

**Implementación de un prototipo orientado a la web para la gestión de televisión con IPTV
en el municipio de Soacha Cundinamarca**

Manuel Alejandro Ramírez Jaramillo

Trabajo de grado presentado para optar al título de Ingeniero de Sistemas

Director

Arellys de Jesús Correa Rodríguez

Universidad Nacional Abierta y a Distancia

Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Ingeniería de Sistemas

Bogotá, Cundinamarca

2023

Resumen

Actualmente la televisión en Colombia se destaca por la presencia dominante de dos corporaciones de medios privados, Caracol y RCN, que promueven una televisión principalmente comercial. Esto ha llevado a la desaparición de pequeños canales comunitarios debido a la falta de recursos para competir en términos de audiencia y producción. En orden a lo anterior, y teniendo en cuenta que actualmente la tecnología IPTV está en desarrollo permitiendo acceder a televisión utilizando una dirección IP para conectarse a través de Internet, este proyecto tiene como objetivo la implementación de un prototipo basado en IPTV para la gestión de televisión en el municipio de Soacha, ofreciendo accesibilidad a los pequeños proveedores de televisión comunitaria y así estos puedan dar contenido variado a sus clientes, su desarrollo fue posible mediante la metodología ágil XP junto con el lenguaje de programación PHP, obtener un prototipo de software orientado a la web para la gestión de la televisión en el municipio de Soacha.

Palabras Clave: Internet TV, PHP, Programación Web, Protocolo IPTV, Televisión Comunitaria

Abstract

Currently, television in Colombia stands out for the dominant presence of two private media corporations, Caracol and RCN, which promote primarily commercial television. This has led to the disappearance of small community channels due to a lack of resources to compete in terms of audience and production. In light of this, and considering that IPTV technology is currently being developed, allowing access to television using an IP address to connect via the Internet, this project aims to implement an IPTV-based prototype for television management in the municipality of Soacha. It offers accessibility to small community television providers so they can provide diverse content to their clients. Its development was made possible through the agile XP methodology, along with the PHP programming language, to obtain a web-oriented software prototype for television management in the municipality of Soacha.

Keywords: Internet TV, PHP, Web Programming, IPTV Protocol, Community Television

Tabla de Contenido

Resumen.....	2
Abstract.....	3
Tabla de Contenido.....	4
Lista de Tablas.....	9
Lista de Figuras.....	11
Introducción.....	14
Generalidades de la Investigación.....	16
Planteamiento del Problema.....	16
Formulación del Problema.....	18
Justificación.....	19
Alcance y Delimitación.....	21
Objetivos.....	22
Objetivo General.....	22
Objetivos Específicos.....	22
Marco Referencial.....	23
Antecedentes.....	23
Marco Conceptual.....	28
Servicio de Suscripción de Audio y Video (Televisión Paga).....	28

<i>Plataforma IPTV</i>	28
<i>Proceso de Transmisión IPTV</i>	29
<i>Protocolo de Transmisión</i>	29
<i>Protocolo TCP</i>	29
<i>Protocolo UDP</i>	30
<i>Protocolo RTP</i>	30
<i>Protocolo SDP</i>	30
<i>Protocolo RTSP</i>	30
<i>Hardware y Software para la Ejecución de IPTV</i>	30
<i>Condiciones de Transmisión de Bits</i>	30
<i>Software Interno del Sistema IPTV</i>	31
<i>Almacenamiento y Hosting del Software</i>	32
<i>Servicios de Transmisión Libre</i>	32
Marco Teórico.....	33
Redes Comunitarias	33
Ingeniería de Software	33
Programación Estructurada.....	34
Desarrollo Ágil.....	34
Marco Legal	36
Metodología	37

Metodología XP	37
<i>Planificación</i>	38
<i>Diseño</i>	41
<i>Codificación</i>	41
<i>Pruebas</i>	42
Planificación	44
Historia de Usuario	44
Roles del Proyecto	51
Plan de Entrega del Proyecto	52
Velocidad del Proyecto	53
Reuniones con la Comunidad	54
Reunión Inicial de Iteración	54
Reunión Matinal	63
Diseño	64
Simplicidad	64
Tarjetas CRC (Clase-Responsabilidad-Colaboración)	64
Metáfora del Sistema	68
Spike Solution	68
Refactoring	69
Codificación	70

Frontend	70
<i>Login</i>	70
<i>Login Index e Index2</i>	72
<i>User Panel</i>	73
<i>Panel</i>	74
<i>Servicios Facturación</i>	75
<i>Editar Usuario</i>	76
<i>Signin</i>	77
<i>Env Fact</i>	78
<i>Resend Password</i>	79
<i>CSS</i>	80
<i>Dependencias</i>	81
<i>Img</i>	81
Backend.....	82
<i>Conexión</i>	82
<i>Log</i>	83
<i>Reg</i>	84
<i>Actualizar Usuario</i>	85
<i>Eliminar Usuario</i>	86
<i>Killsess</i>	87

Base de Datos.....	88
<i>Usuarios</i>	89
<i>Servicios</i>	89
<i>Ventas</i>	90
Pruebas.....	91
Pruebas Unitarias	91
<i>Test Actualizar Usuario</i>	92
<i>Test Conexión</i>	93
<i>Test Eliminar Usuario</i>	94
<i>Test Killsess</i>	95
<i>Test Log</i>	96
<i>Test Reg</i>	97
Pruebas Funcionales.....	99
<i>Pruebas Funcionales para el Administrador</i>	100
<i>Pruebas Funcionales para el Usuario</i>	106
Resultados	110
Conclusión	112
Recomendaciones	114
Referencias Bibliográficas	115

Lista de Tablas

Tabla 1 Historia de Usuario Login de ingreso	45
Tabla 2 Historia de Usuario Recuperación de clave.....	45
Tabla 3 Historia de Usuarios de Cambio de contraseña	46
Tabla 4 Historia de Usuario Activación del servicio	46
Tabla 5 Historia de Usuario menú	47
Tabla 6 Historia de Usuario búsqueda de canales	47
Tabla 7 Historia de Usuario Pantalla completa.....	48
Tabla 8 Historia de Usuario Generar factura	48
Tabla 9 Historia de Usuario Registro de la información personal del cliente	49
Tabla 10 Historia de Usuario estado de cuenta.....	50
Tabla 11 Historia de Usuario Cerrar sesión	50
Tabla 12 Roles del Proyecto	51
Tabla 13 Plan de Entregas de las Historias de Usuario	52
Tabla 14 Velocidad del Proyecto	53
Tabla 15 Tareas.....	55
Tabla 16 Tarea 1 Historia de Usuario 1	56
Tabla 17 Tarea 2 Historia de Usuario 1	56
Tabla 18 Tarea 3 Historia de Usuario 2	57
Tabla 19 Tarea 4 Historia de Usuario 3	57
Tabla 20 Tarea 5 Historia de Usuario 4	58
Tabla 21 Tarea 6 Historia de Usuario 5	58
Tabla 22 Tarea 7 Historia de Usuario 6	59

Tabla 23 Tarea 8 Historia de Usuario 6	59
Tabla 24 Tarea 9 Historia de Usuario 7	60
Tabla 25 Tarea 10 Historia de Usuario 8	60
Tabla 26 Tarea 11 Historia de Usuario 9	61
Tabla 27 Tarea 12 Historia de Usuario 9	61
Tabla 28 Tarea 13 Historia de Usuario 10	62
Tabla 29 Tarea 14 Historia de Usuario 11	62
Tabla 30 Cronograma de Reuniones.....	63
Tabla 31 Login de Ingreso	65
Tabla 32 Recuperación de Clave	65
Tabla 33 Cambio de Contraseña	65
Tabla 34 Activación del Servicio.....	66
Tabla 35 Menú	66
Tabla 36 Búsqueda de Canales	66
Tabla 37 Pantalla Completa	67
Tabla 38 Generar Factura.....	67
Tabla 39 Base de Datos de los Clientes	67
Tabla 40 Estado de Cuenta	68
Tabla 41 Cerrar Sesión.....	68

Lista de Figuras

Figura 1 Gráfico transmisión de IPTV a través de una red.....	29
Figura 2 Login.....	71
Figura 3 Visualización de la Aplicación del Login de Ingreso.....	71
Figura 4 Index e Index2	72
Figura 5 Visualización del Reproductor Multimedia.....	73
Figura 6 User Panel.....	73
Figura 7 Visualización del Panel del Usuario.....	74
Figura 8 Panel	74
Figura 9 Panel de Gestión de Usuarios	75
Figura 10 Servicios Facturación	75
Figura 11 Visualización de Servicios y Facturación.....	76
Figura 12 Editar Usuario.....	76
Figura 13 Editar Usuario.....	77
Figura 14 Signin.....	77
Figura 15 Visualización de Registro de Cliente	78
Figura 16 Env Fact.....	78
Figura 17 Visualización de Factura Enviada	79
Figura 18 Resend Password.....	79
Figura 19 Formulario de Recuperación de Contraseña.....	80
Figura 20 CSS	80
Figura 21 Submódulo Dependencias	81
Figura 22 submódulo Img.....	81

Figura 23 Conexión.....	82
Figura 24 Codificación de la Conexión	82
Figura 25 Parámetros del Login.....	83
Figura 26 Codificación Log.....	83
Figura 27 Reg.....	84
Figura 28 Codificación Reg	84
Figura 29 Actualizar Usuario.....	85
Figura 30 Codificación de Actualizar Usuario	85
Figura 31 Eliminar Usuario	86
Figura 32 Codificación de la Opción Eliminar Usuario	86
Figura 33 Killsess	87
Figura 34 Codificación de Killsess	87
Figura 35 Base de Datos	88
Figura 36 Tabla Usuario	89
Figura 37 Tabla servicios.....	89
Figura 38 Tabla Ventas	90
Figura 39 test Actualizar Usuario	93
Figura 40 Test Conexión.....	94
Figura 41 Test Eliminar Usuario.....	95
Figura 42 Test Killsess.....	96
Figura 43 Test Log.....	97
Figura 44 Test Reg.....	99
Figura 45 Inicio de sesión	100

Figura 46 Navegación del panel de administración	101
Figura 47 Creación de usuario	102
Figura 48 Servicios y facturación	103
Figura 49 Editar usuario.....	104
Figura 50 Eliminar usuario	105
Figura 51 Inicio de sesión	106
Figura 52 Servicio de IPTV	107
Figura 53 Servicio de RADIO	108
Figura 54 Facturación y datos del servicio contratado	109

Introducción

La llegada sorpresiva del COVID-19 nos impactó demasiado a pesar de las advertencias previas. La televisión enfrenta el desafío de llegar al público y ampliar su cobertura a zonas rurales con recursos limitados. En Colombia, los canales privados dominados por grandes grupos económicos han llevado al cierre de canales públicos y comunitarios. Sin embargo, la promulgación de leyes de promoción de las TIC y la economía creativa en el país brinda oportunidades para impulsar la industria audiovisual y mejorar la conectividad en áreas remotas. Actualmente, la televisión en Colombia tiende a ser más comercial que cultural, mientras que los pequeños canales comunitarios luchan por sobrevivir debido a la falta de financiamiento. Ante esto, se requieren soluciones innovadoras que brinden acceso a televisión con contenido cultural y educativo.

El uso de la tecnología IPTV permite acceder a programas sin horarios establecidos, lo que beneficia a empresas y usuarios. El entretenimiento digital evoluciona constantemente y la transición a la tecnología digital es cada vez más común en los operadores de televisión. En este caso se propone desarrollar una aplicación de IPTV para que los proveedores de televisión comunitaria puedan llegar no solo al municipio de Soacha, sino también a los lugares más remotos donde exista cobertura de internet inalámbrica.

Con este proyecto se busca resolver un problema local, real y para ello se desarrolló un prototipo web basado en la metodología ágil XP, para gestionar la televisión IPTV en el municipio de Soacha, Cundinamarca. Esta metodología se divide en las fases de planificación, diseño, codificación y pruebas, y se enfoca en el desarrollo rápido de software y la adaptación a los cambios de requisitos para garantizar la calidad del producto entregado al cliente.

Durante el desarrollo del proyecto se identificaron 11 historias de usuarios para satisfacer las necesidades del administrador y el cliente. Se llevaron a cabo reuniones de planificación y seguimiento del progreso del proyecto. Se establecieron roles basados en la metodología XP, y se utilizó el diseño con tarjetas CRC para comprender el proceso de gestión de televisión con IPTV. Se siguió el paradigma de programación estructurada en la fase de codificación, utilizando PHP, JS, HTML y CSS3. También se realizaron pruebas unitarias y funcionales, que resultaron exitosas. Los usuarios valoraron la interfaz como intuitiva y atractiva, y la funcionalidad cumplió con sus expectativas en términos de apariencia y facilidad de uso.

En conclusión, el proyecto se desarrolló utilizando la metodología XP, lo que facilitó su ejecución y desarrollo. La planificación y recopilación de información fueron clave, y los requisitos funcionales se identificaron con éxito. Se diseñó una arquitectura escalable y eficiente, y se obtuvo un prototipo con una interfaz visualmente agradable. Finalmente, se realizaron pruebas exhaustivas que resultaron satisfactorias tanto en la ejecución del código como en la experiencia del usuario final.

Generalidades de la Investigación

Planteamiento del Problema

Teniendo en cuenta que se habían dado advertencias sobre una posible pandemia que afectaría al mundo entero, la llegada del COVID-19 nos tomó a todos por sorpresa, sin embargo, debemos señalar que, como menciona (Guevara Arroba & S. A., 2021), es importante analizar los siguientes factores: el espectáculo debía darse a pesar de que el mundo se refugió en casa. En este contexto histórico, el desafío para los medios de comunicación como la televisión y los programas de televisión es llegar al público, pero más allá de esto, es abrir cobertura a los lugares rurales donde el acceso y los recursos son mínimos. Además de acuerdo con (López Toapanta et al., 2022), el uso de proyectos de televisión por suscripción tiene altos costos de inversión, impide que los pequeños proveedores de TV por cable actualicen su tecnología, genera pérdida de clientes y la posibilidad de que sus servicios se cierren, lo que genera desempleo.

Otra cara de la moneda es que, en Colombia, de acuerdo con (Banrepcultural, 2012), en el año 2000 el advenimiento de los canales privados propiedad de los dos bloques económicos más grandes del país (Santo Domingo y Ardila Lülle) ha permitido a los programadores que anteriormente se habían adherido a las pautas comerciales abandonar estos grupos y mantener su canal en funcionamiento. Luego de varios intentos de supervivencia, programadores como Tevecine, Punch, JES y Cenpro cesaron sus actividades, muchos otros fueron intervenidos por su situación económica.

Por lo que, por un tiempo, los canales de televisión pública y de entidades independientes que no pertenecen a los dos gigantes empresariales privados, tendieron a desaparecer. (Rincón et al., 2020) menciona que la televisión rural incluye acceso a estos dos grandes canales privados

ya que han ampliado su cobertura a través de la tecnología TDT y a pequeños canales comunitarios que se limitan a la zona donde funcionan y pueden variar de municipio a municipio ya que son pequeños conglomerados que apuestan por llevar apuestas culturales y educativas por medio de la televisión y radio comunitarios.

De acuerdo con (Rincón et al., 2020) en Colombia se promulgó la Ley N° 1278 de promoción de las TIC y la Ley N° 1834 de promoción de la economía creativa y la Ley Naranja de 2019 son los marcos legales destinados para impulsar la industria audiovisual y llevar internet 5G a zonas remotas de Colombia. Si las radios y proveedores de televisión comunitarias deciden ingresar a la industria, pueden aprovechar los incentivos establecidos por estas regulaciones para producir contenido de medios cruzados y mejorar la conectividad a Internet. De lo contrario, ambos podrían desaparecer de las localidades en las que operan.

Por lo que actualmente en el país, se puede decir que la televisión ha tomado un rumbo más comercial que cultural y también de acuerdo con (Banrepcultural, 2012) un enfoque más político de acuerdo con los poderes privados que los representan. Es preocupante que Caracol y RCN al año 2014 sustenten entre ambos una cuota de audiencia del 44,58%, de televidentes en todo el país, mientras que los otros 89 canales del país (incluida la televisión pública) tienen el 55,42% restante (Banrepcultural, 2012).

Ante esto, se puede decir que el gran problema del servicio de televisión actualmente en Colombia es el monopolio que mueven dos gigantes corporativos, que promueven una televisión netamente comercial y que puede obedecer a intereses políticos que pueden no siempre ser imparciales. Además, de que pequeños canales de televisión comunitaria o pública han tendido a desaparecer o a ser liquidados, por algunas razones entre las que se encuentra la financiación, porque hacer televisión genera un alto costo, las principales fuentes de ingreso se obtiene por la

publicidad pero con la entrada de nuevos canales privados que cuentan con mayor infraestructura personal , programación y calidad, en ese sentido, los canales comunitarios al no contar con medios para competir en audiencia-mercancía pasaron a desaparecer del mercado, esto va de la mano con la falta de audiencia, con la llegada de la era digital muchos pequeños productores se les dificultó reinventarse para estar a nivel con el mercado. (Ramírez, 2015)

Actualmente, se puede ver televisión de calidad vía internet, por tanto, vale la pena plantear soluciones innovadoras que permitan el acceso a televisión con un contenido cultural y educativo.

Formulación del Problema

¿Cómo Implementar un prototipo orientado a la web para la gestión de televisión con IPTV en el municipio de Soacha de Cundinamarca?

Justificación

El desempleo provocado en el municipio de Soacha, al igual que muchas otras ciudades del país por las medidas del gobierno nacional para contener el COVID 19 es un tema importante que debe estar en la agenda nacional, sobre todo en el sector académico, como lo conocemos, según (Dane, 2022), la cifra de desempleo ronda el 17,3%. Si hay una respuesta del sector privado, utilizaremos herramientas tecnológicas para restaurar la mayoría de los empleos perdidos y restaurar la economía de una manera que sea segura para la vida de los ciudadanos. Además de la necesidad de interconectar a los sectores más alejados de la población rural a bajos costos.

La tecnología IPTV se está desarrollando rápidamente en todo el mundo. Por lo tanto, es importante investigar el mecanismo detrás de la tecnología IPTV que actualmente impulsa Internet. La televisión IPTV permite a los usuarios tener programas sin tener que recurrir a un horario establecido, además una dirección IP pública no requiere que estés en el mismo país para conectarte a ella, solo si tienes acceso a internet puedes acceder a ella, así que llevarlo al mercado resulta beneficioso para diferentes empresas. (López Toapanta et al., 2022)

Hoy en día, el contenido y el entretenimiento digital evolucionan constantemente para satisfacer a una nueva generación de consumidores, la solución de entretenimiento digital es la última adaptación de las plataformas de transmisión de video en constante evolución de manera efectiva, rápida y sustancial. La mayoría de los operadores de televisión del mundo están modernizando sus redes existentes mediante la introducción de plataformas digitales de vanguardia para maximizar su base de suscriptores y pasar del servicio de televisión analógico tradicional a la tecnología digital. Ha aparecido una nueva tecnología llamada Internet TV. IPTV es un término general utilizado para referirse a servicios de varios tipos de contenido de video

que utilizan IP como mecanismo de entrega de contenido. La ventaja de este mecanismo de distribución de señal de TV es que mejora la experiencia del usuario y la compatibilidad con las redes domésticas (Llerena Andrade, 2009).

