

# APLICACIÓN MÓVIL – UNICAUCA ESTÉREO



Juan Sebastian Peña Olave

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Telecomunicaciones

Grupo de Radio e InALámbricas - GRIAL

Sistemas de Comunicaciones Móviles e Inalambricos

Popayán, Cauca

2022



# APLICACIÓN MÓVIL – UNICAUCA ESTÉREO



Trabajo de Grado en Modalidad de Práctica Profesional

Juan Sebastian Peña Olave

Director: Víctor Manuel Quintero Flórez

Codirector: Javier Alexander Hurtado Guaca

Asesor Empresa: María Alejandra Medina Mosquera

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Telecomunicaciones

Grupo de Radio e InALámbricas - GRIAL

Sistemas de Comunicaciones Móviles e Inalambricos

Popayán, Cauca

2022



# TABLA DE CONTENIDO

	Página
<b>1 GENERALIDADES</b>	<b>4</b>
1.1 APLICACIONES MÓVILES . . . . .	4
1.2 TIPOS DE APLICACIONES MÓVILES . . . . .	6
1.2.1 Aplicaciones Nativas . . . . .	6
1.2.2 Aplicaciones Web Progresivas . . . . .	7
1.2.3 Aplicaciones Híbridas . . . . .	7
1.2.4 Aplicaciones Multiplataforma . . . . .	8
1.3 INTERACCIÓN HUMANO COMPUTADOR . . . . .	8
1.4 DISEÑO CENTRADO EN EL USUARIO . . . . .	11
1.4.1 Principios del Diseño Centrado en el Usuario . . . . .	12
1.4.2 Fases del Diseño Centrado en el Usuario . . . . .	13
1.5 USABILIDAD . . . . .	14
1.5.1 Conceptos de la Usabilidad . . . . .	14
<b>2 METODOLOGÍA</b>	<b>16</b>
2.1 CICLO DE VIDA DEL DESARROLLO SOFTWARE . . . . .	16
2.2 MODELO ITERATIVO E INCREMENTAL . . . . .	16
2.3 DEFINICIÓN DE REQUERIMIENTOS . . . . .	18
2.4 DEFINICIÓN DE CRITERIOS DE USABILIDAD . . . . .	20
2.4.1 Diez Heurísticas de Usabilidad de Nielsen . . . . .	20
<b>3 DISEÑO DEL SISTEMA</b>	<b>25</b>
3.1 ESTADO INICIAL DEL SISTEMA . . . . .	25
3.2 DISEÑO FUNCIONAL DEL SISTEMA . . . . .	26
3.3 DESCRIPCIÓN DE ARQUITECTURA . . . . .	27
3.3.1 Vista de Escenarios . . . . .	27
3.3.2 Vista de Lógica . . . . .	30
3.3.3 Vista de Procesos . . . . .	31
3.3.4 Vista de Desarrollo . . . . .	33
3.3.5 Vista Física . . . . .	34
3.4 DIAGRAMA DE FLUJO DE NAVEGACIÓN . . . . .	35
3.5 DISEÑO DE INTERFACES GRÁFICAS . . . . .	39
3.5.1 Vista Splash . . . . .	39
3.5.2 Menú Principal . . . . .	40

3.5.3	Reproductor de Audio . . . . .	40
3.5.4	Sección de Inicio . . . . .	41
3.5.5	Sección de Programas . . . . .	41
3.5.6	Sección de Agenda . . . . .	43
3.5.7	Sección de Favoritos . . . . .	43
3.5.8	Menú Lateral . . . . .	45
3.5.9	Sección de Parrilla de Programación . . . . .	45
3.5.10	Sección de Contacto . . . . .	46
3.5.11	Sección de Información sobre la Emisora . . . . .	46
<b>4</b>	<b>IMPLEMENTACIÓN</b>	<b>48</b>
4.1	HERRAMIENTAS DE DESARROLLO . . . . .	48
4.1.1	React Native . . . . .	48
4.1.2	Redux . . . . .	49
4.1.3	Git . . . . .	50
4.1.4	Visual Studio Code . . . . .	52
4.1.5	Drupal . . . . .	53
4.2	CONFIGURACIÓN SERVIDOR UNICAUCA ESTÉREO . . . . .	54
4.3	IMPLEMENTACIÓN APLICACIÓN MÓVIL . . . . .	57
4.3.1	Repositorio . . . . .	57
4.3.2	Instalación y Configuración del Entorno de Desarrollo . . . . .	57
4.3.3	Módulo de Reproducción de Audio . . . . .	60
4.3.4	Manejo de Estados de Carga y Error . . . . .	60
4.3.4.1	Componente de carga . . . . .	61
4.3.4.2	Componente de error . . . . .	62
4.3.5	Adaptación a Múltiples Tamaños de Pantalla . . . . .	63
<b>5</b>	<b>PRUEBAS</b>	<b>64</b>
5.1	MODELO DE USABILIDAD . . . . .	64
5.2	ESTRATEGIA DE INVESTIGACIÓN . . . . .	65
5.3	SELECCIÓN DE MÉTRICAS DE USABILIDAD . . . . .	66
5.3.1	Medida de Eficacia . . . . .	66
5.3.2	Medida de Eficiencia . . . . .	66
5.3.3	Medida de Satisfacción . . . . .	67
5.4	DESCRIPCIÓN DEL AMBIENTE DE PRUEBAS . . . . .	68
5.5	RESULTADOS . . . . .	69
5.5.1	Eficacia . . . . .	69

5.5.2	Eficiencia Basada en el Tiempo . . . . .	70
5.5.3	Eficiencia Relativa Global . . . . .	71
5.5.4	Satisfacción . . . . .	71
<b>6</b>	<b>CONCLUSIONES Y TRABAJOS FUTUROS</b>	<b>73</b>
6.1	CONCLUSIONES . . . . .	73
6.2	TRABAJOS FUTUROS . . . . .	74
	<b>Anexos</b>	<b>83</b>
<b>A</b>	<b>RESULTADOS ENCUESTA INICIAL</b>	<b>83</b>
<b>B</b>	<b>MANUAL DE USUARIO APLICACIÓN MÓVIL UNICAUCA ESTÉREO</b>	<b>88</b>
B.1	VISTA SPLASH . . . . .	88
B.2	VISTA DE INICIO . . . . .	88
B.3	MENÚ PRINCIPAL . . . . .	89
B.4	MENÚ LATERAL . . . . .	89
B.5	VISTA DE CONTACTO . . . . .	91
B.6	VISTA ACERCA DE UNICAUCA ESTÉREO . . . . .	92
B.7	VISTA DE PODCAST . . . . .	93
B.8	VISTA AGENDA . . . . .	94
B.9	VISTA FAVORITOS . . . . .	95
B.10	REPRODUCTOR DE AUDIO . . . . .	96
<b>C</b>	<b>FORMATO ENCUESTA PRUEBAS DE USABILIDAD</b>	<b>98</b>
C.1	PARAR LA REPRODUCCIÓN DE LA SEÑAL EN VIVO . . . . .	98
C.2	REPRODUCIR UN PODCAST DEL PROGRAMA LLAMADO FACTOR RH . . . . .	98
C.3	VISUALIZAR EL DETALLE DE UNA NOTICIA . . . . .	99
C.4	VISUALIZAR EL DETALLE DE UN TWEET . . . . .	99
C.5	VISUALIZAR EL DETALLE DE UN EVENTO DENTRO DE LA SECCION “AGENDA” . . . . .	99
C.6	VISUALIZAR EL DETALLE DE UN PROGRAMA DE LA SECCION “PODCAST” . . . . .	100
C.7	VISUALIZAR LA PARRILLA DE PROGRAMACIÓN DE LA EMISORA DESDE EL MENÚ LATERAL . . . . .	100

C.8	AGREGAR EL PROGRAMA FACTOR RH A FAVORITOS DESDE EL DETALLE DE UN PROGRAMA . . . . .	100
C.9	ACTIVAR NOTIFICACIONES PARA EL PROGRAMA FAVORITO AGRE- GADO ANTERIORMENTE . . . . .	101
C.10	ELIMINAR EL PROGRAMA AGREGADO ANTERIORMENTE DE LA LISTA DE FAVORITOS DESDE LA SECCIÓN “FAVORITOS” . . . . .	101
C.11	AGREGAR UN PODCAST DEL PROGRAMA FACTOR RH A FAVORITOS DESDE EL DETALLE DE UN PROGRAMA . . . . .	101
C.12	ELIMINAR EL PODCAST AGREGADO ANTERIORMENTE DE LA LISTA DE FAVORITOS . . . . .	102
C.13	VISUALIZAR LA HISTORIA E INFORMACIÓN DE LA EMISORA DESDE EL MENÚ LATERAL . . . . .	102
C.14	VISUALIZAR LA PÁGINA WEB DE LA EMISORA DESDE EL MENÚ LATERAL . . . . .	102
C.15	VISUALIZAR EL INSTAGRAM DE LA EMISORA DESDE EL MENÚ LATERAL . . . . .	103



# LISTADO DE ACRÓNIMOS

- ACM** *Association for Computing Machinery*, Asociación para Maquinaria de Computación. 9
- API** *Application Programming Interface*, Interfaz de Programación de Aplicaciones. 37, 53
- ASQ** *After Scenario Questionnaire*, Cuestionario del Escenario Posterior. 67
- CD** *Continuous Delivery*, Despliegue Continuo. 57
- CI** *Continuous Integration*, Integración Continua. 57
- CLI** *Command Line Interface*, Interfaz de Línea de Comandos. 57
- GPS** *Global Positioning System*, Sistema de Posicionamiento Global. 5, 7
- HCI** *Human Computer Interaction*, Interacción Humano Computador. 4, 9–11
- HTTP** *Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertextos. 27, 60
- IDE** *Integrated Development Environment*, Entorno de Desarrollo Integrado. 57
- RUP** *Rational Unified Process*, Proceso Unificado de Desarrollo. 27
- SCM** *Source Control Manager*, Gestor de Código Fuente. 51
- SDLC** *Software Development Life Cycle*, Ciclo de Vida del Desarrollo de Software. 16
- SEQ** *Single Ease Question*, Pregunta de Facilidad Única. 67, 68
- SIGCHI** *Special Interest Group in Computer Human Interaction*, Grupo de Interés Especial en Interacción Humano Computador. 9
- SMEQ** *Subjective Mental Effort Questionnaire*, Cuestionario de Esfuerzo Mental Subjetivo. 67

**UCD** *User Centered Design*, Diseño Centrado en el Usuario. 11–13

**UI** *User Interface*, Interfaz de Usuario. 50

**UME** *Usability Magnitude Estimation*, Estimación de Magnitud de Usabilidad. 67

## LISTADO DE TABLAS

Tabla 1	Información de participantes . . . . .	69
---------	--	----

## LISTADO DE FIGURAS

Figura 1	Capas de un ecosistema móvil. Obtenido de [9] . . . . .	5
Figura 2	Ecosistema de la Interacción Humano Computador, Obtenido de [22] . . . . .	10
Figura 3	Modelo iterativo . . . . .	17
Figura 4	Modelo incremental . . . . .	18
Figura 5	Arquitectura del estado inicial ecosistema web Unicauca Estéreo . . . . .	25
Figura 6	Ecosistema móvil en arquitectura de sistema Unicauca Estéreo . . . . .	26
Figura 7	Modelo de 4+1 vistas del Proceso Unificado de Desarrollo . . . . .	27
Figura 8	Diagrama de casos de uso menú principal . . . . .	29
Figura 9	Diagrama de casos de uso menú lateral . . . . .	30
Figura 10	Diagrama de casos de uso módulo de favoritos . . . . .	31
Figura 11	Diagrama de clases aplicación móvil Unicauca Estéreo . . . . .	32
Figura 12	Diagrama de proceso sección de inicio . . . . .	33
Figura 13	Diagrama de proceso navegador principal . . . . .	34
Figura 14	Diagrama de proceso navegador lateral . . . . .	35
Figura 15	Diagrama de componentes aplicación móvil Unicauca Estéreo . . . . .	36
Figura 16	Diagrama de despliegue aplicación móvil Unicauca Estéreo . . . . .	37
Figura 17	Diagrama de flujo de navegación aplicación móvil Unicauca Estéreo . . . . .	38
Figura 18	Vista Splash aplicación móvil Unicauca Estéreo . . . . .	39
Figura 19	Diseño de menú principal aplicación móvil Unicauca Estéreo . . . . .	40
Figura 20	Diseño de reproductor de audio. . . . .	41
Figura 21	Diseño sección de inicio. . . . .	42
Figura 22	Diseño sección de programas. . . . .	42
Figura 23	Diseño sección de agenda. . . . .	43
Figura 24	Diseño de sección de programas favoritos. . . . .	44
Figura 25	Diseño de sección de podcast favoritos. . . . .	44
Figura 26	Diseño del menú lateral . . . . .	45
Figura 27	Diseño de sección de parrilla de programación. . . . .	46
Figura 28	Diseño de sección de contacto con la emisora. . . . .	47
Figura 29	Diseño de sección de historia e información. . . . .	47
Figura 30	Funcionamiento de compilador en React Native . . . . .	49
Figura 31	Flujo de Redux . . . . .	51

Figura 32	Flujo de de ramificación de Git . . . . .	52
Figura 33	Arquitectura de construcción modular de Drupal . . . . .	53
Figura 34	Portal de administración del portal web Unicauca Estéreo . . . . .	54
Figura 35	Panel de administración del módulo Services . . . . .	55
Figura 36	Lista de recursos de la ruta de servidor API REST . . . . .	56
Figura 37	Consola de administración de Gitlab . . . . .	58
Figura 38	Estructura de archivos y carpetas del proyecto en React Native . . . . .	59
Figura 39	Vista con componente de carga. . . . .	61
Figura 40	Reproductor de audio con componente de carga. . . . .	62
Figura 41	Vista con componente de error. . . . .	62
Figura 42	Modelo de usabilidad ISO 9241-11 . . . . .	64
Figura 43	Tasa de finalización por tarea . . . . .	70
Figura 44	Eficiencia basada en el tiempo por tarea . . . . .	71
Figura 45	Eficiencia relativa global por tarea . . . . .	72
Figura 46	Puntuación media de satisfacción . . . . .	72
Figura 47	Profesión o cargo en la Universidad del Cauca . . . . .	83
Figura 48	Género . . . . .	84
Figura 49	Uso de teléfono inteligente . . . . .	84
Figura 50	Frecuencia de uso de un teléfono inteligente . . . . .	85
Figura 51	Género . . . . .	85
Figura 52	Consumo de radio . . . . .	86
Figura 53	Consumo de programas en formato podcast . . . . .	86
Figura 54	Medios usados para escuchar radio . . . . .	87
Figura 55	Consumo de noticias . . . . .	87
Figura 56	Vista splash . . . . .	88
Figura 57	Inicio de la aplicación móvil Unicauca Estéreo . . . . .	89
Figura 58	Menú de navegación principal . . . . .	89
Figura 59	Menú de navegación lateral . . . . .	90
Figura 60	Parrilla de programación . . . . .	90
Figura 61	Vista de detalle de un programa radial . . . . .	91
Figura 62	Vista de contacto con la emisora . . . . .	92
Figura 63	Vista de acerca de la emisora . . . . .	93
Figura 64	Vista de programas . . . . .	94
Figura 65	Vista de agenda universitaria . . . . .	94
Figura 66	Vista de detalle de un evento . . . . .	95
Figura 67	Vista de programas favoritos . . . . .	95
Figura 68	Vista de podcast favoritos . . . . .	96

Figura 69	Reproductor principal de la aplicación móvil . . . . .	96
-----------	--	----

## INTRODUCCIÓN

En los últimos años se ha dado una evolución importante en las tecnologías móviles, los dispositivos y aplicaciones móviles han cambiado el mundo de las telecomunicaciones y la vida de las personas. Cada vez más personas utilizan los dispositivos y aplicaciones móviles con el fin de intercambiar mensajes, navegar en internet, realizar llamadas telefónicas, participar en redes sociales, ver vídeos, escuchar audios y tomar fotos, entre otras actividades, las cuales se utilizan cada vez con mayor frecuencia [1].

Debido a la masificación del uso de las aplicaciones móviles por parte de los usuarios, se hace necesario mejorar su desempeño y lograr con ello una mayor eficiencia de las tareas para las cuales fueron creadas, por medio de mejoras en aspectos relacionados con la experiencia del usuario en su uso y su facilidad en la resolución de tareas. Por lo anterior, es importante estudiar todos los aspectos relacionados con la interacción del usuario con las aplicaciones móviles o un software en general. De esta manera, nace el concepto de diseño centrado en el usuario, el cual es un enfoque que analiza la forma en la que un programa software es usado, con el fin de establecer parámetros de uso dentro de un contexto específico y que sean considerados en el proceso de desarrollo software [2]. Aplicar metodologías como la descrita anteriormente resulta en un aumento de la calidad de la aplicación, como por ejemplo, la usabilidad, la cual es una característica de calidad de la experiencia o facilidad con la que un usuario puede utilizar una interfaz gráfica de usuario, una página web o una aplicación móvil [3].

Unicauca Estéreo es un medio de comunicación institucional de carácter universitario y de interés público, que busca compartir contenido de carácter educativo y cultural a la comunidad que se encuentre en su zona de cobertura, además de esto, promueve los valores humanos, la formación ciudadana y el debate público. Unicauca Estéreo está presente en más de 10 municipios del departamento del Cauca y cuenta con un promedio de 54 programas en emisión [4], [5].

Actualmente, Unicauca Estéreo hace uso de dos canales de difusión que se usan para transmitir su parrilla de programación a la comunidad universitaria y al público en general: la frecuencia 104.1 MHz en Frecuencia Modulada (FM) y a través de su página web. Esta última, además de ofrecer la transmisión en vivo de la programación e información relevante sobre su historia, programas activos y normatividad, cuenta con una biblioteca donde se almacenan los *podcast* de los diferentes programas que emite.

El objetivo de este trabajo de grado es el diseño e implementación de una aplicación móvil para la emisora institucional Unicauca Estéreo, para complementar los canales de difusión existentes con una plataforma que se profile como un nuevo canal de comunicación, que permita a los usuarios de Unicauca Estéreo acceder de manera más efectiva

a los contenidos brindados, llegar a un número mayor de personas y lograr una mayor interacción con los radioescuchas.

El desarrollo de la aplicación móvil aplica el diseño centrado en el usuario, para obtener una aplicación que cumpla con los parámetros de usabilidad, los cuales se establecen como referencia al inicio del trabajo de grado.

