los niveles de interactividad, todo está en la forma en que este funcione. Para los juegos del nivel 3, deben ser juegos participativos en la que 2 o más usuarios intervengan.

T-Salud. En el ámbito de la salud se ofrecen varias nuevas posibilidades. Una de ellas es la posibilidad de reservar consulta con su médico de cabecera, pudiendo introducir algunos de los síntomas para avanzar el trabajo del doctor. La tele-consulta es otra posibilidad que consiste en estar en contacto con su doctor sin necesidad de desplazarse al centro de salud. Este sistema de tele-consulta permite la realización de estadísticas que permitirían una mejor actuación de cara, por ejemplo, a la detección de un brote de gripe.

1.2 Elementos fundamentales de las plataformas de televisión sobre IP

En la actualidad los usuarios de las tecnologías de la información y telecomunicaciones (TIC's) cada vez son más exigentes respecto a los servicios y los diferentes equipos que presenta la tecnología. Esta exigencia ha dado lugar a la convergencia de servicios telemáticos, la cual ha permitido que distintos

dispositivos que recientemente se han convertido de uso cotidiano, puedan ser capaces de realizar distintas operaciones, mientras que antes eran necesarios varios equipos para realizar las mismas tareas.

El autor **(Bustamante de los Ríos, 2014)** plantea que para comprender con mayor facilidad el funcionamiento de un sistema interactivo bajo la red de datos es importante conocer los diferentes elementos que conforman una plataforma de televisión sobre IP.

Una plataforma de televisión sobre IP contiene los siguientes bloques:

- *Head End* o cabecera.
- *VoD* o sistema de video bajo demanda.
- *Middleware* o mediador.
- *Digital Rights Management* o sistema de control de derechos.
- *Set-Top Box* o caja decodificadora.
- *Home Gateways* o puerta de enlace.

De los elementos mencionados anteriormente, la plataforma web interactiva que se desarrolla posee el mediador y la caja decodificadora o *STB*⁴(responde a las siglas del inglés *Set-Top-Box*). A continuación se describen las características de cada uno de ellos que son utilizadas en el desarrollo de la plataforma.

***Middleware* o mediador**

El *Middleware* es una plataforma de gestión de aplicaciones que interactúa con la red de acceso y los dispositivos utilizados por los usuarios finales para permitir el aprovisionamiento y la distribución de servicios de televisión interactivos. **(Telecomunicaciones, 2005)**

Es una plataforma informática que provee las herramientas y tecnologías necesarias para ofrecer servicios de video haciendo uso de su infraestructura. El *middleware* interactúa con los STB por lo que ambos deberán contar con aplicaciones diseñadas en la modalidad cliente (en el STB) servidor (en el *Middleware*).

⁴ STB: caja decodificadora

El *Middleware* es el responsable de asegurar la completa interoperabilidad del servicio de video. No está limitado a una única operación en el sistema, pero debe ser capaz de comunicarse directamente con cada componente para proveer soluciones de video. En el servicio de PVR, el usuario que desea ordenar la grabación en un determinado día y hora, realiza la solicitud al *middleware*, estableciendo una sesión. El *middleware* lo autorizará o no en función de la cuenta del usuario y su perfil. En caso afirmativo le ordenará al servidor de video que realice la grabación de la media seleccionada en el horario solicitado por el cliente. Todo esto le permite a la plataforma contar con seguridad para sus servicios, aplicaciones y funciones que ofrece.

STB

Son dispositivos de terminales de abonado, los cuales serán los que adapten las señales provenientes de la red de datos en señales visibles en un televisor convencional. Los STB presentan una interfaz Ethernet hacia la red de datos, que en general estará conectada al módem ADSL⁵. En algunos casos los propios STB cuentan con el módem ADSL integrado. A su vez esos dispositivos poseen un conector RCA⁶, el cual permite la conexión con el televisor.

1.3 Servicios Interactivos

Se pueden establecer tres categorías o tipos de servicios interactivos:

- **Servicios de información:** Son aquellos que ofrecen una información independiente de la programación audiovisual que se está emitiendo en ese momento.
- **Servicios interactivos con conexión:** Son aquellos que complementan con información suplementaria la programación audiovisual emitida.
- **Servicios transaccionales:** Son aquellos que ofrecen la posibilidad de enviar y recibir información de forma personalizada y exclusiva.

⁵ ADSL: codificador –decodificador de señales digital-analógicas utilizado para conectarse a internet y que tiene la propiedad de poder usar el teléfono y la computadora al mismo tiempo para conectarse por el mismo canal.

⁶ RCA: conector utilizado para los televisores analógicos conformados principalmente para dos señales de audio y uno de video.

Por otra parte, los servicios interactivos se implementan por medio de aplicaciones interactivas. Se define como una aplicación interactiva aquellos programas adicionales a los contenidos de televisión a los que puede accederse y puede ejecutar un descodificador interactivo. El usuario es el que decide si quiere o no ver las aplicaciones interactivas. Con el fin de comunicar al usuario la posibilidad de acceso a aplicaciones interactivas los operadores o canales de televisión presentan un pequeño menú interactivo que indica al usuario que puede ver una aplicación o un grupo de aplicaciones interactivas.

En la figura 1 que se muestra a continuación se detallan los tipos de servicios interactivos planteados anteriormente, así como algunos ejemplos (SiTvi, 2009) .

■ Tipos de servicios			
<p>Informativos / interactivos</p>	<p>No necesitan canal de retorno</p> <p>Información unidireccional</p> <p>Sincronizado o no con Audio /Video</p>		<p>Ejemplos de aplicaciones</p> <ul style="list-style-type: none"> -Información de tráfico -Información tiempo -Información de empleo -Agenda de eventos
<p>Informativos / interactivos con conexión</p>	<p>Comunicación bidireccional</p> <p>Ampliar información</p> <p>Sincronizado o no</p>		<p>Ejemplos de aplicaciones</p> <ul style="list-style-type: none"> -Información catastral -Solicitud de padrón -Solicitud recogida de muebles
<p>Transaccionales</p>	<p>Información por canal seguro y personalizada</p> <p>Comunicación bidireccional</p> <p>Sincronizado o no</p>		<p>Ejemplos de aplicaciones</p> <ul style="list-style-type: none"> -Servicios que requieren el acceso con DNI-E -Información de la vida laboral

Fig 1. Tipos de servicios interactivos.

1.4 Servicios interactivos que ofrecen los sistemas IPTV

Desde hace algún tiempo la televisión sobre IP es la tecnología que está revolucionando el mercado audiovisual. La misma por sus características físicas y técnicas permite ofrecer a los usuarios interesantes prestaciones. Entre las principales ventajas y servicios de IPTV se destacan (Lindquist, 2010):

Televisión Interactiva

La interactividad constituye uno de los mayores retos de la televisión. Como se ha dicho con anterioridad, en IPTV el video es transmitido por la red de datos, la cual es usada a su vez como canal de retorno permitiendo que el cliente interactúe de manera directa y en tiempo real con el proveedor de los servicios. La posibilidad de contar con un canal de retorno de banda ancha basado en IP permite un amplio número de servicios y la personalización de estos teniendo en cuenta los gustos y preferencias del usuario. La televisión interactiva puede adoptar muchas formas, que van desde la simple pulsación de un botón en un menú hasta planes más sofisticados.

De acuerdo al nivel de interacción puede clasificarse de dos formas: local y global.

- **Interactividad local:** se refiere a los servicios interactivos que no necesitan del uso del canal de retorno.
- **Interactividad global:** se refiere a los servicios interactivos con la utilización del canal de retorno.

Algunos ejemplos de televisión interactiva son:

- Selección de ángulo de cámara, donde el espectador puede escoger diferentes ángulos de cualquier evento que se transmita en vivo.
- Votación/sondeos de opinión, donde el resultado de un caso que se esté abordando en directo a través de la televisión está determinado por una votación de la audiencia.
- Anuncios/respuesta, donde los espectadores pueden solicitar más información sobre un producto o servicio anunciado.
- Publicidad personalizada, el hecho de que en IPTV exista un canal bidireccional permite a los usuarios determinar y seleccionar las áreas de interés sobre las que les gustaría recibir ofertas de publicidad.

Broadcast y Multicast

La IPTV trae asociado los servicios de *Broadcast* y *Multicast*. El *broadcast* permite realizar la transmisión de los contenidos a todos los nodos de una red a la misma vez. Por su parte la tecnología *multicast* garantiza un uso óptimo de la red del operador en función de los consumos del usuario. El uso eficiente de los recursos de red evita la replicación innecesaria de flujos. Esta tecnología está directamente orientada hacia las aplicaciones multimedia y tiene como característica la escalabilidad puesto que permite que los servicios y aplicaciones sean accesibles para un gran número de participantes.

Medición de audiencia

IPTV al tratarse de un perfil cerrado, donde el proveedor maneja tanto la red de transmisión como los contenidos, permite tener control sobre la audiencia de toda la comunidad sobre materiales específicos. La medición de audiencia dice cuántas personas están viendo un programa de televisión en un momento dado e indica la preferencia relativa de los espectadores hacia un determinado espacio con respecto a otros que se emiten simultáneamente.

Navegación Web

Haciendo uso del televisor y el STB, la Televisión sobre IP posibilita la realización de funciones normalmente asociadas a un navegador web de una computadora, como son acceso al correo electrónico, canales de chat, compras y consulta de facturas en línea, votaciones, juegos interactivos, publicidad interactiva y servicio al cliente.

Videoconferencia y Educación a distancia

IPTV permite desarrollar servicios como la videoconferencia. La videoconferencia es una tecnología que proporciona la comunicación directa entre dos o más personas que se encuentran ubicadas en diferentes lugares a través del envío y recepción de audio, video y datos.

Esta tecnología permite realizar una conexión a cualquier parte del mundo, sin la necesidad de trasladarse a un punto de reunión y es muy aplicada en la educación a distancia. Hoy en día esta modalidad educativa funciona enviando al estudiante el material de estudio a través de diferentes medios como son el correo electrónico, internet e IPTV. A la enseñanza desarrollada mediante las nuevas tecnologías de telecomunicaciones se le conoce como *e-learning*.

La televisión de alta definición

La televisión de alta definición se caracteriza por emitir las señales televisivas en una calidad digital superior a los sistemas ya existentes. La pantalla utiliza una proporción del aspecto 16:9. La alta resolución de las imágenes (1920 píxeles × 1080 líneas o 1280 píxeles × 720 líneas) permite mostrar mucho más detalles comparado con la televisión analógica o de definición estándar.

Vista previa de los canales (Mosaico)

IPTV posibilita la presentación de programas en mosaico proporcionando una vista previa de cada uno de los canales preferidos por el usuario.

Servicios personalizados

Con IPTV se pueden obtener servicios personalizados a la altura de las demandas actuales de los clientes. Crear y reproducir la lista de los programas de videos favoritos y acceder a los programas más populares o más vistos, es ya una realidad en estos tipos de televisión.

Streaming de video

El video *streaming* permite la transmisión de contenido multimedia sin necesidad de descargarlo en la computadora del usuario. Para la reproducción del mismo el usuario accede a la página que lo contiene y desde su computadora inicia la reproducción, el servidor envía la transmisión de forma continua al tiempo que el video se visualiza. Con la recepción de una pequeña parte, el usuario podrá comenzar a visionar el contenido, mientras continúa recibiendo la corriente de datos que irá visualizando posteriormente de forma continuada e ininterrumpida. El *streaming* de video se puede usar de muchas formas, un ejemplo de ello es la educación a distancia.

Integración

Es una forma de ofrecer varios servicios en un paquete integrado para muchas empresas. La IPTV permite combinar la transmisión de televisión con la transferencia de datos sobre la misma red, esto significa reducción de los gastos en redes y que el consumidor tiene que tratar con un proveedor en lugar de dos.

1.5 Soluciones existentes

Para lograr una mejor comprensión de las características de la plataforma a desarrollar, se hace necesario realizar un análisis de algunas plataformas con características similares. A continuación se expone el análisis desarrollado de los sistemas homólogos.

Plataforma Punto Azul:

Se trata de una solución integral y flexible para plataformas de televisión sobre redes IPTV. Además, permite el acceso a servicios de valor añadido y aporta interactividad a los servicios de televisión convencional. Con la tecnología multi-pantalla el usuario podrá tener acceso al sistema donde y cuando quiera, lo que unido a los servicios interactivos, potencia el uso del sistema y transforma una operación de televisión en un sistema global de entretenimiento muy atractivo para el usuario.

La plataforma al permitir el uso de servicios de televisión interactiva en redes IP, facilita el acceso a contenidos exclusivos a aquellos usuarios que no poseen su propia infraestructura de red. De este modo, es posible proporcionar servicios como EPG, redes sociales, video conferencia, PVR a través de redes gestionadas o incluso mediante la red global de internet.