Por otro lado, los canales de grandes consorcios privados como Caracol y RCN tienen una participación de audiencia mucho mayor que los demás concesionarios de televisión entre la audiencia del país, y que los proveedores de televisión pública del país, entonces para compensar estos números, el Ministerio de Tecnologías de la Información y las Comunicaciones (TIC) lanzó la campaña "Lanzamiento al público": asociarse con productoras de clase mundial para producir programación interesante y de alta calidad para mejorar el contenido del canal público nacional (Banrepcultural, 2012).

Además, con el fin de dar apoyo a que se continúe desarrollando la televisión comunitaria en el país que es la mayor proveedora de la televisión en zonas rurales pues el mercado de la televisión actualmente está más volcado hacia lo comercial y no tanto a la programación cultural y educativa que es allí donde marcará la diferencia el diseño y desarrollo de la aplicación de televisión que se propone. Pues los proveedores de televisión pública podrán publicar su contenido tanto comercial como educativo y de esta manera puedan atraer a más audiencia ofreciendo variedad en su programación. Por lo tanto, se propone desarrollar e implementar un aplicativo web para los servicios de IPTV basada en servidores que brinden accesibilidad a los pequeños proveedores de televisión comunitaria para que puedan ofrecer variedad de programación a sus clientes, y así de esta manera contribuir un poco en el mejoramiento del desempleo en Colombia.

Alcance y Delimitación

El alcance del proyecto está dado por el objetivo general, donde se establece que se realizará el diseño de un primer prototipo de implementación de IPTV en el municipio de Soacha Cundinamarca que no tienen acceso al servicio de televisión gratuita pero que sí se encuentran cobijados por programas gubernamentales de acceso a internet.

La delimitación en el desarrollo del presente proyecto está en las condiciones de red móvil que posea el municipio y las condiciones atmosféricas, de acceso y de consecución de equipos que reciban la señal con especificaciones técnicas ideales para la prestación del servicio y evitar el retraso en la señal, imagen pixelada o desconexiones frecuentes para ofrecer un servicio con calidad que beneficie el municipio de Soacha Cundinamarca.

Objetivos

Objetivo General

Implementar un prototipo orientado a la web implementando IPTV para la gestión de televisión comunitaria en el municipio de Soacha Cundinamarca.

Objetivos Específicos

Identificar los requisitos funcionales para un prototipo orientado a la web para la gestión de televisión con IPTV en el municipio de Soacha Cundinamarca.

Diseñar la arquitectura de un prototipo orientado a la web para la gestión de televisión con IPTV en el municipio de Soacha Cundinamarca.

Desarrollar el prototipo orientado a la web para la gestión de televisión con IPTV en el municipio de Soacha Cundinamarca.

Evaluar el prototipo orientado a la web para la gestión de televisión con IPTV en el municipio de Soacha Cundinamarca.

Marco Referencial

Para lograr el objetivo de este proyecto, fue necesario contar con la comprensión del contexto teórico y conceptual de la problemática y establecer las bases para el diseño y la implementación de la solución. En este sentido, se debe abordar diferentes disciplinas del conocimiento, que permitirán el éxito del proyecto, y llegar al diseño de una solución adecuada a las necesidades relacionadas con el acceso a la televisión, de la población del municipio de Soacha.

A continuación, se desarrollarán con mayor detalle los diferentes aspectos que conforman el marco referencial de dicho proyecto.

Antecedentes

Partiendo de la búsqueda realizada en diferentes bases de datos, es adecuado iniciar con la premisa de que Internet Protocol TV o IPTV debido a la facilidad de su tecnología e infraestructura es uno de los sistemas de distribución o transmisión de video más viable del mercado logrando mayor acceso en la población general, mejor calidad de imagen (condicionado a la limitación propias de cada red) y rápido acceso desde cualquier dispositivo mediante aplicativo o navegador web. Esto se explica a continuación a partir de la lectura analítica de los antecedentes encontrados sobre estudios implementados de esta tecnología de TV sobre diferentes sistemas y redes de internet.

En el estudio de (Ricardo Alonso Ferro Bolívar & Cesar Hernández, 2011), se realizó un conteo de características y atributos que tienen los sistemas de IPTV que ponen esta tecnología en ventaja sobre algunos sistemas de televisión actuales como lo son la TDT, el triple play, televisión móvil, pay per view (ppv), video on demand (vod), personal video recording (pvr) y switched digital broadcast (SDB), teniendo en cuenta la misma infraestructura de red que

utilizan. Esta comparación se realizó mediante una metodología de análisis bajo los parámetros de Calidad de Servicio (QoS) y de la Calidad de Experiencia del usuario (QoE). Llegó a la conclusión de que el servicio de IPTV ofrece más ventajas con respecto a otros servicios de televisión existentes porque no requiere el uso de una infraestructura diferente para su implementación y que sirve para una mejor personalización y administración de los servicios digitales.

En un proyecto aplicado realizado por (Zárate Torres, 2012), se propuso el diseño e implementación de una aplicación interactiva para implementar un modelo de formación a distancia por medio de la televisión denominado T-Learning, usando un protocolo GingaNCL. En este diseño se analizó tanto la factibilidad por medio de la televisión digital y sus características como el modelo IPTV y sus bondades al momento de diseñar un canal de televisión que sea transmitido por medio de la televisión digital con cobertura en los colegios intervenidos y el modelo IPTV para extender la cobertura en zonas rurales del país. El estudio concluye con que, en los lugares de baja cobertura y difícil acceso del servicio de televisión digital, se recomienda el uso de IPTV además de la tecnología GingaNCL para países con bajas condiciones sociales de acceso a la tecnología; además de una recomendación que hacen los autores sobre la necesidad que hay de que tanto Estado como La academia se unan en pro de fomentar el conocimiento sobre la implementación de televisión digital como acceso a la educación en países como Perú.

De acuerdo con el proyecto planteado (Caizaluisa Caizaluisa & Valle Alberca, 2011), el cual tuvo como objetivo principal recopilar y entregar un conocimiento más profundo acerca del beneficio y el funcionamiento de la tecnología IPTV, además de brindar un modelo para una posible implementación del servicio por parte de cualquier proveedor de servicios de internet y

telecomunicaciones; se encuentra el diseño de un prototipo de transmisión de datos que permite optimizar el funcionamiento de los protocolos IP para lograr prestar un servicio de televisión de calidad sin la necesidad de inversiones adicionales en redes de infraestructura y que ofrece las bondades de calidad de audio y video que ofrece la IPTV, teniendo en cuenta que debe ser un servicio que cuente con un buen ancho de banda para su óptimo funcionamiento. Los autores concluyen que este servicio será el que determine la prestación del servicio en el futuro por su calidad en la imagen y por la facilidad que representa para los proveedores de servicios de telecomunicaciones el implementarlo, monitorearlo y controlarlo.

De acuerdo con el trabajo de (Ramos Rodríguez & Pepinosa, 2013), el principal objetivo que persiguieron con este fue el de realizar un recuento de los principales estudios que han utilizado la tecnología IPTV sobre redes inalámbricas y móviles con Calidad en el Servicio (Qos). Donde concluyó que la IPTV es un servicio que depende de las buenas condiciones de la red para evitar problemas comunes como retardo, variación en el retardo y pérdida de paquetes que pueden ser solucionado desde la QoS, para asegurar la correcta prestación del servicio la solución se alberga en tecnologías móviles como la LTE y Wimax para lograr dicho propósito.

En el artículo de (Durán Dorado & Arciniegas, 2013), proponen una solución arquitectónica alternativa para implementar y soportar servicios de Video on Demand (VoD) para contenido de televisión en Internet Protocol Television (IPTV). Encontraron que, en comparación con otras arquitecturas, la solución entregada facilita la convergencia entre los estándares abiertos en el entorno de TV y las tecnologías Web 2.0 actuales, como servicios web, REST (transferencia de estado representacional) y XML (lenguaje de marcado extensible). Una serie de pruebas de la arquitectura prototipo los llevó a concluir que se puede lograr un buen rendimiento en términos de tiempo de respuesta en un entorno web real bajo estrés. En términos

de ancho de banda, el servicio puede distribuir todas las funciones a la velocidad de acceso disponible y, dado que la arquitectura está diseñada para redes no administradas, la calidad del servicio no puede ser garantizada.

Además, de acuerdo con otro estudio posterior llevado a cabo por (Ramos Rodríguez & Pepinosa, 2013), donde el objetivo principal fue el de realizar una simulación de tráfico para servicios móviles de IPTV en un modelo de red LTE. Tal modelo, explican los autores, se deriva de la arquitectura funcional para brindar servicios de IPTV en esta tecnología. Para la simulación, propusieron un escenario donde los recursos de la red se comparten entre las aplicaciones y los servicios móviles se adaptan según sea necesario. En este caso, se evaluó el comportamiento de los parámetros clave de rendimiento de la red (latencia, pérdida de paquetes y rendimiento) según el número de terminales móviles conectados a la red, los diferentes niveles de movilidad de los usuarios y los cambios en la configuración de la red. Por lo tanto, se propone un algoritmo que asegure el comportamiento óptimo de los parámetros de desempeño del tráfico de IPTV.

En este artículo los autores (Avellaneda et al., 2014), evaluaron las características de la entrega de contenidos de televisión y video por demanda como la pérdida de paquetes, retardo, fluctuación o variación del retardo sobre una plataforma IPTV-IMS, utilizando protocolo de flujo en tiempo real (RTSP) y protocolo de transferencia de hipertexto (HTTP) a pequeña escala. Su principal objetivo fue determinar cuál de estos presenta un mejor desempeño. Para esto, estudiaron múltiples escenarios en los cuales se varió tanto el servicio entregado como la estructura de la red y el flujo de datos. El análisis de los resultados les permitió concluir que, con las condiciones dadas, el protocolo HTTP presentó un mejor desempeño, mostrando menores pérdidas y valores de retardo y fluctuación dentro de los estándares dados para IPTV.

Otro estudio que se realizó en Colombia, por parte de (Martelo et al., 2015), tuvo como objetivo principal realizar un análisis sobre la implementación de TV por Internet (IPTV) en algunas organizaciones de la ciudad de Cartagena. Se identificó que las tecnologías de red más utilizadas en la ciudad (WAN), facilitaron el diseño y la implementación de dicha topología de red, que al mismo tiempo se evaluó bajo los parámetros de ingeniería de Calidad en el servicio (QoS). Finalmente, los autores elaboraron una guía que contiene información sobre las aplicaciones que brinda IPTV en la tecnología WAN, para facilitar la obtención de información técnica y adecuada para la implementación de la tecnología IPTV en su portafolio de servicios con el uso de dichas redes.

Finalmente, en términos generales, se concluye que debido a la simplicidad de la tecnología e infraestructura IPTV, es la más viable por encima de otros sistemas para la transmisión de TV digital, inclusive en lugares remotos. Además, por su naturaleza digital puede llegar a ofrecer mucho mejor calidad de imagen, pero también ligado y condicionado directamente a las condiciones de la red. Por otra parte, su protocolo de distribución (HTTP) le permite al usuario final, acceder desde cualquier tipo de dispositivo con un aplicativo nativo, genérico o desde algo tan simple como un navegador web.

Marco Conceptual

En este apartado, se presentan los conceptos más relevantes y necesarios para el desarrollo del proyecto.

Servicio de Suscripción de Audio y Video (Televisión Paga)

Es un servicio que recibe imágenes, audio, señales multimedia y datos solo para un suscriptor o audiencia específica. Considerando que el hogar promedio tiene 3.8 personas conectadas a la cuenta de cada suscriptor, el número estimado de usuarios del sitio es de 48.258.494 residentes del estado (Dane, 2022).

El nuevo servicio de streaming multimedia tiene muchas ventajas frente a la televisión de pago. En primer lugar, el precio es más bajo. Por ejemplo, en el país, el plan básico de televisión de pago cuesta alrededor de \$70.000/mes, mientras que el servicio de transmisión más popular cuesta \$16.000/mes para el plan Básico y entre 30.000 y 40.000 para el plan Premium. Otra razón por la que cada vez más personas eligen plataformas de transmisión. (Santi Deliani Rahmawati & Halimatus Saidah, 2020)

Plataforma IPTV

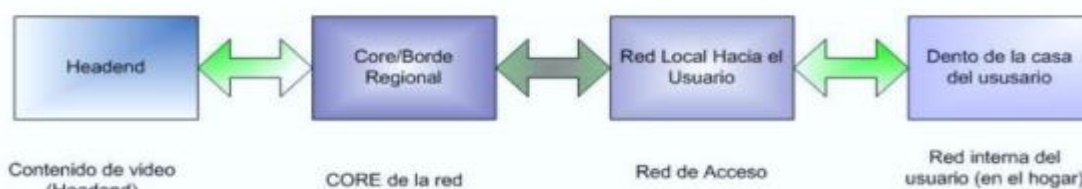
Con el desarrollo del mundo de Internet, la industria de la televisión ha sufrido muchos cambios, por lo que han aparecido nuevas actividades, como la combinación de datos del mercado de contenido de TV y el mercado de contenido de Internet. Esta tecnología se llama IPTV. Internet TV es un método de transmisión de televisión a través de Internet mediante la entrega y transmisión bajo demanda de contenido de TV o video a una red de banda ancha (Yúbal Fernández, 2022).

Proceso de Transmisión IPTV

En la Figura 1, se describe este proceso donde primero, el punto central de la red, también conocido como cabecera, es responsable de amplificar y ajustar el contenido para una distribución adecuada a través de los sistemas IP. Luego, todo el contenido del sistema se enruta a través de la red CORE que transporta el tráfico de datos. Luego, la red de acceso proporciona conectividad de red a los consumidores de servicios de IPTV y, finalmente, en la intranet del usuario, servicios de transmisión a los hogares a través de enrutadores. (Vinicio & Meza, 2008).

Figura 1

Gráfico transmisión de IPTV a través de una red.



Nota. adaptado del Gráfico transmisión de IPTV a través de una red. (Vinicio & Meza, 2008)

fuelle <https://dspace.ups.edu.ec/bitstream/123456789/21914/1/UPS%20-%20TTS620.pdf>

Protocolo de Transmisión

Al igual que la tecnología utilizada en Internet, la tecnología IPTV utiliza protocolos de transmisión y control de versiones para acceder a este servicio. Protocolo TCP, protocolo UDP, protocolo RTP, protocolo SDP, protocolo RTSP (Mina Gómez, 2014).

Protocolo TCP

De acuerdo con (Tablado & F, 2021), el protocolo TCP o Transfer Control Protocol, consiste en un protocolo estandarizado sobre el cual se realiza la transferencia de datos entre los participantes de la red informática. Los programas que componen una red de datos en una red informática utilizan el protocolo TCP para establecer conexiones entre ellos, garantizando así el

flujo de datos entre las partes.

Protocolo UDP

User Datagram Protocol, este protocolo se limita a recopilar mensajes y enviar paquetes a través de la red sin establecer una conexión, porque el datagrama en sí contiene suficiente información de direccionamiento en su encabezado. Para garantizar la llegada, el protocolo requiere que el anfitrión del paquete envíe el mensaje (Still Puebla & Simon Antony, 2018).

Protocolo RTP

Real-Time Transport Protocol, este protocolo se usa para transportar datos de flujos de multimedia sobre redes IP de conmutación de paquetes (Boronat Segui, 2019).

Protocolo SDP

Session Description Protocol, este protocolo proporciona un formato de tipo texto para la descripción de sesiones multimedia y es utilizado en una variedad de entornos de red y aplicaciones (Boronat Segui, 2019).

Protocolo RTSP

Real Time Streaming Protocol, permite al cliente de streaming controlar la transmisión de flujos multimedia (Oliver Gil, 2018).

Hardware y Software para la Ejecución de IPTV

En un sistema de IPTV, la red debe estar diseñada para que pueda manejar una amplia variedad de servicios y aplicaciones, al tiempo que proporciona mecanismos de control adecuados para garantizar la calidad de servicio del contenido entregado para los usuarios del sistema de IPTV sin afectar la operación. de otros. (Ramos Rodríguez & Pepinosa, 2013)

Condiciones de Transmisión de Bits

El contenido de TV se puede entregar en 4:3 para SDTV o 16:9 para HDTV y a altas

tasas de bits de 1,75 Mbps a 15 Mbps utilizando tecnología de video H.262 con MPEG-2-Main para transmisión SDTV de 2,5 Mbps y HDTV de 15 Mbps (720p /1080i) corrientes. Si usa H.264, también conocido como MPEG-4, la velocidad de codificación de video se reducirá a 1,75 Mbps para SDTV y 10 Mbps para HDTV. En el caso de tecnologías como MPEG-2 AAC o MPEG-4 para la codificación de contenidos de audio para SDTV y HDTV, se utilizan el primer y segundo método para conseguir 128 kbps y 96 kbps para sonido estéreo. Asimismo, la calidad del contenido depende del rendimiento del receptor y determina si el dispositivo puede decodificar una señal SDTV o HDTV. Por esta razón, Scalable Video Coding (SVC) debe utilizarse en los sistemas de IPTV para ajustar el bitrate de acuerdo con las características del equipo del usuario y el bitrate disponible en el cable de enlace (Ramos Rodríguez & Pepinosa, 2013)

Software Interno del Sistema IPTV

El sistema de IPTV comienza con el servidor donde se almacena el contenido de video fuente; señales de televisión por satélite, contenido pregrabado o eventos en vivo, las señales almacenadas en la fuente deben ser convertidas en datos digitales por el decodificador, y los datos digitales encriptados se colocarán en un paquete IP para su entrega; esta señal se envía a la unidad superior para decodificar y convertir la señal transmitida en el televisor del usuario. (Mina Gómez, 2014).

Los sistemas de IPTV tienen la ventaja de brindar a los usuarios una interfaz que les permite interactuar con el sistema; En otras palabras, la capa de aplicación del sistema IPTV es el middleware que gestiona las aplicaciones entregadas a los usuarios como parte de los servicios contratados como EPG Program Guide, DVR Personal PVR y TVTS TV con evaluación meteorológica. (Mina Gómez, 2014)

Almacenamiento y Hosting del Software

Para el desarrollo del proyecto, cualquier plan de hosting compartido es idóneo dado que sería la primera fase y el script para desarrollarse es supremamente liviano (no más de unos cuantos megas). La carga principal de los contenidos (Las señales de IPTV) provienen de forma independiente de cada uno de los canales de tv que la transmiten, de esta forma, si tenemos 1000 canales de tv, serán 1000 servidores diferentes soportando las diferentes transmisiones. Si en dado caso en una fase de producción el servicio de hosting del software se ve afectado por el exceso de demanda, siempre será posible escalar el servicio a un servidor dedicado.