El presente documento se encuentra organizado de la siguiente manera: en el Capítulo 1 se presenta el marco teórico el cual se compone de la definición de los conceptos de usabilidad a nivel general y en el contexto de las aplicaciones móviles, de la interacción humano computador, además de algunos conceptos relacionados con el desarrollo de aplicaciones móviles; en el Capítulo 2, se presenta la definición de los requerimientos del sistema y los criterios de usabilidad; en el Capítulo 3, se presenta el proceso de diseño e implementación de la aplicación; en el Capítulo 4, se presentan los resultados y su análisis; y por último, en el Capítulo 5, las conclusiones y trabajos futuros.

## PLANTEAMIENTO DEL PROBLEMA

Unicauca Estéreo es un medio de comunicación institucional perteneciente a la Universidad del Cauca que funciona como emisora de interés público cuyo objetivo es elevar el nivel educativo y cultural de sus oyentes, difundir los valores humanos, la formación ciudadana y promover el debate público. Para lograr esto, Unicauca Estéreo requiere de todos los medios posibles para acceder a un porcentaje mayor de audiencia por lo que surge la necesidad de contar con medios o herramientas de comunicación que vayan acordes a las tendencias de los tiempos actuales, sobre todo en materia de tecnología.

Si bien Unicauca estéreo cuenta actualmente con dos medios de difusión con los cuales proporciona su contenido a su audiencia (104.1 MHz FM y su página web), los tiempos modernos y la tendencia actual donde una gran parte de la población posee por lo menos un dispositivo móvil ya sea un teléfono inteligente o una tableta sugieren que contar con una aplicación móvil supone una oportunidad para ampliar su radio de cobertura y proveer mayor comodidad y facilidad a sus radioescuchas. Complementario a lo anterior, una consecuencia directa de la gran masificación del uso de las aplicaciones móviles es una exigencia de calidad en términos de experiencia de usuario, debido a la alta competencia de este mercado.

Existen numerosas estaciones de radio institucionales que a través de una aplicación móvil comúnmente llamada *app* transmiten sus contenidos ya sea en vivo (mediante un servicio de streaming) o mediante programación grabada previamente (*podcast*). En Colombia,

algunos ejemplos son: Radio UTCH de la Universidad Tecnológica del Chocó, Univalle Estéreo, UNRadio de la Universidad Nacional y Javemóvil de la Universidad Javeriana. Sin embargo, en el ámbito interno de la Universidad del Cauca no existe un proyecto que documente de manera detallada el proceso de desarrollo de una aplicación móvil para una emisora institucional y que además implemente un proceso de diseño centrado en el usuario que sustente su usabilidad.

Dicho lo anterior, en el presente proyecto se propone el diseño e implementación de una aplicación móvil para la emisora institucional Unicauca Estéreo donde se busca complementar los canales de difusión existentes con una plataforma que se profile como un nuevo canal de comunicación que permita a los usuarios de Unicauca Estéreo acceder de manera más efectiva a los contenidos brindados y en consecuencia llegar a un número mayor de oyentes. Durante el proceso de desarrollo del presente proyecto se aplica el diseño centrado en el usuario para obtener así una aplicación que cumpla con los parámetros de usabilidad que son establecidos como referencia en su etapa inicial.



# 1 GENERALIDADES

Este trabajo de grado esta relacionado con el desarrollo de una aplicación móvil para la emisora universitaria Unicauca Estéreo, proceso que aplicará criterios de usabilidad. En el proceso de desarrollo de sistemas de software, es de vital importancia plantear una metodología, la cual permite definir una hoja de ruta de los pasos y procesos a seguir en cada una de las etapas del proyecto, permitiendo a los involucrados tener un flujo de trabajo más adecuado y tener una trazabilidad del avance del proyecto. Además de lo anterior, teniendo en cuenta que cada una de las etapas del proceso de desarrollo y aún más el diseño de la aplicación deben cumplir con parámetros específicos de usabilidad planteados al inicio del trabajo de grado, es necesario definir los conceptos que engloban a la usabilidad como enfoque de diseño de sistemas de software.

A continuación se define el concepto de Interacción Humano Computador (HCI, *Human Computer Interaction*), el cual aporta una base teórica sobre la cual se fundamentan diferentes enfoques y metodologías de desarrollo, entre los cuales se encuentra la usabilidad y el diseño centrado en el usuario, los cuales brindarán una serie de parámetros que van a ser empleados en el desarrollo la aplicación para dispositivos móviles.

## 1.1 APLICACIONES MÓVILES

Los dispositivos móviles, como herramientas de comunicación y entretenimiento, juegan un papel muy importante en la vida diaria de los seres humanos, impactando la sociedad a nivel global en actividades y tareas que pueden ser realizadas desde un unico dispositivo. Lo anterior es posible también, gracias a las aplicaciones móviles. Las *apps* han estado presentes en los dispositivos móviles desde hace varios años, por ejemplo, en los sistemas operativos de Nokia y de Blackberry antes de la creación de las tiendas de aplicaciones [6], cuando los teléfonos móviles eran dispositivos con pantallas reducidas y carecían de tecnologías táctiles en contraposición a los *smartphones* usados hoy en día. En esencia, una aplicación móvil es un programa software que funciona de manera autónoma y que ha sido diseñado y desarrollado para ser ejecutado en un dispositivo móvil [7]. Para entender de manera más fácil el concepto, se puede decir que las *apps* son para los móviles lo que los programas son para los computadores de escritorio.

Actualmente, se pueden encontrar *apps* de todo tipo, desde juegos y entretenimiento, pasando por el comercio electrónico y compras en línea, hasta llegar a aplicaciones para la educación y el sector empresarial, sin embargo, en los primeros teléfonos móviles estaban enfocadas en mejorar la productividad personal (alarmas, calendarios, calculadoras, correo electrónico, etc.). La llegada del iPhone al mercado en junio de 2007 cambió drásticamente

la forma en la cual concebimos a los dispositivos móviles y ello generó nuevos modelos de negocio y ecosistemas que hicieron de las aplicaciones algo más rentable y atractivo tanto para usuarios como para desarrolladores [8].

Gran parte del éxito que han alcanzado las aplicaciones móviles en los últimos años, se debe a todo un ecosistema de tecnologías tanto en hardware como en redes de conexión que han ido evolucionando en conjunto con los dispositivos móviles, y que han servido como soporte para brindar mejores funcionalidades a sus usuarios. En sus inicios los dispositivos móviles contaban con características limitadas, mientras que en la actualidad se pueden encontrar dispositivos con mayor velocidad de procesamiento y comunicación, memorias más rápidas y con mayor capacidad, conexión inalámbrica de alta velocidad con WiFi y redes celulares, mejoras en sensores de los dispositivos como cámaras, receptores GPS, acelerómetro, giróscopo, etc [7]. Todas estas partes han hecho del ecosistema móvil algo mucho más complejo, lo cual se puede representar como un conjunto de capas que dependen entre sí para crear una experiencia punto a punto, tal como se puede apreciar en la Figura 1. Dentro de este ecosistema, dependiendo del producto o servicio que se analiza y de la arquitectura que este implemente, se pueden encontrar capas que presentan algún cambio o que son removidas completamente.

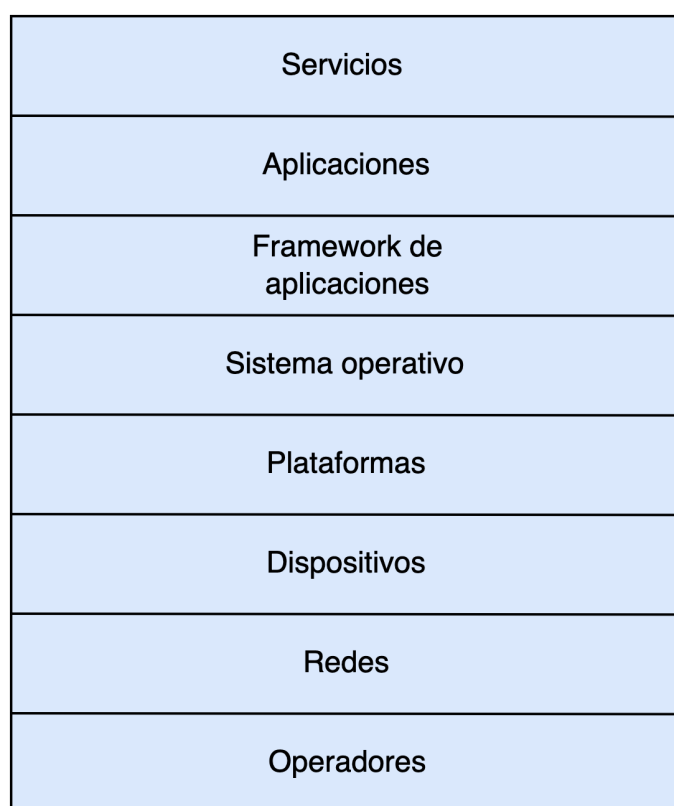


Figura 1: Capas de un ecosistema móvil. Obtenido de [9]

De este sistema de capas más comunes que representa el ecosistema de un servicio o producto móvil, se puede destacar que las primeras tres capas correspondientes a los *frameworks*, las aplicaciones y los servicios, representan los factores principales que tienen relación directa con las aplicaciones móviles en sí, donde los primeros se refieren todo el conjunto de herramientas y lenguajes de programación que son empleados para el desarrollo de la aplicación móvil, las segundas representando a los archivos que son instalados en los dispositivos con los cuales interactúan los usuarios y los últimos los servicios que brinda una aplicación móvil que suplen una necesidad o cumplen con una función específica para el usuario final.

## 1.2 TIPOS DE APLICACIONES MÓVILES

Teniendo en cuenta las consideraciones tecnológicas que envuelven el proceso de desarrollo existen cuatro tipos principales de aplicaciones móviles: aplicaciones nativas, aplicaciones web progresivas, aplicaciones híbridas y un nuevo enfoque denominado aplicaciones multiplataforma. Adicionalmente, es importante aclarar que para el presente documento sólo se consideran los sistemas operativos de iOS y Android, dejando de lado otras opciones dado que abarcan una sección muy pequeña del mercado de dispositivos móviles, e.g., Windows Phone OS [10] o son muy recientes y no cuentan con una base sólida en su ecosistema de desarrollo, e.g., Harmony OS [11]).

### 1.2.1 Aplicaciones Nativas

Las aplicaciones nativas son aquellas diseñadas y desarrolladas de manera exclusiva para una plataforma en concreto, por ejemplo, una aplicación es nativa para iOS, si ha sido creada solamente para la plataforma de Apple. Las aplicaciones creadas de forma exclusiva para un sistema operativo pueden tener una excelente integración con el equipo móvil y ofrecen al usuario una muy buena experiencia. Estas aplicaciones deben ser descargadas desde una tienda de aplicaciones del proveedor del sistema operativo [12]. Actualmente, la mayoría de aplicaciones nativas han sido diseñadas e implementadas para los sistemas operativos iOS y Android, y en una menor medida Windows Phone. Las aplicaciones nativas son desarrolladas en un lenguaje de programación específico acorde a la plataforma destino, siendo Swift y Objective C para iOS, Java y Kotlin para Android, y C# para Windows Phone.

Las aplicaciones móviles nativas permiten aprovechar totalmente las funcionalidades de los dispositivos móviles, ofreciendo entre otras las siguientes ventajas [13]:

- Alto desempeño y eficiencia.

- Acceso total a los recursos del sistema operativo (cámara, álbum de fotos, GPS, etc.).
- Excelente experiencia de usuario.
- Acceso a contenidos sin la necesidad de estar conectado a internet.

### 1.2.2 Aplicaciones Web Progresivas

Las aplicaciones web progresivas son aplicaciones de software que han sido creadas usando tecnologías web como lo son HTML5 CSS3 y JavaScript. De manera general, este tipo de aplicaciones son páginas web que han sido adaptadas para que puedan ser accedidas desde dispositivos móviles aplicando un concepto de programación web conocido como *Responsive Design*, con el fin de encajar correctamente en un tamaño de pantalla reducido [14]. Al ser páginas web, las aplicaciones web progresivas sólo pueden ser usadas a través de una aplicación de navegación como Google Chrome, Mozilla FireFox, Safari, etc.

Dentro de las ventajas que conlleva usar aplicaciones web progresivas se pueden destacar las siguientes [13]:

- No es necesario instalar ninguna aplicación adicional.
- Pueden ser accedidas desde cualquier dispositivo móvil: teléfonos inteligentes o tabletas).
- Las actualizaciones se realizan de manera instantánea sin depender de aprobaciones de una tienda de aplicaciones.
- Se puede aprovechar la información previa de la aplicación web.

### 1.2.3 Aplicaciones Híbridas

Estas aplicaciones son una variante muy similar a las aplicaciones web progresivas, ya que al igual que estas parten de una aplicación desarrollada con tecnologías web (HTML, CSS y JavaScript), sin embargo, la diferencia más importante es que se empaquetan dentro de un navegador web generando una aplicación que se puede instalar en Android o en iOS. Estas aplicaciones permiten que sean transformadas de aplicaciones web a aplicaciones móviles de manera rápida.

Las aplicaciones híbridas llegaron por primera vez en 2012 [15], para ese momento, debido a que las funcionalidades y la interfaz de usuario eran desarrolladas usando tecnologías destinadas a web, no era posible acceder a características propias del dispositivo móvil (nativas). Si bien, con el tiempo este tipo de aplicaciones han evolucionado hasta

el punto de hacer uso de recursos y funcionalidades asociadas a la cámara, el giroscopio, el acelerómetro, etc., usando un concepto propio llamado *Plugins*. Este tipo de aplicaciones se consideran comúnmente inferiores en términos de experiencia de usuario y desempeño en comparación a sus contrapartes nativas [16].

#### 1.2.4 Aplicaciones Multiplataforma

De manera general, el concepto de aplicaciones móviles multiplataforma se refiere al desarrollo de una aplicación que puede ser ejecutada en iOS o en Android, para lo cual emplean tecnologías que son independientes de la plataforma para la cual está destinada o de su sistema operativo y teniendo una sola base de código [17].

Existen herramientas multiplataforma que permiten desarrollar aplicaciones móviles utilizando tecnologías previamente orientadas a web como JavaScript. Estas aplicaciones transforman las instrucciones de JavaScript a instrucciones nativas del sistema operativo sobre el cual se esté ejecutando. La transformación ocurre a través de diferentes procesos como *cross-compilation*<sup>1</sup> o el *native scripting*<sup>2</sup>. En el caso en que este proceso sea muy lento para una parte específica de una aplicación, es posible desarrollar en código nativo las instrucciones que sean necesarias y acoplarlas al resto del código.

El uso de aplicaciones multiplataforma ofrecen diferentes ventajas entre las cuales se destacan las siguientes:

- Se puede escribir el código de una aplicación una sola vez.
- Se minimizan los costos.
- Se reduce el tiempo de desarrollo.
- El lanzamiento al mercado es más rápido.

### 1.3 INTERACCIÓN HUMANO COMPUTADOR

El computador es una herramienta que fue creada con el fin de automatizar diferentes tareas que anteriormente eran realizadas por las personas de manera manual. Los computadores han evolucionado a un ritmo acelerado desde su creación en 1938, evolución impulsada por la necesidad del ser humano de contar con una herramienta más útil para satisfacer sus necesidades. De esta manera también fue creada una herramienta de carácter complejo y por lo tanto en sus inicios debía ser manipulada por personas capacitadas

---

<sup>1</sup>*cross-compilation* es el proceso de transformar un código creado en un lenguaje de programación en otro que pueda ser ejecutado en una plataforma distinta [18]

<sup>2</sup>*native scripting* es un proceso en el cual el código escrito en un lenguaje de programación diferente se interpreta en tiempo de ejecución para ser convertido a código nativo, ya sea para iOS o Android [17].

en el tema o que entendieran su funcionamiento, y es por lo anterior que nace la necesidad de estudiar la interacción entre los sistemas de computo y las personas que los operan.

No ha sido fácil definir el concepto de HCI, existe multiplicidad de definiciones al respecto, sin embargo, debido a la necesidad de caracterizar este campo de investigación en el ámbito del presente trabajo de grado, se adopta la definición de la HCI como la disciplina que se centra en el estudio de las personas, los sistemas computacionales y la forma en la cual estos interactúan entre sí [19].

La HCI fue creada con diferentes objetivos, pero para efectos del presente documento se destacan dos objetivos principales: el primero es aplicar los conocimientos actuales sobre las capacidades y limitaciones tanto de los humanos como de los computadoras o máquinas en general, con el fin de crear nuevas aplicaciones para su uso; el segundo objetivo de la HCI es el de mejorar la calidad de la interacción entre las personas y los computadores [20]. Debido a lo anterior, la HCI es una disciplina que se debe tener en cuenta durante el desarrollo de cualquier sistema computacional que vaya a ser manipulado por un usuario, ya que un análisis del comportamiento entre las personas y los computadores puede brindar herramientas importantes para la definición tanto de los parámetros de diseño como de los criterios de evaluación, los cuales marcan la pauta del proceso de desarrollo de las funcionalidades, prestaciones y limitaciones de un sistema de software.

La HCI estudia los sistemas donde una o más personas interactúan con uno o más computadores. Un sistema de este tipo se encuentra compuesto principalmente por un usuario o persona, un computador o máquina en general, y la interacción que se genera cuando el usuario hace uso del computador para realizar una tarea o cumplir un objetivo en específico; y es en este intercambio de información donde se pueden observar la mayoría de los fenómenos que se presentan entorno a la HCI. Sin embargo, entorno a estos tres componentes principales se encuentra todo un ecosistema de conceptos que describen de una manera más detallada las variables presentes cuando una persona interactúa con un computador. En la Figura 2, el Grupo de Interés Especial en Interacción Humano Computador (SIGCHI, *Special Interest Group in Computer Human Interaction*), el cual es un grupo perteneciente a la Asociación de Maquinas Computacionales<sup>3</sup> (ACM, *Association for Computing Machinery*) describe estos conceptos [22], en la cual se puede observar a una persona que tiene tres características principales que son objeto de estudio por parte de la HCI: su capacidad de procesamiento de la información; las características de su lenguaje; y la física (ergonomía) en a interacción con un computador, el cual a su vez tiene sus

---

<sup>3</sup>La ACM es una organización a nivel mundial que reúne a profesionales, estudiantes, investigadores y a cada persona interesada en ciencias de la computación. La ACM es reconocida como la asociación número uno de los profesionales en ciencias de la computación [21]

propias características relacionadas a aspectos tecnológicos (tanto del computador como de su interfaz), considerando los dispositivos de entrada y salida como el medio para interactuar con la ergonomía del usuario.