La plataforma y su portal de televisión, no solo se limitan a STB. La plataforma también es capaz de soportar otros dispositivos como *smartphones*, tabletas, consolas de videojuegos y ordenadores con diferentes sistemas operativos. Esta característica permite ofrecer a los clientes la posibilidad de visionar sus contenidos preferidos en cualquier sitio y cuando lo deseen. **(Safeview, 2015)**

Arantia TV

Engloba en una única plataforma IPTV todos los servicios necesarios para dar respuesta a las necesidades de sus clientes en todo tipo de proyectos, incluyendo hoteles, hospitales, operadores, empresas y construcciones residenciales, proporcionando además nuevas herramientas para mejorar las operaciones del negocio. Arantia ofrece a sus consumidores los elementos necesarios para el despliegue de servicios IPTV, Video Bajo Demanda y todo tipo de servicios interactivos de valor añadido. **(Arantia, 2015)**

Es una plataforma abierta, escalable y totalmente personalizable en función de las necesidades de cada proyecto, la misma posee:

- Arquitectura abierta diseñada para facilitar el despliegue de nuevos servicios IPTV y las labores de gestión y operación de las aplicaciones.
- Arquitectura modular y escalable en número de dispositivos, usuarios y servicios.
- Flexibilidad para soportar diferentes escenarios y modelos de negocio.
- Interfaz de usuario totalmente personalizable y adaptable.

Arantia facilita un nuevo mundo de posibilidades para operadores, empresas y proyectos singulares mediante el despliegue de nuevos servicios IP, permitiendo de esta forma:

- Incrementar la oferta de servicios de valor añadido como una nueva fuente de generación de ingresos.
- Incrementar la satisfacción de los usuarios.

- Fácil despliegue de servicios y contenidos en directo, programados y bajo demanda en una única plataforma.
- Desarrollo de todo tipo de servicios interactivos.
- Integración total en una única interfaz de usuario de fácil manejo de todas las aplicaciones potencialmente demandadas.

Plataforma web para la televisión digital interactiva:

El STCV cuenta actualmente con una plataforma web para la televisión digital interactiva. La misma es adaptable a los diferentes tipos de STB y *middleware*, que tengan nivel 3 de interactividad gracias a su arquitectura orientada a *plugins*, la cual permite la utilización de diversos STB. Dicha plataforma no es escalable, es decir, si se desea añadir una aplicación de valor agregado, los desarrolladores deberán modificar el código fuente de la misma. Dicha plataforma no está orientada a funcionar en una gran diversidad de dispositivos, principalmente móviles, esto atenta contra la gran demanda que existe en el mundo, de su utilización.

1.5.1 Análisis de las soluciones existentes.

A continuación se muestra una tabla resumen de las principales características que se tomaron en cuenta para el estudio de las plataformas.

Tabla 1. Características principales de las plataformas analizadas

Plataformas	Interactividad	Accesibilidad (dispositivos móviles)	Incorporación de servicios interactivos	Integración con el STCV
Plataforma Punto Azul	X	X	X	
Arantia TV	X	X	X	
Plataforma web para la televisión digital interactiva	X			X

La Plataforma Interactiva para la Televisión Digital no permite la agregación de funcionalidades, ya que es necesario modificarle el código, pues la misma no se diseñó para que se adaptara y reconociera aplicaciones nuevas sin que perdiera la calidad de sus servicios. Dicha plataforma solo puede ser visualizada correctamente a través de computadoras y televisores analógicos, este último conectado a

una caja decodificadora; lo que representa una desventaja para los usuarios que posean dispositivos móviles, tales como: teléfonos celulares o tabletas, ya que no posee un diseño adaptable a dichos dispositivos.

Las demás plataformas cumplen con los objetivos que se persiguen, pero ninguna se integra al Sistema de Transmisión de Canales Virtuales, ya que para eso deben contar con una aplicación de televisión que se conecte al subsistema de transmisión para obtener el flujo televisivo que emite en ese momento. Este flujo también es utilizado por las aplicaciones de cartelera y vista de canales en mosaico, las cuales están presentes en la plataforma a desarrollar.

El subsistema antes mencionado, transmite según la planificación obtenida del subsistema de programación, y le brinda la posibilidad al usuario de seleccionar los canales disponibles de su preferencia para ubicarlos en la vista de canales en mosaico. Por tanto, las plataformas Punto Azul y Arantia TV, al no contar con la aplicación de televisión, no se contemplan dentro de la solución.

Por lo anteriormente expuesto se hace necesario desarrollar una plataforma web interactiva que les permita a los usuarios del Sistema de Transmisión de Canales Virtuales acceder desde distintos dispositivos móviles a los servicios interactivos.

1.6 Conclusiones del capítulo

En este capítulo se abordaron todos los elementos teóricos que sustentan la solución del problema, los cuales conforman la guía metodológica a seguir en el desarrollo del presente trabajo. Igualmente, con el estudio de los elementos presentes en este capítulo, se solucionaron las tareas de investigación 1, 2 y 3, lo que se evidencia en:

- El estudio y caracterización de los niveles de interactividad permitió identificar las principales características que son necesarias para que el sistema tenga los tres niveles de interactividad integrado en su funcionamiento básico.
- La caracterización de los elementos que conforman los sistemas IPTV permitió determinar las posibles funcionalidades del sistema a desarrollar.
- Debido a que no se encontró un sistema que solucione lo planteado en la situación problemática, es necesario la creación de una Plataforma Web Interactiva que le permita a los usuarios del Sistema de

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN PROPUESTA

Transmisión de Canales Virtuales acceder desde distintos dispositivos móviles a los servicios interactivos.

Capítulo 2. Características de la solución propuesta

En el presente capítulo se presentan las herramientas, lenguajes y tecnologías utilizadas para la construcción de la solución. A su vez, se describen los principales artefactos que comprenden la modelación del dominio y se realiza la captura de los requisitos mediante una técnica de recopilación de información. Se detallan además los requisitos no funcionales presentes en el sistema.

2.1 Lenguajes, tecnologías y herramientas

Para desarrollar la plataforma propuesta en la presente investigación, se hace necesario indagar sobre los lenguajes, herramientas y tecnologías a ser utilizados en el desarrollo de la propuesta de solución. A continuación se procede con el estudio y selección de las mismas.

2.1.1 *Metodología de desarrollo*

Una metodología de desarrollo de software representa un marco de trabajo que tiene entre sus funciones guiar, planificar, estructurar, controlar, manipular y dirigir el proceso de desarrollo de sistemas de información. Surge ante la necesidad de trabajar mediante el uso de procedimientos, técnicas, herramientas y documentos durante el desarrollo de software. **(Pressman, 2010)**

Las metodologías se clasifican en dos tipos: ágiles o ligeras y pesadas o tradicionales. Las ágiles tienen como principios el trabajo en equipo como arma fundamental; el avance del trabajo enmarcándose solamente en los elementos necesarios que este exige. Se enfocan en la constante interacción con el cliente haciéndolo parte del equipo de trabajo y en la posibilidad de cambiar todo lo que debe ser cambiado de forma que se alcance la mayor fiabilidad y calidad en el producto que se desarrolla.

Por su parte, las metodologías tradicionales o pesadas imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Estas se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas, notaciones para el modelado y documentación detallada. A su vez, se enfocan en el detalle de cada proceso y tarea que se debe desarrollar, en las herramientas a utilizar, genera una documentación extensa que debe justificar a cada paso de avance y además adelanta la puesta en práctica de la solución.

Teniendo en cuenta las necesidades para la solución del problema investigativo, la línea de desarrollo del proyecto STCV y la sugerencia por parte del cliente, se utiliza la metodología de desarrollo Proceso Racional Unificado o *RUP* (responde a las siglas del inglés *Rational Unified Process*). La misma define fases, principios, etapas o flujos de trabajo, disciplinas de soporte, entre otros elementos, para mejorar y organizar el trabajo desde el comienzo **(Anónimo, 2015)**. Además, desde un inicio están bien definidos los requisitos de software, sobre los que no ocurrirán cambios constantes.

2.1.2 Framework

En el desarrollo de software, un *framework* o “marco de trabajo”, es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Selección del *framework*

Se elige el *framework* Express ya que es muy útil para desarrollar aplicaciones web sobre node.js. Además de ser robusto, rápido, flexible y muy simple. Entre otras facilidades que ofrece se utiliza para motores de plantillas (Jade, EJS, JinJS), *Middleware* via *Connect* **(Anónimo, 2015)**. En la aplicación se utiliza especialmente para el ruteo de las URL (*GET* y *POST*).

2.1.3 Node JS v0.10.24

Node JS es una plataforma construida en tiempo de ejecución de JavaScript para construir fácilmente aplicaciones rápidas de red escalables. Node es basado en eventos, esto hace que sea ligero y eficiente, ideal para aplicaciones en tiempo real de datos intensivos que se ejecutan a través de dispositivos distribuidos. En algunos casos de construcción de sistemas que utilizan Node, se puede ver la utilización de clases, lo que quiere decir que en su funcionamiento no es absolutamente orientado a eventos.

Node provee un entorno de ejecución y un conjunto de librerías básicas, o módulos nativos, a partir de las cuales se pueden crear aplicaciones orientadas principalmente a las redes de comunicación, aunque una parte de estas librerías permite interactuar con componentes del sistema operativo. **(Kiessling, 2010)**

2.1.4 Lenguaje de Marcado de Hipertexto

HTML es un lenguaje de composición de documentos que cuenta con instrucciones especiales que son interpretadas por el navegador y como resultado se muestran textos, imágenes, videos y enlaces. Utilizando *HTML* se puede definir la apariencia de las páginas creadas de diferentes formas. Contiene una gran variedad de etiquetas que son las encargadas de mantener la estructura e indicar la semántica de un documento, convirtiéndose en un lenguaje para presentación con soporte de aspectos visuales (W3school, 2015).

Se decide utilizar *HTML 5* ya que este presenta mejoras con respecto a las anteriores versiones de *HTML*. Además incluye un nuevo conjunto de etiquetas para dar estructura a un documento *HTML*, abandonando el uso de tablas y etiquetas *div* para la maquetación de las páginas. En esta versión, los formularios son mejorados con una redefinición de nuevos tipos de elementos para la etiqueta *input*, eliminando la sequedad o la falta de tipos de datos que existía para los formularios en *HTML 4*.

2.1.5 Lenguaje Unificado de Modelado v2.0

Un lenguaje de modelado es un conjunto de símbolos y reglas que están estandarizados y se utilizan para modelar parte de un diseño de software orientado a objetos. Comúnmente son utilizados en combinación con una metodología de desarrollo de software para llegar de una especificación inicial a la implementación. (Anónimo, 2015)

Para el desarrollo del sistema se utilizó el Lenguaje de Modelado Unificado en su versión 2.0 por ser el estándar más utilizado para especificar y documentar cualquier sistema de forma precisa. A su vez es el lenguaje que se utiliza para visualizar, construir y documentar los artefactos de sistemas de software.

2.1.6 Hoja de estilos en cascadas

Hojas de estilo en cascada o *CSS* (responde a las siglas del inglés *cascading style sheets*) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con *HTML* y *XHTML* que separa los contenidos y su presentación (Eguíluz, 2015). En la actualidad se utiliza el lenguaje *CSS* para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, así como la posición de cada elemento dentro de la página.

Ventajas:

- Flexibilidad.
- Compatibilidad y continuidad.
- Separación del contenido y presentación.
- Unificación del diseño de las páginas del sitio.
- Optimización de los tiempos de carga y el tráfico del servidor.
- Permite diferenciar estilos para imprimir/visualizar en pantalla.

Se selecciona CSS en su versión 3 para la aplicación de estilos a los contenidos que se muestran en el sistema a desarrollar. Esto hace al entorno de la aplicación más atractivo y amigable para los usuarios y permite a los diseñadores reducir los costos de mantenimiento de las páginas web al disminuir la complejidad de las mismas.

2.1.7 *JavaScript*

Provee el comportamiento, tercer pilar en el actual paradigma de las aplicaciones web, el cual define a las páginas web como entes consistentes de tres partes claramente distinguibles: el contenido (*HTML*), la presentación (*CSS*) y el comportamiento (*JavaScript*). El código *JavaScript* se ejecuta sobre un ambiente anfitrión, habitualmente los navegadores web, quienes son los más comunes pero no los únicos. Esto quiere decir que posee carácter multiplataforma y además puede ser empleado en la programación de todo tipo de aplicaciones y entornos. **(Stefanov, 2008)**

JavaScript es un poderoso lenguaje prototipado orientado a objetos que actualmente se puede ejecutar en el lado servidor, en aplicaciones de escritorio y en medios enriquecidos (rich media).

2.1.8 *Entorno de Desarrollo Integrado*

Netbeans 8.0

Es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en *Java*, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso. Es un proyecto de código abierto con una gran base de usuarios, una comunidad en constante crecimiento **(Anónimo, 2015)**.

NetBeans es una base modular y extensible, usada como una estructura de integración para crear aplicaciones. Ofrece servicios comunes a las aplicaciones, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación (**Jiménez, 2015**). Entre las características de la plataforma están:

- Administración de las interfaces de usuario (menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- Administración de ventanas.