Servicios de Transmisión Libre

El término Over The Top (OTT) se refiere a los servicios disponibles en línea, en este caso Internet, pero no proporcionados por un proveedor de servicios de Internet (ISP). (Margaret Rouse, 2019). La forma de acceder a los contenidos audiovisuales ha evolucionado continuamente, desde la forma de distribución, como la aparición de la televisión, hasta la forma de recepción de vídeo sobre IP (Protocolo de Internet), así como los dispositivos de recepción han ampliado su visión. estos contenidos a medida que se mueve desde un dispositivo como un televisor hay muchas formas de acceder a este contenido como una tableta, teléfono inteligente o Smart TV. (D. Pablo Casco Castillo, 2018).

Marco Teórico

A continuación, se presentan las teorías que van a dar soporte a esta investigación.

Redes Comunitarias

Las redes comunitarias son redes de telecomunicaciones (Internet, Intranet o Telefonía Móvil) de propiedad y gestión colectiva de las comunidades, sin finalidad de lucro y con fines comunitarios. Se unen al tejido comunitario y buscan contribuir a los procesos educativos, económicos, de salud, políticos de acuerdo con una estrategia de lo que se quiere comunicar.

En las redes comunitarias la comunidad participa activamente en el diseño, construcción y operación, con el fin de conectar, comunicar y mejorar la calidad de vida en su territorio; disminuyendo la brecha digital, fortaleciendo la inclusión social y logrando el acceso a las nuevas tecnologías de la información y la comunicación (TIC) a través de la autonomía comunitaria.(Colnodo, 2019)

Ingeniería de Software

La Ingeniería del Software es una rama de la ingeniería que se enfoca en el desarrollo y mantenimiento de software utilizando tecnologías y prácticas de diversas disciplinas. Algunas personas opinan que debería llamarse "Desarrollo del Software", ya que el término ingeniería implica un mayor nivel de rigor y pruebas que lo que se suele hacer actualmente. Según la definición del IEEE, el software es un conjunto de programas, procedimientos, reglas, documentación y datos que pertenecen a un sistema de cómputo. La Ingeniería del Software se define como un enfoque sistemático para el desarrollo, operación, mantenimiento y retirada de software diseñado para usuarios, y su nombre fue acuñado por primera vez Fritz Bauer.

(Barzanallana Rafael, 2014)

Programación Estructurada

Es un paradigma de programación que se estableció en los años 70 con el objetivo de mejorar las prácticas de programación y facilitar la creación de programas más complejos y fáciles de mantener. Este paradigma consiste en un estilo de programación que utiliza estructuras básicas como ciclos y condicionales, añadiendo la posibilidad de ejecutar subrutinas o funciones. Antes de la programación estructurada, se usaba una instrucción denominada "GOTO" para controlar el flujo de ejecución de las sentencias, lo que hacía que los programas fueran difíciles de entender y mantener. La programación estructurada propuso en su lugar el uso de subrutinas, lo que permitió organizar el código de una manera más clara, además de permitir también crear pequeñas piezas de código fácilmente utilizables y mantenibles.

Las ventajas de la programación estructurada incluyen una mayor capacidad de organización jerárquica del código, la capacidad de abstracción, una mayor capacidad de depuración, un aumento en la velocidad de desarrollo y una reducción de la complejidad del software, motivo por el cual se decide implementar dicho paradigma en el presente proyecto, dada la simplicidad de los requerimientos del mismo. (Álvarez Miguel Ángel, 2022)

Desarrollo Ágil

La gestión ágil de proyectos es un enfoque diferente a la gestión de proyectos tradicional, que se centra en la adaptabilidad y la evolución junto con el software que se está desarrollando. Es un conjunto de prácticas, métodos, principios, técnicas y herramientas que tienen como objetivo mejorar el desempeño del equipo y producir mejores resultados durante el proyecto, especialmente en el desarrollo de software. Los dos enfoques principales para la gestión de proyectos son ágil y predictivo (tradicional), y el primero incluye varias metodologías, como Scrum, Kanban y eXtreme Programming, siendo esta última la que se aplica en el presente

proyecto. La gestión ágil de proyectos se enfoca en entregar software con el mayor valor posible, mientras que la gestión tradicional de proyectos se enfoca en aspectos predefinidos como el alcance, el costo y el tiempo. El enfoque ágil se basa en cuatro valores y 12 principios que se derivan del Manifiesto Ágil, que dan prioridad a las personas y las interacciones, el software funcional, la colaboración con el cliente y la respuesta al cambio. La gestión ágil de proyectos también enfatiza la autogestión de los equipos de desarrollo, quienes se encargan de entregar software 100% funcional en períodos iterativos de 2 a 4 semanas, priorizando la calidad de los profesionales sobre los procesos y herramientas propuestas en la gestión tradicional de proyectos. (Manchón Eduardo, 2006)

Marco Legal

De acuerdo con el marco regulatorio del sector de las telecomunicaciones colombiano para el período 1990-2009 estuvo determinado por la Ley N° 142 de 1994 (Ley de Servicios Públicos Locales), que además de la regulación del sector estudiado, también regula el suministro de agua y alcantarillado y energía eléctrica, distribución de gas, teléfono fijo universal, telefonía móvil local en zonas rurales; la ley introduce y activa el principio de libertad de empresa (artículo 10) y con la ayuda del artículo 11, se especifica la función social de la propiedad de una empresa de servicios públicos.

Metodología

El presente proyecto de investigación es de tipo aplicado, es decir que el propósito es ayudar a resolver un problema local existente del mundo real, razón por la cual los proyectos aplicados también se denominan estudios aplicados. De esta manera, el propósito de un proyecto aplicado es recopilar información para ayudar a abordar un problema identificable en un entorno específico. (Bastis Consultores, 2022)

En este capítulo se presenta la metodología utilizada para el desarrollo del prototipo orientado a la web para la gestión de televisión con IPTV en el municipio de Soacha, Cundinamarca, basada en la metodología XP (eXtreme Programming). es una metodología ágil que se enfoca en el desarrollo rápido de software y la adaptación a los cambios en los requisitos del proyecto, se divide en cuatro fases: planificación, diseño, codificación y pruebas. En cada fase, se llevan a cabo una serie de prácticas que ayudan a garantizar la calidad del software y a maximizar el valor entregado al cliente.

Metodología XP

La metodología que se va a implementar para llevar a cabo el presente proyecto, es la metodología Extreme Programming o Metodología XP, esta metodología que se da en el desarrollo de las metodologías ágiles para el diseño de software proporciona grandes ventajas para el desarrollo de la aplicación web que se desea llevar a cabo, pues brinda la flexibilidad de ajustarse a los requerimientos cambiantes de sus clientes y brinda los métodos y herramientas necesarias para adaptar el diseño a las exigencias del proyecto (Bello, 2021).

Este proyecto propone realizar una aplicación práctica de dicha metodología XP para el desarrollo de software y de esta manera contribuir a determinar la idoneidad del método para las condiciones del presente caso, donde el desarrollador es una sola persona.

Planificación

La planificación es la primera fase de cualquier proyecto XP. En este punto, se comienza a interactuar con los clientes y otros miembros del equipo de desarrollo para explorar los requisitos del sistema. Esta fase identifica los cambios metodológicos necesarios en función del número y tamaño de las iteraciones y las características del proyecto (Echeverry Tobón & Delgado Carmona, 2007). Esta sección cubrirá los siguientes puntos: 1. historias de usuario, 2. progreso del proyecto, iteraciones, entregas pequeñas, reuniones, roles en XP.

Historias de Usuarios. De acuerdo con (Echeverry Tobón & Delgado Carmona, 2007), las historias de usuario se utilizan como una herramienta para informar al equipo de desarrollo sobre los requisitos del sistema. Estos son pequeños textos de cliente que describen las acciones que realizará el sistema; están escritos en términos de clientes, no de desarrolladores, por lo que son claros y fáciles de repetir.

Progreso del Proyecto. De acuerdo con (Echeverry Tobón & Delgado Carmona, 2007), el progreso del proyecto se mide en el número de historias de usuario que se pueden evacuar dentro de una Iteración, ya que esto nos dará un estimado de tiempo total en el que se puede gestionar una historia del usuario y el tiempo total que se demoran las Iteraciones. Realizando este cálculo, se pudo comprobar que es posible realizar 4 historias de usuario por Iteración.

Iteraciones. En el método XP, la creación del sistema se divide en varias fases para facilitar la implementación. A menudo, los proyectos contienen más de tres fases llamadas iteraciones, de ahí el concepto de enfoque iterativo. Al final de cada iteración, se proporcionarán dispositivos compatibles que deben pasar las pruebas de aceptación creadas por el cliente para verificar el cumplimiento.

Las tareas que no se completan en una iteración se considerarán en la próxima iteración según lo determine el cliente, independientemente de si deben completarse o eliminarse del diseño del sistema. (Echeverry Tobón & Delgado Carmona, 2007)

Entregas Pequeñas. De acuerdo con (Echeverry Tobón & Delgado Carmona, 2007), el ciclo de iteración es de una a tres semanas, y al final se entregará un producto de muestra que debería ser completamente funcional.

Reuniones. La planificación es esencial para cualquier enfoque, por lo que XP requiere una revisión constante del plan de negocios. Si bien esta es una forma de evitar el exceso de documentación, es muy estricta cuando se trata de organizar su trabajo y las reuniones son básicamente revisiones periódicas del progreso del proyecto (Echeverry Tobón & Delgado Carmona, 2007). De acuerdo con lo anterior, y siguiendo lo postulado por la metodología XP, las reuniones que deben llevarse a cabo son las de: plan de entregas, inicial de iteración, diarias o stand up meeting.

Roles en XP. En este enfoque, el concepto de roles se utiliza para organizar a las personas responsables de cada actividad durante el proyecto. Cada uno de estos roles es realizado por uno o más miembros del equipo, y no se excluye el cambio de roles entre equipos durante la implementación del sistema. El director del proyecto es responsable de dirigir y organizar reuniones a lo largo del proyecto, una de sus responsabilidades es decidir qué hacer, cuándo hacerlo y revisar cómo se desarrolla el sistema, con el apoyo de los clientes, seguidores y otros miembros del equipo (Echeverry Tobón & Delgado Carmona, 2007). Esta metodología cuenta con los siguientes roles.

El jefe de proyecto tiene de como responsabilidad la dirección y organización las reuniones que se realizan durante el proyecto. Es erróneo afirmar que entre sus tareas se

encuentra decir que hacer, cuando hacer y de revisar cómo se desarrolla el sistema, para ello se cuenta con el apoyo del cliente, el tracker y los demás miembros del grupo.

El usuario o cliente determina qué se va a construir en el sistema, además de decidir el orden en que se entregarán cada segmento del proyecto. Es parte fundamental del equipo XP (se menciona su importancia como una de las prácticas), en todo proyecto debe existir un cliente. Además, tiene como tarea establecer las pruebas de aceptación, las cuales determinan si el sistema cumple con los requerimientos del usuario.

En el grupo de los programadores se encuentran además los diseñadores y los analistas. Los programadores son quienes construyen el sistema y realizan las pruebas correspondientes a cada módulo o unidad de código. Cuando surgen dudas o preguntas que afectan decisiones sobre la funcionalidad del sistema (las decisiones técnicas son solucionadas gracias a las habilidades de los programadores), el programador no debe hacer suposiciones acerca de lo que el cliente quiere; en este caso, debe dirigirse al mismo y aclarar la situación.

El entrenador (coach) es el responsable de que el proceso se realice de forma correcta. Se asegura de que los conceptos de la metodología se apliquen al proyecto, además de brindar ayuda continua a los demás integrantes del equipo.

El tester o quien realiza las pruebas, colabora en la realización de las pruebas de aceptación y es quien muestra los resultados de las mismas. En este proceso, ayuda al cliente a diseñar tales pruebas y a verificar que las pruebas sean aprobadas.

El rastreador (tracker) tiene como tarea observar la realización del sistema. Varias veces por semana cuestiona a los integrantes del equipo para anotar sus logros y avances. Mantiene datos históricos del proyecto.

Diseño

El diseño de la metodología XP se basa en establecer los códigos y recursos necesario para el desarrollo del proyecto, donde se diseñan las ideas, procesos, entregables fijados en la planeación y se implementa la creación de tarjetas CRC (Clase-Responsabilidad-Colaboración) estas permiten obtener una imagen visual de cómo se relacionan e interactúan las funciones de una manera global por medio de la programación utilizadas en cada una de las historias. (Alicia Raeburn, 2022) .

Las tarjetas CRC incluyen:

Clase (conjunto de elementos similares)

Responsabilidades (según la clase)

Colaboradores (clase que interactúa con el colaborador en cuestión)

Codificación

La programación es un proceso paralelo al diseño y está influenciado por comentarios de XP que algunos expertos consideran controvertidos, como la rotación de programadores o la programación en pareja. Además de los temas anteriores, los lectores encontrarán explicaciones de los siguientes temas: atender siempre al cliente y probar el código primero (Echeverry Tobón & Delgado Carmona, 2007).

Atender Siempre al Cliente. Uno de los requisitos de XP es que el cliente esté siempre disponible. Necesitas ser parte del equipo de desarrollo, no solo despejar sus dudas. En este sentido, es muy útil para despejar las dudas que puedan surgir (sobre todo a nivel personal) para asegurar que el trabajo realizado responde a las necesidades expresadas en la historia de usuario (Echeverry Tobón & Delgado Carmona, 2007).

Probar el Código Primero. La creación de una prueba primero le ahorra mucho tiempo al escribir el código para pasar la prueba, lo que lleva menos tiempo que crear el código solo. Una de las ventajas de probar antes del código es que le permite establecer requisitos para el mismo, en otras palabras, escribir las pruebas primero hace que sea más fácil y manejable encontrar todos los casos especiales en los que se debe usar el código. De esa forma, el desarrollador puede estar seguro de que ha terminado porque todas las pruebas han pasado (Echeverry Tobón & Delgado Carmona, 2007).

Pruebas

De acuerdo con (Echeverry Tobón & Delgado Carmona, 2007) la metodología XP toma muy en serio los aspectos relacionados con las pruebas, clasificándolos según diferentes tipos y funciones específicas, especificando quién debe usarlos e implementarlos, cuándo y cómo. El buen uso de las pruebas depende del éxito de otras aplicaciones, como la propiedad del paquete de código y la refactorización. Cuando las pruebas se realizan bien, no hay necesidad de preocuparse de que otros programadores cambien su código, es decir, si una parte está rota, la prueba mostrará un error y le permitirá encontrarla. Los mismos criterios se aplican a las conversiones. Una de las cosas que puede evitar que un programador cambie una pieza de código funcional es evitar que funcione por completo, este miedo se alivia en gran medida si realiza una serie de pruebas para asegurarse de que funcionan correctamente. Bajo XP, tienes que ser muy estricto con tus pruebas. Se puede lanzar una nueva versión solo después de pasar el 100% de todas las pruebas, de lo contrario, este resultado se utilizará para detectar el error y solucionarlo mediante un mecanismo predefinido.

Pruebas Unitarias. Estas pruebas se aplican a todos los métodos importantes en todas las fases del proyecto, siempre que se omita cualquier fase sin vincular el conjunto de pruebas correspondiente. Uno de los elementos clave es que, idealmente, los métodos se crean antes que ellos, brindando a los programadores la mejor comprensión de lo que necesitan programar y cada clase de prueba que deben pasar. Esto mejorará el trabajo y el código será de mayor calidad. Los desarrolladores deben construirlo utilizando un mecanismo que les permita automatizar la implementación y la ejecución de la manera más pequeña y eficiente posible (Echeverry Tobón & Delgado Carmona, 2007).

Verificación de Aceptación. El cliente gestiona las pruebas de aceptación, también conocidas como pruebas funcionales, en función de los requisitos de la historia de usuario. En todas las iteraciones, cada historia de usuario elegida por el cliente debe tener una o más pruebas de aceptación donde se deben definir los casos de prueba y corregir los errores (Echeverry Tobón & Delgado Carmona, 2007).

Cuando se Encuentra un Error. Debe escribir una prueba antes de intentar solucionarlo. De esa forma, tanto el cliente sabe qué es y dónde está, como el equipo de desarrollo puede enfocarse mejor en (Echeverry Tobón & Delgado Carmona, 2007).

Por otro lado, se puede evitar que vuelva a suceder, si un cliente informa un error y el cliente ejecuta pruebas de aceptación para el equipo de desarrollo, el programador responsable también puede generar nuevas pruebas unitarias para que pueda encontrar la parte específica donde está el error (Echeverry Tobón & Delgado Carmona, 2007).

Planificación

A partir de este capítulo se describe las necesidades escritas por el cliente, a través de reuniones con la comunidad para ver la viabilidad de llevar el servicio a los hogares, identificando las historias de usuario por medio de la metodología XP, esta sirve como base metafórica para el sistema, donde el profesional tomará como referencia antes de iniciar la codificación, pueden modificarse, replantearse e incluso surgir nuevas historias en el desarrollo del proyecto.

Historia de Usuario

Las historias de usuario poseen información útil de las necesidades a satisfacer del cliente son descripciones cortas y simples, escrita por el usuario en un lenguaje fácil de comprender, para ser adapta al sistema de la aplicación IPTV.

La historia de usuario de la aplicación IPTV

Login de ingreso

Recuperación de clave

Cambio de contraseña

Activación del servicio

Menú

Búsqueda de canales

Pantalla completa

Generar factura

Base de datos de los clientes

Estado de cuenta

Cerrar sesión

A continuación, la presentación de las tarjetas de usuarios para el desarrollo de la aplicación IPTV.