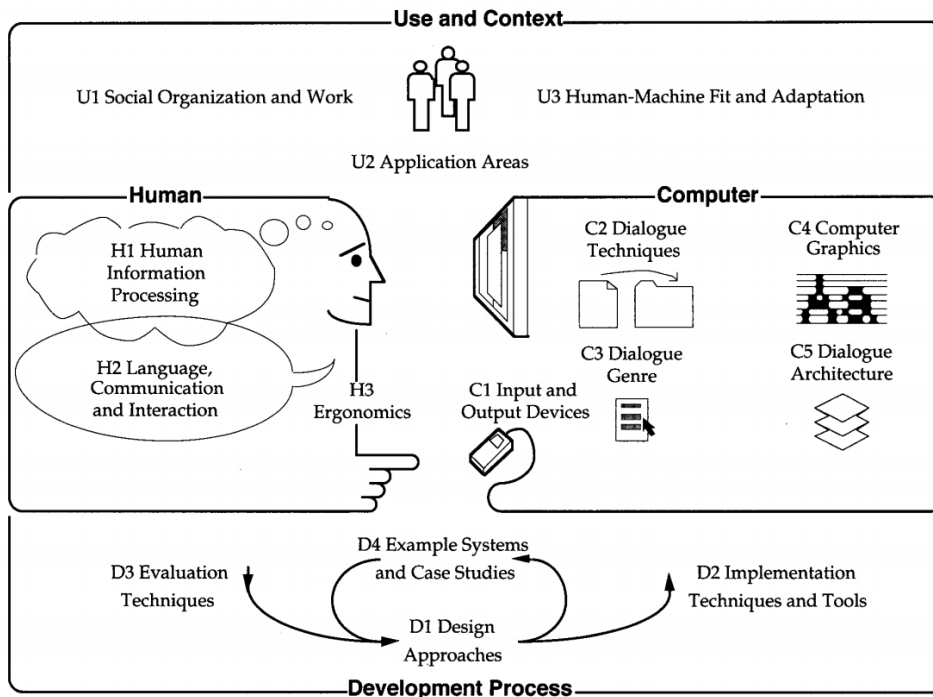


Figura 2: Ecosistema de la Interacción Humano Computador, Obtenido de [22]

La Figura 2 presenta que las interacciones deben tener en cuenta que los sistemas informáticos hacen parte de un entorno social, organizativo y son creados con el fin de cumplir ciertas necesidades específicas (contexto), y para hacer esto posible las aplicaciones deben aplicar diferentes estrategias en el proceso de diseño, considerando el aprendizaje humano y la adaptabilidad del sistema para adaptarse a los aspectos humanos, técnicos o laborales requeridos en cada situación. Esta figura muestra además diferentes factores presentes en el momento de una interacción, y desde el punto de vista de los humanos la HCI estudia los siguientes parámetros:

- Organización social y trabajo(U1): en este factor se tienen en cuenta todas las interacciones y entornos sociales de las personas además de cómo el desarrollo de productos tecnológicos puede mejorar la calidad en la vida y el trabajo de los usuarios.
- Áreas de aplicación(U2): el cual centra su foco de atención en estudiar los diferentes contextos o áreas en los cuales han sido desarrolladas las interfaces de un programa de software.

- Adaptación hombre-máquina(U3): sección en la cual se evalúa el proceso de adaptación de los usuarios con un programa de software con el propósito de realizar ajustes ya sea en la etapa de diseño o al momento del uso por parte del usuario.

Por otra parte, desde el punto de vista del desarrollo a nivel tecnológico la HCI busca centrarse en los siguientes parámetros:

- Técnicas diseño(D1) de los diferentes entornos de interacción para los usuarios.
- Herramientas de desarrollo(D2) con las cuales son implementadas las interfaces gráficas.
- Técnicas de evaluación(D3) aplicadas para dar entendimiento al comportamiento de los humanos durante el proceso de interacción.
- Casos de estudio(D4) previos que sirven como ejemplos ampliados de diseño de interfaces de usuario.

## 1.4 DISEÑO CENTRADO EN EL USUARIO

El Diseño Centrado en el Usuario (UCD, *User Centered Design*) es un enfoque de diseño en el cual se definen procesos con base en las necesidades de los usuarios finales, los cuales juegan un papel central y se involucran durante todas las etapas de los procesos. Teniendo en cuenta la importancia que tienen los usuarios para el UCD, es necesario definir un perfil detallado de los usuarios objetivo, el cual se obtiene a partir de un proceso de investigación y análisis que permite resolver preguntas tales como: ¿Quiénes van a ser los usuarios del producto?, ¿Qué opinan del producto o de productos similares?, ¿Cómo interactúan con el producto?, o ¿En qué condiciones interactúan con el producto y bajo qué circunstancias? [2]. El UCD busca obtener productos que sean de fácil uso e interactivos y para ello se apoya en toda la información obtenida de las personas que se consideran potenciales usuarios, como por ejemplo, el contexto del uso de la aplicación, su interacción con el producto, su comportamiento y/o sus necesidades [23].

El UCD aparece para cambiar y dar un nuevo enfoque al proceso de diseño tradicional que tenían las empresas desarrolladoras de software, en un momento en el cual su foco de atención eran aspectos tales como: las posibilidades técnicas, el desarrollar una interfaz gráfica atractiva o incluso los intereses corporativos. Esto hace que el producto final a menudo no cumpla con el nivel de facilidad de uso esperado por los usuarios o sea muy poco intuitivo, generando finalmente pérdidas económicas y críticas negativas.

El UCD es un proceso en el cual, el usuario se encuentra en el centro de los diferentes procesos realizados, por ende es importante aplicar diferentes técnicas para involucrar al



usuario final en el proceso de diseño. Algunas de las más comunes van desde incluirlo en los procesos de captura de requisitos y pruebas, hasta incluirlo como asesor a lo largo de todas las fases del proceso de diseño.

### 1.4.1 Principios del Diseño Centrado en el Usuario

Existen diferentes autores que sugieren principios para describir lo que se consideran como buenas prácticas en la aplicación del UCD en el proceso de desarrollo, pero para el contexto del presente trabajo de grado se considerarán las expuestas por Don Norman, el cual es considerado el creador de este concepto [24]. Norman describe los siguientes principios como una guía que puede ayudar a los equipos de diseño a aplicar de mejor manera el enfoque de UCD:

1. El diseño se basa en la comprensión del usuario final, sus tareas y su entorno: no basta con tener una vaga idea del público objetivo del producto. El enfoque centrado en el usuario final requiere un conocimiento más profundo de este y sus circunstancias.
2. El usuario final se involucra en todo el proceso de desarrollo: esta es una de las diferencias principales en relación con otros enfoques. No se consulta al usuario final para conocer su opinión acerca de un producto ya terminado, sino que su punto de vista marca el inicio del proceso de desarrollo.
3. La evaluación del usuario final controla el proceso de diseño: el usuario final evalúa todas las versiones beta y todos los prototipos, y toda su realimentación se tiene en cuenta para continuar el desarrollo del producto.
4. Se trata de un proceso iterativo: las distintas etapas del proceso de desarrollo de producto se realizan más de una vez y no de una forma secuencial, teniendo en cuenta la retroalimentación de los usuarios.
5. Se tiene en cuenta toda la experiencia de usuario: el diseño centrado en el usuario no trata de simplificar al máximo el uso de un producto, sino que entiende la experiencia del usuario como un concepto más amplio. La idea es que el producto provoque emociones positivas, ofrezca soluciones auténticas y fomente su uso.
6. El equipo del proyecto es multidisciplinario: el UCD requiere una estrecha colaboración que trascienda las distintas disciplinas. El pensamiento aislado no tiene cabida en el desarrollo de productos software. Solo es posible satisfacer los requisitos de los usuarios si se tienen en cuenta las distintas perspectivas de los programadores, ingenieros y comunicadores.

### 1.4.2 Fases del Diseño Centrado en el Usuario

Si bien el proceso de diseño varía en función de diferentes factores tales como: la aplicación que se está desarrollando, el perfil de los usuarios finales o las capacidades tecnológicas, se pueden definir 4 fases principales en las cuales se divide un proceso UCD: análisis de contexto; definición de requisitos; diseño; y evaluación, los cuales se explican a continuación.

**Análisis de contexto** En esta fase el objetivo principal es descubrir para quién se está diseñando. La idea de esta fase es crear un perfil lo más detallado posible del usuario final, y para ello es importante en esta etapa recolectar la mayor cantidad de información acerca de las necesidades de los potenciales usuarios o de cuales son sus expectativas frente al sistema a diseñar.

**Definición de los requisitos** Como segunda fase se definen los requerimientos del proyecto y es en esta etapa donde se debe tener un mayor acercamiento con el usuario final a través de diferentes métodos, con el fin de extraer una mayor cantidad de información y entender de mejor manera el problema principal.

Además de involucrar a los usuarios, también es importante realizar un acercamiento con los actores involucrados en el proyecto (*stakeholders*, clientes, asesores, colegas del sector, etc.) para de esta manera tener una perspectiva de los requerimientos a nivel de negocio del sistema.

**Diseño** Una vez se tiene toda la información necesaria, se inicia la fase de diseño en donde el principal objetivo es obtener una versión prototipo del producto, la cual pueda ser evaluada por los usuarios. Durante el proceso de diseño es común encontrar etapas que incluyen procesos de análisis y planeación, ya que es necesario tener definida una ruta de uso con base en la información recolectada de los usuarios.

**Evaluación** Finalizado el proceso de diseño donde se obtiene un prototipo funcional del producto, es necesario recibir una retroalimentación por parte de los usuarios potenciales. El principal objetivo es responder a la siguiente pregunta: ¿El usuario puede resolver el problema planteado haciendo uso del producto?, para lograr esto es necesario aplicar diferentes métodos que faciliten obtener información de los comportamientos y opiniones de los usuarios en diferentes contextos de uso de la aplicación.

## 1.5 USABILIDAD

La norma ISO 9241-11 define el concepto de usabilidad como el grado en que un producto de software puede ser usado por determinados usuarios para lograr sus objetivos con eficacia, eficiencia y satisfacción en un contexto de uso específico [25]. De la definición anterior se puede establecer que la usabilidad por sí sola no es un factor medible, pero que depende de otros conceptos que sí pueden serlo y además la usabilidad es uno de los factores que se busca alcanzar mediante un determinado proceso de diseño.

Tomando como referencia las fases del proceso de desarrollo de software, en especial la etapa de diseño, el objetivo principal de la usabilidad es centrarse en descubrir las necesidades del usuario y suplirlas a través del producto final. Existen diferentes enfoques a la hora de definir lo que se puede considerar como un producto usable. Según Bevana [26] se pueden considerar tres enfoques que buscan definir la usabilidad y la forma de ser medida:

- Desde el punto de vista del producto: donde la usabilidad se mide en términos de la ergonomía y los atributos propios del producto final.
- Desde el punto de vista del usuario: donde la usabilidad es medida en términos del entendimiento y el esfuerzo mental que el usuario invierte para usar el producto final.
- Desde el punto de vista del rendimiento del usuario: donde la usabilidad es medida en términos de la interacción del usuario con el producto, destacando la facilidad de uso y la aceptabilidad del producto final.

### 1.5.1 Conceptos de la Usabilidad

El concepto de usabilidad es de carácter ambiguo, ya que se puede interpretar de diferentes maneras dependiendo de: el contexto en el cual se usa el programa software; el usuario que interactúa con el programa software; y el propósito general para el cual fue desarrollado. Por lo anterior, la literatura define una serie de conceptos que en conjunto conforman la definición de usabilidad: capacidad de aprendizaje, eficiencia, recordación, tasa de error y satisfacción [27].

1. Capacidad de aprendizaje: en qué medida es fácil para los usuarios completar tareas básicas la primera vez que usan el producto.
2. Eficiencia: qué tan rápido los usuarios pueden completar tareas específicas usando el producto.

3. Recordación: qué tan fácil es para los usuarios recordar como usar el producto después de un tiempo sin hacerlo.
4. Tasa de error: cuántos errores cometen los usuarios usando el producto, que tan graves son los errores y que tan fácil los usuarios pueden recuperarse de los errores.
5. Satisfacción: qué tan placentero es para el usuario usar el producto.

## 2 METODOLOGÍA

En el presente capítulo se realiza la definición de los requerimientos para el desarrollo de la aplicación móvil Unicauca Estéreo. De manera inicial se realiza la definición del concepto de ciclo de vida del desarrollo de software en la sección 2.1 Para el desarrollo de las diferentes fases del proyecto se toma como referencia el modelo iterativo e incremental el cual es descrito en la sección 2.2 definiendo cómo se integra dicho modelo a las etapas de desarrollo. En la sección 2.3 se realiza la definición de requerimientos del sistema planteados por el equipo administrativo de la emisora Unicauca Estéreo y en la sección 2.4 se definen los criterios de usabilidad que debe cumplir la aplicación móvil y que deben ser considerados durante el proceso de diseño de la misma.

### 2.1 CICLO DE VIDA DEL DESARROLLO SOFTWARE

El Ciclo de Vida del Desarrollo Software (SDLC, *Software Development Life Cycle*) es un metodología aplicada en el diseño, implementación, validación y mantenimiento de un sistema o aplicación software [28]. Esta metodología se utiliza para desarrollar software de una manera más simple y ordenada, generando una serie de pasos regularmente descritos en un documento, el cual es usado como referencia por los involucrados en el proyecto. De igual manera se puede definir como un conjunto de actividades que realizan el equipo de desarrolladores, diseñadores, analistas, comunicadores, etc., para desarrollar e implementar un sistema software.

Existen diferentes tipos de modelos para el SDLC que han sido propuestos a lo largo de la historia del desarrollo de sistemas software, cada uno de los cuales plantea un enfoque diferente de cómo abordar la organización de un proyecto software, con sus ventajas y desventajas. Entre los más destacados se encuentran el modelo en cascada, el modelo en espiral, el modelo iterativo e incremental, y el modelo en V [29].

### 2.2 MODELO ITERATIVO E INCREMENTAL

El modelo iterativo e incremental define una serie de bloques temporales denominados iteraciones, en el cual cada bloque añade valor al sistema o producto que se encuentra en desarrollo. Para profundizar en la definición de este modelo es necesario detallar por separado los conceptos de desarrollo iterativo y desarrollo incremental.

## Desarrollo iterativo

El desarrollo iterativo toma como referencia el modelo en cascada para realizar varias iteraciones. La Figura 3 muestra una representación de este modelo donde al final de cada iteración se entrega una versión mejorada del sistema. Independiente del actor que esté encargado de la validación, el objetivo principal es obtener correcciones o propuestas para mejorar las funcionalidades del sistema, las cuales serán evaluadas y aplicadas en la siguiente iteración. Las iteraciones se repetirán hasta obtener un sistema que cumpla con los requerimientos establecidos inicialmente en conjunto con el cliente.

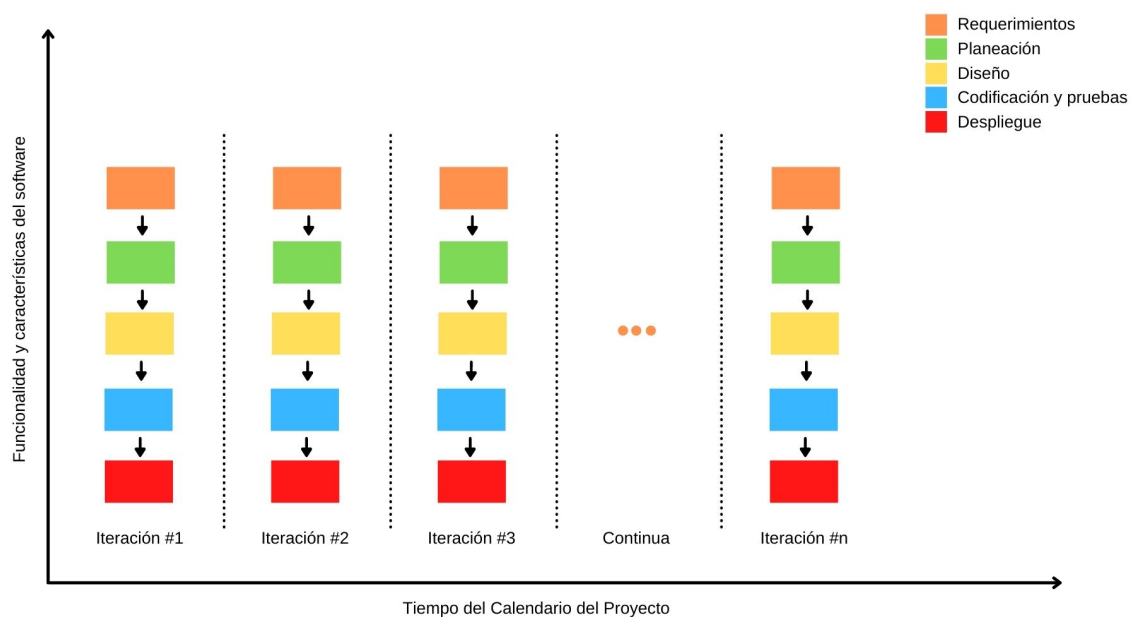


Figura 3: Modelo iterativo

## Desarrollo incremental

Con este enfoque de desarrollo se busca realizar varias entregas a lo largo del proyecto, como se puede observar en la Figura 4, cada una de estas entregas adiciona una funcionalidad al sistema. Cada una de estas entregas debe tener un valor para el usuario final, es decir, el usuario debe ser capaz de probar o interactuar con dicha funcionalidad. Con estas iteraciones se busca refinar el concepto general del sistema que se encuentra en desarrollo.

De esta manera, el modelo iterativo e incremental unifica las ideas de los conceptos descritos anteriormente, con el fin de generar un enfoque de desarrollo en el cual en cada

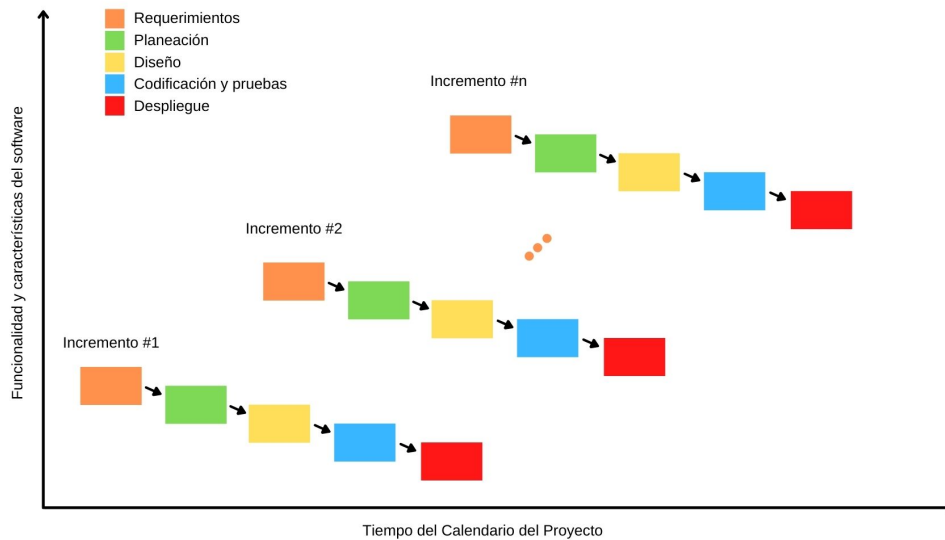


Figura 4: Modelo incremental

iteración se desarrollan funcionalidades o partes del sistema y al mismo tiempo se realicen mejoras o cambios sobre las funcionalidades existentes.