En la plataforma a desarrollar se utiliza el entorno de desarrollo NetBeans para la compilación de código *JavaScript* y para la depuración del mismo, así como para el diseño de las interfaces de usuario.

2.1.9 Visual Paradigm 8.0 (Visual Paradigm for UML v8.0 Enterprise Edition)

Las herramientas CASE (Herramienta de Ingeniería de Software Asistida por Computación) proporcionan ayuda automatizada a los ingenieros, analistas y desarrolladores durante el ciclo de vida del desarrollo de un software, garantizando una mejora en la calidad del mismo. Facilitan la realización de prototipos, el desarrollo conjunto de aplicaciones y mejoran y estandarizan la documentación. Además aumentan la portabilidad de las aplicaciones, facilitan la reutilización de componentes y posibilitan un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos. (**Anónimo, 2015**)

Se utiliza *Visual Paradigm for UML* ya que permite realizar la construcción de diferentes modelos, tales como: despliegue, componente, clases, entre otros, además de aportar una documentación detallada de los mismos, la cual está en correspondencia con la generada por el proyecto STCV.

2.1.10 Protocolo de comunicación

El protocolo de comunicación *WebSocket* permite la comunicación bidireccional entre las aplicaciones web y los servidores web a través de un solo socket TCP, además de ser la base de la comunicación en la plataforma. Esta conexión se produce en tiempo real y se mantiene permanentemente abierta hasta que se cierre de manera explícita. De manera simple, *WebSocket* permite comunicar el cliente y el servidor a través de un canal bidireccional. (**Anónimo, 2015**)

EL protocolo *HTTP* en sus comienzos en el modelo Cliente-Servidor servía las páginas de una manera sincrónica, lo que significa que por cada petición del cliente a través de un *request HTTP*, el cliente quedaba bloqueado hasta que el servidor daba una respuesta. El actual protocolo *HTTP* estándar es lento porque debe solicitar documentos a un servidor y esperar a que se envíe el documento antes de poder

mostrar una página web, mientras que el protocolo *WebSocket* funciona de modo tal que el servidor generalmente recibe las peticiones de los clientes, las procesa y envía posteriormente los resultados a través de la conexión que existe entre ellos.

La aplicación de *WebSocket* trae como beneficios comunicar los procesos del servidor de manera que las aplicaciones que corren en el servidor se comuniquen con el cliente y este último muestre la información procesada y actualizada en cierta parte del sitio web sin tener que recargarlo o sin que el cliente lo solicite explícitamente. Esto permite dotar a la web de aplicaciones escritas para este entorno más interactivo, es decir, sitios con tanta interacción asincrónica, característica que posee la plataforma. **(Anónimo, 2015)**

Características:

- El protocolo *WebSocket* es un protocolo basado en TCP independiente.
- *WebSocket* está diseñado para ser implementado en los navegadores y servidores web, pero puede ser utilizado por cualquier aplicación de cliente o servidor.
- El protocolo *WebSocket* hace posible una mayor interacción entre un navegador y un sitio web, facilitando el contenido en directo y la creación de juegos en tiempo real. Esto es posible al proporcionar una manera estandarizada para el servidor para enviar el contenido al navegador sin ser solicitado por el cliente, y lo que permite que los mensajes se pasan de nuevo adelante y hacia atrás mientras se mantiene la conexión abierta. De esta manera, una conversación de dos vías puede tener lugar entre el navegador y el servidor.
- *Websockets* es un estándar de *HTML 5*.
- La tecnología de *Websockets* se extiende por la mayoría de los servidores web más conocidos como Apache, Jetty, Tomcat, Tornado, Node.js, entre otros.

2.2 Modelo de Dominio

Al no contar con los procesos del negocio bien definidos, se decide realizar un modelo de dominio donde se plasman todos los conceptos que forman parte del mismo, creando así toda una base de conocimientos.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los conceptos que existen o los eventos que suceden en el entorno en el que trabaja el sistema. **(Pressman, 2010)**

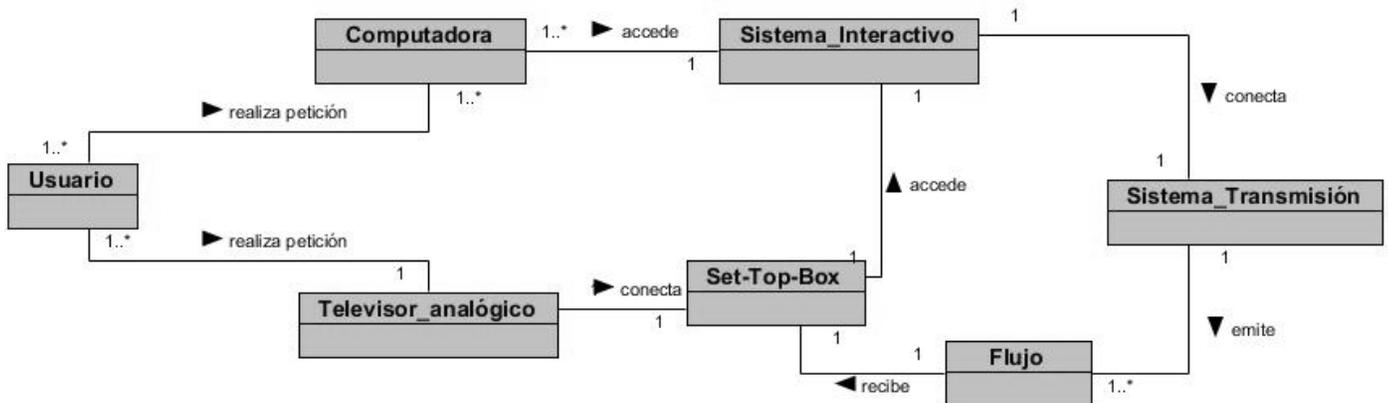


Fig 2. Modelo de dominio

2.2.1 Descripción de los conceptos que intervienen en el dominio del problema

Usuario: Es el actor del sistema encargado de realizar una petición a la computadora o al televisor analógico, para que posteriormente pueda acceder al sistema interactivo.

Computadora: Dispositivo electrónico por el cual los usuarios acceden al Sistema Interactivo.

Televisor_Analógico: Dispositivo destinado a la recepción de señales analógicas de televisión.

Set-Top-Box: Dispositivo encargado de decodificar las señales digitales a analógicas para que los televisores tradicionales puedan disfrutar de la televisión digital. Es fundamental para lograr la interactividad.

Sistema Interactivo: Es el sistema encargado de brindar una serie de servicios interactivos a los usuarios del STCV.

Sistema de Transmisión: Sistema encargado de obtener la planificación de la transmisión guardada en la base de datos y las medias del servidor NFS⁷.

Flujo: Es la transmisión de paquetes de archivos multimedia realizado por el STCV.

2.3 Requisitos

La especificación de requisitos en el proceso de desarrollo del software es de vital importancia. Tener los requisitos bien claros y definidos permite comprender desde un inicio la línea a seguir en el desarrollo y así garantizar la eficiencia y calidad del software. **(Pressman, 2010)**

2.3.1 Requisitos funcionales.

Para poder identificar qué debe hacer el sistema y entender su funcionamiento, es fundamental conocer los requisitos funcionales que el sistema debe cumplir. La técnica utilizada para la obtención de requisitos fue la entrevista al cliente de la plataforma a desarrollar. La especificación de los requisitos se pueden consultar en el Anexo 2: Especificación de requisitos.

A continuación se muestran los requisitos funcionales (RF) identificados con una breve descripción:

RF 1: Autenticar usuario.

Descripción: El sistema debe permitir que los usuarios previamente registrados se autenticuen en el mismo.

RF 2: Adicionar usuario.

Descripción: El sistema debe permitir que los administradores inserten usuarios.

RF 3: Modificar usuario.

Descripción: El sistema debe permitir que los administradores editen los datos asociados al usuario, luego de ser seleccionado.

RF 4: Eliminar usuario.

Descripción: El sistema debe permitir que los administradores eliminen el usuario seleccionado.

⁷ NFS: Sus siglas significan "Sistema de Archivos en Red".

RF 5: Buscar usuario.

Descripción: El sistema debe permitir que los administradores realicen búsqueda de usuarios registrados y conectados en el sistema.

RF 6: Modificar espacio de almacenamiento por usuario.

Descripción: El sistema debe permitir que los administradores modifiquen el espacio de almacenamiento asociado al usuario seleccionado.

RF 7: Eliminar espacio de almacenamiento por usuario.

Descripción: El sistema debe permitir que los administradores eliminen el espacio de almacenamiento asociado al usuario seleccionado.

RF 8: Modificar perfil de usuario.

Descripción: El sistema debe permitir que el usuario previamente registrado modifique su perfil.

RF 9: Eliminar perfil de usuario.

Descripción: El sistema debe permitir que el usuario previamente registrado elimine su perfil.

RF 10: Visualizar perfil de usuario.

Descripción: El sistema debe permitir que los usuarios visualicen su perfil.

RF 11: Crear aplicación de navegación de ficheros.

Descripción: El sistema debe permitir crear la aplicación de navegación de ficheros.

RF 12: Visualizar aplicación de navegación de ficheros.

Descripción: El sistema debe permitir que los usuarios visualicen la aplicación de navegación de ficheros.

RF 13: Almacenar reportes de las trazas.

Descripción: El sistema debe permitir almacenar reportes de las trazas de los usuarios.

RF 14: Visualizar reportes de las trazas.

Descripción: EL sistema debe permitir que los administradores visualicen los reportes de las trazas por usuario.

RF 15: Visualizar aplicación interactiva.

Descripción: El sistema debe permitir la visualización de una aplicación interactiva previamente instalada.

RF 16: Adicionar aplicación interactiva.

Descripción: El sistema debe permitir que los usuarios adicionen una aplicación interactiva.

RF 17: Instalar aplicación interactiva.

Descripción: El sistema debe permitir instalar aplicaciones interactivas luego que los usuarios las adicionen.

RF 18: Registrar usuario.

Descripción: El sistema debe permitir que los usuarios nuevos se registren para que puedan acceder a la plataforma.

RF 19: Visualizar archivos multimedia.

Descripción: El sistema debe permitir mostrar los archivos multimedia previamente almacenados por el usuario.

RF 20: Adicionar archivos multimedia.

Descripción: El sistema debe permitir a los usuarios adicionar archivos multimedia, dígase imágenes, música en formato .mp3 y videos.

RF 21: Eliminar archivos multimedia.

Descripción: El sistema debe permitir a los usuarios eliminar sus archivos multimedia previamente almacenados.

2.3.2 *Requisitos no funcionales.*

Los requisitos no funcionales (RNF) detallan las propiedades o cualidades que el producto debe tener, aumentándole funcionalidad al sistema, pues hacen al producto atractivo, fácil de usar, rápido y confiable, los cuales se encuentran separados por categorías (**Pressman, 2010**) que ahora se mencionarán.

Requisitos de usabilidad:

- **RNF 1 Tipo de aplicación informática:** La aplicación propuesta será usada por personas que pueden o no tener conocimientos básicos de informática, por lo que el sistema debe ser de fácil uso.
- **RNF 2 Guía de Mensajes:** El sistema deberá mostrar información al usuario de las acciones a realizar, esto se logra a partir de mensajes en los íconos que representas dichas acciones.

Requisitos de eficiencia

- **RNF 3 Tiempo de respuesta:** El sistema debe tener como tiempo de respuesta de las peticiones de los usuarios como máximo de 500ms, ya que se desarrolla en tiempo real.

Requisitos de software

- **RNF 4:** La plataforma debe permitir ser ejecutada en el sistema operativo Ubuntu 12.04 y Windows 7, además de versiones superiores a las especificadas.
- **RNF 5:** Los usuarios de la plataforma tendrán acceso al sistema a través de navegadores web, tales como: Mozilla Firefox, Opera y Chrome. Cualquier navegador utilizado debe ser compatible con HTML5. Además deberán contar con el instalador del intérprete de Node en su versión 0.10.24.

Requisitos de Hardware

- **RNF 6 Prestaciones de la PC Cliente:** Las PC clientes deben cumplir con los requisitos mínimos que se especifican a continuación:
 - 1GB RAM o superior.
 - Tarjeta de red de datos.
 - 10 GB de espacio en disco como mínimo.
- **RNF 7 Prestaciones del servidor:** El servidor deberá cumplir con los requisitos mínimos que se especifican a continuación:
 - 1GB RAM o superior.

- Tarjeta de red de datos.
- 1TB de espacio en disco como mínimo.

Requisitos de Seguridad

- **RNF 8: Confidencialidad:** La información tiene que ser accedida solo por las personas autorizadas, por lo que será establecido un nivel de acceso a la plataforma mediante 2 roles, los cuales solo podrán interactuar con determinadas áreas del sistema.
- **RNF 9: Niveles de acceso:** En el sistema existirán diferentes aplicaciones interactivas, las cuales tendrán personas designadas como administradores, las cuales podrán establecer las políticas de seguridad en su aplicación. Debe existir al menos un administrador por cada una.