Tabla 1

Historia de Usuario Login de ingreso

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
1	Login de ingreso	Alta	Administrador, Cliente	Cliente, Administrador	Baja	que me permita ingresar a la aplicación		1	2

Nota. En esta tabla se muestran los actores y la descripción de la historia de usuario login de ingreso. *Fuente.* Elaboración propia

Tabla 2

Historia de Usuario Recuperación de clave

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
2	Recuperación de clave	Alta	Administrador, Cliente	Cliente, Administrador	Baja	que garantice recuperar la contraseña cuando se me olvide		1	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario recuperación de clave. *Fuente.* Elaboración propia

Tabla 3*Historia de Usuarios de Cambio de contraseña*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
3	Cambio de contraseña	Alta	Administrador, Cliente	Cliente , Administrador	Baja	que permita cambiar la contraseña		2	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario cambio de contraseña. *Fuente.* Elaboración propia

Tabla 4*Historia de Usuario Activación del servicio*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
4	Activación del servicio	Alta	Administrador	Administrador	Media	activación de la aplicación de manera instantánea		2	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario activación del servicio. *Fuente.* Elaboración propia

Tabla 5*Historia de Usuario menú*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
5	Establecer el menú	Alta	Cliente	Cliente	Baja	poder navegar los canales desde la lista de opciones de la aplicación		2	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario menú. *Fuente.* Elaboración propia

Tabla 6*Historia de Usuario búsqueda de canales*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
6	búsqueda de canales	Alta	Cliente	Cliente	Baja	variedad en canales y facilidad para su búsqueda	Canales de contenido educativo y de entretenimiento, para la población infantil, jóvenes y adultos.	3	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario búsqueda de canales. *Fuente.* Elaboración propia

Tabla 7*Historia de Usuario Pantalla completa*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
7	Pantalla completa	Alta	Cliente	Cliente	Baja	Quiero ampliar la pantalla durante la programación	Que la aplicación tenga la opción de minimizar y ampliar la pantalla	3	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario pantalla completa. *Fuente.* Elaboración propia

Tabla 8*Historia de Usuario Generar factura*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
8	Generar factura	Alta	Administrador	Administrador	Media	Generar cobro del servicio por medio de la factura		3	3

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario generar factura. *Fuente.* Elaboración propia

Tabla 9*Historia de Usuario Registro de la información personal del cliente*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
9	Base de datos de los clientes	Alta	Administrador	Administrador	Media	una base de datos con la información del cliente, desde la fecha ingreso, información personal, saldos pendientes entre otros.		3	3

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario registro de la información del cliente. *Fuente.*

Elaboración propia

Tabla 10*Historia de Usuario estado de cuenta*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
10	Estado de cuenta	Alta	Cliente	Cliente	Baja	visualizar el estado de mi saldo a pagar		4	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario estado de cuenta. *Fuente.* Elaboración propia

Tabla 11*Historia de Usuario Cerrar sesión*

Numero	Nombre	Prioridad	Responsable	Usuario	Riesgos en desarrollo	Descripción	Observación	Iteración	Esfuerzo
11	Cerrar sesión	Alta	Administrador	Cliente , Administrador	Baja	finalización de la sesión		4	2

Nota. En esta tabla se muestra los actores y la descripción de la historia de usuario cerrar sesión. *Fuente.* Elaboración propia

Roles del Proyecto

Dentro del desarrollo del proyecto de la aplicación IPTV contara con los siguientes roles los cuales son:

Jefe de proyecto

Cliente

Programador

Por otra parte, es importante resaltar que el programador actuara como jefe direccionando y organizando todo lo respectivo con el proyecto, el desarrollo del sistema, realización de las pruebas, observador de las metas cumplidas y de los avances obtenidos en el proceso.

Tabla 12

Roles del Proyecto

Roles	Responsabilidades
Project manager	Dentro de las personalidades del Project manager es supervisar y direccionar la ejecución del proyecto, evaluando cada una de las etapas para alcanzar los objetivos propuestos.
Cliente	Es la persona quien va a adquirir el servicio , este nos brindara los requerimientos básicos para el desarrollo de la aplicación

Nota. En esta tabla muestra los roles desempeñados en el proyecto. *Fuente.* Elaboración propia

Plan de Entrega del Proyecto

Teniendo definidas las historias de usuario proporcionadas por el cliente para el desarrollo de la Aplicación IPTV, se presentará la secuencia de entregas a realizar durante cada iteración, el plan de entregas detalla la prioridad de cada historia, esfuerzo, fecha de inicio y finalización como se representa en la tabla.

Tabla 13

Plan de Entregas de las Historias de Usuario

Historia	Iteración	Prioridad	Esfuerzo	Inicio	Final
Historia 1	1	Alta	2	26 /05/ 2023	2 /06/2023
Historia 2	1	Alta	2	3 /06/ 2023	10 /06/ 2023
Historia 3	2	Alta	2	1 1/06/2023	16/ 06/2023
Historia 4	2	Alta	2	17 /06/ 2023	20 /06/2023
Historia 5	2	Alta	2	21/06/ 2023	24/06/2023
Historia 6	3	Alta	2	30 /06/ 2023	6 /07/2023
Historia7	3	alta	3	7 /07/2023	10 /07/2023
Historia 8	3	Alta	3	11/07/ 2023	14 /07/2023
Historia 9	4	Alta	3	15/07/2023	20 /07/ 2023
Historia 10	4	Alta	2	21 /08/2023	28/08/2023
Historia 11	4	Alta	2	29/08/2023	05/08/2023

Nota. Plan de entregas de las historias de usuario para el diseño de la aplicación. *Fuente.*

Elaboración propia

Velocidad del Proyecto

Tabla 14

Velocidad del Proyecto

HISTORIA	HORAS	SEMANA	HORAS POR SEMANAS
HISTORIA 1	8 horas	1 semana y 5 días	72 horas
HISTORIA 2	8 horas	1 semana	48 horas
HISTORIA 3	8 horas	1 semana	48 horas
HISTORIA 4	8 horas	1 semana	32 horas
HISTORIA 5	8 horas	1 semana	32 horas
HISTORIA 6	8 horas	1 semana	56 horas
HISTORIA 7	8 horas	1 semana	32 horas
HISTORIA 8	8 horas	1 semana	32 horas
HISTORIA 9	8 horas	1 semana	40 horas
HISTORIA 10	8 horas	1 semana	48 horas
HISTORIA 11	8 horas	1 semana	48 horas
TOTAL			488 horas

Nota. Velocidad de desarrollo de la aplicación. *Fuente.* Elaboración propia

Reuniones con la Comunidad

Las reuniones con la comunidad se realizarán antes de empezar el proyecto con el fin de recoger información para el diseño de la aplicación partiendo de la necesidad descrita y obtenida por cliente, a través de métodos de recolección de información para el desarrollo de la aplicación IPTV.

Durante la reunión se interactuó directamente con los clientes que corresponden a la población del municipio de Soacha, a quienes se les planteó el proyecto, se expresó de manera grupal los puntos importantes que les gustaría encontrar en la aplicación, por medio de una actividad lúdica y recreativa para hacer más entretenido el momento las personas que participaban expresaban de manera verbal sus necesidades y que esperaban de la nuestra propuesta.

Reunión Inicial de Iteración

Las reuniones realizadas en el equipo de trabajo se basarán en delegar responsabilidades para el desarrollo de las historias de usuario antes de iniciar cada interacción, con el fin de facilitar las cargas de trabajo, brindar un servicio de calidad.

Tabla 15*Tareas*

NÚMERO DE LA TAREA	NÚMERO DE LA HISTORIA	NOMBRE DE LA TAREA
1	1	Diseño de interfaz de sistema de inicio de sesión
2	1	Validación del sistema de inicio
3	2	Desarrollo del sistema de recuperación de contraseña
4	3	cambio de contraseña desde el panel de usuario
5	4	Sistema de creación y activación del servicio desde el panel del administrador
6	5	Interfaz del diseño del menú
7	6	Integración de los canales para la aplicación
8	6	Interfaz de la búsqueda de canales
9	7	Desarrollo del sistema Fullscreen o pantalla completa
10	8	desarrollo del Módulo de generación de factura
11	9	Creación de la base de datos para la gestión de usuarios
12	9	Validación de la base de datos
13	10	Creación del módulo de Estado de cuenta
14	11	Desarrollo de la fundación de cierre de sesión

Nota. Tareas por desarrollar en las historias de usuario. *Fuente.* Elaboración propia

Tabla 16*Tarea 1 Historia de Usuario 1*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
1	1	Diseño de interfaz de sistema de inicio de sesión	Desarrollo	26 /05/ 2023	1 /06/2023	Manuel Alejandro Ramírez Jaramillo	Establecer sistema de ingreso con usuario y contraseña

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 1 historia de usuario 1. *Fuente.* Elaboración propia

Tabla 17*Tarea 2 Historia de Usuario 1*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
2	1	Validación del sistema de inicio	prueba	2 /06/2023	2 /06/2023	Manuel Alejandro Ramírez Jaramillo	Garantizar el acceso a la aplicación

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 2 historia de usuario 1. *Fuente.* Elaboración propia

Tabla 18*Tarea 3 Historia de Usuario 2*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
3	2	Desarrollo del sistema de recuperación de contraseña	Desarrollo	3 /06/2023	10 /06/2023	Manuel Alejandro Ramírez Jaramillo	Establecer sistema de recuperación de clave por medio de correo

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 3 historia de usuario 2. *Fuente.* Elaboración propia

Tabla 19*Tarea 4 Historia de Usuario 3*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
4	3	Módulo de cambio de contraseña desde el panel de usuario	Desarrollo	11/06/2023	16/06/2023	Manuel Alejandro Ramírez Jaramillo	Desarrollo del módulo de cambio de contraseña

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 4 historia de usuario 3. *Fuente.* Elaboración propia

Tabla 20*Tarea 5 Historia de Usuario 4*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
5	4	Sistema de creación y activación del servicio desde el panel del administrador	Desarrollo	17/06/2023	20/06/2023	Manuel Alejandro Ramírez Jaramillo	Activación del servicio para que el cliente disfrute de todo el contenido de la aplicación

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 5 historia de usuario 4. *Fuente.* Elaboración propia

Tabla 21*Tarea 6 Historia de Usuario 5*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
6	5	Diseño de la interfaz del menú	Desarrollo	21/06/2023	24/06/2023	Manuel Alejandro Ramírez Jaramillo	Brindar facilidad al usuario de visualizar las diferentes opciones y manejos que ofrece la aplicación

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 6 historia de usuario 5. *Fuente.* Elaboración propia

Tabla 22*Tarea 7 Historia de Usuario 6*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
7	6	Integración de los canales para la aplicación	Desarrollo	30/06/2023	4/07/2023	Manuel Alejandro Ramírez Jaramillo	Canales de contenido educativo y de entretenimiento, pensando en la población infantil, jóvenes y adultos

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 7 historia de usuario 6. *Fuente.* Elaboración propia

Tabla 23*Tarea 8 Historia de Usuario 6*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
8	6	Interfaz de la búsqueda de canales	Desarrollo	5/07/2023	6/07/2023	Manuel Alejandro Ramírez Jaramillo	visualización de búsqueda de canales

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 8 historia de usuario 6. *Fuente.* Elaboración propia

Tabla 24*Tarea 9 Historia de Usuario 7*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
9	7	Desarrollo del sistema fullscreen o pantalla completa	Desarrollo	7/07/2023	10/07/2023	Manuel Alejandro Ramírez Jaramillo	Permitir la función de pantalla completa permitiendo ocultar el menú principal

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 9 historia de usuario 7. *Fuente.* Elaboración propia

Tabla 25*Tarea 10 Historia de Usuario 8*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
10	8	Desarrollo del Módulo generación de factura	Desarrollo	11/07/2023	14/07/2023	Manuel Alejandro Ramírez Jaramillo	Generar factura digital

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 10 historia de usuario 8. *Fuente.* Elaboración propia

Tabla 26*Tarea 11 Historia de Usuario 9*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
11	9	Creación de la base de datos para la gestión de usuarios	Desarrollo	15/07/2023	19/07/2023	Manuel Alejandro Ramírez Jaramillo	Registro de la información personal del cliente

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 11 historia de usuario 9. *Fuente.* Elaboración propia

Tabla 27*Tarea 12 Historia de Usuario 9*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
12	9	Validación de la base de datos	Desarrollo	20/07/2023	20/07/2023	Manuel Alejandro Ramírez Jaramillo	Funcionalidad de la base de datos

Nota. En esta tabla se muestra el nombre, responsable y la descripción de la tarea 12 historia de usuario 9. *Fuente.* Elaboración propia

Tabla 28*Tarea 13 Historia de Usuario 10*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
13	10	Creación del módulo de Estado de cuenta	Desarrollo	21/07/2023	28/07/2023	Manuel Alejandro Ramírez Jaramillo	Visualización de las facturas pagadas

Nota. En esta tabla se muestra el nombre, responsable y descripción de la tarea 13 historia de usuario 10. *Fuente.* Elaboración propia

Tabla 29*Tarea 14 Historia de Usuario 11*

Número Tarea	Número Historia	Nombre	Tipo de tarea	Fecha de inicio	Fecha de fin	Programador responsable	Descripción
14	11	Desarrollo de la función de cierre de sesión	Desarrollo	29/07/2023	05/08/2023	Manuel Alejandro Ramírez Jaramillo	Finalización de la sesión

Nota. En esta tabla se muestra el nombre, responsable y descripción de la tarea 14 historia de usuario 11. *Fuente.* Elaboración propia

Reunión Matinal

Siguiendo la metodología XP las reuniones deben realizarse al comenzar el día laboral, para facilitar la comunicación y mantener un buen clima laboral, al no contar con estructura física de trabajo, se emplea la modalidad de teletrabajo, exigiendo resultados de la evolución constante del proyecto para brindar la asesoría necesaria para obtener los resultados esperados como se plantea en el siguiente cronograma.

Tabla 30

Cronograma de Reuniones

Reunión matinal	Iteración	Lugar	Duración	Fecha
Reunión 1	1	Virtual a través de Teams.	1 hora	26 /05/ 2023
Reunión 2	1	Virtual a través de Teams.	1hora	3 /06/ 2023
Reunión 3	2	Virtual a través de Teams.	1hora	11 /06/2023
Reunión 4	2	Virtual a través de Teams.	1hora	17 /06/ 2023
Reunión 5	2	Virtual a través de Teams.	1hora	21/06/ 2023
Reunión 6	3	Virtual a través de Teams.	1hora	30 /06/ 2023
Reunión 7	3	Virtual a través de Teams.	1hora	7 /07/2023
Reunión 8	3	Virtual a través de Teams.	1hora	11/07/ 2023
Reunión 9	4	Virtual a través de Teams.	1hora	15/07/2023
Reunión 10	4	Virtual a través de Teams.	1hora	21 /08/2023
Reunión 11	4	Virtual a través de Teams.	1hora	29/08/2023

Nota. En esta tabla se muestra el registro de reuniones. *Fuente.* Elaboración propia

Diseño

En la elaboración y ejecución del proyecto, el diseño es modificado de acuerdo con las situaciones que se presenta en el tiempo de vida, implementando los recursos necesarios para su funcionamiento. partiendo desde una programación flexible e integral utilizando tarjetas CRC(Clase-Responsabilidad-Colaboración) para describir las clases utilizadas en la programación y que tenga suficiente capacidad para cubrir con la historia del usuario.

Entre los elementos a destacar en la metodología XP en el capítulo del diseño se encuentra la simplicidad, tarjeta CRC, metáfora del sistema, Spike Solution, el refactoring.

Simplicidad

Para el desarrollo del software se acogió a las recomendaciones realizadas por la metodología XP por lo que se implementó un diseño simple y sencillo, fácil de ejecutar, simplificando el diseño para agilizar su proceso, en un tiempo determinado para cada tarea, brindado como finalidad una buena visualización del software para el cliente con interfaces sencillas y claras de acuerdo con las necesidades descritas en las historias de usuario.

Tarjetas CRC (Clase-Responsabilidad-Colaboración)

La inclusión de esta técnica permitió la presentación y comprensión del sistema orientado en tarjetas CRC donde se describe las responsabilidades en lugar de los métodos y atributos, las clases que representan una colección del objeto y los colaboradores que representa la petición de un cliente a un servidor para cumplir la responsabilidad del cliente, fue necesario tener claro los requerimientos del proyecto, además es posible implantar sin necesidad de software.

En las tablas se muestran las tarjetas CRC, las cuales fueron de gran utilidad.

Tabla 31*Login de Ingreso*

Cliente y administrador	
Responsable	Colaboradores
Validar el ingreso a la aplicación	Capa de acceso de datos Método de validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de login de ingreso. *Fuente.* Elaboración propia

Tabla 32*Recuperación de Clave*

Cliente y administrador	
Responsable	Colaboradores
Validar recuperación de clave	Capa de acceso de datos Método de validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de recuperación de clave. *Fuente.* Elaboración propia

Tabla 33*Cambio de Contraseña*

Cliente y administrador	
Responsable	Colaboradores
Validación de cambio de contraseña	Capa de acceso de datos Método de validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de cambio de contraseña. *Fuente.* Elaboración propia

Tabla 34*Activación del Servicio*

Administrador	
Responsable	Colaboradores
Parámetros de activación del servicio	Capa de acceso de datos
Validación de activación del servicio	Método de validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de activación del servicio. *Fuente.* Elaboración propia

Tabla 35*Menú*

Cliente	
Responsable	Colaboradores
Crear la interfaz del menú	Listado de canales y botones

Nota. En esta tabla se muestra la tarjeta CRC de menú. *Fuente.* Elaboración propia

Tabla 36*Búsqueda de Canales*

Cliente	
Responsable	Colaboradores
Parámetro de búsqueda de canales	Capa de acceso de datos
Validar búsqueda de canales	Métodos Validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de búsqueda de canales. *Fuente.* Elaboración propia

Tabla 37*Pantalla Completa*

Cliente	
Responsable	Colaboradores
Validar pantalla completa	Capa de acceso de datos Métodos Validación de Datos

Nota. En esta tabla se muestra la tarjeta CRC de pantalla completa. *Fuente.* Elaboración propia

Tabla 38*Generar Factura*

Administrador	
Responsable	Colaboradores
Parámetros para generar factura	Capa de acceso de datos
Validar la generación de factura	Métodos Validación de Datos

Nota. En esta tabla se muestra la tarjeta CRC de generar factura. *Fuente.* Elaboración propia

Tabla 39*Base de Datos de los Clientes*

Administrador	
Responsable	Colaboradores
Parámetros para Base de datos de los clientes	Capa de acceso de datos
Validar Base de datos de los clientes	Métodos Validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de base de datos de los clientes. *Fuente.*

Elaboración propia

Tabla 40*Estado de Cuenta*

		Cliente	
Responsable			Colaboradores
Parámetros	Estado de cuenta		Capa de acceso de datos
	Validar Estado de cuenta		Métodos Validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de estado de cuenta. *Fuente.* Elaboración propia

Tabla 41*Cerrar Sesión*

		Cliente y administrador	
Responsable			Colaboradores
Parámetros	Cerrar sesión		Capa de acceso de datos
	Validar Cerrar sesión		Métodos Validación de datos

Nota. En esta tabla se muestra la tarjeta CRC de cerrar sesión. *Fuente.* Elaboración propia

Metáfora del Sistema

Gracias al proceso desarrollado con la metodología XP, el proyecto se plasma de una forma sencilla y fácil de comprender, facilitando su diseño de arquitectura de acuerdo con las historias usuario con métodos eficaces que brindan conceptos claros sobre las funciones a ejecutar.

Spike Solution

Dentro del diseño, las soluciones se presentaron de dos tipos funcionales y técnicos, en los técnicos se basaron en determinar la viabilidad de cada tarea e historia de usuario, mientras que los funcionales analizaba los riesgos, complejidad y funcionalidad del software brindando

una estimación más exacta de los objetivos planteados. El Spike Solution ayuda a adaptar al software a las situaciones cambiantes que se presentan, mejorando su función, sin perder la finalidad con el cual va dirigido.