De acuerdo a [30], el uso del modelo iterativo e incremental tiene diferentes ventajas entre las cuales se destacan:

- la flexibilidad de realizar cambios a lo largo del proyecto con base en las necesidades o propuestas del cliente.
- La reducción de riesgos debido al constante acompañamiento al cliente y los usuarios finales.
- La posibilidad de realizar un desarrollo paralelo de nuevas funcionalidades y mejoras o cambios en funcionalidades previas.
- Modularidad para desarrollo y pruebas, ya que las fases se realizan durante cada iteración para cada una de las funcionalidades del sistema.

Con base en lo descrito anteriormente, se plantea una adaptación del modelo iterativo e incremental para el desarrollo de la aplicación móvil Unicauca Estéreo en el presente trabajo de grado.

## 2.3 DEFINICIÓN DE REQUERIMIENTOS

Con base en el proceso de captura y definición de requerimientos, proceso realizado en conjunto con el equipo técnico y administrativo de la emisora institucional Unicauca Es-

téreo, se planteó realizar el desarrollo de una aplicación móvil para los sistemas operativos iOS y Android, la cual debe contar con la funcionalidad principal de reproducción de la señal en vivo de la emisora. Adicionalmente, la aplicación móvil de Unicauca Estéreo tendrá las siguientes funcionalidades:

- La aplicación debe permitir reproducir los episodios almacenados de los diferentes programas, los cuales se deben visualizar en una sección destinada y organizada para este fin.
- La aplicación debe contar con una sección de noticias, en la cual se encuentren las noticias más recientes publicadas por Unicauca Estéreo.
- La aplicación debe contar con una sección destinada para la agenda universitaria, en la cual sea posible acceder a los eventos de la agenda universitaria publicados por parte de la emisora.

Otras funcionalidades secundarias que se definen para la aplicación son las siguientes:

- La aplicación debe contar con una sección de parrilla de programación semanal de la emisora.
- La aplicación debe contar con una sección con información general de la emisora Unicauca Estéreo y su historia.
- La aplicación debe contar con una sección de contacto con la emisora a través de diferentes medios como correo electrónico y redes sociales.

El diseño de las interfaces gráficas de usuario y el flujo de navegación de la aplicación se desarrollan cumpliendo con una lista de criterios de usabilidad los cuales se definen en la sección 2.4.

Adicional al proceso realizado para la definición de requerimientos, se realiza una encuesta libre a los miembros de la comunidad de la Universidad del Cauca con el propósito de evaluar la sección del mercado a la cual está enfocada la aplicación móvil y recolectar información relevante que pueda servir para la toma de decisiones en el proceso de diseño e implementación del presente proyecto. Dentro de esta encuesta se realizan a los participantes preguntas relacionadas con el perfil demográfico y el consumo y acceso a servicios relacionados con la radio, *podcast* y noticias. Los resultados obtenidos así como un análisis de los mismos se presentan en el anexo A.



## 2.4 DEFINICIÓN DE CRITERIOS DE USABILIDAD

La evaluación basada en heurísticas es un método de resolución de problemas utilizado para analizar los inconvenientes de usabilidad en una interfaz de usuario y que permite resolver ciertas carencias relacionadas con las interacciones entre los usuarios y la interfaz de un sistema. Esta alternativa es uno de los métodos de usabilidad más rápidos, fáciles y de bajo costo al aplicar en el área de la evaluación de usabilidad de sistemas.

Para definir los criterios de usabilidad en los cuales va a estar basado el desarrollo de la aplicación móvil Unicauca Estéreo, se toma como referencia las 10 heurísticas de usabilidad descritas por Nielsen [31], las cuales han sido ampliamente usadas en la práctica y son una guía para el diseño de interfaces de usuario.

### 2.4.1 Diez Heurísticas de Usabilidad de Nielsen

Las diez heurísticas de usabilidad fueron planteadas por Jakob Nielsen en el libro “Usability Engineering” en 1994 [27], las cuales son un conjunto de pautas generales que deben ser interpretadas en múltiples contextos en el diseño de sistemas que incorporan interfaces de usuario.

**Visibilidad del estado del sistema.** El sistema siempre debe informar al usuario sobre el estado actual de las diferentes actividades que se realizan en éste a través de la interfaz. Un ejemplo de la visibilidad del sistema se puede observar cuando el color o las características de un botón cambian cuando éste es presionado. Una interacción predecible para el usuario genera confianza en el sistema.

**Coincidencia entre el sistema y el mundo real.** Debe existir una concordancia entre los términos que se utilizan en el sistema y los que se usan en el mundo real, además de lo anterior, también es necesaria la existencia de una lógica de procesamiento de modo que los flujos de los procesos del sistema sean acordes a los flujos lógicos que se siguen en la vida real. El sistema debe ser lo más lógico y natural posible. Esto le permite a los usuarios familiarizarse de una manera sencilla con el sistema, recordarlo y que su proceso de aprendizaje sea más ágil.

**Libertad para operar y controlar.** Es muy común que los usuarios comentan errores al interactuar con un sistema, por ejemplo, es fácil encontrar que un usuario presione un botón mientras explora una interfaz y que esto lo lleve a una nueva vista que no necesitaba. Esta heurística plantea que se debe brindar a los usuarios facilidades para tener un control en la navegación del sistema. Si los usuarios perciben este tipo de facilidades a la hora de

interactuar con el sistema, van a sentir libertad y seguridad de operar cualquier acción o funcionalidad dentro de éste.

**Coherencia y estándares.** Esta heurística sugiere que se deben seguir las convenciones de la plataforma y la industria en la cual se encuentra categorizado el sistema. Cada vez que se use o aplique un color, un icono o una palabra en las sucesivas pantallas del sistema, este o esta debe significar lo mismo. Esta heurística hace mención a los textos o a la forma en la cual se expresan los mensajes.

**Prevención de errores.** Si bien es importante presentar de manera correcta y coherente mensajes frente a algún eventual error dentro del sistema o brindar al usuario la facilidad para revertir estos errores, un buen diseño debe centrarse en su prevención. Esta heurística sugiere que el sistema debe aplicar herramientas como sugerencias, limitaciones y flexibilidad con el fin de prevenir que el usuario incurra en algún error al momento de interactuar con el sistema.

**Reconocer en lugar de recordar.** La memoria de los seres humanos es limitada y debería estar enfocada en otras tareas más que en recordar cómo se usa un sistema. Esta heurística propone que un sistema correctamente diseñado debe brindar al usuario ayudas o guías como mínimo en el proceso inicial de interacción con la aplicación o aún mejor presentar una interfaz intuitiva que ayude a los usuarios a desenvolverse fácilmente en su uso.

**Flexibilidad y eficiencia en el uso.** Se debe permitir que los usuarios completen las tareas de diferentes maneras, ya que los usuarios de un sistema son diferentes, es por ello que esta heurística plantea la idea de brindar al usuario maneras de personalizar el sistema para que su uso sea más eficiente.

**Diseño estético y minimalista.** Esta Heurística propone que un diseño correcto no debe presentar una sobrecarga excesiva de las interfaces de usuario, y para ello es necesario que el sistema evite contener información irrelevante para el usuario o que casi nunca se use. Un diseño con unidades extra de información puede disminuir la visibilidad de las unidades relevantes, lo cual genera una carga de memoria adicional para el usuario.

**Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores.** El sistema debe permitir al usuario estar enterado de los errores en cada oportunidad que uno ocurra, así mismo, se deben notificar los errores de una manera natural para el usuario,

y por ello es necesario evitar usar lenguajes técnicos o códigos de error. La información debe ser presentada en un lenguaje de fácil entendimiento para el usuario y de preferencia sugerir una solución al mismo.

**Ayuda y documentación.** Si bien la idea principal es que el sistema y las interfaces de usuario sean intuitivas y fáciles de usar de modo que el usuario pueda interactuar de manera empírica y sencilla con estas, en ocasiones y dependiendo del tipo de sistema que se está desarrollando va a ser necesario contar con un manual de usuario, una guía de uso o un sistema o plataforma de soporte que permita a los usuarios resolver cualquier duda o problema que se pueda presentar.

Sin embargo, a pesar de las ventajas que tiene aplicar el conjunto de heurísticas de Nielsen, es necesario realizar una adaptación a las características específicas de las aplicaciones móviles, tales como: tamaños de pantalla reducidos, la ergonomía de los dispositivos móviles o su constante evolución en términos de tecnología, lo cual permite lograr resultados más efectivos en la evaluación de la usabilidad, lo que implica una mayor calidad en la experiencia de usuario [32], [33].

Durante proceso de definición de los criterios de usabilidad del presente proyecto, es importante considerar otras investigaciones o artículos que planteen conjuntos de heurísticas adaptados específicamente al contexto de las aplicaciones móviles. Además de lo anterior, el conjunto de heurísticas definido debe estar regido por los requerimientos planteados por la Emisora Unicauca Estéreo descritos en la sección 2.3, en aras de contar en el proceso de diseño con una referencia que se adapte no solo al contexto de las aplicaciones móviles sino también al del presente proyecto en específico.

Para el presente proyecto se definen diez heurísticas las cuales son planteadas por los autores Joyce y Lilley en "*Towards the Development of Usability Heuristics for Native Smartphone Mobile Applications*" [34]. En este documento se realiza un proceso metodológico, en el cual se analizan múltiples artículos relacionados con la usabilidad de aplicaciones móviles para proponer un conjunto de heurísticas propio [35]–[38], lo anterior con el fin de presentar un conjunto de trece heurísticas, las cuales son una versión personalizada basada en las propuestas por Nielsen, pero adaptada específicamente al contexto de las aplicaciones móviles.

De la lista propuesta por Joyce y Lilley se excluyen tres heurísticas: en primer lugar, "*Mostrar una superposición que señale las características principales cuando sea apropiado o se solicite*", debido a que el cumplimiento de esta heurística implica realizar una funcionalidad que no se encuentra en los requerimientos planteados por la emisora Unicauca Estéreo descritos en la sección 2.3; en segundo lugar, "*Facilitar la entrada de datos*", debido a que los contenidos de la aplicación móvil son una imagen o un reflejo de la in-

formación en el portal web de la emisora, la cual tiene un comportamiento informativo, y en ninguna sección de la aplicación se requiere que el usuario ingrese datos mediante el teclado del dispositivo móvil; finalmente, "*Utilizar los sensores cuando sea necesario para reducir la carga de trabajo de los usuarios*", se excluye debido a que no es necesario el uso de los sensores del dispositivo móvil para implementar ninguna de los requerimientos funcionales de la aplicación.

A continuación se presenta la lista de heurísticas definidas como referencia para el diseño e implementación de la aplicación móvil Unicauca Estéreo. En la lista se denota un código para cada heurística además de una descripción.

**H1 - Proporcionar una notificación inmediata del estado de la aplicación.** El usuario de la aplicación móvil debe estar constantemente e inmediatamente enterado del estado de los procesos que se llevan a cabo en las diferentes secciones.

**H2 - Utilizar un tema y términos coherentes, así como convenciones y normas conocidas por el usuario.** Se debe manejar un tema general para las diferentes vistas de la aplicación. El tema debe contener un conjunto de textos, iconos, expresiones y colores que sean homogéneos y que se adapten al contexto de la aplicación y mayormente al de los usuarios.

**H3 - Prevenir los errores siempre que sea posible. Ayudar a los usuarios en caso de que se produzca un error.** La aplicación debe ser en la medida de lo posible a prueba de errores y en caso de que se presente un error se debe notificar al usuario de una manera que sea entendible y si es posible, presentar una forma de solucionar el error.

**H4 - Cada interfaz debe centrarse en una tarea.** Centrarse en una tarea garantiza que cada una de las vistas de la aplicación tengan un aspecto menos recargado y simple de manera que se incluyan sólo los elementos absolutamente necesarios en pantalla para completar dicha tarea.

**H5 - Diseñar una interfaz visualmente agradable.** Las vistas que cuentan con un aspecto más atractivo y minimalista son mucho más memorables para el usuario y por ende más utilizadas. Los usuarios también suelen ser mas permisivos con las interfaces que poseen estas características.

**H6 - Las interfaces intuitivas facilitan los desplazamientos de los usuarios.** Las interfaces móviles deben ser fáciles de aprender. Esto permite a los usuarios completar

sus tareas con mayor facilidad.

**H7 - Diseñar un camino claro y navegable para completar la tarea.** Los usuarios deben ser capaces de ver de inmediato cómo pueden interactuar con la aplicación y navegar hasta completar la tarea.

**H8 - Permitir opciones de configuración y accesos directos.** Dependiendo del usuario al que se dirija, la aplicación móvil puede permitir opciones de configuración y accesos directos a la información más importante y a las tareas más frecuentes, incluyendo la posibilidad de configurar según las necesidades contextuales.

**H9 - Adaptarse a diversos entornos móviles.** Los entornos diversos consisten en diferentes tipos de contextos de uso, como las condiciones de mala iluminación y el elevado ruido ambiental, que son dolencias comunes a las que los usuarios de móviles tienen que enfrentarse cada día.

**H10 - Crear un icono estético e identificable.** Un icono para una aplicación móvil debe ser estético e identificable, ya que es lo que un usuario ve cuando busca en el dispositivo y cuando busca en las tiendas de aplicaciones. Será el primer elemento que vea antes del título de la aplicación, la descripción y las capturas de pantalla.

### 3 DISEÑO DEL SISTEMA

En el presente capítulo se realiza una descripción del proceso de diseño de la aplicación móvil Unicauca Estéreo. Se define un diseño que no solo cumple con los requerimientos descritos en la sección 2.3 si no que también se basa en los criterios de usabilidad establecidos en la sección 2.4. Además de lo anterior, durante el presente capítulo se realiza una justificación de las decisiones tomadas para cada uno de los procesos de diseño planteados.

El presente capítulo se divide en: estado inicial del sistema; arquitectura del sistema; diseño técnico y de componentes; diagrama de flujo de navegación; y diseño de interfaces de la aplicación.

#### 3.1 ESTADO INICIAL DEL SISTEMA

La emisora universitaria Unicauca Estéreo en el estado inicial del presente proyecto cuenta con un ecosistema web como se puede observar en la Figura 5. Este ecosistema está constituido por un servidor Apache en su versión 2.2.162.2.22, el cual es soportado por una base de datos relacional MySQL con una versión 5.1.4.9.3. Sobre el servidor se ejecuta un sistema Drupal, sobre el cual está construido el portal web de la emisora, desde el cual se brindan servicios informativos a la comunidad universitaria.

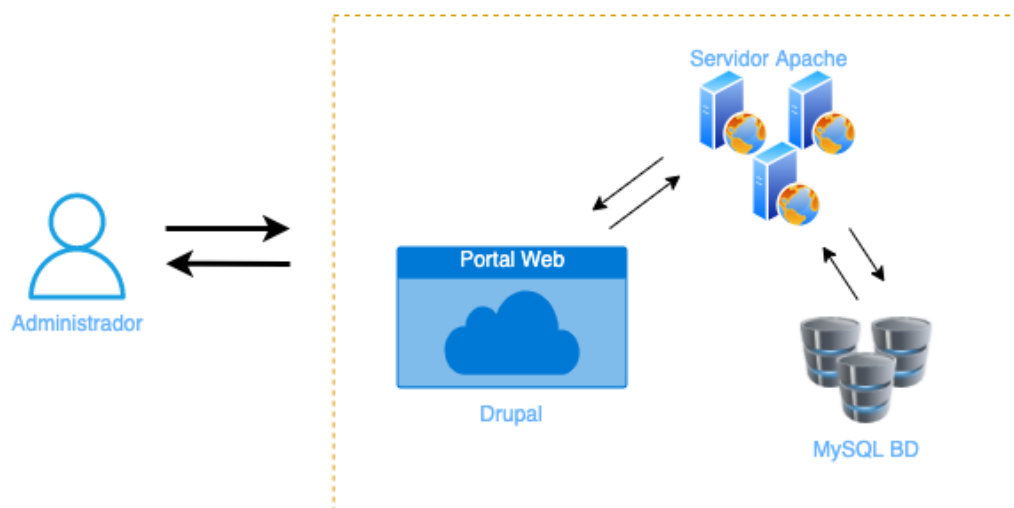


Figura 5: Arquitectura del estado inicial ecosistema web Unicauca Estéreo

Con base en la propuesta de valor del presente proyecto, el cual comprende a nivel general el desarrollo de la aplicación móvil para la emisora Unicauca Estéreo, y teniendo en cuenta que dicha aplicación debe ofrecer los mismos servicios que se encuentran en

el portal web, se plantea adicionar un nuevo ecosistema de tipo móvil a la arquitectura descrita anteriormente.

### 3.2 DISEÑO FUNCIONAL DEL SISTEMA

El sistema planteado para el desarrollo de la aplicación móvil Unicauca Estéreo se basa en el modelo cliente servidor, el cual describe que las tareas delegadas por parte del cliente se deben distribuir entre los servidores disponibles. La aplicación móvil constituye el ecosistema denominado cliente como se puede observar en la Figura 6. Es importante destacar que la aplicación puede ser ejecutada tanto en el sistema operativo Android como en el sistema operativo de iOS para el caso de un iPhone. El usuario interactúa con la aplicación mediante sus interfaces gráficas con las cuales puede acceder a las diferentes funcionalidades descritas en los requerimientos del sistema.