2.4 Casos de Uso del Sistema

Según (**Pressman, 2010**) los casos de uso son un conjunto de escenarios que identifican una línea de utilización para el sistema que va a ser desarrollado, facilitando una descripción de cómo se usará el sistema. Un diagrama de caso de uso muestra la relación entre los actores del sistema y los casos de uso.

A continuación se presentan los casos de uso del sistema, así como el diagrama asociado.

CU 1: Autenticar usuario.

CU 2: Gestionar usuario.

CU 3: Administrar espacio de almacenamiento por usuario.

CU 4: Gestionar perfil de usuario.

CU 5: Administrar aplicación de navegación de ficheros.

CU 6: Administrar reportes de las trazas.

CU 7: Administrar aplicaciones interactivas.

CU 8: Registrar usuario.

CU 9: Administrar archivos multimedia.

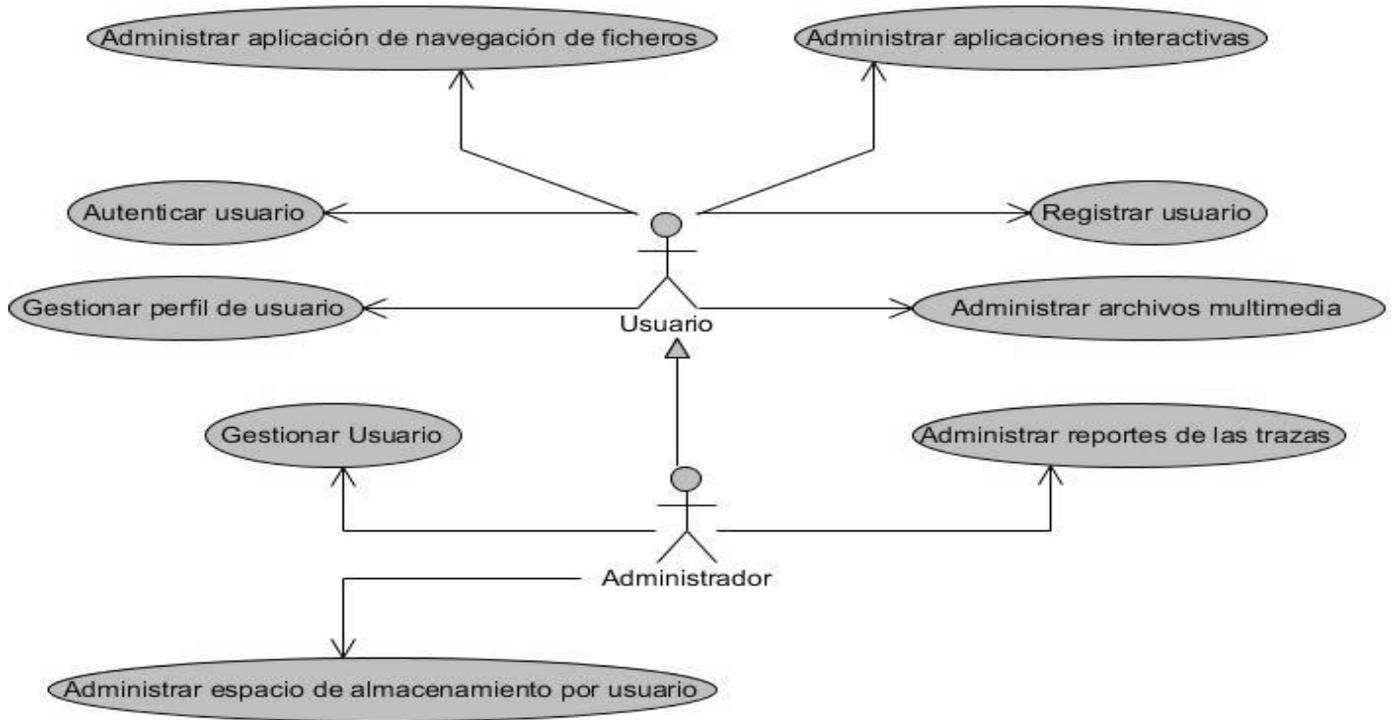


Fig 3. Diagrama de caso de uso del sistema

2.4.1 Descripción de los actores del sistema

Los actores del sistema intercambian información con él, aunque no forman parte de este. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. A continuación se muestran los actores y la justificación que tienen en el sistema.

Tabla 2. Descripción de los actores del sistema.

Actor	Descripción
Usuario	Es la persona que interactúa con el sistema; tiene acceso a las distintas funcionalidades que se brindan luego de estar registrado y autenticado.
Administrador	Es la persona encargada de la administración y control del sistema; tiene acceso a todas las funcionalidades que tiene el actor usuario, así como gestionar usuario y mostrar reportes de las trazas, entre otras.

2.4.2 Descripción de los casos de usos del sistema.

La descripción amplia de los casos de uso del sistema permite realizar el diseño, implementación y las pruebas de la interfaz. A continuación se describe en las tablas 3 y 4 los casos de uso principales del sistema. Se pueden consultar el resto de las especificaciones de casos de uso en el Anexo 3: Especificación de casos de uso.

Tabla 3. Descripción de caso de uso: Autenticar Usuario.

Objetivo	Autenticar usuario en el sistema.	
Actores	Usuario (inicia).	
Resumen	El caso de uso inicia cuando el actor desea autenticarse en el sistema.	
Complejidad	Baja.	
Prioridad	Crítico.	
Precondiciones	El usuario debe estar registrado previamente en el sistema.	
Postcondiciones	El usuario queda autenticado en el sistema.	
Referencia	RF 1: Autenticar usuario.	
Flujo de eventos		
Flujo básico: autenticación satisfactoria.		
	Actor	Sistema
1	Introduce su nombre de usuario y contraseña y selecciona la opción autenticar.	
2		Verifica que los datos estén completos.
3		Verifica que los datos entrados del usuario coincidan con los que tiene registrados.
4		Asigna privilegios según el tipo de usuario y visualiza la página principal con las opciones a las que el usuario tiene privilegios de acuerdo a su rol, finalizando así el caso de uso.

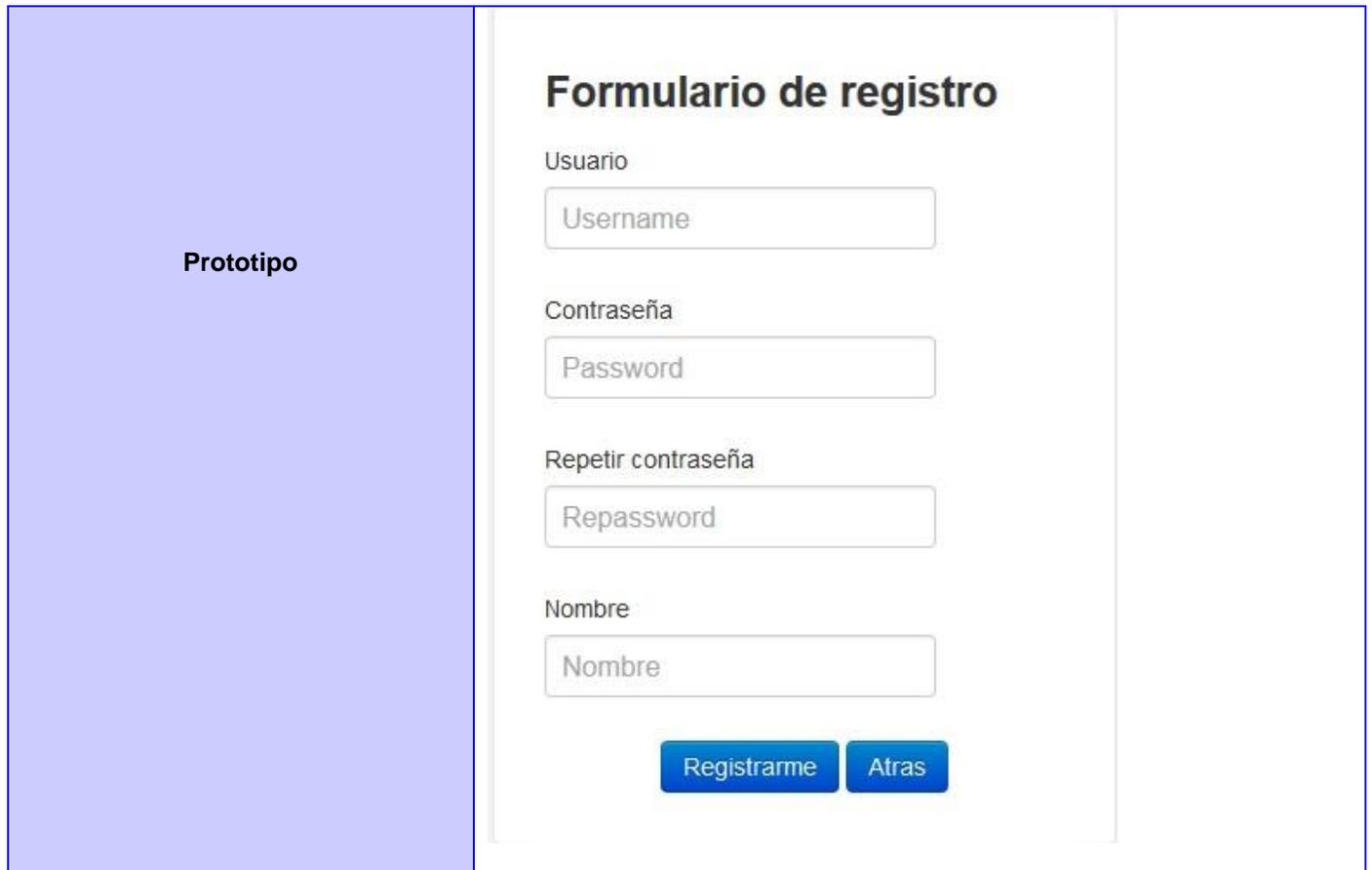
Flujos alternos		
Nº Evento 2. Los datos necesarios para la autenticación no están completos		
	Actor	Sistema
2.1		Muestra mensaje: "Es necesario que introduzca usuario y contraseña".
Nº Evento 3. Los datos de la autenticación no coinciden con los registrados		
	Actor	Sistema
3.1		Muestra mensaje: "Usuario y/o contraseña incorrectos".
Relaciones	CU Incluidos	No procede
	CU Extendidos	No procede
Prototipo		

Tabla 4. Descripción de caso de uso: Registrar Usuario.

Objetivo	Registrar usuario en el sistema.
Actores	Usuario (inicia).
Resumen	El caso de uso inicia cuando el actor desea autenticarse en el sistema.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA

Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones		
Postcondiciones	El usuario queda registrado en el sistema.	
Referencia	RF 14: Registrar usuario.	
Flujo de eventos		
Flujo básico: registro satisfactorio.		
	Actor	Sistema
1.	Introduce los siguientes datos: nombre y usuario, contraseña, confirmar contraseña.	
2.		Verifica que los datos estén completos.
3.		Verifica que los datos entrados sean correctos.
Flujos alternos		
Nº Evento 2. Los datos necesarios no están completos.		
	Actor	Sistema
2.1		Muestra mensaje: "No pueden existir campos vacíos".
Nº Evento 3. Los datos entrados no son correctos		
	Actor	Sistema
3.1		Muestra mensaje: "Datos incorrectos".
Nº Evento 4. La selección de las aplicaciones de preferencia no es correcto		
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.



Prototipo

Formulario de registro

Usuario

Contraseña

Repetir contraseña

Nombre

2.5 Conclusiones del capítulo

En este capítulo se abordaron diferentes elementos que sustentan la descripción de la solución propuesta, dándole a su vez cumplimiento a la tarea 4 de la investigación. Arribándose finalmente, a las siguientes conclusiones:

- La selección de la metodología de desarrollo permitió estructurar, planificar y controlar el proceso de construcción de la solución.
- El estudio realizado de las posibles tecnologías y lenguajes a utilizar, permitió determinar la base tecnológica necesaria para desarrollar el sistema propuesto.
- La utilización de la técnica de obtención de información permitió obtener requisitos funcionales y no funcionales del sistema para su posterior implementación.

Capítulo 3: Diseño e implementación de la solución propuesta

En el siguiente capítulo se describen los elementos relacionados con el diseño e implementación. Se definen los patrones de diseño y arquitectónico. Dentro de los artefactos desarrollados en el capítulo se encuentran: el diagrama de diseño, el de componentes y el modelo de despliegue de la plataforma.