Refactoring

Este se encarga de mejorar el diseño, sin cambiar su comportamiento por lo tanto su funcionalidad sigue siendo simple, sencilla y entendible para cada recodificación que se presente en el proceso. por consiguiente, la metodología XP recomienda realizarlo cada vez que sea necesario, aunque puede quitar tiempo, los resultados a futuro son satisfactorios.

Codificación

La codificación para este proyecto siguiendo la metodología XP implica un enfoque ágil y colaborativo que se centra en la satisfacción del cliente y en la entrega continua de un producto funcional y de alta calidad.

Como lenguaje de programación se utilizó PHP ya que es multiplataforma lo que permite una libertad de operación permitiendo desarrollar un sistema de software que incluye la transmisión de señales de televisión en tiempo real a través de internet, lo que garantiza la visualización de contenidos en una gran variedad de dispositivos.

A continuación, se describe como se estructura el desarrollo del aplicativo por medio de dos módulos, Backend y Frontend.

Frontend

Desde el Front se desarrolla la parte visual, tanto de contenido para el cliente, como del panel de administración.

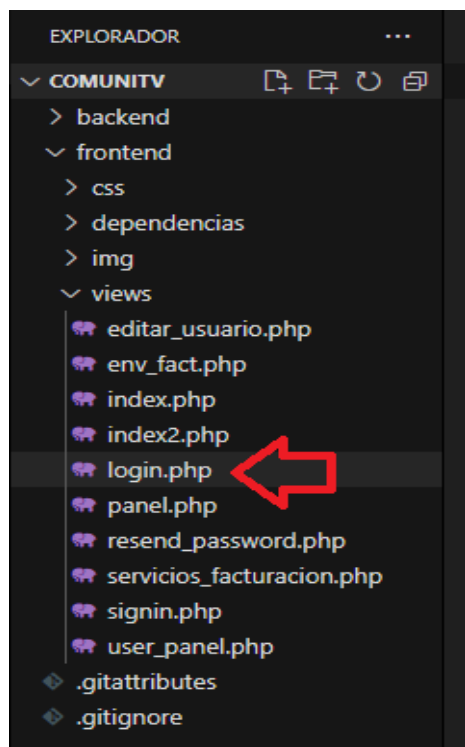
Iniciando con la sección de **Vistas** (views), tenemos aquí todos los archivos que contienen el entorno visual de cada una de las secciones del aplicativo. (Ordenadas por la línea de navegación)

Login

Corresponde al inicio de sesión tanto para cliente, como para el área administrativa, es la primera sección del aplicativo, y consta de un formulario que se comunica con el backend, enviándole por parámetros el usuario y contraseña del cliente para ser validados en la base de datos y posteriormente permitir el ingreso.

Figura 2

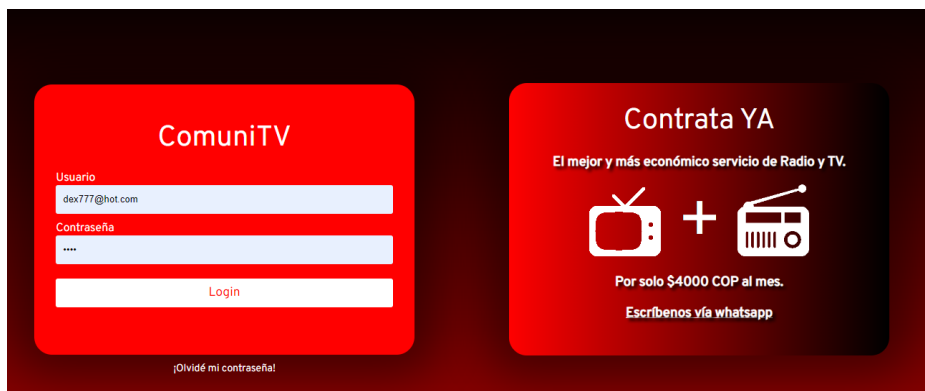
Login



Fuente. Elaboración propia

Figura 3

Visualización de la Aplicación del Login de Ingreso



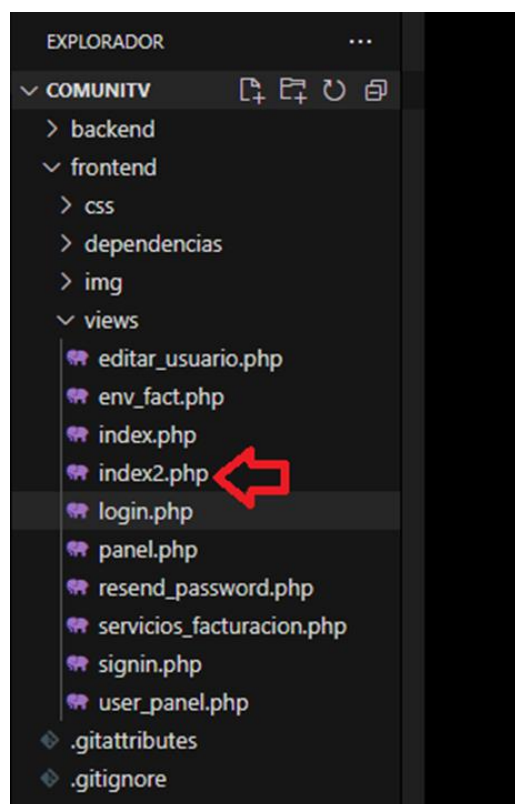
Fuente. Elaboración propia

Login Index e Index2

Luego de ser validado el usuario, por el lado del usuario cliente, se accede al área de servicios prestados, en este caso, Index contiene los servicios de IPTV y en el caso de index2 contiene los servicios de RADIO. Estas dos secciones contienen un reproductor multimedia basado en JavaScript, el cual recibe por parámetros las señales de streaming en formato mp3 para radio, o en m3u8 para IPTV. También se valida con el backend el tipo de servicio contratado ya que el usuario puede contratar solo RADIO, solo IPTV o ambos servicios

Figura 4

Index e Index2



Fuente. Elaboración propia

Figura 5

Visualización del Reproductor Multimedia



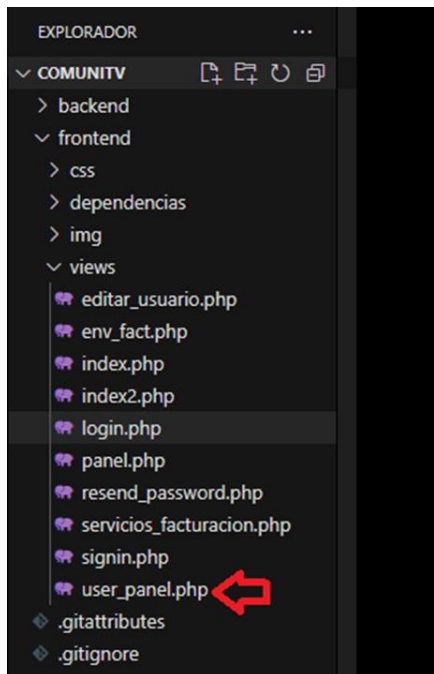
Fuente. Elaboración propia

User Panel

Es el panel de usuario y nos trae la información de facturación del usuario logueado

Figura 6

User Panel



Fuente. Elaboración propia

Figura 7

Visualización del Panel del Usuario

◀

Hola juan pablo 333 , estos son tus servicios contratados.

Cédula	Nombres	Apellidos	Servicios Contratados	Costo Servicios	Fecha Registro	Fecha de Corte	Estado
7687678	Juan pablo 333	Gonzales	IPTV + RADIO	\$4000 / Mes	2022-10-15	2023-03-02	ACTIVO

Y estas son tus Facturas pagadas.

Cédula	Código Servicio	Servicios	Valor Venta	Descuento	Impuesto	Fecha Venta
7687678	333	IPTV + RADIO	40000	0%	0%	2022-10-25 00:00:00
7687678	111	IPTV	5000	0%	0%	2022-10-25 00:00:00
7687678	222	RADIO	4000	0%	0%	2022-10-25 00:00:00
7687678	333	IPTV + RADIO	4000	0%	0%	2022-10-25 00:00:00

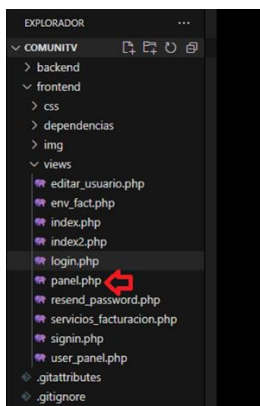
Fuente. Elaboración propia

Panel

Pasando ahora al entorno administrativo, luego de loguearnos iniciamos con el panel de administración de usuarios. Este panel nos carga una consulta específica por base de datos, acerca de los usuarios que se encuentran registrados con tres opciones en cada registro mostrado, los cuales son, “Servicios y facturación”, “editar” y “eliminar”. Adicional tenemos un botón que nos llevará a la sección de “Crear usuario”

Figura 8

Panel



Fuente. Elaboración propia

Figura 9

Panel de Gestión de Usuarios



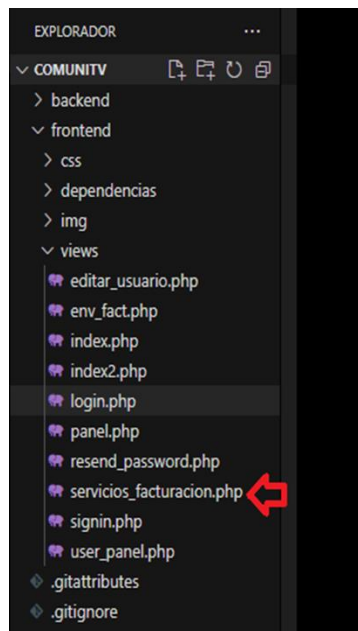
Fuente. Elaboración propia

Servicios Facturación

En esta sección le pedimos al backend que nos muestre que plan tiene registrado el usuario, sus facturas pagadas, y si su plan está vencido, la opción de renovar el plan.

Figura 10

Servicios Facturación



Fuente. Elaboración propia

Figura 11

Visualización de Servicios y Facturación



Servicios y Facturación de Juan Pablo 111 Gonzales

Cédula	Nombres	Apellidos	Servicios Contratados	Costo Servicios	Fecha Registro	Fecha de Corte	Estado	
78687687	Juan Pablo 111	Gonzales	IPTV + RADIO	\$4000 / Mes	2022-10-08	2024-01-26	ACTIVO	Activar / Renovar

Plazo

Plan

[Aplicar](#)

[Ver Facturas](#)

Cédula	Código Servicio	Servicios	Valor Venta	Descuento	Impuesto	Fecha Venta	
78687687	333	IPTV + RADIO	48000	0%	0%	2023-01-31 16:52:44	Enviar Factura

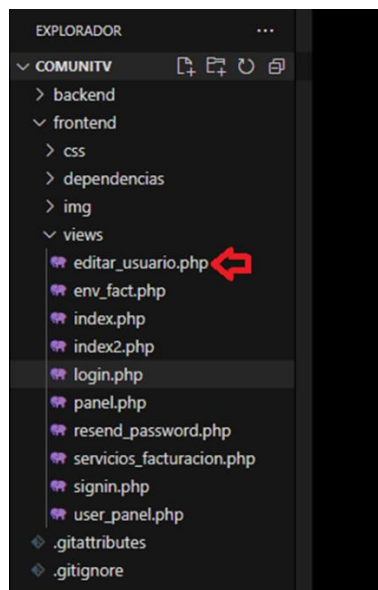
Fuente. Elaboración propia

Editar Usuario

En esta sección tenemos solo un formulario que invoca los datos del usuario a editar. Este mismo formulario que invoca datos también puede enviarlos a la base de datos dada su naturaleza de formulario.

Figura 12

Editar Usuario



Fuente. Elaboración propia

Figura 13

Editar Usuario



Editar Usuario

Nombres : Juan pablo 111

Apellidos : Gonzales

Email : dex111gmail.com

Municipio : Soacha

Departamento : Cundinamarca

Telefono : 767686876

Actualizar Cancelar

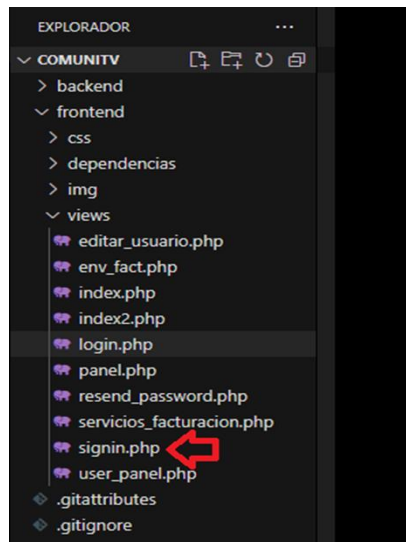
Fuente. Elaboración propia

Signin

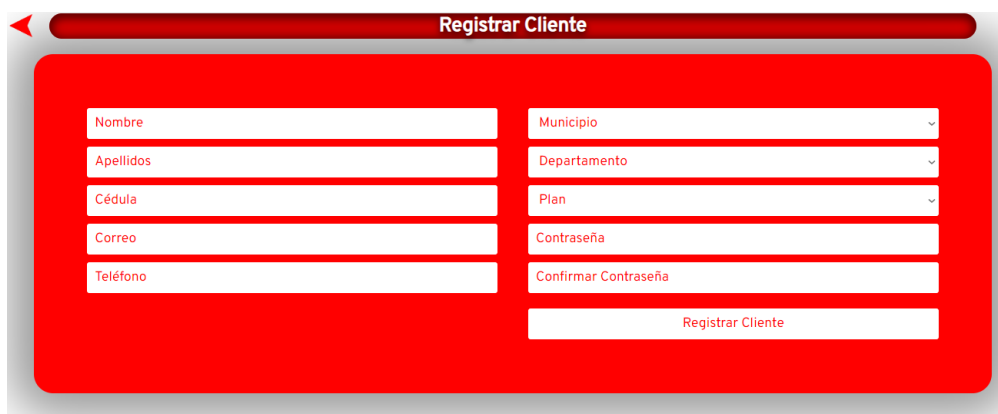
En esta sección tenemos otro formulario parecido al de editar usuario, pero con aún más campos, debido a que en este formulario creamos el perfil del usuario y de una vez se selecciona un plan de servicio de entrada.

Figura 14

Signin



Fuente. Elaboración propia

Figura 15*Visualización de Registro de Cliente*

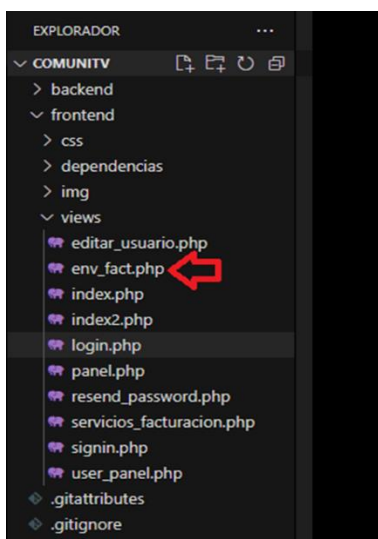
The image shows a web form titled "Registrar Cliente" with a red header. The form is set against a dark red background and contains the following fields:

Nombre	Municipio
Apellidos	Departamento
Cédula	Plan
Correo	Contraseña
Teléfono	Confirmar Contraseña
Registrar Cliente	

Fuente. Elaboración propia

Env Fact

Es un bloque de respuesta que se muestra luego de ser enviada la factura al correo del usuario. Este bloque recibe los parámetros del usuario por método GET desde el panel de administración y con dichos parámetros se consultan los datos de facturación y de contacto del usuario, para luego enviarlos al correo registrado por el usuario.

Figura 16*Env Fact*

Fuente. Elaboración propia

Figura 17

Visualización de Factura Enviada



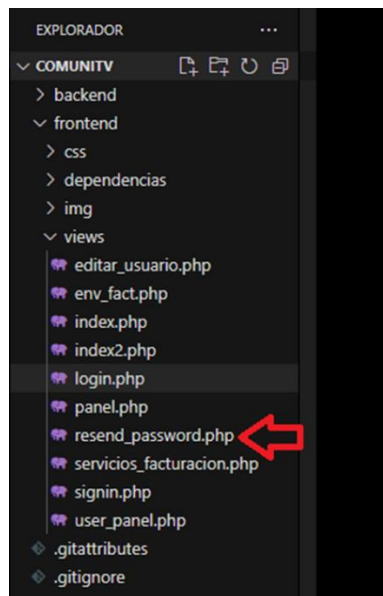
Fuente. Elaboración propia

Resend Password

Finalmente cerramos la sección de vistas con el envío de contraseña al usuario. Este servicio consta de un pequeño formulario que nos pide el email registrado, si dicho correo existe en la base de datos, enviará la contraseña registrada al correo registrado por el cliente, si dicho correo no existe, saltará el respectivo mensaje de error

Figura 18

Resend Password



Fuente. Elaboración propia

Figura 19

Formulario de Recuperación de Contraseña



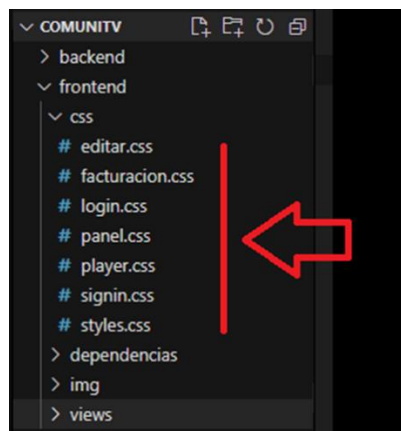
Fuente. Elaboración propia

CSS

Desde este submódulo tenemos todos los archivos de estilos para cada sección de las vistas. Desde aquí se diseña gran parte de la interfaz de usuario definiendo colores, tipos de fuentes, fondos, bordes, entre otros atributos

Figura 20

CSS



Fuente. Elaboración propia

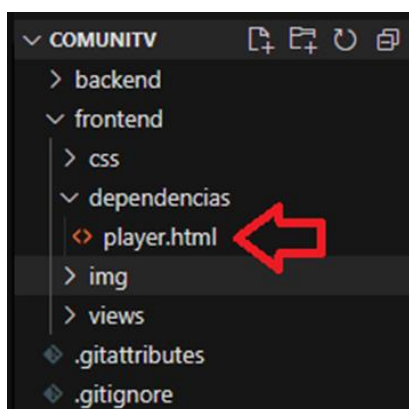
Dependencias

En este submódulo tenemos dentro el archivo HTML (player.html) que contiene el reproductor de audio y video que reproduce las señales de IPTV y RADIO. Este archivo recibe por método GET el parámetro de la señal.

Ej. ../dependencias/player.html? file=https://iptv.com/senal.m3u8

Figura 21

Submódulo Dependencias



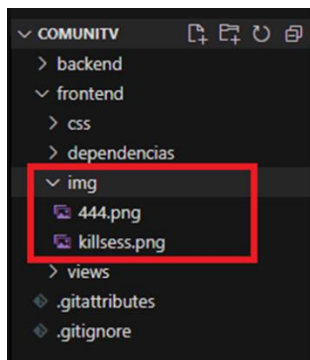
Fuente. Elaboración propia

Img

Para finalizar el módulo Frontend cerramos con el submódulo img, el cual contiene las imágenes que serán invocadas por las vistas para ser mostradas en las interfaces de usuario.