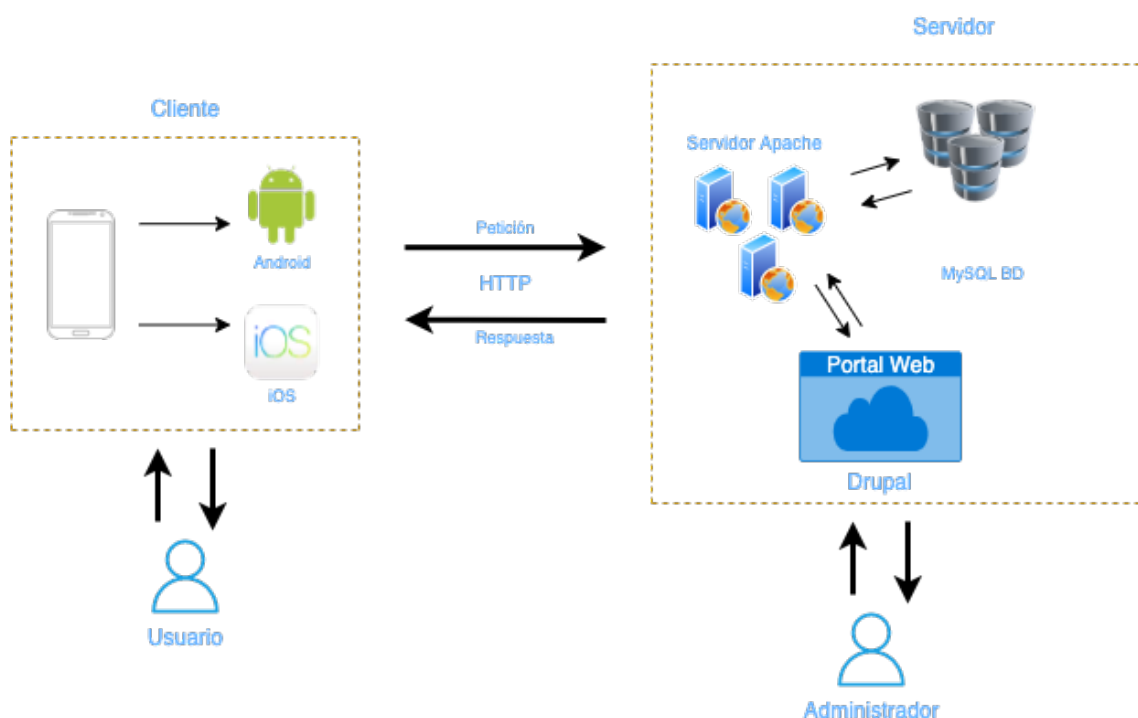


Figura 6: Ecosistema móvil en arquitectura de sistema Unicauca Estéreo

En el ecosistema del servidor no se presentan cambios de arquitectura frente a lo descrito en la sección 3.1, sin embargo, es importante recalcar que se hace necesaria la instalación y configuración de nuevos módulos a nivel de software, los cuales se describen en la sección 3.3, lo anterior con el fin de brindar el soporte necesario para la adecuación

del nuevo ecosistema móvil. La comunicación entre los dos ecosistemas (Cliente/Servidor) se realizan mediante el Protocolo de Transferencia de Hipertextos (HTTP, *Hypertext Transfer Protocol*), en el cual el cliente realiza peticiones al servidor y el cual responde con la información necesaria en un formato específico, y esta información es procesada e interpretada por el cliente para ejecutar un cambio en el contexto de la aplicación móvil.

### 3.3 DESCRIPCIÓN DE ARQUITECTURA

La arquitectura de la aplicación se describe a través del modelo “4+1 vistas” planteado por el profesor Philippe Kruchten, el cual describe la arquitectura de un sistema software aplicando diferentes vistas que son concurrentes en el desarrollo en general [39]. El objetivo principal de este modelo es describir un sistema basado en los diferentes puntos de vista de los interesados en el proyecto, tales como, gerentes de proyecto, desarrolladores o usuarios finales. El modelo describe cuatro vistas principales, las cuales son: vista lógica, vista de desarrollo, vista de procesos y vista física, como se observa en la Figura 7, la cual se basa en el Proceso Unificado de Desarrollo (RUP, *Rational Unified Process*). Adicionalmente, se describe el sistema usando casos de uso.

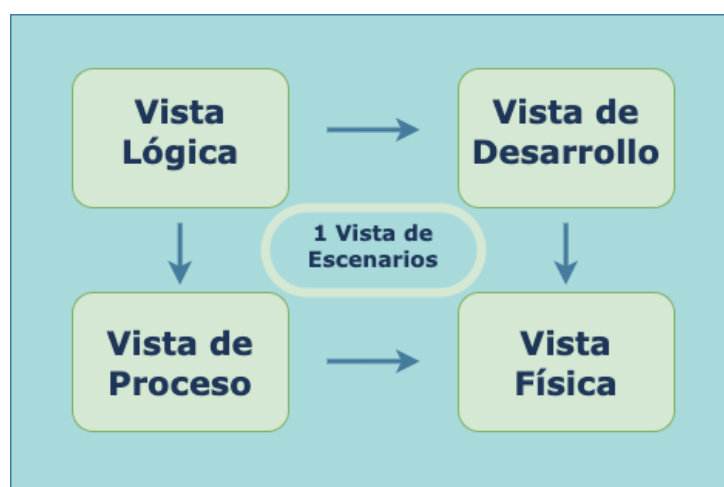


Figura 7: Modelo de 4+1 vistas del Proceso Unificado de Desarrollo

#### 3.3.1 Vista de Escenarios

La vista de escenarios se relaciona con las otras cuatro vistas de este modelo y su función principal es la de representar una abstracción de los requerimientos funcionales más importantes del sistema, siendo el punto de partida que deriva en las demás vistas. El diseño de la vista de escenarios se describe a través de diagramas de casos de uso. A



continuación, se presentan los requerimientos funcionales de la aplicación móvil, para dar paso al planteamiento de la vista de escenarios del sistema.

- Reproducir señal en vivo.
- Reproducir un *podcast* de un programa.
- Visualizar noticias.
- Visualizar *tweets*.
- Visualizar eventos.
- Visualizar programas.
- Visualizar parrilla de programación.
- Agregar un programa a favoritos.
- Eliminar un programa de favoritos.
- Agregar un *podcast* a favoritos.
- Eliminar un *podcast* de favoritos.
- Activar notificaciones para un programa favorito.
- Visualizar información de la emisora.
- Contactar a la emisora.
- Visualizar página web.
- Visualizar redes sociales.

Los requerimientos funcionales se dividen con base en el sistema de navegación de la aplicación, específicamente, se considera a qué menú pertenece cada uno, exceptuando el diagrama de flujo presentado en la Figura 5, el cual expresa las funcionalidades relacionadas con el sistema de favoritos de la aplicación.

En la Figura 8 se describen los casos de uso que representan las funcionalidades principales del sistema. La aplicación debe permitir a los usuarios reproducir la señal en vivo de la emisora Unicauca Estéreo, así como también reproducir un capítulo o *podcast* de un programa perteneciente a la emisora. La aplicación debe permitir igualmente visualizar la lista de dichos programas, así como un detalle de cada uno, donde se listen los *podcast* publicados por el programa. Debe ser posible para el usuario visualizar noticias, eventos

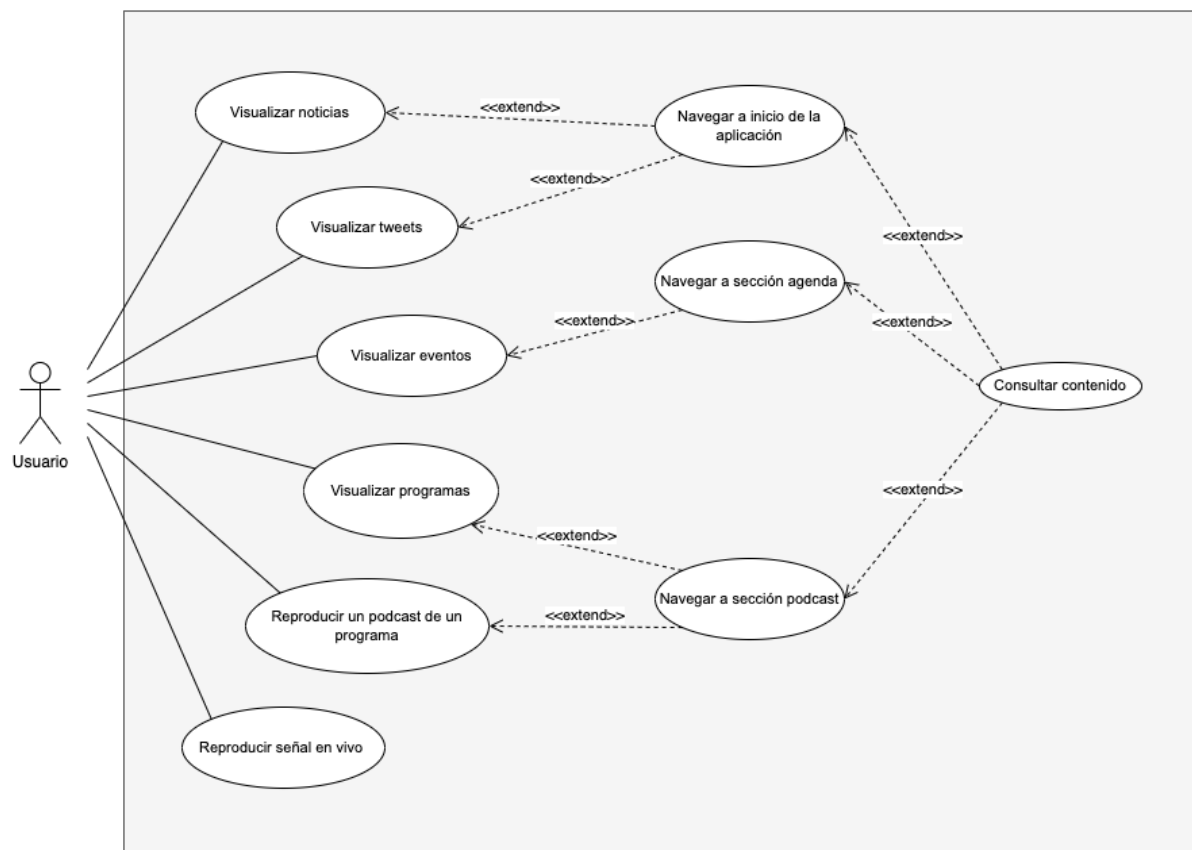


Figura 8: Diagrama de casos de uso menú principal

y *tweets* publicados por la emisora. Esta información se debe encontrar en diferentes secciones de la aplicación.

Las funcionalidades pertenecientes al menú lateral de la aplicación son representadas mediante casos de uso en la Figura 9. En este menú, el usuario debe ser capaz de navegar hacia la pantalla que despliega la parrilla de programación, en la cual se debe mostrar al usuario los programas que se presentan cada día en la emisora universitaria. La aplicación debe permitir también contactar a la emisora mediante correo electrónico o número de teléfono, opciones que el usuario debe poder elegir, y para lo anterior el dispositivo móvil hará uso de la aplicación que se use para cada uno de estos dos propósitos. Por otra parte, la aplicación debe permitir acceder tanto a la página web de la emisora, así como a sus diferentes redes sociales. El usuario debe poder escoger una de estas opciones y de igual manera el dispositivo móvil hará uso de la la aplicación de contacto requerida (navegador en caso de la página web y aplicación en caso de una red social).

El módulo de favoritos, al tener una complejidad mayor se describe en un diagrama de casos de uso independiente descrito en la Figura 10. Los requerimientos funcionales

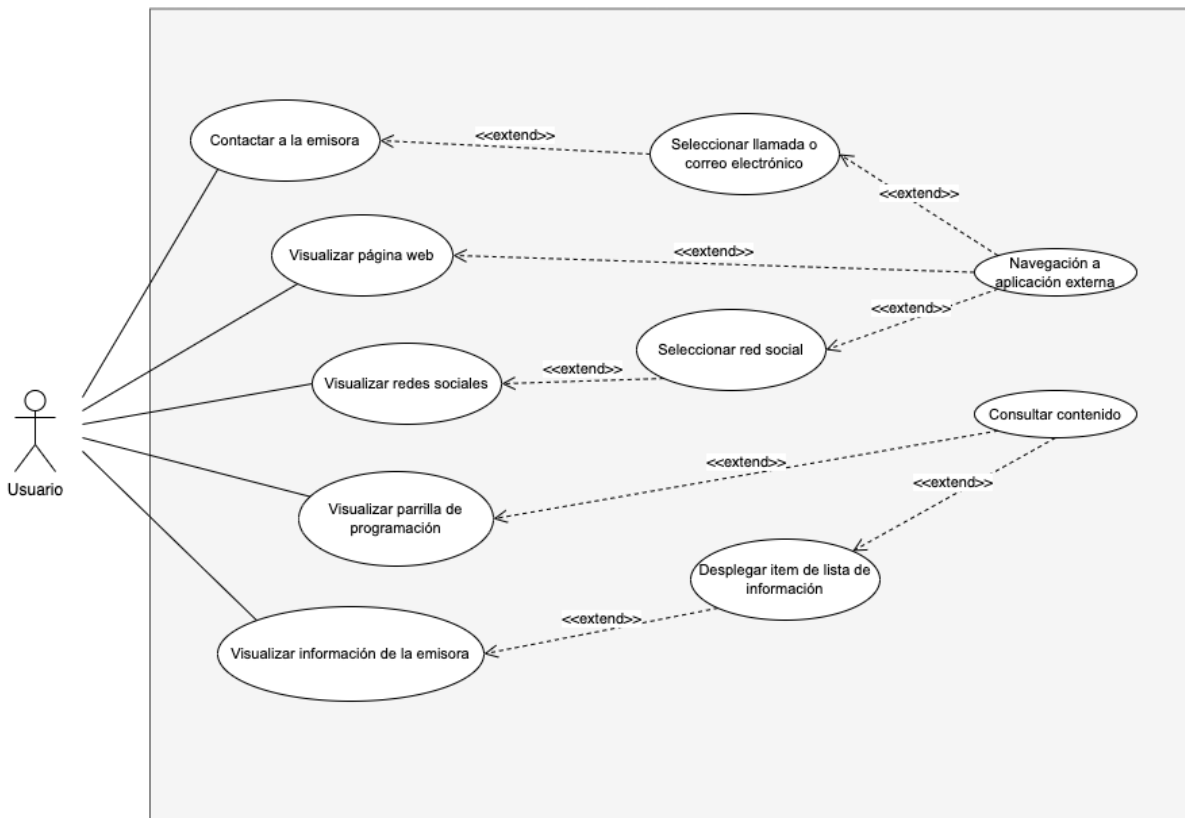


Figura 9: Diagrama de casos de uso menú lateral

de éste módulo se basan en la posibilidad del usuario de agregar y eliminar de una lista de favoritos, tanto a los programas de la emisora universitaria como a cada uno de los *podcast* de dichos programas. La aplicación móvil debe permitir al usuario visualizar cada una de estas listas y además activar una notificación al usuario de un programa que esta pronto a iniciar su transmisión en vivo.

### 3.3.2 Vista de Lógica

La vista lógica busca describir la estructura de los requerimientos funcionales del sistema. Esta vista es representada mediante clases de objetos, aplicando conceptos del desarrollo de software como la abstracción, encapsulamiento y herencia. El diagrama que usualmente se usa para describir esta vista es el diagrama de clases.

La Figura 11 detalla el diagrama de clases de la aplicación móvil Unicauca Estéreo, en el cual se puede observar la clase abstracta "Usuario" que implementa una interfaz "IUsuario", en la cual se gestionan todos los métodos generales que describen los requerimientos funcionales de las interfaces gráficas del sistema. La clase abstracta "Usuario"

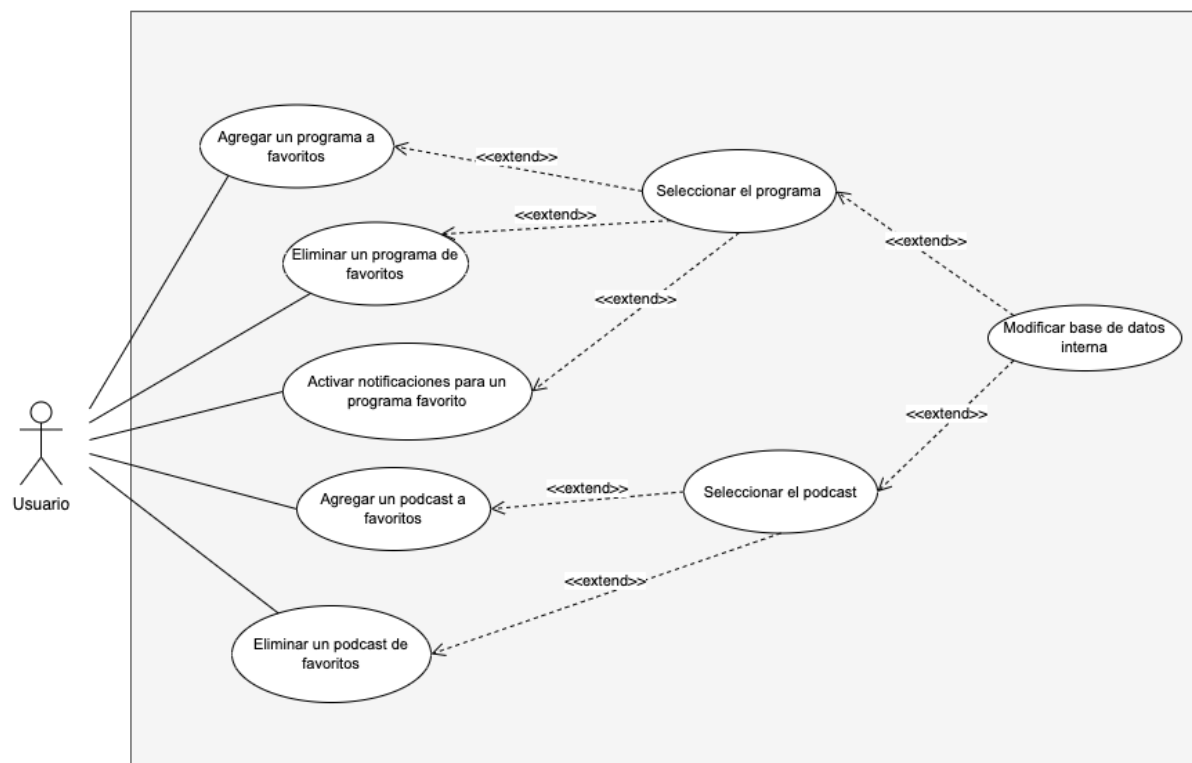


Figura 10: Diagrama de casos de uso módulo de favoritos

está asociada a la clase "Visualizacion", la cual tiene como función gestionar la navegación en la aplicación, permitiendo la visualización de las diferentes pantallas y la cual a su vez se encuentra asociada a la clase "Consulta", encargada de realizar las peticiones al servidor de la emisora Unicauca Estéreo a través de los diferentes *endpoints* dispuestos para cada servicio. Dependiendo de los parámetros que se envían en la petición, la clase "Consulta" puede retornar una o varias instancias de las clases "Evento", "Noticia" o "Programa". La clase "Programa" hereda sus atributos a la clase "Podcast" y a su vez ambas están asociadas a la clase "Favorito", la cual se encarga de gestionar el sistema de favoritos de la aplicación.