3.1 Propuesta de solución.

En la siguiente figura se muestra propuesta de solución:

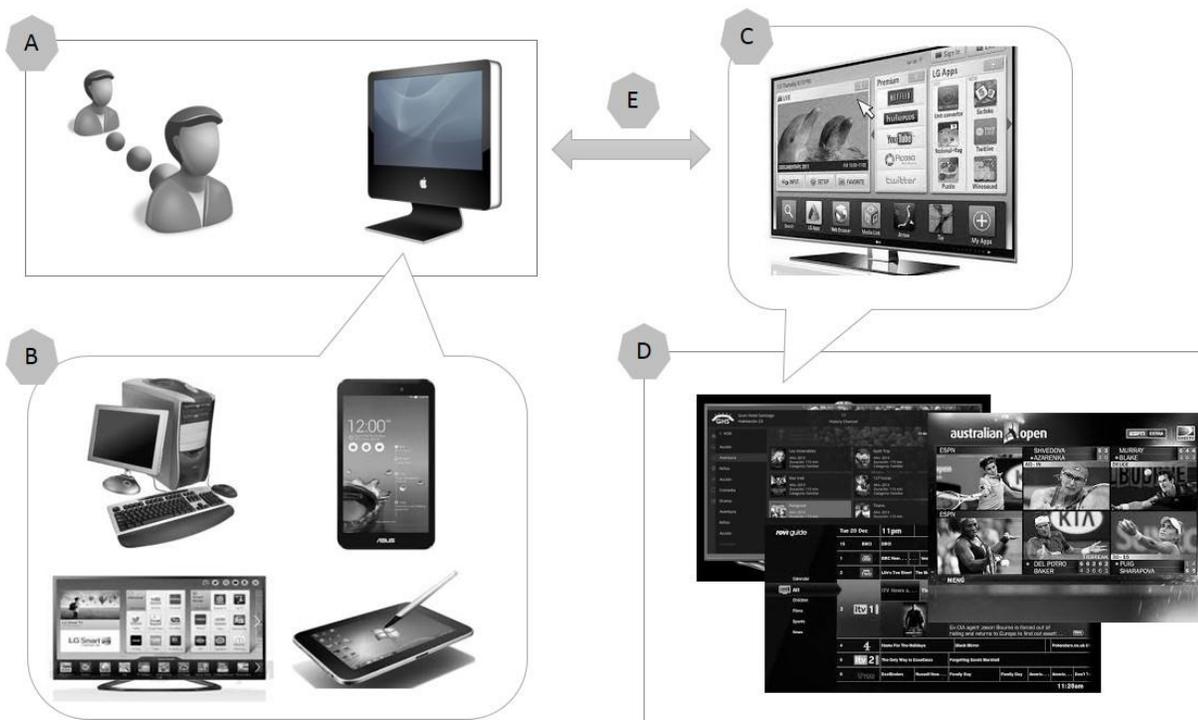


Fig 4. Propuesta de solución.

A: Los usuarios registrados acceden a los servicios interactivos que ofrece la plataforma mediante diferentes dispositivos.

B: Los dispositivos pueden ser computadoras, teléfonos móviles, televisores inteligentes y tabletas.

C: La plataforma que se desarrolla, el subsistema interactivo del Sistema de Transmisión de Canales Virtuales.

D: El Sistema de Transmisión de Canales Virtuales.

E: Protocolo de comunicación *WebSocket*.

3.2 Patrones

Un patrón detalla un problema que ocurre una y otra vez en el ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que se puede usar esa solución un millón de veces más, sin repetir lo mismo dos veces. **(Krucchten, 2000)**

Los patrones son utilizados en el desarrollo del software para establecer una manera de organizar y estructurar el desarrollo. Estos son una guía para realizar algunas acciones dentro del proceso de desarrollo, especificando un conjunto de subsistemas predefinidos y las funcionalidades de cada uno. Además facilitan la comprensión de la situación en que se encuentre algún problema que se quiera solucionar. **(Krucchten, 2000)**

3.2.1 Patrones de arquitectura

Los patrones de arquitectura de software son patrones de diseño de software que constituyen una vía en la solución de problemas de arquitectura de software. Los mismos poseen un nivel de abstracción mucho mayor que los patrones de diseño. Además brindan una descripción de los elementos y el tipo de relación que tienen, así como las restricciones para su uso. Los patrones se especifican describiendo los componentes, con sus responsabilidades y relaciones. **(Gamma, 2013)**

El patrón arquitectónico seleccionado para el desarrollo de la plataforma fue el Modelo- Vista-Controlador (MVC). El mismo separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres capas diferentes. (Ver figura 5)

Patrón MVC está compuesto por 3 capas:

- **Modelo:** Es la capa encargada de encapsular toda la lógica de negocio de la aplicación.
- **Controlador:** Es el eje central de la arquitectura, encargada de gestionar todas las peticiones, validar los datos recibidos y dirigir cualquier petición de cualquier tipo. Solo se comunica con el modelo y responde a través de vistas.
- **Vista:** Es la respuesta de cada controlador y lo que se le presenta al usuario final, se puede comunicar con el controlador y el modelo (en algunas ocasiones).

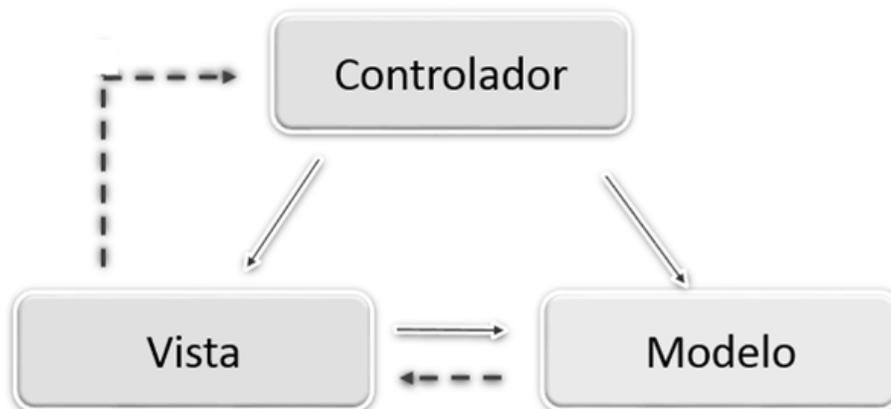


Fig 5. Patrón arquitectónico Modelo-Vista-Controlador.

3.2.2 Patrones de diseño

Un patrón de diseño constituye una solución estándar para un problema común de programación en el desarrollo del software. Además es una técnica muy eficaz para flexibilizar el código haciéndolo satisfacer ciertos criterios, así como permite una manera más práctica de describir ciertos aspectos de la organización de un programa. (Gamma, 2005)

Patrones GRASP

Los Patrones de Principios Generales para Asignar Responsabilidades (*GRASP* por sus siglas en inglés) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. A continuación se muestran los patrones evidenciados en la plataforma.

- **Experto:** Las responsabilidades deben ser asignadas a las clases que poseen la información para realizar dicha responsabilidad. La plataforma hace uso de este patrón y se evidencia cuando se desea obtener toda la información de los usuarios almacenados en el sistema, ya que la única clase con la responsabilidad de conocer esta información es *HandlerFile.js*.
- **Creador:** Se utiliza para asignarle a una clase la responsabilidad de crear una instancia de otra. Dentro de la plataforma este patrón se evidencia en las acciones de los controladores, las cuales crean objetos del modelo o los formularios que representan las entidades. Ejemplo de ello es, la clase *UserController.js*.
- **Controlador:** Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Se evidencia el uso de este patrón en la plataforma, ya que para cada

petición o evento que se genere en el mismo, existe una clase controladora llamada *Routing.js*, con la responsabilidad de manejar todos los eventos del sistema es decir, obtener y devolver una respuesta.

- **Alta Cohesión:** Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen asignadas solo las responsabilidades que les corresponde y mantienen una estrecha relación con el resto de las clases, es decir, una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. Ejemplo de ello en la aplicación es la clase *HandlerFile.js*.

- **Bajo Acoplamiento:** Determina el nivel de dependencia de una clase con respecto a otras. Una clase con bajo acoplamiento no depende de muchas otras. Ejemplo de ello en la aplicación es la clase *Log.js*.

Patrones GoF

Los patrones GoF (*Gang of Four* o “Pandilla de los Cuatro” en español), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación clases y la formación de estructuras de mayor complejidad. Nos permiten crear grupos de objetos para ayudarnos a realizar tareas complejas. (TACII, 2009), (Tello, 2014),

- **Singleton:** Asegura que una clase posea una sola instancia y proporciona un punto de acceso global a ella. Este patrón se ve evidenciado en la clase *Log.js*, debido a que siempre se hace una sola instancia de esta clase.

- **Inyección de Dependencias:** Una dependencia es otro objeto necesario para que una clase funcione. Por ejemplo, tenemos la clase llamada *User.js*, que necesita el objeto *handler* para guardar información sobre las acciones del usuario. Bajo esta definición, el objeto *handler* es una dependencia de la clase *User.js*.

3.3 Diseño de la interfaz

La metodología *RUP* propone la realización del análisis y el diseño para lograr un acercamiento de los desarrolladores con los procesos a implementar. Según (Pressman, 2010) el análisis tiene como misión el estudio de los requisitos, así como el estructurarlos y darle el refinamiento pertinente a los mismos, con el objetivo de conseguir una comprensión más precisa de ellos. Por su parte, el diseño es una representación abstracta de lo que se va a construir, contribuye a formar una arquitectura sólida y ayuda a

crear un plano para la implementación. Mediante él se modela el sistema para que soporte todos los requisitos incluyendo los no funcionales.

Uno de los principios de *RUP* a los cuales **(IBM, 2007)** hace referencia, precisa que la metodología debe de adaptarse al tamaño del proceso y a las necesidades del proyecto. A su vez el análisis es una etapa de la metodología *RUP* que debe de evidenciarse en la construcción del software. De esta se puede prescindir de acuerdo a la manera en que se haya decidido implementar el sistema, pasando a analizar los requisitos como parte del diseño o como parte integrada de la captura de los requisitos, lo que deja en evidencia que la manera de emplear el análisis es consecuente con la construcción del sistema.

En consideración a lo anteriormente expuesto y bajo el principio de que la metodología es altamente configurable, pudiendo obviarse artefactos durante el desarrollo, se decide pasar del flujo de requisitos directamente al diseño, sin la implementación del análisis. Esta decisión es tomada luego de realizar un estudio y constatar que para el desarrollo de la plataforma web interactiva, no se contempló el análisis como parte de la implementación de las soluciones. Además, los requisitos a implementar de la plataforma son bien conocidos y se cuenta con una comprensión de los mismos.

3.4 Modelo de Diseño

3.4.1 Diagrama de Clases del Diseño

El diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema. Este muestra sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de diseño. A continuación en la figura 5 se representa el diagrama de clases del diseño para el CU Registrar Usuario. Los demás diagramas se encuentran representados en el Anexo 4: Diagramas de Clases del Diseño.

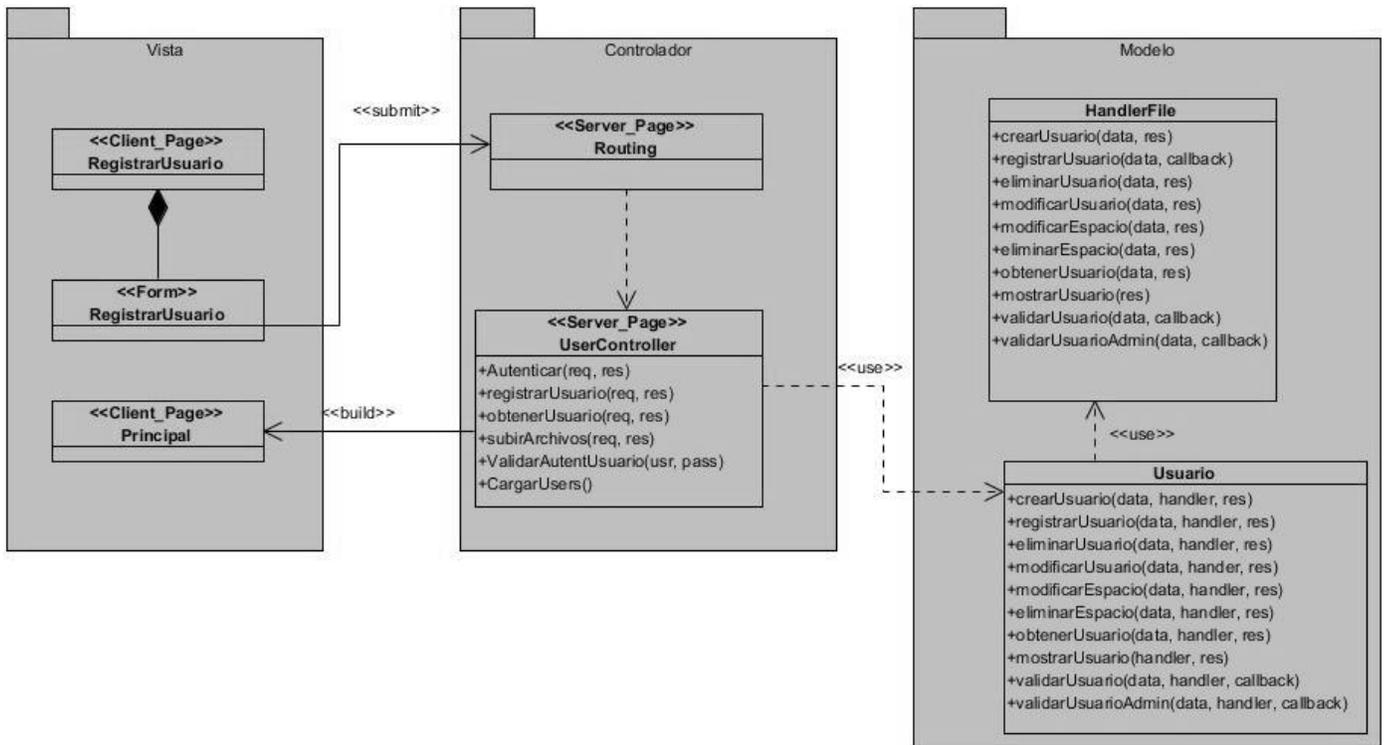


Fig 6. Diagrama de clases del diseño: Registrar usuario

3.5 Modelo de Implementación

En este modelo se describe cómo se implementan en términos de componentes, ficheros de código fuente o ejecutables, los elementos del modelo, tales como las clases, entre otros. Este modelo describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y cómo dependen los componentes unos de otros. Se realizan con el objetivo de poseer una vista de forma general del sistema a partir de las dependencias e integraciones de los componentes y módulos. **(Pressman, 2010)**