Figura 22

submódulo Img



Fuente. Elaboración propia

Backend

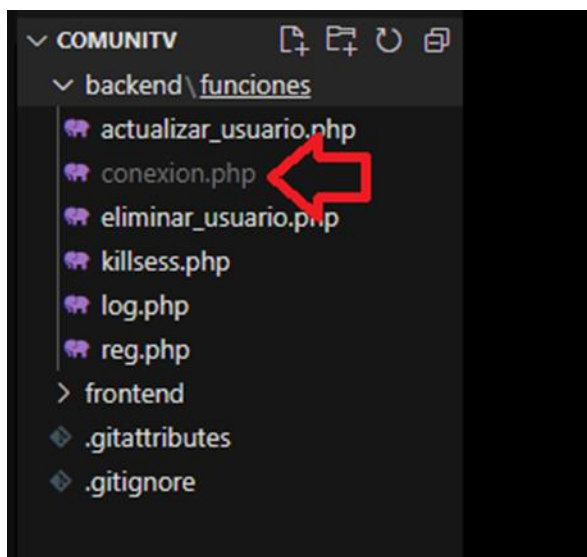
Pasando ahora al módulo Backend, tenemos un único submódulo llamado funciones, el cual contiene todos los archivos php que ejecutan las peticiones del cliente (frontend).

Conexión

Iniciando en orden de ejecución tenemos el archivo “conexion.php” con el cual conectamos toda la parte Back del aplicativo con la base de datos.

Figura 23

Conexión



Fuente. Elaboración propia

Figura 24

Codificación de la Conexión

```
conexion.php x
backend > funciones > conexion.php > ...
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "12345678";
5 $dbname = "comunitv";
6
7 // Crear conexión
8 $conn = mysqli_connect($servername, $username, $password, $dbname);
9 // verificar conexión
10 if (!$conn) {
11     die("Connection failed: " . mysqli_connect_error());
12 }
13
14 ?>
```

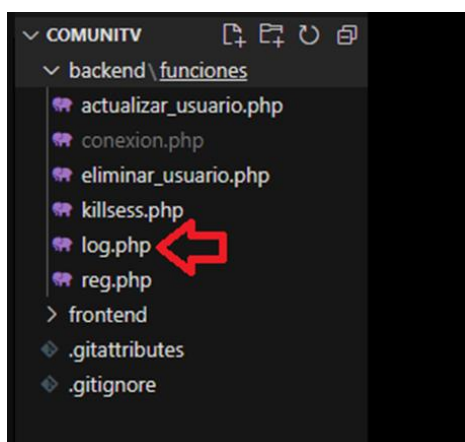
Fuente. Elaboración propia

Log

Desde este archivo recibimos los parámetros del login por medio del método POST, para posteriormente validar los datos ingresados por el usuario con la base de datos y finalmente dar una respuesta, ya sea un mensaje de error por usuario no válido, o la redirección a la vista de servicios si es un usuario válido.

Figura 25

Parámetros del Login



Fuente. Elaboración propia

Figura 26

Codificación Log



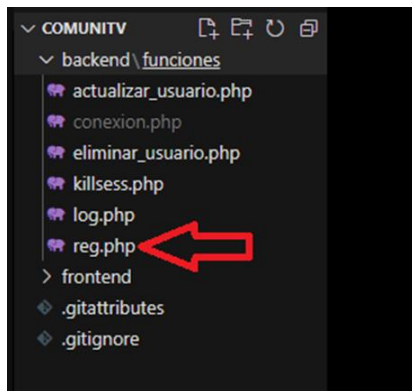
Fuente. Elaboración propia

Reg

Pasando al entorno administrativo, el archivo “reg.php” recibe por POST los datos del formulario del registro de usuarios, para luego ser validados e ingresados a la base de datos.

Figura 27

Reg



Fuente. Elaboración propia

Figura 28

Codificación Reg

```

reg.php x
backend > funciones > reg.php > ...
1  <?php
2
3  include 'conexion.php';
4
5  $nombre = $_POST["nombre"];
6  $apellidos = $_POST["apellidos"];
7  $cedula = $_POST["cedula"];
8  $email = $_POST["email"];
9  $telefono = $_POST["telefono"];
10 $municipio = $_POST["municipio"];
11 $departamento = $_POST["departamento"];
12 $codigo_servicio = $_POST["codigo_servicio"];
13
14
15 $contrasena = $_POST["contrasena"];
16 $con_contrasena = $_POST["con_contrasena"];
17
18 if($contrasena == $con_contrasena){
19     }
20 }
21 else {
22     session_start();
23     $error = "Las contraseñas NO coinciden";
24     $_SESSION['error'] = $error;
25     header("Location: ../../frontend/views/signin.php");
26     die();
27 }
28
29

```

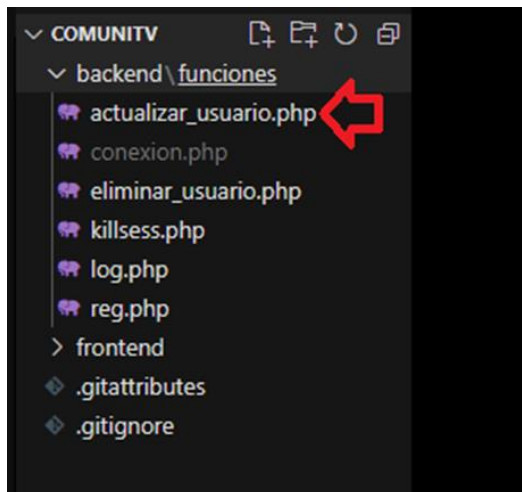
Fuente. Elaboración propia

Actualizar Usuario

Desde este archivo se valida la información enviada por el cliente desde el formulario de actualización de datos del usuario.

Figura 29

Actualizar Usuario



Fuente. Elaboración propia

Figura 30

Codificación de Actualizar Usuario

```

actualizar_usuario.php x
backend > funciones > actualizar_usuario.php > ...
1  <?php
2
3  require "conexion.php";
4
5  $id      = $_POST['id'];
6  $nombres = $_POST['nombres'];
7  $apellidos = $_POST['apellidos'];
8  $email   = $_POST['email'];
9  $municipio = $_POST['municipio'];
10 $departamento = $_POST['departamento'];
11 $telefono  = $_POST['telefono'];
12
13
14 $editar_usuario = mysqli_query($conn, "UPDATE usuarios SET nombres='$nombres', apellidos='$apellidos', email='$email', municipio='$municipio'");
15
16
17 if (mysqli_affected_rows($conn) > 0) {
18     header("Location: ../../frontend/views/panel.php");
19 }
20 else{
21     header("Location: ../../frontend/views/panel.php");
22 }
23 ?>

```

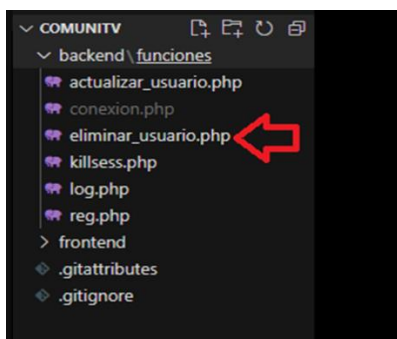
Fuente. Elaboración propia

Eliminar Usuario

Es un archivo simple que recibe por parámetros el id del usuario desde el panel de usuarios (Frontend) para posteriormente ejecutar una sentencia DELETE en la tabla de usuarios de la base de dato

Figura 31

Eliminar Usuario



Fuente. Elaboración propia

Figura 32

Codificación de la Opción Eliminar Usuario

```

eliminar_usuario.php X
backend > funciones > eliminar_usuario.php > ...
1  <?php
2
3  require "conexion.php";
4
5  session_start();
6  $email = $_SESSION['email'];
7
8  $verify_sql_2 = mysqli_query($conn, "SELECT email, rol FROM usuarios WHERE email='$email' and rol='ADMIN'");
9  if(mysqli_affected_rows($conn) > 0){
10
11  }else{
12  header("Location: ../../frontend/views/index.php");
13  die;
14  }
15
16
17
18
19
20  $id = $_GET['id'];
21
22  $eliminar_usuario = mysqli_query($conn, "DELETE FROM usuarios WHERE id='$id'");
23
24  if (mysqli_affected_rows($conn)> 0) {
25  header("Location: ../../frontend/views/panel.php");
26  }
27
28  ?>
  
```

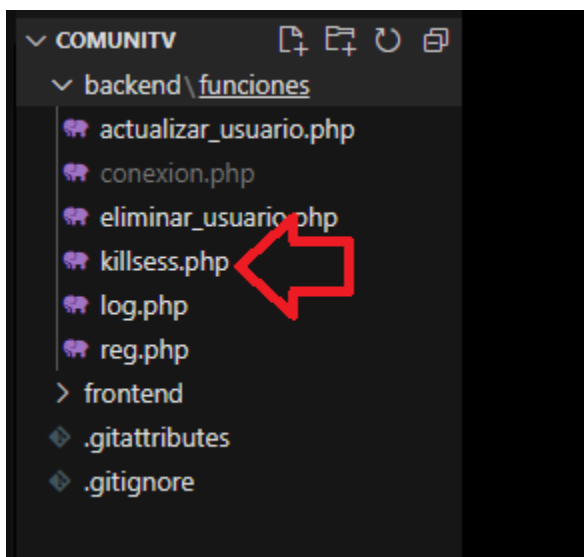
Fuente. Elaboración propia

Killsess

Finalizamos el módulo **Backend** con el archivo “killsess.php” el cual se encarga de ejecutar una simple función “session_destroy();” para finalizar la sesión de cualquier usuario, tanto cliente como administrativo.

Figura 33

Killsess



Fuente. Elaboración propia

Figura 34

Codificación de Killsess

```
killsess.php X
backend > funciones > killsess.php
1  <?php
2  session_start();
3  session_destroy();
4  header("Location: ../../frontend/views/login.php");
5  ?>
```

Fuente. Elaboración propia

Base de Datos

En cualquier sistema de software, la estructura de la base de datos es fundamental para garantizar la correcta gestión y manipulación de la información. En el caso del aplicativo que se describe en este apartado, se utilizan tres tablas que conforman dicha estructura y que son esenciales para el manejo de la información.

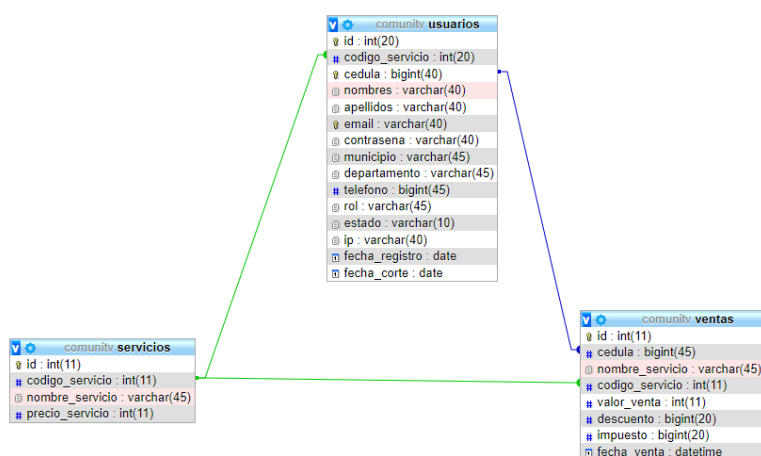
En esta sección se presentará una descripción detallada de la estructura de estas tablas y su relación entre sí. Se explicará el propósito de cada una de ellas, así como los campos que las conforman. De esta manera, se pretende brindar una visión general del manejo de la información en el aplicativo, para que los desarrolladores o usuarios del sistema puedan entender mejor la lógica del funcionamiento y la forma en que se almacenan y procesan los datos.

En definitiva, es una parte fundamental del diseño y de la arquitectura del aplicativo, y conocerla en detalle permitirá entender mejor su funcionamiento y aprovechar de manera eficiente las funcionalidades que ofrece.

Estas tablas son: **servicios, usuarios y ventas**

Figura 35

Base de Datos



Fuente. Elaboración propia

Usuarios

Desde esta tabla directamente gestionamos los datos personales del usuario, su ROL, su estado actual y el servicio contratado (Relacionando la tabla “servicios”).

Campos que Contiene. id, codigo_servicio, cedula, nombres, apellidos, email (correo electrónico), contraseña, municipio, departamento, teléfono, rol, estado, ip, fecha_registro, fecha_corte.

Figura 36

Tabla Usuario

id	codigo_servicio	cedula	nombres	apellidos	email	municipio	contrasena	departamento	telefono	rol	estado	ip	fecha_registro	fecha_corte
38	333	765765	Andres	López	dex@hot	soacha	\$2y\$10\$38Mv	cund	8676867	ADMIN	ACTIVO	127	2022-10-07	9999-10-11
44	333	786876	juan pablo	Gonzales	dex111gr	Soacha	\$2y\$10\$df564	Cundinamarca	767686876	USER	ACTIVO	127	2022-10-08	2024-01-26
46	333	765675	juan pablo	Gonzales	dex222@	Soacha	\$2y\$10\$asca:	Cundinamarca	4564645	USER	INACTIVO	127	2022-10-08	2022-02-22
57	333	768767	juan pablo	Gonzales	dex777@	Soacha	\$2y\$10\$if5gBf	Cundinamarca	876768	USER	ACTIVO	127	2022-10-15	2023-03-02

Fuente. Elaboración propia

Servicios

Desde la tabla “servicios” relacionamos el servicio contratado por el usuario en su perfil de usuario, y con su historial de facturación en la tabla “ventas”.

Campos que Contiene. id, codigo_servicio, nombre_servicio, precio_serv

Figura 37

Tabla servicios

	id	codigo_servicio	nombre_servicio	precio_servicio
<input type="checkbox"/> Editar Copiar Borrar	1	111	IPTV	3000
<input type="checkbox"/> Editar Copiar Borrar	2	222	RADIO	1000
<input type="checkbox"/> Editar Copiar Borrar	3	333	IPTV + RADIO	4000

Fuente. Elaboración propia

Ventas

Finalmente con la tabla “ventas” generamos y almacenamos el historial de facturas del cliente, relacionando el perfil del usuario desde su “cedula” y el servicio contratado desde el “codigo_servicio”

Campos que Contiene. id, cedula, nombre_servicio, codigo_servicio, valor_venta, descuento, impuesto, fecha_venta.

Figura 38

Tabla Ventas

id	cedula	nombre_servicio	codigo_servicio	valor_venta	descuento	impuesto	fecha_venta
74	7687678	IPTV + RADIO	333	40000	0	0	2022-10-25 00:00:00
75	7687678	IPTV	111	9000	0	0	2022-10-25 00:00:00
76	7687678	RADIO	222	4000	0	0	2022-10-25 00:00:00
77	7687678	IPTV + RADIO	333	4000	0	0	2022-10-25 00:00:00
78	7687678	IPTV + RADIO	333	4000	0	0	2022-10-25 09:23:17
79	7687678	IPTV + RADIO	333	8000	0	0	2022-10-25 21:23:38
80	7656757	IPTV + RADIO	333	16000	0	0	2022-10-25 23:49:39
81	7687678	IPTV + RADIO	333	48000	0	0	2022-10-26 00:26:17
82	7687678	RADIO	222	1000	0	0	2022-10-26 00:39:50
83	7687678	RADIO	222	12000	0	0	2022-10-26 15:34:00
84	7687678	IPTV + RADIO	333	4000	0	0	2022-10-26 15:40:28
85	7687678	RADIO	222	12000	0	0	2022-10-27 15:49:38
86	7687678	IPTV + RADIO	333	48000	0	0	2023-01-31 16:45:13
87	78687687	IPTV + RADIO	333	48000	0	0	2023-01-31 16:52:44
88	7687678	IPTV + RADIO	333	4000	0	0	2023-01-31 18:14:47
89	78687687	IPTV	111	18000	0	0	2023-03-01 14:43:59

Fuente. Elaboración propia

Pruebas

Las pruebas son una parte fundamental del desarrollo de software, estas ayudan a garantizar que el código se ejecute correctamente y se cumplan los requisitos del proyecto. En este caso, las unitarias son tipos de pruebas que se usan frecuentemente en el desarrollo y se centran en probar pequeñas unidades de código, como funciones o métodos, para cerciorarse de que funcionan correctamente de forma individual. Por otra parte, las pruebas funcionales se enfocan en probar el software de forma global, simulando situaciones de uso real en compañía de los usuarios del sistema para comprobar que se cumple con los requisitos de funcionalidad planteados.

Pruebas Unitarias

Estas son una técnica de programación que consiste en escribir pruebas automatizadas para verificar que cada fragmento de código, como funciones o métodos, funcionan correctamente de forma individual. En el caso de PHP, las pruebas unitarias son una práctica cada vez más frecuente en el desarrollo de aplicaciones web, ya que permiten detectar errores de manera temprana, acortar el tiempo de depuración y garantizar que los cambios en el código no introduzcan regresiones en la funcionalidad existente. En la era de la programación ágil, las pruebas unitarias se han convertido en una práctica esencial para asegurar la calidad del software. (Soler Julieth, 2022)

En este caso se usó PHPUnit, el cual es un Framework de pruebas unitarias para PHP que se ha convertido en una herramienta imprescindible para cualquier desarrollador de PHP. Con PHPUnit, podemos crear fácilmente pruebas unitarias para nuestras aplicaciones PHP y verificar que el código funciona correctamente. Es una herramienta indispensable para la implementación

de pruebas unitarias en proyectos de cualquier tamaño y para el presente, lo usaremos para validar los resultados que devuelve el backend del mismo. (Nacienceno Carlos, 2013)

A continuación, se muestran los resultados de las respectivas pruebas de cada uno de los archivos que hacen parte del Backend.

Test Actualizar Usuario

En el método "_execute", se establecen valores para las variables \$_POST que simulan los datos que el usuario enviaría al servidor cuando se actualiza un usuario. Luego, se llama a la función "actualizar_usuario" en el archivo correspondiente y se captura la salida utilizando la función ob_get_clean().

El método "actualizarUsuarioTest", se utiliza la función assertStringContainsString() de PHPUnit para comprobar si la salida de la función "actualizar_usuario" contiene la cadena "actualizado correctamente". Si la cadena está presente, el test se considera correcto, de lo contrario, se producirá un error. La figura 38 Test Actualizar Usuario, muestra en las líneas 6 a 20 el código para el método "actualizarUsuarioTest", y la línea de comandos la ejecución del test junto con su resultado satisfactorio. Se puede observar en la figura test Actualizar Usuario, en la línea de comandos que el resultado es satisfactorio.