### 3.3.3 Vista de Procesos

La vista de procesos describe los aspectos dinámicos del sistema, específicamente, los procesos que se llevan a cabo y cómo estos se comunican en tiempo de ejecución. Para describir esta vista se emplean diagramas de flujo, también llamados diagramas de actividades. Al igual que en la vista de escenarios descrita en la sección 3.3.1, se dividen los diagramas en dos módulos: el primero haciendo alusión a los procesos de la pantalla de inicio; y el segundo al navegador principal de la aplicación móvil Unicauca Estéreo.

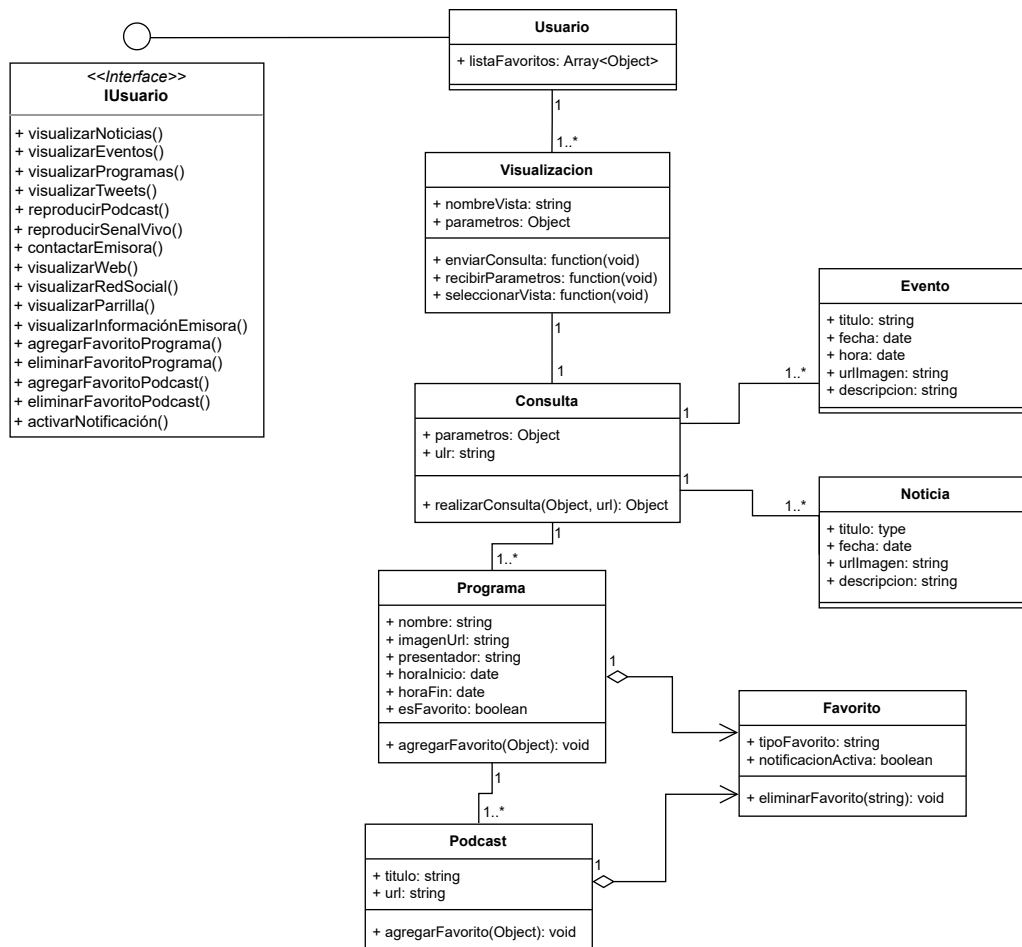


Figura 11: Diagrama de clases aplicación móvil Unicauca Estéreo

La Figura 12 presenta el proceso de la pantalla de inicio de la aplicación, donde se inicia desde la presentación de la vista de tipo *splash*, la cual tiene como función permitir la inicialización de la aplicación mientras el sistema valida el estado de las diferentes variables de entorno. Posteriormente se da paso a la vista de inicio, dentro de la cual el usuario puede visualizar una lista recortada de las diferentes secciones de la aplicación (señal en vivo, programas a continuación, noticias recientes, *tweets* recientes de la emisora, entre otros) y navegar al detalle en cada sección.

La vista de inicio descrita en la sección 3.3.2 hace parte del menú principal de la aplicación. En la Figura 13 se describen las demás vistas que pertenecen a este menú (vista de programas, vista de agenda, vista de favoritos), además de lo anterior, la barra de navegación contiene un reproductor que permite controlar el audio de la señal en vivo

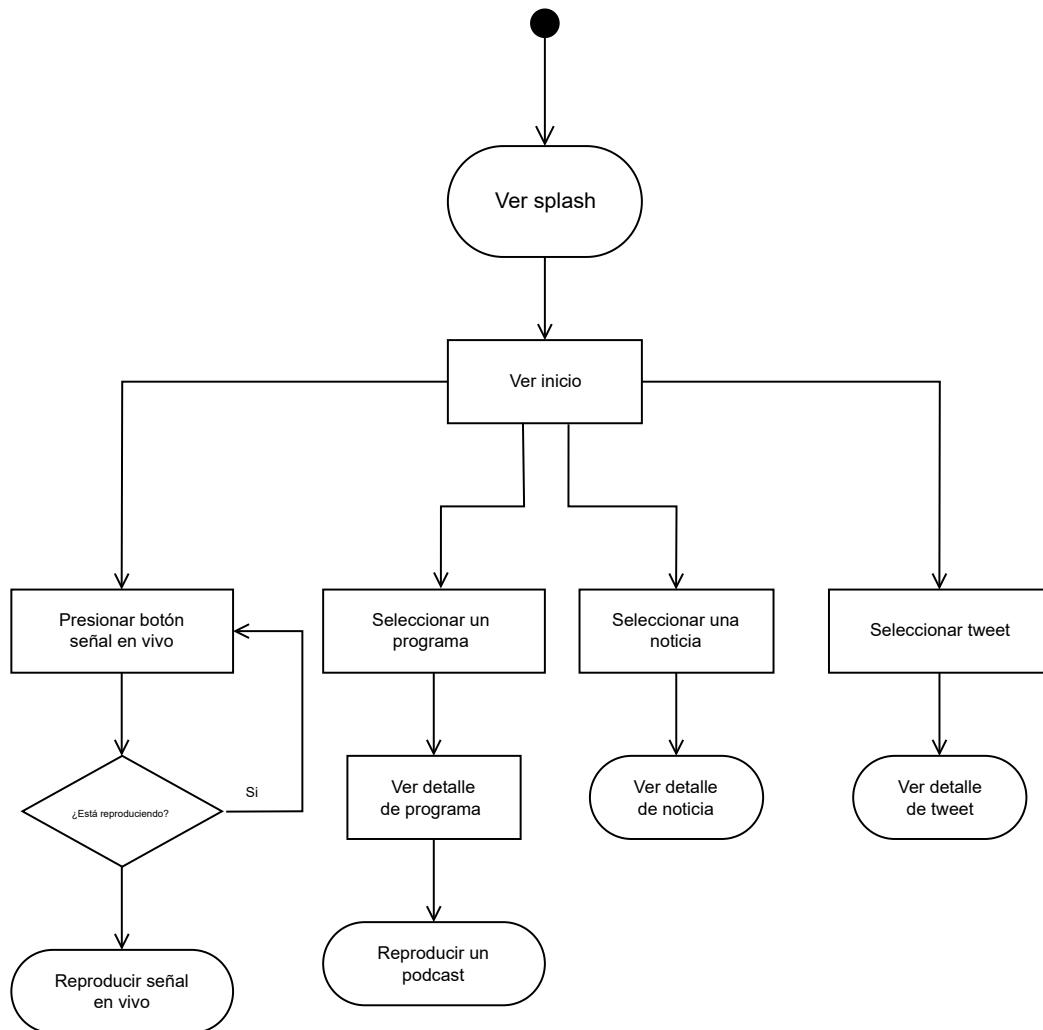


Figura 12: Diagrama de proceso sección de inicio

y de los *podcast*.

En la Figura 14 se visualiza los procesos que se llevan a cabo por parte del usuario en el navegador lateral. En éste se encuentran las funcionalidades secundarias de la aplicación, tales como: visualizar la parrilla de programación de la emisora; contactar mediante un correo electrónico o una llamada telefónica; visualizar información mas detallada sobre su identidad e historia; y visitar tanto su portal web como sus redes sociales.

### 3.3.4 Vista de Desarrollo

La vista de desarrollo describe la organización de los diferentes módulos del sistema, también llamados artefactos, considerando de manera estática su conexión y comunicación.

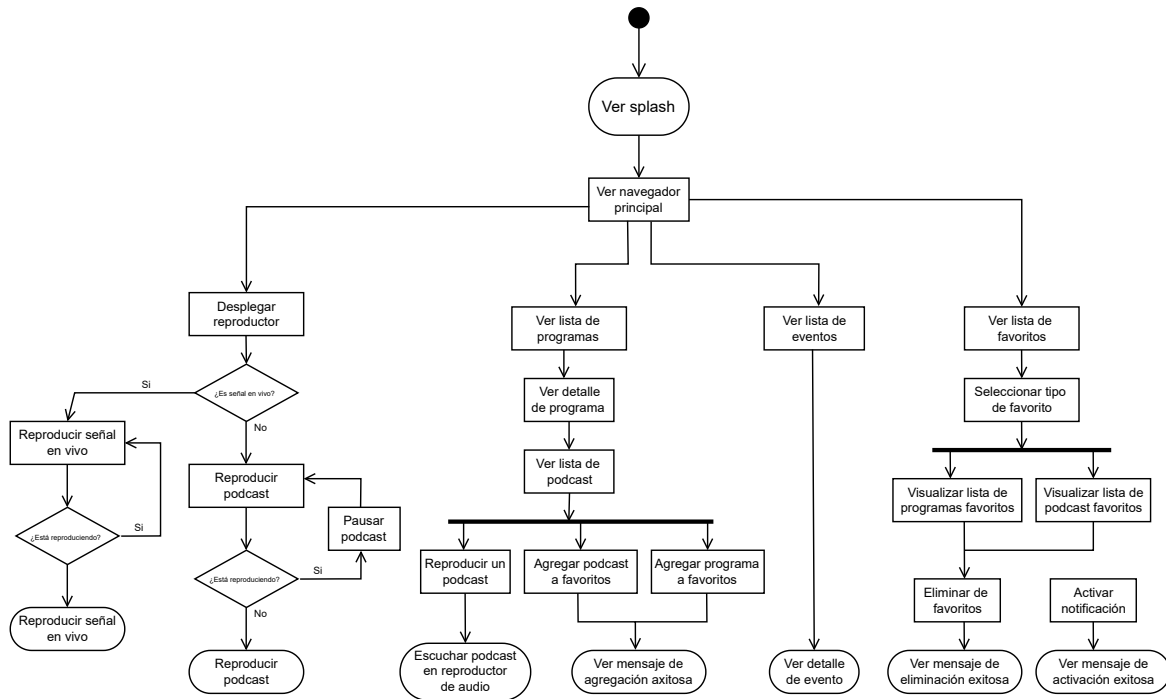


Figura 13: Diagrama de proceso navegador principal

Para implementar esta vista se usa un diagrama de componentes el cual se presenta en la Figura 15.

### 3.3.5 Vista Física

La vista física describe el sistema desde un punto de vista ingenieril y representa la topología de los componentes a nivel físico. Dicho de otra manera, representa la arquitectura de un sistema en tiempo de despliegue. El diagrama de despliegue se emplea para describir esta vista.

En la Figura 16 se presenta la configuración para el ecosistema de Unicauca Estéreo. Por un lado, se puede observar los componentes que representan los servidores asociados a los servicios del sistema, los cuales incluyen un servidor de aplicaciones dentro del cual se ejecuta el sistema de gestión de contenido y un servidor de base de datos. Por otra parte, se puede observar los componentes que soportan la ejecución de los sistemas cliente descritos como un computador y un dispositivo móvil.

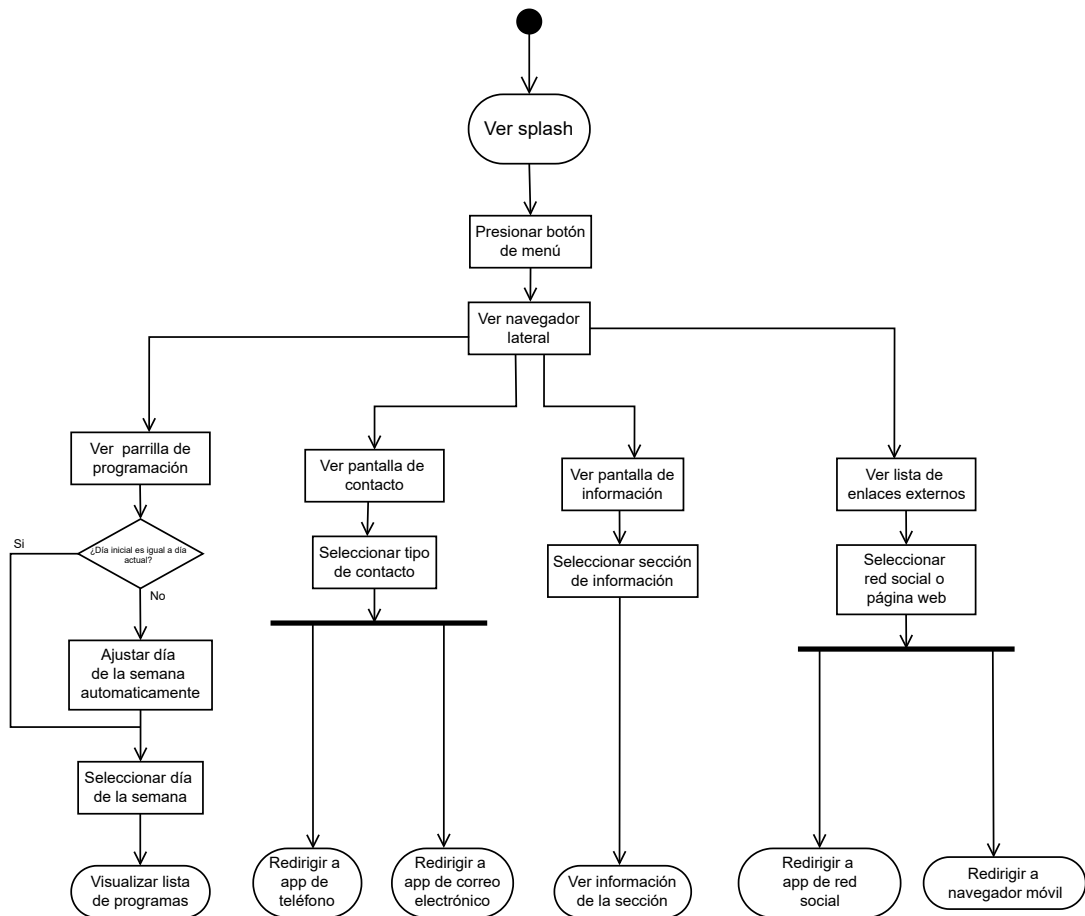


Figura 14: Diagrama de proceso navegador lateral

### 3.4 DIAGRAMA DE FLUJO DE NAVEGACIÓN

Con el fin de presentar una visión general de la aplicación y específicamente la relación entre las diferentes pantallas que la componen, se presenta el diagrama de flujo de navegación en la Figura 17 el cual se detalla a continuación.

Como primera instancia, una vez el usuario abre la aplicación por primera vez en el ciclo de ejecución del dispositivo móvil, se presenta la pantalla introductoria denominada comúnmente como “*Splash*”. Posteriormente, se da continuidad a las pantallas principales de la aplicación. De manera general, la aplicación móvil Unicauca Estéreo cuenta con un navegador principal, en el cual el usuario puede navegar entre las secciones de mayor prioridad: inicio de la aplicación; sección de programas radiales denominado “*Podcast*”; sección de agenda universitaria; y sección de favoritos. Además y de manera adyacente es



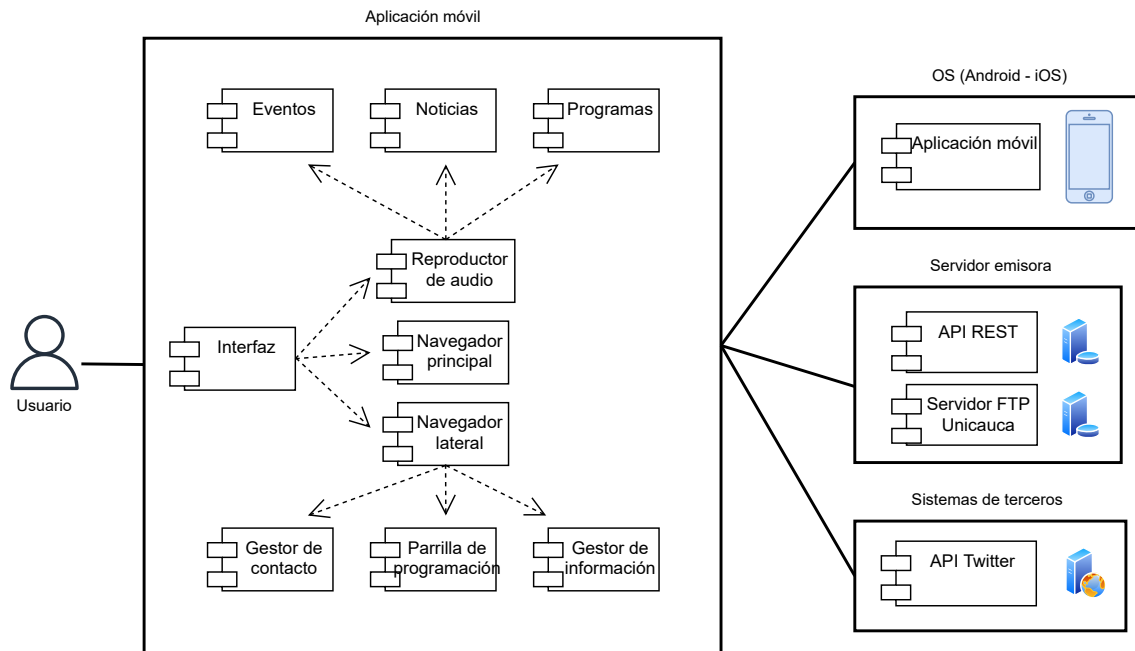


Figura 15: Diagrama de componentes aplicación móvil Unicauca Estéreo

posible acceder al reproductor multimedia de la aplicación desde el navegador principal.