A continuación se muestra el diagrama de componentes donde se integra el sistema propuesto:

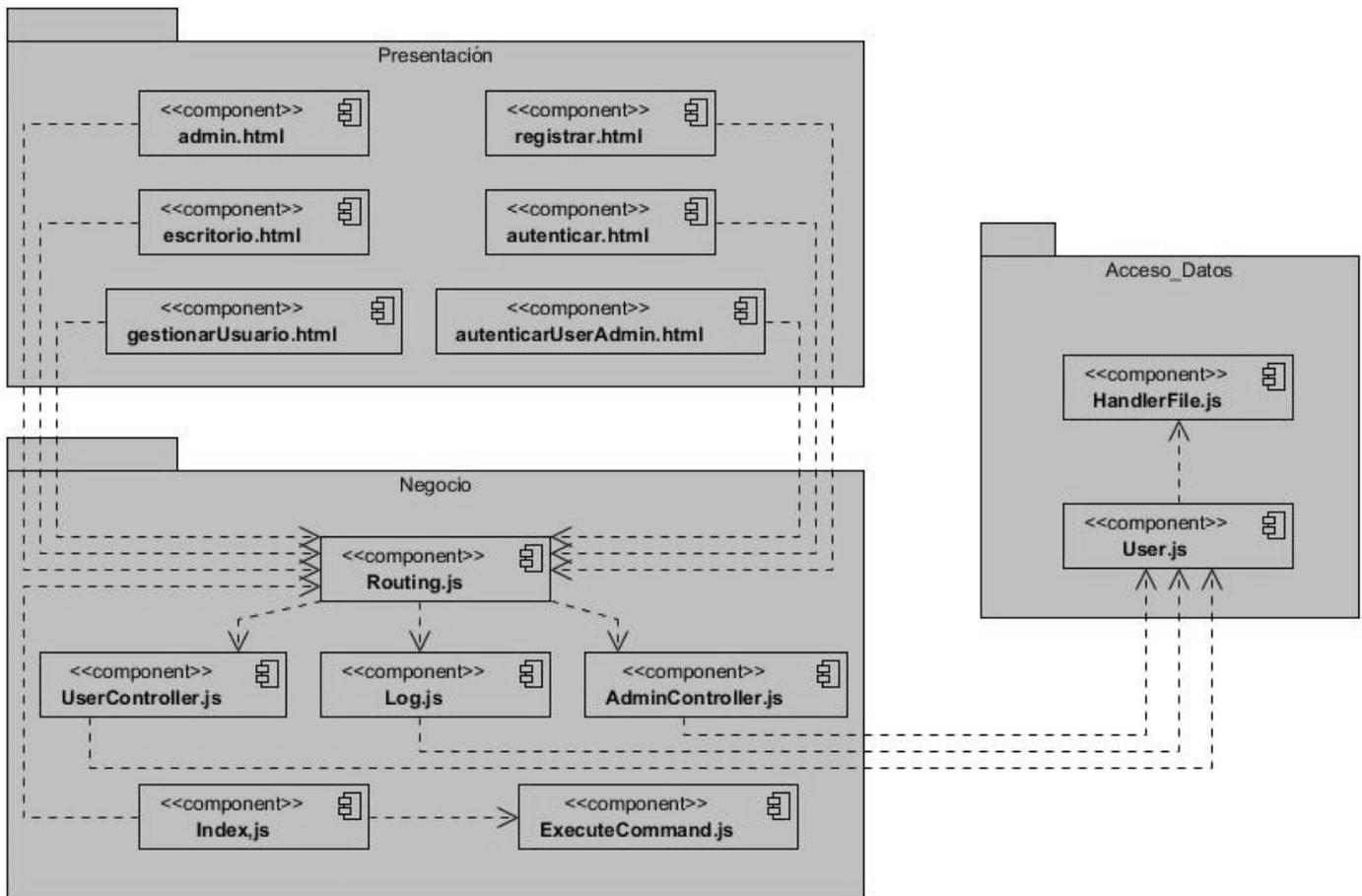


Fig 7. Diagrama de Componentes de la plataforma

3.6 Diagrama de Despliegue

La vista de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes. El diagrama está compuesto por nodos, dispositivos y conectores. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. (Pressman, 2010)

A continuación se presenta el modelo de despliegue de la plataforma web interactiva. El mismo está compuesto por nodos físicos, conectados a través del protocolo de red TCP/IP, excepto los nodos Televisores y STB que son conectados mediante conexión RCA. Algunos nodos están integrados por el

componente Navegador web. El servidor de aplicaciones, está compuesto a su vez por el componente Sistema de Transmisión de Canales Virtuales y por la Plataforma Web Interactiva.

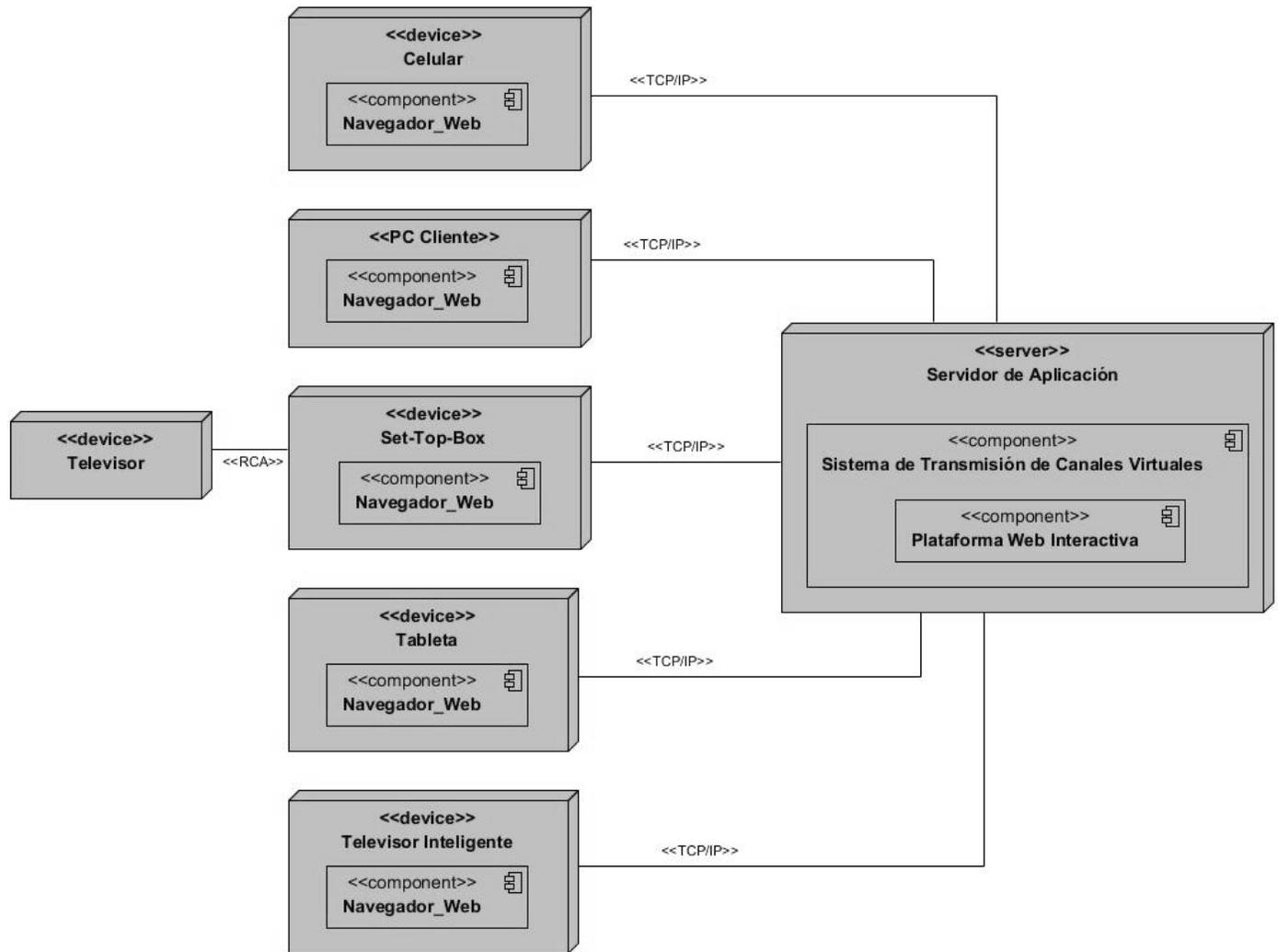


Fig 8. Diagrama de Despliegue

3.7 Código fuente

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un software es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto en el código fuente de un programa está escrito su funcionamiento. Estas instrucciones son escritas en un lenguaje de programación que consiste en un

conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

3.7.1 **Estándares de codificación**

Un estándar de codificación comprende los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Además, si se aplica de forma continua un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas y, posteriormente, se efectúan revisiones del código de rutinas, existen posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener.

Estilos de codificación empleados:

Durante la implementación de la plataforma, fue necesario utilizar algunos estilos de codificación en busca de un estándar que aportara una mayor organización al código.

- **Comentarios, separadores, líneas, espacios en blanco y márgenes:** Establecer un modo común para comentar el código, utilizar separadores, línea, espacios en blanco y márgenes, de forma tal que sea comprensible el código, siguiendo las pautas que a continuación se muestran:

- Comentar al inicio de la clase o función especificando el objetivo de la misma.
- Dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.
- Evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.

- **Clases y métodos:**

Para la definición de las clases y métodos en el código de la aplicación es utilizado el estándar CamelCase. Este es un estilo de escritura que se aplica a frases o palabras compuestas.

Existen dos tipos de estándares de CamelCase:

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Este estándar se utilizó para los nombres de las clases.

Ejemplo: EjemploDeUpperCamelCase.

- lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Este estándar se utilizó para los nombres de los métodos.

Ejemplo: ejemploDeLowerCamelCase.

Ejemplo de ello se evidencia en la siguiente figura del código de la aplicación:

```

exports.User = function () {

  this.crearUsuario=function(data,handler,res) {
    handler.crearUsuario(data,res);
  }

  this.registrarUsuario=function(data,handler,res) {
    handler.registrarUsuario(data,res);
  }

  this.eliminarUsuario=function(data,handler,res) {
    handler.eliminarUsuario(data,res);
  }

  this.validarUsuario=function(data,handler,callback) {
    handler.validarUsuario(data,callback);
  }

  this.validarUsuarioAdmin=function(data,handler,callback) {
    handler.validarUsuario(data,callback);
  }

  this.modificarUsuario=function(data,handler,res) {
    handler.modificarUsuario(data,res);
  }

  this.modificarPerfil=function(data,usuario,handler,res) {
    handler.modificarPerfil(data,usuario,res);
  }
}

```

Fig 9. Uso del estándar de codificación CamelCase.

3.8 Interfaces de la plataforma

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA

La interfaz de una aplicación permite el flujo de información entre el usuario y el sistema (Anónimo, 2015). A continuación se muestran algunas interfaces del sistema.



Fig 10. Interfaz principal: autenticar usuario.

☰ Lista de usuarios registrados en la Plataforma

Filtro de usuario

Nombre	Usuario	Password	Tipo	Capacidad	Cap_Ocupada	Cap_Disponible
Admin	admin	admin	Administrador	15728640 KB	0 KB	15728640 KB
Loko	loko	loko	Basico	5242880 KB	0 KB	5242880 KB
Pablo E	pablo	pablo	Basico	10485760 KB	0 KB	10485760 KB
Yenisel	yenisel	yenisel	Administrador	20971520 KB	0 KB	20971520 KB

Fig 11. Interfaz correspondiente a la búsqueda de usuarios.