Figura 39

test Actualizar Usuario

The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'backend', 'frontend', and 'tests'. The code editor displays the content of 'testActualizarUsuario.php'.

```

1 <?php
2
3 use PHPUnit\Framework\TestCase;
4
5
6 final class testActualizarUsuario extends TestCase
7 {
8     private function _execute()
9     {
10         $_POST['id'] = 57;
11         $_POST['nombres'] = 'Test User ' . time();
12         $_POST['apellidos'] = 'For Test';
13         $_POST['email'] = 'test@test.net';
14         $_POST['municipio'] = 'Tocancipá';
15         $_POST['departamento'] = 'Cundinamarca';
16         $_POST['telefono'] = '5555555';
17         ob_start();
18         include 'backend/funciones/actualizar_usuario.php';
19         return ob_get_clean();
20     }
21
22     /**
23      * @test
24      */

```

Below the code editor, the terminal output shows the execution of the test:

```

PS C:\WAMP\htdocs\comunitv> ./vendor/bin/phpunit ./tests/testActualizarUsuario.php
PHPUnit 9.6.5 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.075, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\WAMP\htdocs\comunitv>

```

Fuente. Elaboración propia

Test Conexión

En el método "setUp", se comprueba si la extensión MySQLi está cargada. Si la extensión no está cargada, se produce un error.

En el método "_execute", se llama a la función "conexion.php" que contiene las instrucciones para establecer una conexión con la base de datos. La salida de la función se captura utilizando la función ob_get_clean().

En el método "testDb", se utiliza la función assertStringNotContainsString() de PHPUnit para comprobar si la salida de la función "conexion.php" no contiene la cadena "Connection failed". Si la cadena no está presente, el test se considera correcto, de lo contrario, se producirá un error. Se puede observar en la figura test conexión, en la línea de comandos que el resultado es satisfactorio.

Figura 40

Test Conexión

```

1  <?php
2
3  use PHPUnit\Framework\TestCase;
4
5
6  final class testConexion extends TestCase
7  {
8      public function setUp(): void
9      {
10         if(!extension_loaded('mysqli')) {
11             $this->hasFailed('MySQLi Extension is not loaded');
12         }
13     }
14
15     public function _execute()
16     {
17         ob_start();
18         include 'backend/funciones/conexion.php';
19         return ob_get_clean();
20     }
21
22     public function testDb() {
23         // ...
24     }
25 }

```

```

PS C:\WAMP\htdocs\comunitv> ./vendor/bin/phpunit ./tests/testConexion.php
PHPUnit 9.6.5 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.050, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\WAMP\htdocs\comunitv>

```

Fuente. Elaboración propia

Test Eliminar Usuario

En el método "setUp", se crea una conexión con una base de datos MySQL y se inserta un registro en la tabla "usuarios" para utilizarlo en la prueba.

En el método "_execute", se establecen los valores de las variables \$_SESSION y \$_GET que simulan la sesión de usuario y la solicitud para eliminar el usuario con el ID "1". Luego, se llama a la función "eliminar_usuario" en el archivo correspondiente y se captura la salida utilizando la función ob_get_clean().

En el método "eliminarUsuarioTest", se utiliza la función assertStringContainsString() de PHPUnit para comprobar si la salida de la función "eliminar_usuario" contiene la cadena "eliminado correctamente". Si la cadena está presente, el test se considera correcto, de lo

contrario, se producirá un error. A continuación, se puede observar en la figura Test Eliminar Usuario que en la línea de comandos se ejecutó el test y su resultado fue acertado.

Figura 41

Test Eliminar Usuario

The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'backend', 'frontend', and 'tests'. The code editor displays the content of 'testEliminarUsuario.php'. The code includes a PHP opening tag, a use statement for PHPUnit\Framework\TestCase, and a final class 'testEliminarUsuario' extending 'TestCase'. The class has a 'setUp()' method that connects to a MySQL database and inserts a user. It also has an '_execute()' method that starts a session, sets environment variables, and includes the 'eliminar_usuario.php' file. Below the code editor, the terminal shows the command to run the test using phpunit, and the output indicates that the test passed successfully with 1 assertion.

```

1  <?php
2
3
4
5  use PHPUnit\Framework\TestCase;
6
7
8  final class testEliminarUsuario extends TestCase
9  {
10     protected function setUp(): void
11     {
12         parent::setUp();
13         $conn = mysqli_connect("localhost", "root", "12345678", "comunitv");
14         mysqli_query($conn, "INSERT INTO usuarios (id, codigo_servicio, cedula, nombres,
15
16     private function _execute()
17     {
18         session_start();
19         $_GET['env'] = "test";
20         $_SESSION['email'] = 'dex@hotmail.com';
21         $_GET['id'] = '1';
22         ob_start();
23         include 'backend/funciones/eliminar_usuario.php';
24         return ob_get_clean();
25     }
26 }

```

```

PS C:\WWW\htdocs\comunitv> ./vendor/bin/phpunit ./tests/testEliminarUsuario.php
PHPUnit 9.6.5 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.402, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\WWW\htdocs\comunitv>

```

Fuente. Elaboración propia

Test Killsess

En el método "_execute", se llama a la función "killsess" en el archivo correspondiente y se captura la salida utilizando la función ob_get_clean().

En el método "eliminarUsuarioTest", se utiliza la función assertStringContainsString() de PHPUnit para comprobar si la salida de la función "killsess" contiene la cadena "Sesión cerrada". Si la cadena está presente, el test se considera correcto, de lo contrario, se producirá un error.

Además, el atributo "@runInSeparateProcess" indica a PHPUnit que se debe ejecutar el test en un proceso separado para evitar posibles interferencias con otras pruebas. Se puede observar en la figura test Killsess, en la línea de comandos que el resultado es satisfactorio.

Figura 42

Test Killsess

```

1  <?php
2
3  use PHPUnit\Framework\TestCase;
4
5
6  final class testKillsess extends TestCase
7  {
8      private function _execute()
9      {
10         ob_start();
11         include 'backend/funciones/killsess.php';
12         return ob_get_clean();
13     }
14
15     /**
16      * @test
17      * @runInSeparateProcess
18      */
19     public function eliminarUsuarioTest() {
20         $this->assertStringContainsString("Sesión cerrada", $this->_execute(), "Correcto");
21     }
22 }

```

```

PS C:\WAMP\htdocs\comunitv> ./vendor/bin/phpunit .\tests\testKillsess.php
PHPUnit 9.6.5 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:00.340, Memory: 4.00 MB

OK (1 test, 1 assertion)
PS C:\WAMP\htdocs\comunitv>

```

Fuente. Elaboración propia

Test Log

En la clase testLog, se definen cuatro métodos de prueba, cada uno de los cuales comprueba un caso de uso diferente.

El método `_execute` se encarga de configurar los datos de entrada necesarios para cada prueba y de invocar la funcionalidad de inicio de sesión a través del archivo `log.php`. El resultado se captura en un búfer de salida y se devuelve como una cadena.

Cada método de prueba utiliza el método `assertStringContainsString` para verificar que la cadena de resultado contenga el mensaje esperado. Si el mensaje no está presente, la prueba falla

y se muestra el mensaje "Correcto" como una indicación de que se esperaba que la prueba fallara. Si el mensaje está presente, la prueba pasa. Se puede observar en la figura test Log, en la línea de comandos que el resultado es satisfactorio.

Figura 43

Test Log

The screenshot shows a code editor with a file named `testLog.php`. The code defines a class `testLog` that extends `PHPUnit\Framework\TestCase`. It includes a private method `_execute` that simulates an HTTP request and a public method `usuarioActivoTest` that uses `PHPUnit` assertions to verify the response.

```

1 <?php
2
3 use PHPUnit\Framework\TestCase;
4
5
6 final class testLog extends TestCase
7 {
8     private function _execute($userEmail, $userPass)
9     {
10         ob_start();
11         $_POST["email"] = $userEmail;
12         $_POST["contrasena"] = $userPass;
13         include 'backend/funciones/log.php';
14         return ob_get_clean();
15     }
16
17     /**
18      * @test
19      * @runInSeparateProcess
20      */
21     public function usuarioActivoTest() {
22         $this->assertStringContainsString("Sesión iniciada correctamente", $this->_execute('dex111@gmail.com', '1234'), "Correcto");
23     }
24 }

```

The terminal output shows the command `PS C:\WAMP\htdocs\comunitv> ./vendor/bin/phpunit .\tests\testLog.php` and the result `OK (4 tests, 4 assertions)`.

Fuente. Elaboración propia

Test Reg

La clase `testReg` extiende `TestCase` y define tres métodos de prueba. Cada método es una prueba independiente que comprueba un caso de uso específico. El objetivo de cada método es ejecutar el método `_execute` que realiza la simulación de una petición HTTP con diferentes parámetros de entrada y el estado de la sesión.

Cada método de prueba tiene una anotación `@test` que indica a `PHPUnit` que debe ejecutar la prueba. La anotación `@runInSeparateProcess` indica que cada prueba debe ejecutarse en un proceso separado para evitar interferencias entre ellas.

La primera prueba, `registroCorrectoTest`, verifica que se pueda registrar un usuario correctamente. Se espera que se muestre un mensaje que indica que el usuario ha sido registrado correctamente. Para esto, se llamará al método `_execute` con los parámetros correctos.

La segunda prueba, `registroConfirmacionContrasenaTest`, verifica que el sistema muestre un mensaje de error si las contraseñas no coinciden. Se espera que se muestre un mensaje que indique que las contraseñas no coinciden. Para esto, se llamará al método `_execute` con contraseñas diferentes.

La tercera prueba, `registroVacioTest`, verifica que el sistema muestre un mensaje de error si hay campos vacíos. Se espera que se muestre un mensaje que indique que hay campos vacíos. Para esto, se llamará al método `_execute` con campos vacíos.

Cada prueba utiliza el método `assertStringContainsString` para verificar si un mensaje esperado se encuentra en la respuesta del sistema. Si el mensaje esperado no se encuentra, la prueba falla. En caso contrario, la prueba se considera exitosa. Se puede observar en la figura `test Reg`, en la línea de comandos que el resultado es satisfactorio.

Figura 44

Test Reg

The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'backend', 'frontend', and 'tests'. The code editor displays the content of 'testReg.php', which is a PHPUnit test case. The test case sets up a mock request with various parameters and includes the 'reg.php' file. The terminal at the bottom shows the command to run the test, the output of PHPUnit, and the successful execution of 3 tests.

```

1  <?php
2
3  use PHPUnit\Framework\TestCase;
4
5
6  final class testReg extends TestCase
7  {
8      private function _execute($cont, $cont2, $opc = null)
9      {
10         ob_start();
11         $_POST["nombre"] = "Usuario";
12         $_POST["apellidos"] = $opc;
13         $_POST["cedula"] = time();
14         $_POST["email"] = "test".microtime()."@test.test";
15         $_POST["telefono"] = time();
16         $_POST["municipio"] = "Prueba";
17         $_POST["departamento"] = "Prueba";
18         $_POST["codigo_servicio"] = 111;
19
20         $_POST["contrasena"] = $cont;
21         $_POST["con_contrasena"] = $cont2;
22         $_SERVER['REMOTE_ADDR'] = '192.168.1.1';
23         include 'backend/funciones/reg.php';
24         return ob_get_clean();

```

```

PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE
PS C:\MAMP\htdocs\comunitv> ./vendor/bin/phpunit .\tests\testReg.php
PHPUnit 9.6.5 by Sebastian Bergmann and contributors.

...                                     3 / 3 (100%)

Time: 00:00.991, Memory: 4.00 MB

OK (3 tests, 3 assertions)
PS C:\MAMP\htdocs\comunitv>

```

Fuente. Elaboración propia

Pruebas Funcionales

Las pruebas funcionales se enfocan en los requisitos empresariales de una aplicación. Su objetivo principal es verificar el resultado de una acción sin evaluar los estados intermedios del sistema durante la ejecución de dicha acción.

En ocasiones, existe confusión entre las pruebas de integración y las pruebas funcionales, ya que ambas requieren la interacción de varios componentes. Sin embargo, se diferencian en

que una prueba de integración puede limitarse a verificar la capacidad de realizar consultas en la base de datos, mientras que una prueba funcional espera obtener un valor específico de la base de datos, según los requisitos establecidos para el producto. (STEN PITTET, 2023)

A continuación, se muestran los resultados de las pruebas ejecutadas por los respectivos actores en el proyecto.

Pruebas Funcionales para el Administrador

Prueba: Inicio de sesión.

Figura 45

Inicio de sesión

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_001			Prueba diseñada por: Manuel A. Ramírez J.			
Prioridad de prueba(Bajo, medio, alto): Media			Fecha de prueba de diseño: 5/05/23			
Nombre del módulo: Pantalla de inicio de sesión.			Prueba ejecutada por: José Luis Silva			
Título de la prueba: Probar la funcionalidad de inicio de sesión.			Fecha de ejecución de la prueba: 6/05/23			
Descripción: Verifique el inicio de sesión con un correo y contraseña válidos.						
Condiciones previas: El usuario tiene un correo y contraseña válidos en el sistema.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Navegar página de inicio	NA	El usuario debe poder iniciar sesión	La página abrió correctamente	Pasa	
2	Ingresar email de usuario	dex@hotmail.com	Se puede ingresar credencial de usuario	El email se puede digitar sin problemas	Pasa	Notifica si los datos son incorrectos
3	Ingresar contraseña	1234	Se puede ingresar credencial	La contraseña se puede digitar sin dificultad	Pasa	Notifica si los datos son incorrectos
4	Hacer clic en el botón de inicio de sesión	NA	El usuario Accede al panel de administración	Se inició la sesión, puede acceder y visualizar el panel de administración	Pasa	

Fuente. Elaboración propia

Prueba: Navegación del panel de administración.**Figura 46***Navegación del panel de administración*

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_002				Prueba diseñada por: Manuel Ramírez		
Prioridad de prueba(Bajo, medio, alto): Alta				Fecha de prueba de diseño: 5/05/23		
Nombre del módulo: Panel de administración.				Prueba ejecutada por: José Luis Silva		
Título de la prueba: Probar la funcionalidad del panel de administración.				Fecha de ejecución de la prueba: 6/05/23		
Descripción: Verifique que todos los módulos del panel estén funcionales y permitan acceder a los entornos que proporcionan.						
Condiciones previas: El usuario debe haber iniciado sesión.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Navegar por los diferentes módulos del panel de administración	NA	El usuario debe poder ingresar a cada módulo de la aplicación	Pude ingresar correctamente a todos los módulos del panel de administración	Pasa	Todos los botones y accesos están disponibles

Fuente. Elaboración propia

Prueba: Creación de usuario.

Figura 47

Creación de usuario

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_003			Prueba diseñada por: Manuel A. Ramírez J.			
Prioridad de prueba(Bajo, medio, alto): Alta			Fecha de prueba de diseño: 5/05/23			
Nombre del módulo: Crear Usuario			Prueba ejecutada por: José Luis Silva			
Título de la prueba: Probar la funcionalidad de creación de usuario.			Fecha de ejecución de la prueba: 6/05/23			
Descripción: Verifique que todos los campos del registro sean funcionales y permitan registrar correctamente a los usuarios.						
Condiciones previas: El administrador debe estar previamente logueado. También se deben tener todos los datos del usuario a registrar.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Dar clic en el botón crear usuario	NA	El administrador visualiza el formulario para crear usuarios	Se visualiza el formulario correctamente	Pasa	
2	Ingresar los datos del usuario en los campos requeridos.	Nombre: Maoly Apellidos: García García Correo: malolyjos2000@gmail.com Cédula: 1090401308 Teléfono: 3102391165 Contraseña: 1234	El formulario permite ingresar los datos para registrar un usuario	Pude diligenciar todos los datos del usuario	Pasa	
3	Clic en el botón registrar cliente	NA	Se visualiza el panel de clientes con el nuevo cliente registrado	Pude visualizar el nuevo cliente en el panel de clientes	Pasa	

Fuente. Elaboración propia

Prueba: Servicios y facturación.

Figura 48

Servicios y facturación

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_004			Prueba diseñada por: Manuel A. Ramírez J.			
Prioridad de prueba(Bajo, medio, alto): Alta			Fecha de prueba de diseño: 5/05/23			
Nombre del módulo: Servicios y facturación.			Prueba ejecutada por: José Luis Silva			
Título de la prueba: Hacer una prueba por cada opción: activar o renovar, ver factura, enviar factura			Fecha de ejecución de la prueba: 6/05/23			
Descripción: Verifique que el sistema permite cambiar y/o renovar el plan del usuario, verifique que el sistema permite mostrar y enviar al cliente sus facturas de venta.						
Condiciones previas: El administrador debe estar previamente logueado. También se debe tener previamente el plan deseado por el cliente y el nuevo plazo de renovación.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Hacer clic en "Activar/Renovar"	NA	Muestra el formulario para cambiar el plazo y el plan.	Pude visualizar el formulario para cambiar el plazo y el plan.	Pasa	
2	Elegir un nuevo plan y aumentar el plazo del plan.	NA	Los desplegables permiten cambiar de plan y/o cambiar el plazo	Pude cambiar el plan y el plazo	Pasa	
3	Clic en el botón "Aplicar"		El servicio contratado debe cambiar al nuevo asignado y plazo debe cambiar la fecha de corte correspondiente.	El servicio contratado cambió para el usuario designado	Pasa	
4	Clic en el botón "Ver facturas"	NA	Muestra el listado de facturas generadas al cliente.	Pude observar el historial de facturas generados para el usuario	Pasa	
5	Clic en el botón "Enviar factura"	NA	El sistema envía la factura al correo del cliente.	El sistema envió la factura sin problemas y el usuario la pudo ver.	Pasa	

Fuente. Elaboración propia

Prueba: Editar usuario.