La sección de inicio de la aplicación funciona a nivel de navegación como un enlace a diferentes pantallas que pertenecen a otros módulos dentro de la aplicación. Desde esta sección es posible navegar en ambos sentidos hacia las siguientes pantallas: detalle de *podcast*; detalle de noticia; lista de Tweets de la emisora; y detalle de un Tweet. En la sección de programas radiales (*podcast*) es posible navegar en ambos sentidos hacia la pantalla de detalle de programa. En la sección de agenda universitaria es posible navegar en ambos sentidos a la pantalla de detalle de un evento publicado por la emisora. En la sección de favoritos, si bien no es posible realizar una navegación a otra pantalla de la aplicación, se cuenta con un menú interno de dos opciones que le permiten al usuario cambiar entre los módulos de programas favoritos y capítulos favoritos.

Finalmente, el menú principal cuenta con la opción de ir hacia otro menú alternativo, descrito en la Figura 17 como menú lateral. Dicho menú permite al usuario navegar hacia las funcionalidades de menor prioridad: parrilla de programación, e información sobre la universidad y contacto. En la sección de parrilla de programación es posible navegar en ambos sentidos hacia la pantalla de detalle de programa.

Es importante destacar que los módulos descritos en la Figura 17 que cuentan con un icono alusivo a una base de datos, denotan las pantallas que requieren una consulta a

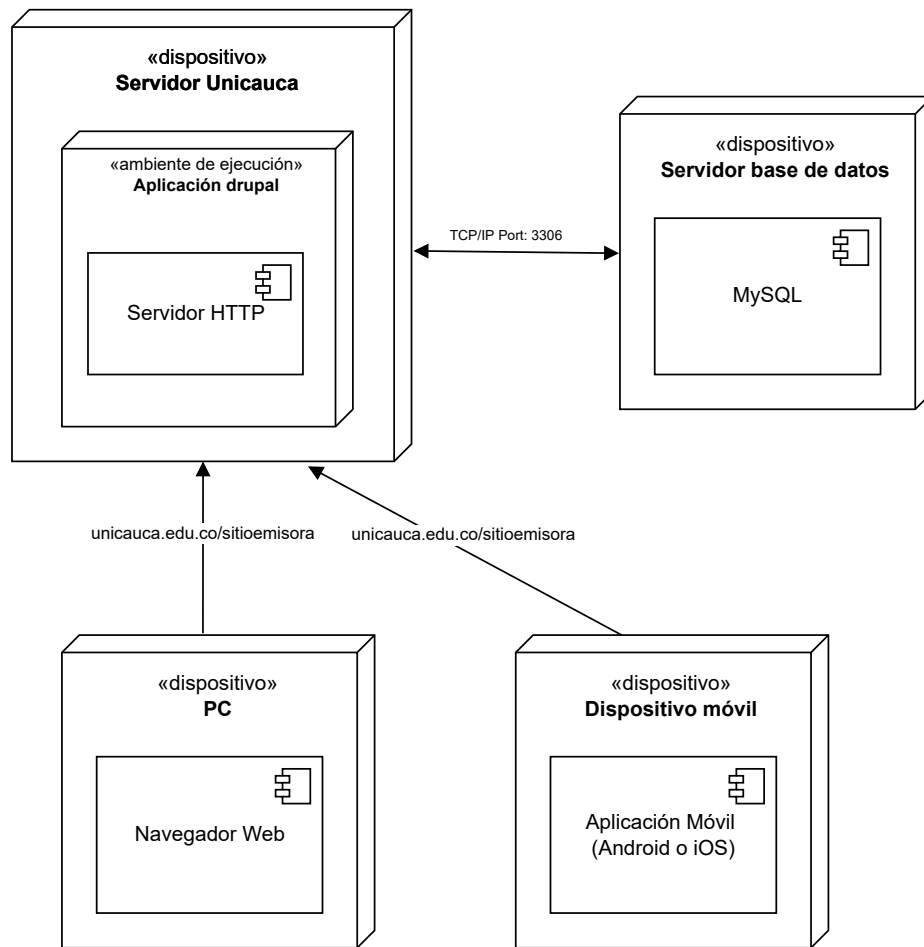


Figura 16: Diagrama de despliegue aplicación móvil Unicauca Estéreo

los registros almacenados en la base de datos del servidor de la emisora, mientras que los módulos que cuentan con un icono alusivo a la nube, hacen referencia a aquellas pantallas en las cuales se realiza una consulta de un servicio, ya sea externo como en el caso de lista de *tweets* y detalle de un *tweet* donde se consume la API de Twitter, o interno como en el caso del reproductor, en el cual se consume el servicio de *streaming* en vivo de la emisora.

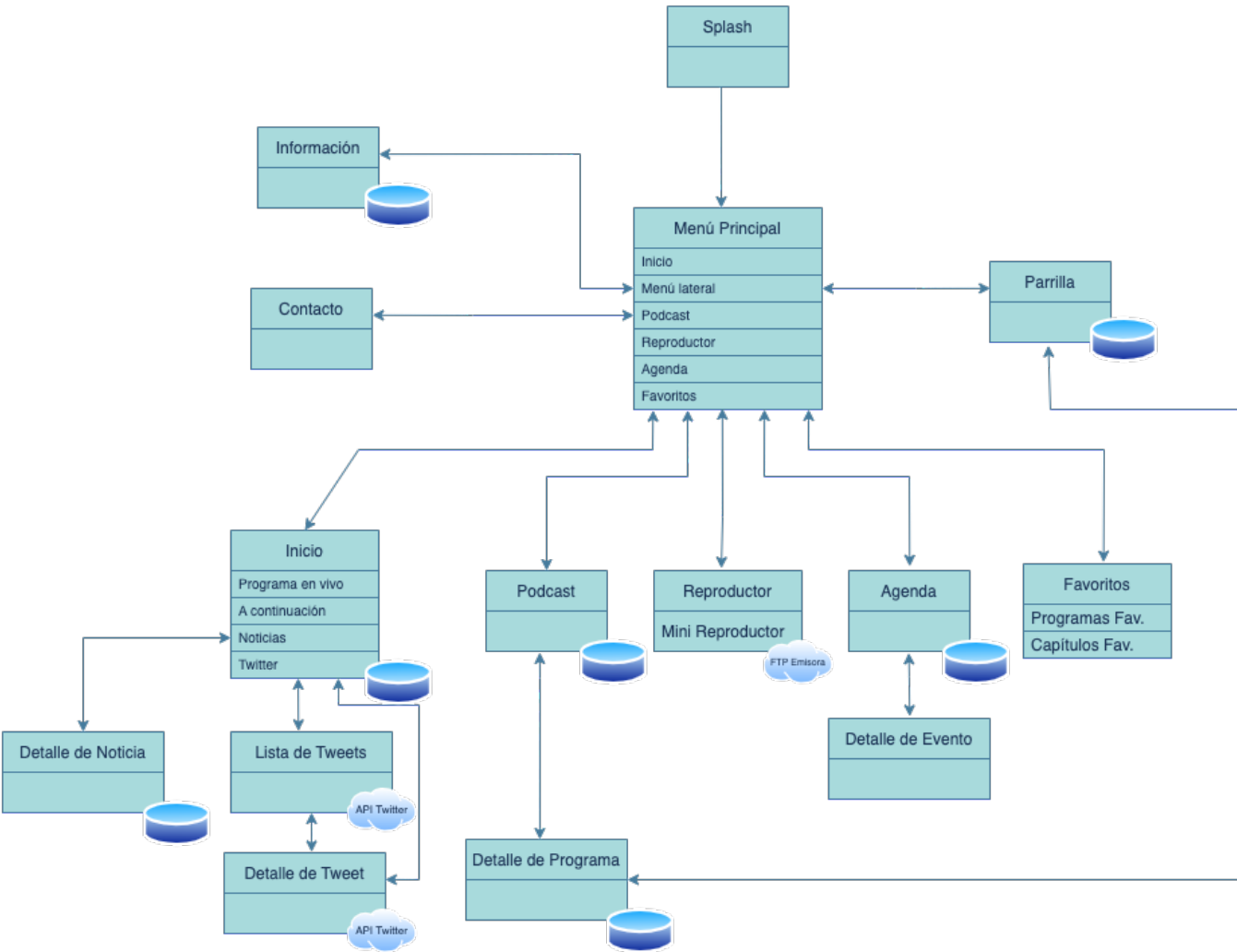


Figura 17: Diagrama de flujo de navegación aplicación móvil Unicauca Estéreo

## 3.5 DISEÑO DE INTERFACES GRÁFICAS

Las interfaces gráficas del sistema han sido diseñadas con el fin de cumplir con los requerimientos funcionales descritos en la sección 2.3, pero además tomando como referencia los criterios de usabilidad descritos en la sección 2.4. El objetivo general del diseño de las interfaces de la aplicación móvil Unicauca Estéreo es brindar una experiencia intuitiva, simple y que a su vez continúe con la línea de colores propuesta por la emisora universitaria. El diseño de las diferentes pantallas se realizó usando la herramienta Adobe Illustrator [40].

A continuación, se presentan las interfaces gráficas de la aplicación móvil Unicauca Estéreo, en las cuales se realiza un análisis del proceso de diseño y las funciones que tienen cada uno de los componentes de las mismas.

### 3.5.1 Vista Splash

Esta es la vista que se presenta al usuario al iniciar la aplicación móvil, la cual consta de una imagen centrada correspondiente al logotipo de Unicauca Estéreo. El propósito general de esta pantalla es permitir al sistema preparar los datos necesarios para que la aplicación funcione correctamente. La emisora es responsable de proveer los respectivos archivos que contienen la línea gráfica para ser aplicados a lo largo del proyecto.



Figura 18: Vista Splash aplicación móvil Unicauca Estéreo

### 3.5.2 Menú Principal

De acuerdo a los requerimientos planteados en la sección 2.3, se definió en conjunto con la emisora que las funcionalidades principales que la aplicación debe proporcionar a los usuarios son la reproducción de audio, seguido por una sección donde los usuarios puedan visualizar los programas de la aplicación y la agenda de eventos universitarios. Para cumplir con lo anterior y teniendo como referencia la heurística H7, se hizo necesario definir una estructura de navegación principal para el usuario, de tal manera que le permita de una manera intuitiva acceder a las funcionalidades descritas anteriormente. Se definió entonces un navegador tipo *BottomTabNavigator* como se presenta en la Figura 19. Adicionalmente, se decidió incluir dentro del navegador una sección de inicio que le permita al usuario de manera rápida acceder a diferentes contenidos de la aplicación y una sección para el sistema de favoritos. En la Figura 19 se puede observar de igual manera la barra de reproducción de audio encima del navegador, la cual permite al usuario tener un control en todo momento sobre las acciones relacionadas con el audio de la señal en vivo y los *podcast* de la emisora.



Figura 19: Diseño de menú principal aplicación móvil Unicauca Estéreo

### 3.5.3 Reproductor de Audio

La reproducción de audio, siendo la funcionalidad principal de la aplicación, debe estar soportada por una interfaz que permita al usuario tener un control total sobre las diferentes acciones. Para esto se realiza el diseño del reproductor de audio con dos componentes: el primero es una barra de reproducción, la cual está posicionada junto al navegador principal de la aplicación, y permite al usuario visualizar la información del estado actual de reproducción y realizar acciones como parar la señal en vivo, pausar, reproducir y agregar a favoritos un *podcast*. Este componente debe permitir además desplegar un segundo componente denominado como reproductor principal, el cual se puede visualizar en la Figura 69, y consta de una vista superpuesta que ocupa toda la pantalla y permite realizar todas las acciones de reproducción sobre el audio de la aplicación, y por lo tanto debe permitir parar y reproducir la reproducción, y para el caso de un *podcast* debe permitir adelantar y atrasar el audio y tener una barra de progreso del mismo.



Figura 20: Diseño de reproductor de audio.

### 3.5.4 Sección de Inicio

La vista de inicio de la aplicación se presenta en la Figura 57 y muestra al usuario: el programa que se encuentra actualmente en emisión en la señal en vivo; una sección que lista los programas que vienen a continuación del programa actual; una sección en la cual se listan las últimas noticias publicadas por la emisora; y una sección que lista las últimas publicaciones realizadas por la emisora en su cuenta de Twitter.

### 3.5.5 Sección de Programas

A la sección de programas se llega desde el menú principal y permite a los usuarios visualizar la lista de programas de la emisora como se presenta en la Figura 64. Al presionar en un programa específico se va a la vista de detalle del programa, la cual presenta la lista de *podcast* que se encuentran publicados para ser reproducidos por el usuario, además de esto, permite agregar a favoritos tanto el programa como cada uno de los *podcast* de manera individual.



Figura 21: Diseño sección de inicio.

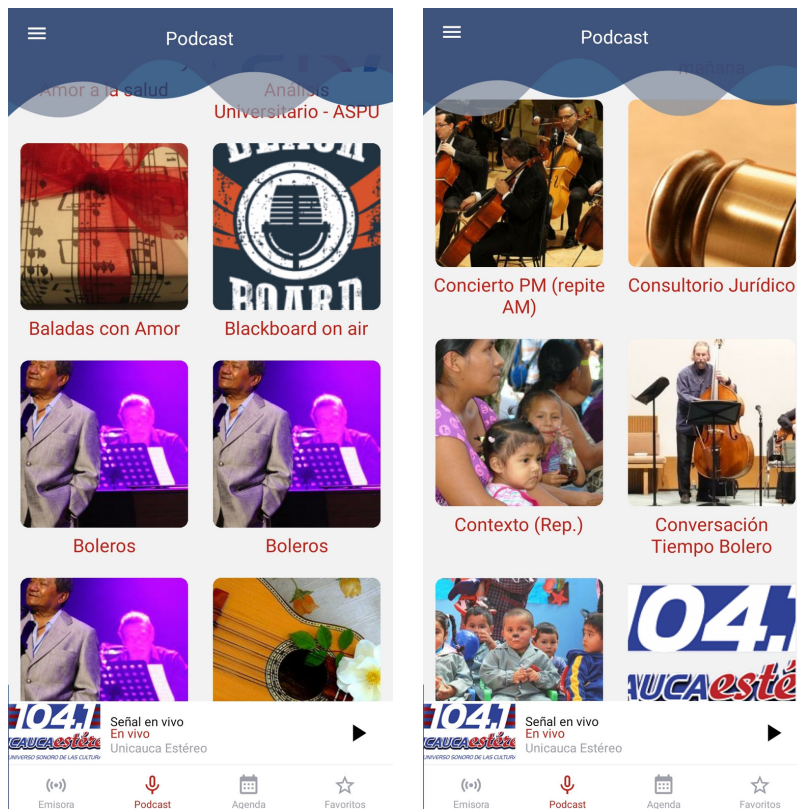


Figura 22: Diseño sección de programas.

### 3.5.6 Sección de Agenda

La sección de agenda se presenta en la Figura 65. Esta permite al usuario visualizar la lista de eventos publicados por la emisora y por la Universidad en orden cronológico y descendente. Al seleccionar un evento de la lista, la aplicación va hacia la vista en detalle, en la cual se presenta la información completa del evento.

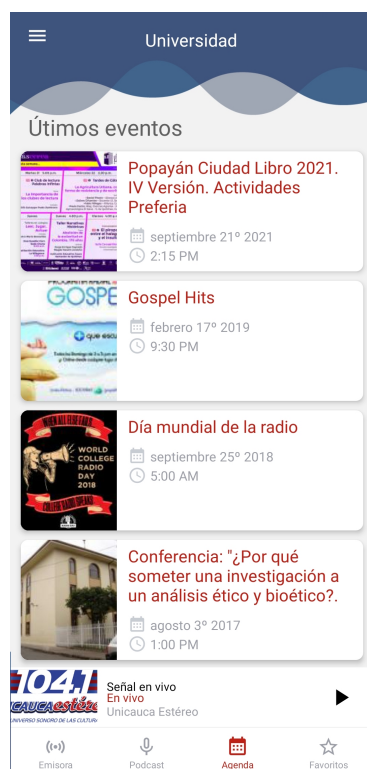


Figura 23: Diseño sección de agenda.

### 3.5.7 Sección de Favoritos

El sistema de favoritos surge de la necesidad de permitir al usuario personalizar la aplicación durante su uso. De acuerdo a la heurística H8, una aplicación altamente usable debe permitir al usuario realizar configuraciones propias y tener accesos directos con el fin de completar tareas de una manera más eficiente. Como se describe en la sección 2.4, la aplicación debe permitir agregar a una lista personalizada de favoritos tanto programas como *podcast* individuales. Por lo anterior, esta sección se divide en dos módulos principales.

Como primera instancia, la lista de programas favoritos se presenta en la Figura C.15. El usuario visualizará los programas marcados como favoritos y cuenta con la posibilidad de activar un recordatorio que genera una notificación en el momento que el programa en cuestión inicie, dentro de la señal en vivo de la emisora. Por otro lado está la lista de



podcast favoritos como se presenta en la Figura 68. Desde esta lista, el usuario visualiza los diferentes capítulos de programas que haya seleccionado como favoritos, con el fin de reproducirlos en el momento deseado por el usuario, sin tener que realizar todo el flujo de navegación desde la sección de programas.

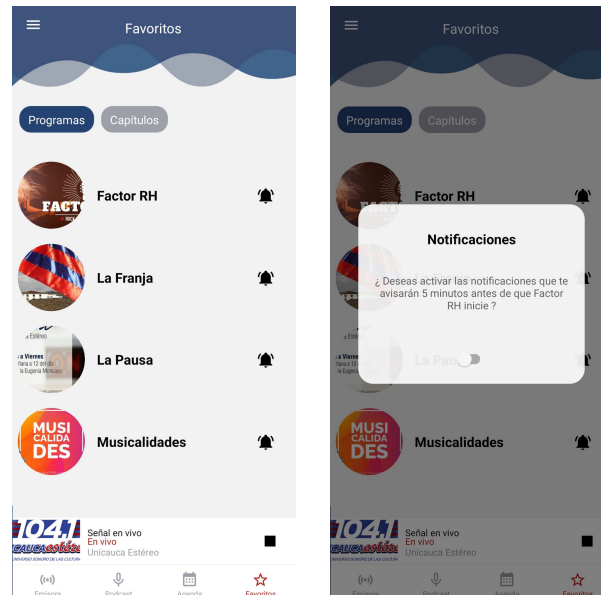


Figura 24: Diseño de sección de programas favoritos.