Crear Usuario

Usuario

Contraseña

Nombre

Tipo

Capacidad

Crear Modificar Eliminar

Fig 12. Interfaz correspondiente a la gestión de Usuarios.

Modificar Espacio a un Usuario

Usuario

Capacidad

Modificar Eliminar

Fig 13. Interfaz correspondiente a la administración de espacio de almacenamiento por usuarios.

GESTIONAR USUARIO Y ESPACIO		ADMINISTRAR LOGS
☰ Lista de Logs registrados en la Plataforma		
Filtro de log		
Nombre	Datos	
11-5-2015(juan).txt	Usuario juan logueado a las 03:46AM del 11-5-2015Usuario juan logueado a las 03:48AM del 11-5-2015Usuario juan logueado a las 04:59AM del 11-5-2015Usuario juan logueado a las 04:59AM del 11-5-2015	
12-5-2015(juan).txt	Usuario juan logueado a las 01:36AM del 12-5-2015Usuario juan logueado a las 01:50AM del 12-5-2015Usuario juan logueado a las 01:58AM del 12-5-2015Usuario juan logueado a las 01:58AM del 12-5-2015	
12-5-2015(loko).txt	Usuario loko logueado a las 02:45AM del 12-5-2015Usuario loko logueado a las 06:55AM del 12-5-2015Usuario loko logueado a las 06:56AM del 12-5-2015Usuario loko logueado a las 06:56AM del 12-5-2015	
12-5-2015(pablo).txt	Usuario pablo logueado a las 02:46AM del 12-5-2015	
13-5-2015(juan).txt	Usuario juan logueado a las 01:45AM del 13-5-2015Usuario juan logueado a las 01:45AM del 13-5-2015Usuario juan logueado a las 02:01AM del 13-5-2015Usuario juan logueado a las 02:01AM del 13-5-2015	
13-5-2015(loko).txt	Usuario loko logueado a las 06:18AM del 13-5-2015	
13-5-2015(root).txt	Usuario root logueado a las 06:19AM del 13-5-2015	
14-5-2015(juan).txt	Usuario juan logueado a las 02:45AM del 14-5-2015Usuario juan logueado a las 02:46AM del 14-5-2015Usuario juan logueado a las 06:02AM del 14-5-2015Usuario juan logueado a las 06:02AM del 14-5-2015	
14-5-2015(loko).txt	Usuario loko logueado a las 02:45AM del 14-5-2015	

Fig 14. Interfaz correspondiente a la visualización de las trazas por usuarios.

3.9 Conclusiones del capítulo

El presente capítulo ha generado toda la información necesaria para la confección de la plataforma web interactiva, dando respuesta a las tareas de la investigación 5 y 6. También se puede concluir que:

- La utilización de los patrones arquitectónicos y de diseño permitió aplicar una guía para la construcción de la plataforma.
- La realización de los artefactos ingenieriles permitió documentar todo el proceso de construcción del software.
- La realización del flujo de trabajo de Implementación permitió poner en práctica algunos estándares de codificación para una mejor comprensión del código, ayudando de esta forma a que futuros programadores se sientan familiarizados con el producto.

Capítulo 4: Pruebas del sistema

En el presente capítulo se expondrá todo lo referente a la fase de pruebas realizadas a la plataforma desarrollada, así como los resultados obtenidos. Dentro de esta fase se desarrollan distintos tipos de pruebas en función de los objetivos de las mismas. Algunas de ellas son las pruebas de Caja Negra, Caja Blanca y pruebas de rendimiento.

4.1 Pruebas realizadas

- **Prueba de Caja Blanca:** Se basa en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
- **Prueba de Caja Negra:** Se realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

Cada prueba tiene técnicas asociadas, las cuales se aplicaron Camino Básico para Caja Blanca y Partición Equivalente para Caja Negra. **(Pressman, 2001)**

4.1.1 Resultados finales de las pruebas realizadas a la plataforma.

Las pruebas fueron realizadas y documentadas en el anexo 6: Plan de Pruebas. En este apartado solo se mostrarán los resultados finales de las mismas:

- **Pruebas de caja negra:**

A continuación se presentan los casos de prueba correspondientes al caso de uso "Registrar usuario".

Tabla 5. Caso de prueba para el caso de uso: Registrar usuario

Sección “Registrar usuario”							
Escenario	Descripción	Nombre	Usuario	Contraseña	Repetir contraseña	Respuesta del sistema	Flujo central
EC 1.1 Registrar correctamente	Se registra correctamente el usuario.	V ('Pepe')	V ('pepe')	V ('pepe')	V ('pepe')	Se registra un nuevo usuario en el sistema.	<ol style="list-style-type: none"> 1. Se selecciona la opción “Registrarse” en la página inicial de la plataforma. 2. Se llenan todos los campos. 3. Se selecciona la opción “Registrarse”.
EC 1.2 Dejar campos vacíos	Se intenta registrar un usuario dejando campos vacíos.	V ('Pepe')	I	V ('pepe')	I	Muestra el mensaje de error: <i>No puede dejar campos vacíos.</i>	<ol style="list-style-type: none"> 1. Se selecciona la opción “Registrarse” en la página inicial de la plataforma. 2. Se llenan los campos. 3. Se selecciona la opción “Registrarse”. 4. Se muestra mensaje de error.

<p>EC 1.3 Insertar datos incorrectos</p>	<p>Se inserta un campo incorrecto.</p>					<p>Redirige hacia el formulario de registro y muestra un mensaje error.</p>	<ol style="list-style-type: none"> 1. Se selecciona la opción "Registrarse" en la página inicial de la plataforma. 2. Se llenan los campos del formulario de registro. 3. Se selecciona la opción "Registrarse". 4. Se muestra mensaje de error.
---	--	--	--	--	--	---	--

A partir del diseño y ejecución de los casos de prueba, introduciendo juegos de datos, tanto correctos como incorrectos, se detectaron errores tales como: opciones que no funcionan, errores de idioma, validación de los datos de entradas, errores de redacción y funcionalidades incorrectas, para un total de 9 no conformidades.

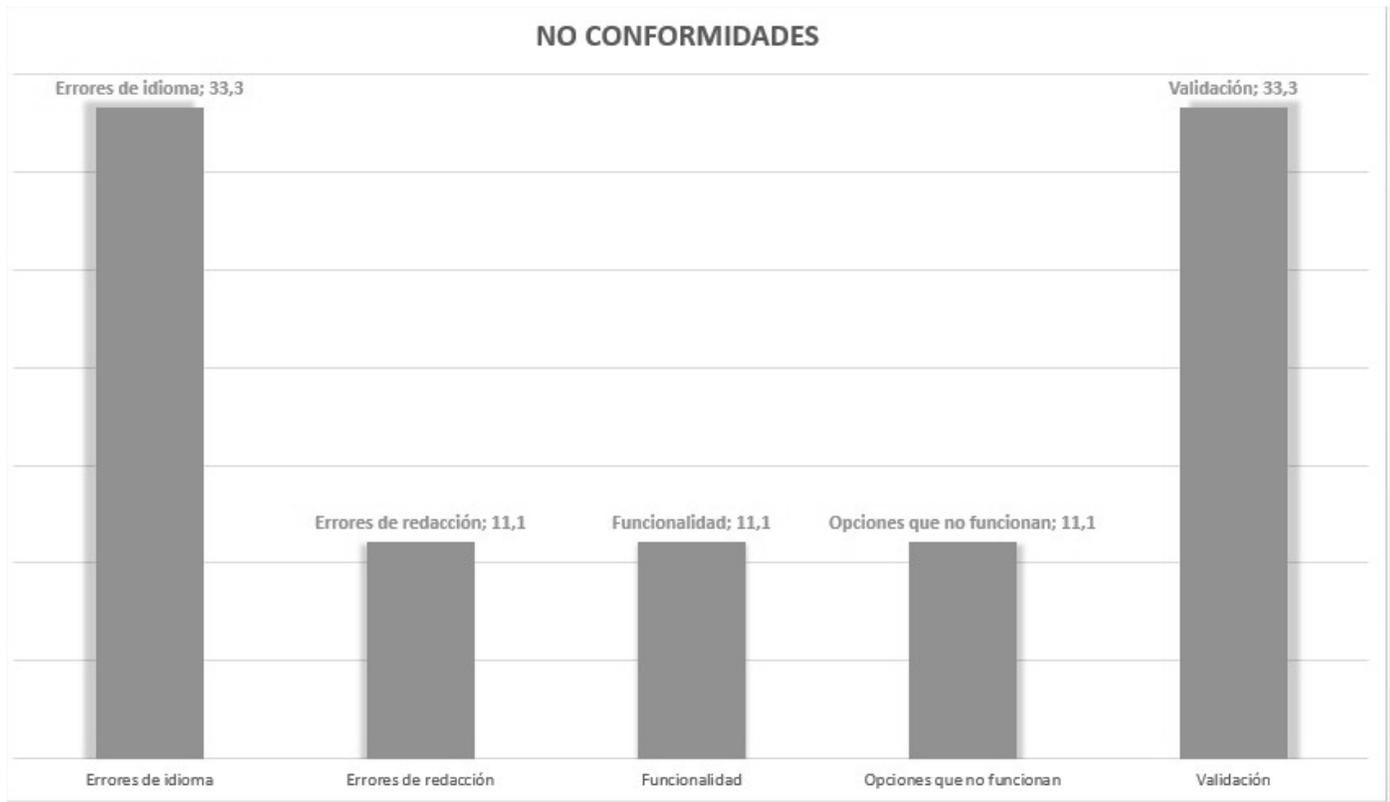


Fig 15. Gráfica de No Conformidades de las pruebas funcionales.

La gráfica anterior representa los resultados alcanzados en las pruebas funcionales, representando en porcentaje las no conformidades detectadas. Se detectaron un 33,3 % en errores de validación de datos y errores de idioma, las funcionalidades incorrectas, los errores de redacción y las opciones que no funcionan simbolizan un 11,1%. Al realizar un análisis de los resultados, se concluye que la mayor cantidad de no conformidades se refieren a los errores de validación de datos y errores del idioma.

Las no conformidades (ver Anexo 5) fueron corregidas, dándole paso a una segunda iteración de pruebas, en la cual se evaluaron nuevamente las funcionalidades, obteniendo resultados satisfactorios.

- **Pruebas de Caja Blanca:**

Para la aplicación de las pruebas de Caja Blanca se utilizó la técnica del Camino Básico ya que permite obtener una medida de la complejidad lógica del diseño procedimental sobre un conjunto básico de caminos de ejecución.

A continuación se muestra en la Figura 13 el código, que pertenece a la funcionalidad de Subir e instalar aplicaciones interactivas.

```

1  this.subirApps=function(req,res){
    var fs = require('fs');
    var unzip = require('unzip');
    var datos;
    if (req.files.archivo.type.indexOf('zip')==12 || req.files.archivo.type.indexOf('x-zip')==13) {
2      fs.createReadStream(req.files.archivo.path).pipe(unzip.Extract({path:dir+'/apps/system/'}));
      var name=req.files.archivo.name.split('.');
      setTimeout(function(){
3        fs.readFile(dir+ConvertPath("/models/users/"+req.session.user+"/System/config.json"),function(err,obj){
          if(err) throw err;
          datos=JSON.parse(obj);
          fs.renameSync(dir+'/apps/system/'+name[0], dir+'/apps/system/'+count);
          datos.apps.push(count);
          count++;
          restarEspacio(req.session.user.req.files.archivo.size);
          fs.writeFile(dir+ConvertPath("/models/users/"+req.session.user+"/System/config.json"),JSON.stringify(datos));
          res.send('subido');
        });
      }, 200);
    }
4  else res.send('no');
5  }

```

Fig 16. Pruebas de Caja Blanca con la técnica del Camino Básico.

En la figura 14 se muestra el grafo obtenido:

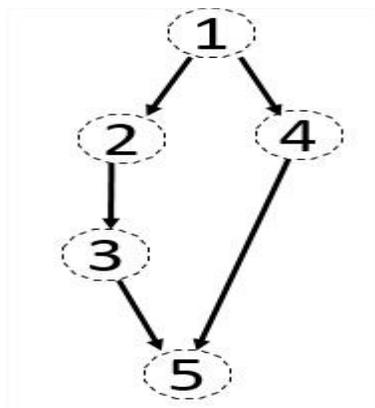


Fig 17. Grafo obtenido del código seleccionado.