Figura 49

Editar usuario

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_005			Prueba diseñada por: Manuel A. Ramírez J.			
Prioridad de prueba(Bajo, medio, alto): Alta			Fecha de prueba de diseño: 5/05/23			
Nombre del módulo: Editar usuario.			Prueba ejecutada por: José Luis Silva			
Título de la prueba: Probar la funcionalidad del formulario de edición de usuarios.			Fecha de ejecución de la prueba: 6/05/23			
Descripción: Verifique que todos los campos del formulario sean funcionales y permitan los cambios en el registro del usuario.						
Condiciones previas: El administrador debe estar previamente logueado. También se deben tener los nuevos datos del usuario a modificar.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Hacer clic en el botón "editar"	NA	Muestra el formulario con los datos del usuario, disponibles para ser editados.	Pude visualizar el formulario de edición de usuarios.	Pasa	
2	Ingresar los datos que se desean actualizar del usuario en los campos requeridos	malolyjos2011@gmail.com	El formulario valida los datos ingresados, y si son tipos de datos correctos, debe permitir la actualización del registro.	Se pudieron cambiar sin problemas los datos del usuario	Pasa	
3	Hacer clic en el botón "actualizar"	NA	Se visualiza el panel de clientes con el cliente actualizado	Pude visualizar el cliente actualizado en el panel de clientes	Pasa	

Fuente. Elaboración propia

Prueba: Eliminar usuario.**Figura 50***Eliminar usuario*

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_006			Prueba diseñada por: Manuel A. Ramírez J.			
Prioridad de prueba(Bajo, medio, alto): Alta			Fecha de prueba de diseño: 5/05/23			
Nombre del módulo: Eliminar usuario.			Prueba ejecutada por: José Luis Silva			
Título de la prueba: Probar la funcionalidad del botón de eliminación de usuarios.			Fecha de ejecución de la prueba: 6/05/23			
Descripción: Verifique que al hacer clic en el botón "eliminar usuario" elimine el usuario correspondiente.						
Condiciones previas: El administrador debe estar previamente logueado. También se deben tener los datos del usuario a eliminar.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Hacer clic en el botón "eliminar usuario"	NA	El sistema debe preguntar si desea eliminar el usuario, con opciones de "si" o "no"	Pude visualizar El mensaje antes de eliminar el usuario.	Pasa	
2	Hacer clic en la opción "si" y luego en el botón "eliminar"	NA	El usuario se elimina del sistema	Al dar clic en la opción "si" y luego en el botón "eliminar" el usuario fue borrado.	Pasa	
3	Se redirecciona al panel de usuarios.	NA	Se debe visualizar el panel de usuarios sin el registro del cliente que se ha eliminado	En el panel de usuarios ya no se muestra el usuario eliminado.	Pasa	

Fuente. Elaboración propia

Pruebas Funcionales para el Usuario

Prueba: Inicio de sesión.

Figura 51

Inicio de sesión

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_001B				Prueba diseñada por: Manuel A. Ramírez J.		
Prioridad de prueba(Bajo, medio, alto): Media				Fecha de prueba de diseño: 5/05/23		
Nombre del módulo: Pantalla de inicio de sesión.				Prueba ejecutada por: Maoly García		
Título de la prueba: Probar la funcionalidad de inicio de sesión.				Fecha de ejecución de la prueba: 6/05/23		
Descripción: Verifique el inicio de sesión con un correo y contraseña válidos.						
Condiciones previas: El usuario tiene un correo y contraseña válidos en el sistema.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Ingresar al sitio mediante la url	https://marleneaza.com/-comunitv/frontend/views/login.php	El usuario debe poder ingresar al formulario de inicio de sesión	Al entrar al enlace pude ver el formulario de inicio de sesión	Pasa	
2	Ingresar email de usuario	malolyjos2011@gmail.com	Se puede ingresar credencial de usuario	Pude digitar mi correo correctamente	Pasa	
3	Ingresar contraseña	1234	Se puede ingresar credencial	Pude ingresar la contraseña correctamente	Pasa	
4	Hacer clic en el botón de inicio de sesión	NA	El usuario Accede	Al dar clic en el botón pude entrar al sistema y visualizar el servicio.	Pasa	

Fuente. Elaboración propia

Prueba: Servicio de IPTV.

Figura 52

Servicio de IPTV

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_002B			Prueba diseñada por: Manuel A. Ramírez J.			
Prioridad de prueba(Bajo, medio, alto): Alta			Fecha de prueba de diseño: 5/05/23			
Nombre del módulo: Servicio de IPTV.			Prueba ejecutada por: Maoly García			
Título de la prueba: Probar todas las funciones del Servicio de IPTV.			Fecha de ejecución de la prueba: 6/05/23			
Descripción: Verifique todos los componentes que hacen parte del módulo de IPTV.						
Condiciones previas: El usuario debe haber iniciado sesión.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Navegar menú de canales	NA	El usuario debe poder navegar y visualizar los canales de IPTV, haciendo clic en el botón de cada canal.	Pude navegar el menú de canales e ingresar al canal WOW TV y luego cambiar al canal WTV	Pasa	
2	Hacer clic en los botones "Fullscreen" y "Minimizar". Para poder minimizar hay que tener la pantalla en Fullscreen.	NA	El contenido se pone en pantalla completa al hacer clic en "Fullscreen" y vuelve a la vista del navegador al hacer clic en "Minimizar".	Pude dar clic en los botones y alternar entre pantalla completa y ventana.	Pasa	
3	Hacer clic en el botón "Radio"	NA	El servicio cambia de IPTV a RADIO.	Pude cambiar correctamente entre el servicio de iptv y el de radio	Pasa	
4	Clic en el botón Menú	NA	El menú de canales se oculta o se muestra.	Al dar clic en el botón de menú el listado de canales se mostraba y ocultaba	Pasa	

Fuente. Elaboración propia

Prueba: Servicio de RADIO.

Figura 53

Servicio de RADIO

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_003B				Prueba diseñada por: Manuel A. Ramírez J.		
Prioridad de prueba(Bajo, medio, alto): Alta				Fecha de prueba de diseño: 5/05/23		
Nombre del módulo: Servicio de RADIO.				Prueba ejecutada por: Maoly García		
Título de la prueba: Probar todas las funciones del Servicio de RADIO.				Fecha de ejecución de la prueba: 6/05/23		
Descripción: Verifique todos los componentes que hacen parte del módulo de RADIO.						
Condiciones previas: El usuario debe tener una cuenta de IPTV + RADIO para poder hacer una prueba completa.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Navegar menú de canales de RADIO.	NA	El usuario debe poder navegar y visualizar los canales de RADIO, haciendo clic en el botón de cada canal.	Pude cambiar entre varias emisoras y escuchar su transmisión	Pasa	
2	Hacer clic en los botones "Fullscreen" y "Minimizar"	NA	El contenido se pone en pantalla completa al hacer clic en "Fullscreen" y vuelve a la vista del navegador al hacer clic en "Minimizar"	Pude dar clic en los botones y alternar entre pantalla completa y ventana	Pasa	
3	Hacer clic en el botón "IPTV"	NA	El servicio cambia de RADIO a IPTV.	Pude cambiar correctamente entre el servicio de radio y el de iptv	Pasa	
4	Clic en el botón Menú	NA	El menú de canales se oculta o se muestra.	El listado de canales se ocultaba y mostraba cuando le di clic al botón	Pasa	

Fuente. Elaboración propia

Prueba: Facturación y datos del servicio contratado.**Figura 54***Facturación y datos del servicio contratado*

Nombre del proyecto: ComuniTV						
Casos de prueba						
ID de caso de prueba: CTV_004B				Prueba diseñada por: Manuel A. Ramírez J.		
Prioridad de prueba(Bajo, medio, alto): Alta				Fecha de prueba de diseño: 5/05/23		
Nombre del módulo: Facturación.				Prueba ejecutada por: Maoly García		
Título de la prueba: Comprobar la información del Cliente.				Fecha de ejecución de la prueba: 6/05/23		
Descripción: Verifique todos los datos de facturación del cliente.						
Condiciones previas: El usuario debe tener una cuenta activa para poder acceder al contenido.						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado actual	Estado (falla o pasa)	Nota
1	Desde el módulo de IPTV o RADIO, hacer clic en "Panel".	NA	Se carga el módulo de Facturación y a continuación todas las facturas pagadas por el cliente.	Al dar clic en el botón, pude ver mi listado de facturas pagadas correctamente.	Pasa	
2	Hacer clic en la flecha de color rojo, (superior izquierda)	NA	Vuelve a IPTV o Radio	Al dar clic en la flecha, retorna al servicio de iptv.	Pasa	

Fuente. Elaboración propia

Resultados

En el desarrollo de la planificación se identificaron once (11) historias de usuarios las cuales describieron las necesidades según el tipo de usuario: administrador y cliente, como son: ingresar, recuperar clave, cambiar contraseña, cerrar sesión. Estas cuatro funcionalidades permiten brindar un servicio adecuado para el cliente, mientras que las opciones de activación del servicio, generación de factura e información del cliente, le garantizan al administrador un control de la información relacionada con los servicios, cliente y estado de cuentas.

En cuanto a las reuniones, se llevan a cabo diferentes tipos de reuniones, como la planificación de entregas y la reunión para revisar el progreso del proyecto de manera satisfactoria.

En cuestión de roles con base en la metodología XP, donde cada miembro del equipo tiene una función específica, para este caso se estableció el jefe del proyecto y el programador como una misma persona y el usuario o cliente las personas de la comunidad.

En la segunda fase de la metodología XP que es el diseño, facilitó comprender el proceso de gestión de televisión con IPTV mediante el uso de las tarjetas CRC al ser una herramienta útil permitió encontrar errores y hacer una evaluación instantánea para su posterior solución.

En la fase de codificación, se siguió el paradigma de programación estructurada teniendo en cuenta la simplicidad del proyecto, y el modelo de datos, esto permitió su desarrollo mediante el lenguaje de programación php para el backend y para el frontend JS junto con HTML y CSS3, generando una aplicación orientada a la web para la gestión de televisión con IPTV en el municipio de Soacha Cundinamarca la cual cuenta con un panel de menú, lista de canales, pantalla completa, generar y consultar el estado de la cuenta de clientes.

Finalmente, en la fase de pruebas, se ejecutaron pruebas unitarias y funcionales, en las primeras se pudieron evidenciar que no existen errores en el código, ya que se obtuvo como resultado una ejecución satisfactoria. Por otra parte en la pruebas funcionales, los usuarios validaron que efectivamente tanto el panel de administración, como el entorno de interacción del usuario cliente funcionan perfectamente sin ningún tipo de errores o fallas en la plataforma.

Además, a los usuarios les agradó su interfaz intuitiva y atractiva, la funcionalidad fue fácil de entender y de usar, permitiendo a los usuarios cambiar entre canales o estaciones de radio y ver sus facturas de pago. Finalmente, la interfaz satisfizo las expectativas de los usuarios en términos de apariencia y facilidad de uso.

Conclusión

Después de desarrollar el presente proyecto es posible concluir que:

La implementación de la metodología XP para el desarrollo de este proyecto facilitó significativamente su ejecución y desarrollo debido a que cuenta con un procedimiento flexible, que brinda una alta calidad en cada tarea realizada por iteración en las historias de usuario permitiendo un aprendizaje continuo ya que se prioriza y se descomponen en mini versiones que facilitan la creación del software basándose en satisfacer las necesidades del cliente.

Gracias a la etapa de la planeación se logró recoger toda la información necesaria para la ejecución del proyecto y desarrollo de las actividades bajo un seguimiento y una retroalimentación constante en los procesos.

La identificación de los requisitos funcionales es un paso crucial para el desarrollo exitoso del proyecto, esto fue posible mediante la fase de planificación de la metodología XP, donde las historias de usuarios dan a conocer lo que se espera de un prototipo orientado a la web para la gestión de televisión con IPTV por parte de los usuarios finales.

Con base en los requisitos funcionales previamente identificados y el paradigma de programación establecido, fue posible elaborar el diseño de la arquitectura del prototipo orientado a la web para la gestión de televisión con IPTV, garantizando así su escalabilidad y eficiencia según el modelo del negocio.

Luego de tener el diseño implementado se logró obtener el prototipo mediante su codificación, haciendo uso de los lenguajes de programación PHP y JS, garantizando la integración de los componentes y con herramientas de visualización como HTML y CSS se obtuvo una interface simple, intuitiva visualmente agradable.

Finalmente se llevan a cabo las pruebas tanto unitarias como funcionales, con las cuales se buscó verificar que las unidades de código funcionen correctamente y que el software cumpla con los requisitos y funcionalidades especificados en los casos de uso o requerimientos del aplicativo, dejando como resultado satisfactorio tanto en la ejecución del código, como los obtenidos por usuario final.

Recomendaciones

Se recomienda para el progresivo crecimiento del proyecto, la integración de nuevos módulos de contenido, como películas de tipo independiente o de reproducción no dependiente de derechos de autor.

También se pueden estudiar planteamientos acerca de materiales de tipo educativo, para posibles módulos educativos.

Referencias Bibliográficas

- Alicia Raeburn. (2022). *¿Qué es la programación extrema (XP)? [2022] • Asana.*
<https://asana.com/es/resources/extreme-programming-xp>
- Álvarez Miguel Ángel. (2022). *Programación estructurada.*
<https://desarrolloweb.com/articulos/programacion-estructurada>
- Avellaneda, J. V., Rodríguez, J. R., & López, D. A. (2014). Servicios de Televisión sobre la Plataforma de Internet (IPTV-IMS) usando Protocolo de Flujo en Tiempo Real (RTSP) y Protocolo de Transferencia de Hipertexto (HTTP). *Información Tecnológica*, 25(1), 67–76.
<https://doi.org/10.4067/S0718-07642014000100008>
- Banrepcultural. (2012). La televisión en Colombia . In
https://enciclopedia.banrepcultural.org/index.php/La_televisi%C3%B3n_en_Colombia.
- Barzanallana Rafael. (2014). *Apuntes Informática Aplicada a la Gestión Pública. Capítulo 1, Ingeniería del software, introduccion. 2011/12. Universidad de Murcia (España). Rafael Barzanallana.* <https://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>
- Bastis Consultores. (2022). *Tesis y Proyectos Aplicados.* <https://online-tesis.com/tesis-y-proyectos-aplicados/>
- Bello, E. (2021). Descubre qué es el Extreme Programming y sus características. *Thinking for Innovation.* <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/>
- Boronat Seguí, F. (2019). *El Protocolo RTP.* <https://riunet.upv.es/handle/10251/122177>
- Caizaluisa Caizaluisa, A. C., & Valle Alberca, M. B. (2011). *Diseño de la infraestructura de la cabecera de un proveedor de servicios de IPTV y la implementación de un prototipo utilizando software de código abierto.* <http://bibdigital.epn.edu.ec/handle/15000/4018>

- Colnodo. (2019). *¿Qué son las redes comunitarias?* <https://redescomunitarias.co/es/que-son-las-redes-comunitarias>
- D. Pablo Casco Castillo. (2018). *EL IMPACTO DE LA TELEVISIÓN BAJO DEMANDA EN EL PANORAMA TELEVISIVO ESPAÑOL.*
- Dane. (2022). *¿Cuántos somos?* . <https://www.dane.gov.co/index.php/estadisticas-por-tema/demografia-y-poblacion/censo-nacional-de-poblacion-y-vivenda-2018/cuantos-somos>
- Durán Dorado, D. F., & Arciniegas, J. (2013). Arquitectura para el Despliegue del servicio de Video bajo Demanda de IPTV, apoyada en Interactividad y Sistemas de Recomendaciones. *Iteckne*, 10(1), 77–89. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-17982013000100010&lng=en&nrm=iso&tlng=es
- Echeverry Tobón, L. M., & Delgado Carmona, L. E. (2007). *Caso práctico de la metodología ágil XP al desarrollo de software*. Pereira : Universidad Tecnológica de Pereira. <https://hdl.handle.net/11059/794>
- Guevara Arroba, & S. A. (2021). *Medios alternativos de publicidad como apoyo a la activación económica de microemprendimientos de calzado en la ciudad de Ambato (Bachelor's thesis, Universidad Técnica de Ambato. Facultad de Diseño y Arquitectura. Carrera de Diseño Gráfico Publicitario).*
- Llerena Andrade, A. P. (2009). *Diseño de una red con plataforma IPTV para Gama TV.* <http://repositorio.espe.edu.ec/jspui/handle/21000/377>
- López Toapanta, E. A., Pérez Rengel, & J. A. (2022). *Desarrollo de una plataforma iptv y vod basado en software libre para proveedores de tv por suscripción (Bachelor's thesis).*
- Manchón Eduardo. (2006). *Desarrollo ágil.* <https://desarrolloweb.com/articulos/desarrollo-agil.html>

Margaret Rouse. (2019). *¿Qué es Over-the-top?*

<https://www.computerweekly.com/es/definicion/Over-the-top>

Martelo, R. J., Orozco, R. Y., & Puello, P. (2015). Guía de Aplicaciones para Implementar

Televisión por Protocolo de Internet (IPTV) en las Organizaciones de Cartagena, Colombia.

Información Tecnológica, 26(4), 97–104. <https://doi.org/10.4067/S0718->

07642015000400013

Mina Gómez. (2014). *Televisión a través de redes IP (Análisis costo - beneficio de la*

implementación de IPTV). <http://repositorio.puce.edu.ec/handle/22000/6372>

Nacianceno Carlos. (2013). *¿Qué es PHPUnit? | Blog - El TallerWeb*.

<http://blog.eltallerweb.com/que-es-phpunit/>

Oliver Gil, J. S. (2018). *EL PROTOCOLO RTSP*. <https://riunet.upv.es/handle/10251/105099>

Ramírez, D. G. (2015). El modelo de televisión regional en Colombia: canales públicos bajo los

parámetros del mercado. *Signo y Pensamiento*, 34(66), 28–42.

<https://doi.org/10.11144/JAVERIANA.SYP34-66.MTRC>

Ramos Rodríguez, & Pepinosa, R. (2013). *Revisión de la Implementación del Servicio de IPTV*

sobre Redes Inalámbricas y Móviles con Calidad de Servicio (QoS) Review to the

Implementation of IPTV Service over Wireless and Mobiles Networks with Quality of

Service (QoS).

Ricardo Alonso Ferro Bolívar, & Cesar Hernández. (2011). *Los sistemas IPTV ¿una amenaza*

inminente para los actuales medios de teledifusión?

http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=s0123-921x2011000100010

- Rincón, L. A., Galicia, M. M., Prada, & J. A. R. (2020). *Metodología para la creación de un modelo de formación transmedia para la radio y televisión comunitarias del departamento del Tolima, Colombia. Comunicación y Métodos*, 2(2), 46-64.
- Santi Deliani Rahmawati, & Halimatus Saidah. (2020). *Número de suscriptores de televisión pagada en Ecuador Cae 18%*.
- Soler Julieth. (2022). *Agile Testing, una práctica ganadora para entregar Software de Calidad | Abstracta Chile*. <https://cl.abstracta.us/blog/agile-testing-buena-practica-entregar-software-calidad/>
- STEN PITTET. (2023). *Los distintos tipos de pruebas en software*.
<https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>
- Still Puebla, & Simon Antony. (2018). *Estudio y diseño de una infraestructura del protocolo UDP/IP para comunicaciones ópticas submarinas*.
<https://repositorio.unican.es/xmlui/handle/10902/14509>
- Tablado, & F. (2021). *Protocolo TCP. ¿Qué es y cómo funciona? | Ayuda Ley Protección Datos*.
<https://ayudaleyprotecciondatos.es/2021/07/29/protocolo-tcp/>
- Vinicio, E., & Meza, S. (2008). *IMPLEMENTACIÓN DE IPTV A TRAVÉS DE ENLACES DE INTERNET DE BANDA ANCHA (TELEVISIÓN SOBRE IP)*.
- Yúbal Fernández. (2022). *“IPTV: qué es, ventajas y desventajas, y qué son las listas de canales.”* <https://www.xataka.com/basics/iptv-que-ventajas-desventajas-que-listas-canales>
- Zárate Torres, D. J. (2012). *Diseño e implementación de una aplicación interactiva para educación a distancia: T-learning a través de un canal TDT universitario y un canal IPTV en la ciudad de Lima*. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/1525>