El cálculo de la complejidad ciclomática proporcionó un límite inferior de la cantidad de pruebas que se deben aplicar a la porción de código, estableciendo una para cada camino independiente que posea el grafo. La fórmula para el cálculo de la complejidad ciclomática es: $V(G) = A - N + 2$, donde $[V(G)]$ es la complejidad ciclomática, $[A]$ representa a la cantidad de aristas y $[N]$ a la de nodos con los que cuenta el grafo de flujo asociado al código. Luego de aplicar la fórmula para los valores del grafo queda de la siguiente manera:

$$V(G) = A - N + 2$$

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

$$V(G) = 2$$

El valor obtenido para la complejidad ciclomática es 2, lo que deja claro que existen 2 posibles caminos a recorrer para el flujo; de igual forma es la cantidad mínima de pruebas que se le realizó al código. La tabla 7 plasma los caminos básicos del flujo.

Tabla 6. Caminos básicos obtenidos del grafo.

No.	Caminos
1	1-2-3-5
2	1-4-5

Dando continuidad al método y contando con el grafo y los caminos identificados previamente se pasa a la creación de casos de pruebas para cada uno de estos caminos. La tabla 8 muestra el caso de prueba para el camino 1-2-3-5, los restantes casos de pruebas se encuentran en el Anexo 7: Plan de Pruebas.

Tabla 7. Caso de prueba para el camino básico No.1

Caso de prueba para el camino básico No.1	
Camino	1-2-3-5
Descripción de los parámetros de entrada	Se le pasa por parámetros una variable <i>req</i> que es la que contiene la información del archivo a subir y una variable <i>res</i> la cual se usara para dar la respuesta.
Resultado esperado	1-2-3-5

- **Pruebas de Rendimiento:**

La realización de estas pruebas está enfocada en determinar la capacidad del sistema de recibir múltiples peticiones sin que se vea afectado, así como la velocidad de respuesta del mismo. Se decidió utilizar JMeter, que es una herramienta de software libre, la cual permite medir la capacidad de carga de una aplicación. Además, con ella es posible conocer los tiempos de respuesta de un sistema dado un número de usuarios determinado y un número real de transacciones procesadas por unidad de tiempo. **(Anónimo, 2015)**

Es preciso realizar las pruebas de carga y estrés pues resulta necesario comprobar el rendimiento del sistema soportando una cantidad máxima de usuarios interactuando con él y su comportamiento al aumentar esta carga con los mismos recursos disponibles.

El entorno de trabajo en el que fueron realizadas las pruebas de rendimiento cumple con las características de los requisitos no funcionales de hardware especificados en el capítulo 2.

En el diseño del plan de pruebas de rendimiento para el sistema se tuvo en cuenta las acciones que el usuario puede realizar al conectarse a la aplicación. En una muestra de 100 usuarios conectados concurrentemente, con un período de subida de 1 segundo (tiempo de espera de cada usuario para realizar una petición). El análisis del resumen brindado por la herramienta arrojó que para un total de 3900 muestras que se le realizaron al sistema se alcanzó un rendimiento de 58,3 peticiones por segundo, con un porcentaje de 0% de errores para cada petición realizada.

4.2 Conclusiones del capítulo

En este capítulo se expusieron los resultados de las principales pruebas realizadas a la plataforma, dándole respuesta a la tarea 7 de la investigación. El capítulo permite concluir que:

- La solución de las no conformidades detectadas permitió aumentar la calidad final de la solución.
- La aplicación de distintas pruebas permitieron comprobar que los resultados obtenidos en el sistema se corresponden con las necesidades iniciales del cliente, quedando implementados todos los requisitos pactados.

Conclusiones Generales

Una vez concluida la presente investigación se arriban a las conclusiones siguientes:

- El estudio de las diversas soluciones existentes en Cuba y el mundo, permitió identificar la necesidad de desarrollar una plataforma web para incorporar servicios interactivos que sean accesibles por los usuarios desde distintos dispositivos móviles.
- Se caracterizaron las principales tecnologías, lenguajes y herramientas empleados para el desarrollo de la aplicación.
- La definición de los requisitos del sistema en conjunto con el diseño de clases del diseño posibilitó la definición de las funcionalidades de la plataforma para cumplir con los objetivos planteados.
- Como resultado de la implementación se obtuvo un sistema funcional que cumple con los requisitos especificados que permite a los usuarios del STCV acceder desde distintos dispositivos móviles a los servicios interactivos.
- La solución fue probada y las deficiencias detectadas en cada una de las iteraciones de prueba fueron corregidas en el tiempo establecido, lo que posibilitó la calidad de la solución.

Por todo lo anteriormente expuesto, se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente ya que la aplicación da solución a la situación problemática que le dio origen.

Recomendaciones

Luego de haber analizado los resultados del presente trabajo de diploma, resulta factible arribar a las siguientes recomendaciones:

- Crear una aplicación de mensajería instantánea, para aumentar la comunicación entre los usuarios del sistema, ya que actualmente es nula.
- Crear una base de datos única y común para todas las aplicaciones que se manejen en la misma.
- Desarrollar un repositorio desde el cual los usuarios puedan descargar e instalar aplicaciones directamente sobre la plataforma.

Bibliografía

- Andreadis, Alessandro, y otros. 2007.** *Towards ITV applications' portability across digital terrestrial television frameworks*. 5th International Conference on Software, Telecommunications . 2007. págs. 1-4.
- Anónimo. 2015.** Herramienta CASE que da soporte al modelado visual con UML 8.0. [En línea] 2015. [Citado el: 12 de Marzo de 2015.] <http://www.visual-paradigm.com>.
- Anónimo. 2015.** <https://www.websocket.org/>. [En línea] 2015. [Citado el: 22 de Enero de 2015.] <https://www.websocket.org/>.
- Anónimo. 2015.** Mejoramiento del Proceso de Pruebas y Corrección de Defectos de Software en un Ambiente Globalizado. [En línea] 2015. [Citado el: 17 de Abril de 2015.] http://chie.uniandes.edu.co/~gsd/index.php?option=com_content&task=view&id=129&Itemid=183..
- Anónimo. 2015.** Metodología RUP. [En línea] 2015. [Citado el: 12 de Marzo de 2015.] <http://rupmetodologia.blogspot.com/>.
- Anónimo. 2015.** Netbeans. [En línea] 2015. [Citado el: 12 de Marzo de 2015.] <https://netbeans.org/community/releases/80>.
- Anónimo. 2015.** npm Inc. [En línea] 2015. [Citado el: 23 de Marzo de 2015.] <https://www.npmjs.com/package/express-sitemap>.
- Arantia. 2015.** [En línea] 2015. [Citado el: 23 de Marzo de 2015.] <http://www.arantia.com/producto/arantia-tv>.
- Arnold, John, Frater, Michael R y Pickering, Mark. 2007.** *Digital Television: technology and standards*. [ed.] NJ. Wiley-Blackwell Hoboken. 2007. pág. 644.
- Aubay. 2011.** Plataforma de Servicios Interactivos TDT. 2011.
- Benoit, Hervé. 2008.** *Digital Television: Satellite, Cable, Terrestrial, IPTV, Mobile TV in the DVB Framework*. 3ra ed. s.l. : Focal Press, 2008. pág. 289.
- Bordignon, Alexandro y Varella, Fernando. 2009.** *Mechanisms for interoperable content production among Web, Digital TV and Mobiles*. Informatica Na Educação: teoria & prática. 2009. págs. 110-180. Vol. 12. 1.
- Brouer, Jesper Dangaard. 2013.** *The IPTV-Analyzer*. 2013.
- Buschmann F, Meunier y Rohnert, R. 1996.** *Patterns., Pattern-Oriented Software Architecture*. . 1996.
- Bustamante de los Ríos, Wilson. 2014.** Universidad Politécnica Salesiana. Repositorio Digital. [En línea] 2014. [Citado el: 12 de Diciembre de 2014.] <http://dspace.ups.edu.ec/handle/123456789/160>.
- Calisoft. 2012.** *Introducción a las pruebas de software*. 2012.
- Chorianopoulos, Konstantinos y Spinellis, Diomidis. 2004.** *Affective usability evaluation for an interactive music television channel*. [ed.] ACM Computers in Entertainment. 2004. Vol. 2. 3.
- Crespo, Angel García. 2012.** *Servicios Interactivos y Accesibilidad en la Televisión Digital, una oportunidad para reducir la Brecha Digital*. 2012.
- Eguíluz, J. 2015.** *Introducción a CSS*. LibrosWeb. [En línea] 2015. [Citado el: 23 de Marzo de 2015.] <http://www.librosweb.es/css/index.html>.

- Gallego, Luis. 2008.** *IPTV Arquitectura*. s.l. : ZTE Coprporation, 2008.
- Gamma. 2013.** *Design patterns*. 2013.
- Gamma, Erich, y otros. 1995.** *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995. Addison-Wesley.
- García, Alberto Garcia. 2010.** *Principios de interactividad: televisión interactiva y realidad virtual*. 2010.
- Gutiérrez, Javier J, y otros. 2006.** *Modelos de pruebas para pruebas del sistema*. 2006. JISBD 2006.
- Helm, Richard. 2013.** *Design Patterns: Elements of Reusable*. 2013.
- IBM, C. 2007.** *Ayuda de Rational Unified Process*. 2007.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 1999.** *El proceso unificado de desarrollo de software*. Madrid : s.n., 1999.
- Jiménez, Alberto Cobo. 2009.** *Nuevas aplicaciones y servicios interactivos de TDT*. 2009.
- Jiménez, Maria Alejandra. 2015.** NETBEANS accesible. [En línea] 2015. [Citado el: 18 de Junio de 2015.] <http://netbeansaccesible.blogspot.com/>.
- Juristo, Natalia, M., Moreno Ana y Vegas, Siras. 2010.** *Técnicas de evaluación de software*. 2010.
- Kernen, Thomas. 2012.** *Home Brew IPTV head-end*. 2012.
- Kiessling, Manuel. 2010.** *El libro para principantes en Node.js*. 2010.
- Kruchten. 2000.** *The Rational Unified Process: An Introduction*. 2000.
- Lacabanne, Raúl. 2012.** ¿Qué entendemos por interactividad? 2012.
- Lindquist, Jan. 2010.** *Open IPTV Forum*. 2010.
- Lu, Karyn Y. 2005.** *Tesis: Interaction Design Principles for Interactive Television Computing*. Georgia Institute of Technology. 2005. pág. 202.
- Monte, Boadilla del. 2013.** *Sistemas y Servicios Interactivos*. 2013.
- Morris, Steven y Smith-Chaigneau, Anthony. 2005.** *Interactive TV Standars*. s.l. : Focal Press, 2005. pág. 585.
- Oñate, Cristina González. 2012.** *El nuevo escenario del medio televisivo: nuevas cadenas de televisión ante el desafío digital*. 2012.
- Pérez, Jorge. 2005.** *Televisión Digital*. s.l. : GAPTEL, 2005.
- Pressman, Roger S. 2010.** *Software engineering: a practitioner's approach*. 7th ed. 2010.
- Pressman, Roger S. 2001.** *Ingeniería del Software. Un enfoque práctico*. [ed.] 5ta Edición. España : s.n., 2001. pág. 323. ISBN 8448132149.
- Punchihewa, Amal y De Silva, Ann Malsha. 2013.** *IPTV Course Notes*. 2013.
- Rost, Alejandro. 2014.** *¿de qué hablamos cuando hablamos de interactividad?* 2014.
- Safeview. 2015.** Punto Azul Ecosistema IPTV/OTT. [En línea] 2015. [Citado el: 23 de Mazo de 2015.] <http://www.safeviewtv.es/es/pages/plataforma-iptv-ott-safeview.html>.

- SiTvi, Centro de Excelencia. 2009.** Servicios interactivos en televisión digital. España : s.n., 2009.
- Stefanov, Stoyan. 2008.** *Object-Oriented JavaScript. Create scalable, reusable high-quality JavaScript applications, and libraries.* 2008.
- Systems, Albentia. 2012.** Televisión sobre IP (IPTV). 2012.
- TACII. 2009.** *Patrones de diseño. Tema 7.* 2009.
- Tektronix. 2014.** *IPTV: la nueva era.* 2014.
- Telecomunicaciones, Union Internacional de. 2005.** *Curso a distancia de la UIT.* 2005.
- Tello, Jesús Cáceres. 2014.** *Design patterns: example of application in the Generative Learning Object.* s.l. : Dpto. Ciencias de la Computación (Universidad de Alcalá) , 2014.
- Tseklevs, Emmanuel, y otros. 2009.** *Converged digital TV services: The role of middleware and future directions of interactive television.* International Journal of Digital Multimedia Broadcasting. s.l. : Hindawi Publishing Corporation, 2009. págs. 1-20.
- W3school. 2015.** THE WORLD'S LARGEST WEB DEVELOPER SITE. [En línea] 2015. [Citado el: 12 de Marzo de 2015.] http://www.w3schools.com/css/css3_intro.asp.
- W3schools. 2015.** THE WORLD'S LARGEST WEB DEVELOPER SITE. [En línea] 2015. [Citado el: 12 de Marzo de 2015.] http://www.w3schools.com/html/html5_intro.asp.