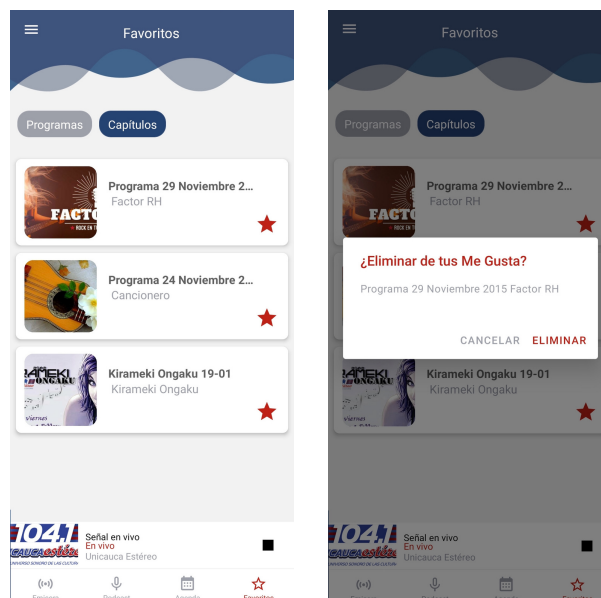


Figura 25: Diseño de sección de podcast favoritos.

### 3.5.8 Menú Lateral

Con el fin de ubicar las funcionalidades consideradas como secundarias dentro de la aplicación móvil y tomando como referencia las heurísticas H4 y H7, las cuales sugieren que cada vista debe centrarse en una funcionalidad y que se debe contar con un camino claro, se decidió realizar el diseño de un menú lateral tipo “Drawer”, tal como se observa en la Figura 59. En este menú se incluye además del acceso al inicio de la aplicación, el cual representa el ecosistema del menú principal, el acceso a la parrilla informativa, contacto con la emisora, información relacionada con la emisora y el enlace directo con su página web y redes sociales.

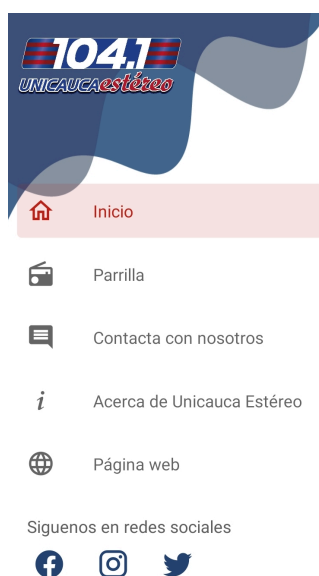


Figura 26: Diseño del menú lateral

### 3.5.9 Sección de Parrilla de Programación

La sección de parrilla, presentada en la Figura 60, permite al usuario visualizar la programación de la emisora Unicauca Estéreo para cada uno de los días de la semana. Esta sección permite seleccionar un día en específico, del cual se despliega una lista que incluya el nombre, la franja horaria y una imagen de cada uno de los programas.

Un aspecto importante a tener en cuenta es que tomando como referencia la heurística H2, la cual expresa la relevancia que tiene presentar términos coherentes y acordes a la realidad de los usuarios, se definió como una funcionalidad adicional que, cuando el usuario ingrese por primera vez a la sección de parrilla de programación, el sistema presente de manera automática el día que corresponde con el día actual.

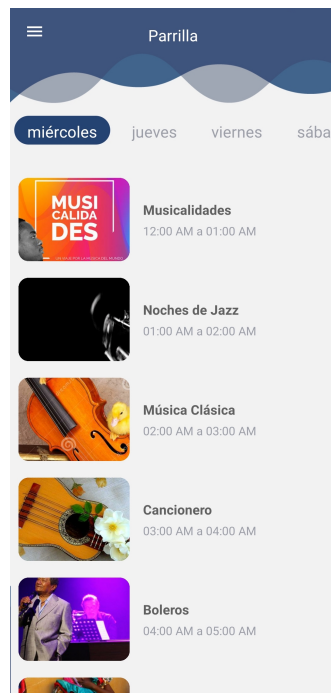


Figura 27: Diseño de sección de parrilla de programación.

### 3.5.10 Sección de Contacto

La sección de contacto, cuya vista se presenta en la Figura 62, permite al usuario realizar un contacto con la emisora mediante una llamada telefónica o un correo electrónico. Para establecer el contacto, la aplicación dispone dos componentes que al presionarlos cierran el flujo de navegación de la sección y realiza la llamada a una aplicación externa del dispositivo móvil, lo cual permitirá realizar una llamada o enviar un correo electrónico según sea el caso.

### 3.5.11 Sección de Información sobre la Emisora

En la sección de información de la emisora, la cual se presenta en la Figura 29, el usuario encontrará información relacionada con la emisora Unicauca Estéreo, la cual incluye: ¿Que es la emisora?; la historia desde su creación; y el proceso llevado a cabo para la regionalización. Estos contenidos son un reflejo de la información a la cual se puede acceder desde el portal web. Cada apartado cuenta con un componente que le permita al usuario desplegar el texto, lo anterior con el fin de brindarle al usuario la posibilidad de ocultar de su vista contenido que no esté consumiendo y para contar con un diseño visualmente agradable de acuerdo a H5.

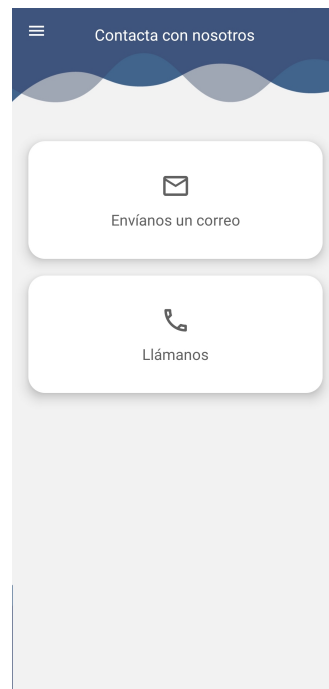


Figura 28: Diseño de sección de contacto con la emisora.

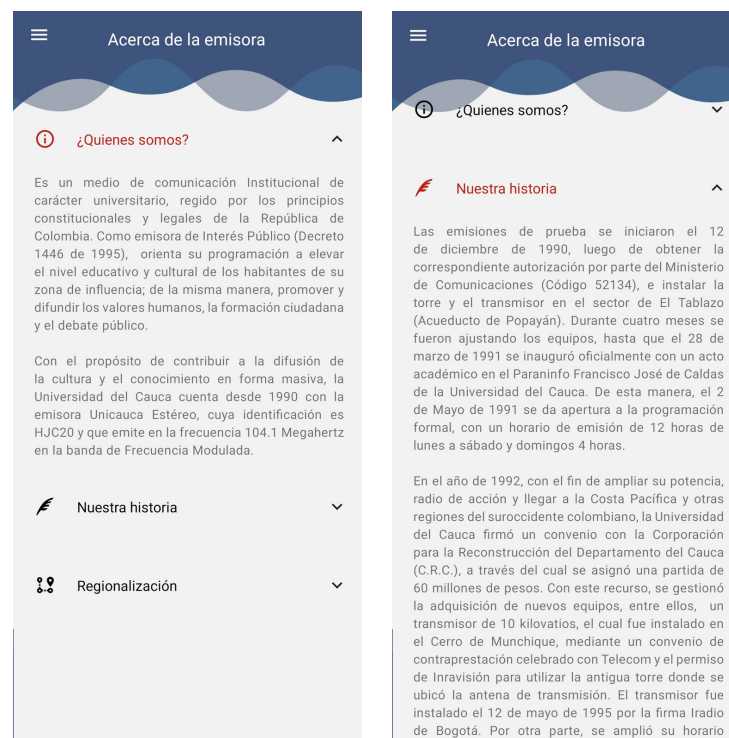


Figura 29: Diseño de sección de historia e información.

## 4 IMPLEMENTACIÓN

Una vez definido el diseño a nivel funcional y de interfaces gráficas de la aplicación móvil Unicauca Estéreo, se procede con su implementación y todos los aspectos que esto implica. En este capítulo se presentan las actividades realizadas en el proceso de implementación de la aplicación.

En la Sección 4.1 se presenta la lista de herramientas que han sido utilizadas en el desarrollo de la aplicación móvil de Unicauca Estéreo, con sus respectivas características técnicas y su descripción. En la Sección 4.2 se realiza la descripción del proceso de configuración del servidor del sistema de información de la emisora, para dar soporte a las consultas realizadas desde la aplicación móvil. En la Sección 4.3 se realiza una descripción del proceso de implementación de la aplicación móvil, haciendo énfasis en los diferentes módulos y librerías que brindan las funcionalidades para satisfacer los requerimientos, con una descripción detallada de su comportamiento a nivel lógico.

### 4.1 HERRAMIENTAS DE DESARROLLO

En esta sección se presentan las herramientas que han sido utilizadas en el desarrollo de la aplicación móvil de Unicauca Estéreo.

#### 4.1.1 React Native

React Native es un *framework* multiplataforma de código abierto basado en React usado para construir aplicaciones nativas con *JavaScript* [41]. Fue creado y es apoyado por Facebook, Inc. React Native permite al desarrollador crear aplicaciones nativas que se ejecutan en los sistemas operativos Android e iOS. React Native utiliza su propio lenguaje llamado JSX, el cual es una extensión de la sintaxis clásica XML que combina algunas de las funcionalidades de ECMAScript con el lenguaje web HTML sin ninguna semántica definida [42]. Normalmente el código de ECMAScript se transforma a *JavaScript* mediante herramientas como Babel<sup>4</sup>. Con JSX y React Native, el desarrollador utiliza componentes de interfaz de usuario declarativos que se compilan como elementos nativos en cada plataforma específica.

En la Figura 30 se observa el funcionamiento general de React Native. Los componentes programados son escritos usando el lenguaje JSX y es mediante el compilador Metro Bundler que el código creado en JavaScript se transforma en elementos o módulos

---

<sup>4</sup>Babel es un transcompilador de JavaScript gratuito y de código abierto que se utiliza para convertir código de versiones nuevas de JavaScript (ECMAScript 2015+) en una versión compatible con versiones anteriores de motores de compilación de código [43]

nativos que pueden ser entendidos y ejecutados tanto en dispositivos móviles Android como iOS. Los módulos generados por React Native tienen un comportamiento y un desempeño similar a los que se generan mediante los lenguajes de programación nativos de cada plataforma.

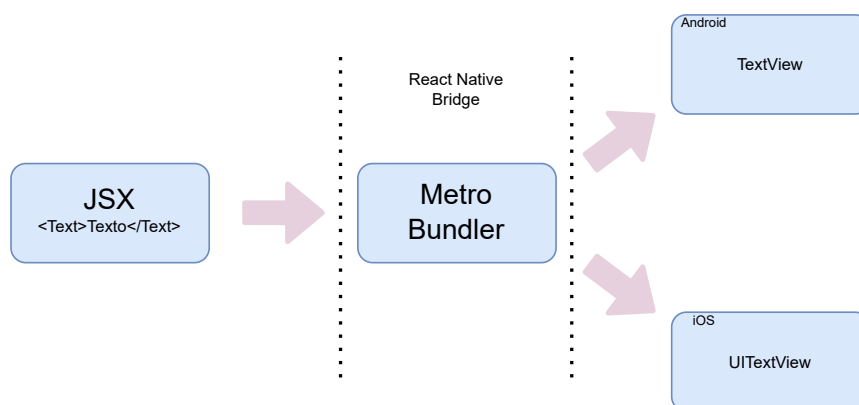


Figura 30: Funcionamiento de compilador en React Native

Una de las características más destacadas de React Native se conoce como Hot reloading, la cual es una forma de recargar la aplicación durante el desarrollo de manera rápida, conservando el estado actual de las variables y componentes [44]. Lo anterior es una característica que beneficia a los desarrolladores, ya que les permite una interacción mucho más eficaz con la aplicación desarrollada y una visualización de los cambios en el código más rápida, lo que a su vez se traduce en un ciclo de producción y compilación más rápido. Anteriormente, se usaba el *Live reloading* para ejecutar cambios y compilación en el código para React Native, lo cual permitía una recarga de la aplicación similarmente rápida, pero sin conservar el estado de la aplicación. Esto generaba en cada refresco que la aplicación volviera a su estado de flujo inicial o de operación inicial. En contraposición, *Hot reloading* permite inyectar nuevas versiones de los archivos editados en la aplicación durante su tiempo de ejecución.

#### 4.1.2 Redux

Redux es un contenedor predecible para el estado de aplicaciones en JavaScript que se usa para el manejo de la información, el cual permite almacenar en una única fuente toda la información de una aplicación en formato de objeto denominado estado, acceder a el y editarlo [43]. Redux permite además saber cómo se encuentra una determinada variable de estado y en el caso de realizarse un cambio de componente en la aplicación, permite

conocer el cambio realizado y la forma en que se hizo. Esta herramienta tiene múltiples ventajas en su aplicación a proyectos de software [45], a continuación se presentan las más relevantes:

- **Previsible:** Redux aporta en la creación de aplicaciones consistentes, se ejecutan en diferentes entornos (cliente, servidor y nativo) y son fáciles de evaluar.
- **Centralizado:** El objetivo de Redux es centralizar el estado y la lógica de una aplicación, lo cual brinda potentes capacidades como deshacer/rehacer y persistencia del estado, entre otras.
- **Depurable:** Redux cuenta con un conjunto de herramientas que permiten realizar trazabilidad de los cambios del estado de la aplicación, permitiendo reducir el tiempo para encontrar y reportar errores.
- **Flexibilidad:** Redux funciona con cualquier capa de Interfaz de Usuario (UI, *User Interface*) y cuenta con un amplio ecosistema de funciones y métodos que permiten adaptar su patrón de cambios de estado de diferentes formas.

En la Figura 31 se pueden visualizar el flujo que detalla el funcionamiento de Redux [46]. En primer lugar, la *store* se encarga de guardar la información que proviene de diferentes fuentes, como por ejemplo, una petición a un servidor generadas por el usuario o de una base de datos interna, lo que se conoce como un súper objeto, lo cual permite entregar datos a las interfaces o *views* de la aplicación, las cuales tienen como función desplegar en pantalla los datos del *store*. En el momento en que se requiera realizar una actualización de los datos almacenados en el *store*, las *views* deben comunicarse con el *store* iniciando una *action*, las cuales son funciones preestablecidas que sirven como un medio para la ejecución de acciones sobre el *store*, al mismo tiempo, estas *actions* son controladas mediante lo que se conoce como *reducers*, las cuales son funciones puras que se encargan de realizar los cambios del estado almacenado en el *store*. Una vez el *store* es actualizado, todos estos cambios se ven reflejados en las *views*.

### 4.1.3 Git

Git es un sistema de control de versiones que fue creado por Linus Torvalds en el año 2005, el cual permite controlar los avances o modificaciones de un proyecto de software [47]. Con Git se puede navegar a través de la línea de tiempo de los cambios del proyecto para visualizar, editar o corregir diferentes archivos del proyecto. Además, Git permite manejar los cambios realizados por varias personas de manera que estas puedan trabajar en un mismo proyecto sin afectarse entre sí. En el caso de presentarse errores, se puede

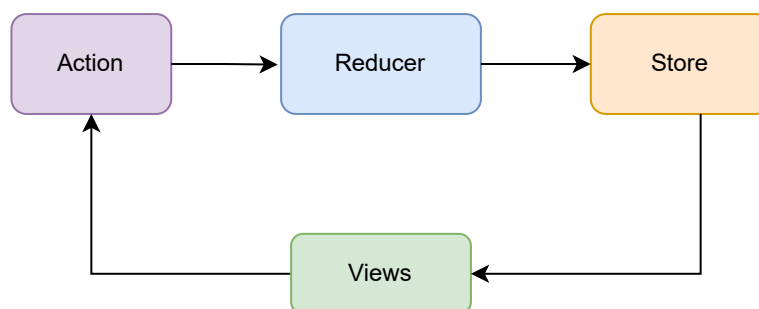


Figura 31: Flujo de Redux

saber de manera precisa quién realizó ese cambio. Si existe algo en una versión vieja que se desea recuperar, es posible hacerlo de manera precisa.

Si bien Git debe funcionar en una máquina local, es posible almacenar los proyectos en la web mediante una plataforma de control de versiones remota como GitHub o GitLab, lo cual permite colaborar con otras personas en un mismo proyecto, usar una interfaz web para revisión de cambios o publicar proyectos en la nube.

La característica de Git que realmente lo distingue de casi todos los demás Gestores de Código Fuente (SCM, *Source Code Manager*) es su modelo de ramas. Git permite contar con múltiples ramas locales que pueden ser totalmente independientes entre sí. La creación, fusión y eliminación de líneas de desarrollo se realiza en segundos, lo cual significa que se pueden hacer cosas tales como [48]:

- **Cambio de contexto sin fricciones:** Crear una rama para probar una idea, generar algunos cambios, volver al lugar desde el que se ramificó, aplicar un parche en el código, y volver al lugar donde se estaba experimentando e integrarlo.
- **Ramas basadas en roles:** Tener una rama que siempre contenga sólo lo que va a producción, otra en la que se fusione el trabajo para las pruebas, y varias ramas más pequeñas para el trabajo diario, como se presenta en el ejemplo de la Figura 32.
- **Flujo de trabajo basado en características:** Crear nuevas ramas para cada nueva función en la que esté trabajando, de modo que se pueda alternar entre ellas sin problemas, y luego eliminar cada rama cuando esa función se fusione con su rama principal.



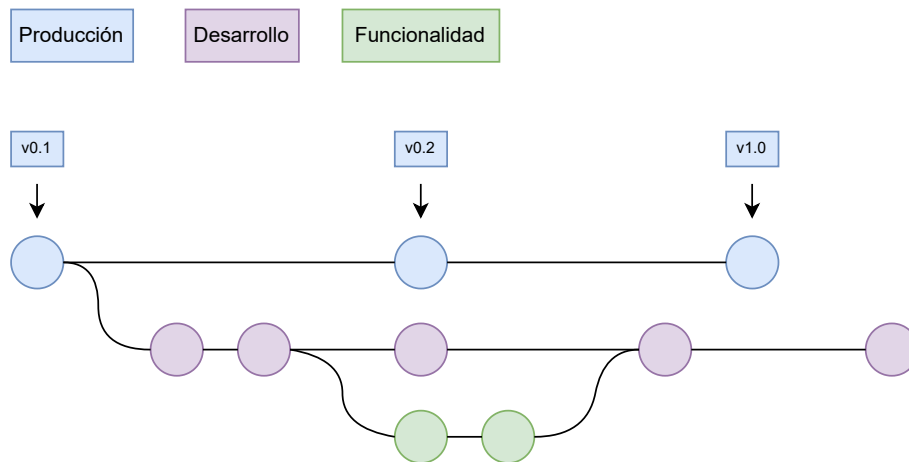


Figura 32: Flujo de de ramificación de Git

#### 4.1.4 Visual Studio Code

Visual Studio Code (VSCode) es un editor de código que cuenta con las herramientas necesarias para realizar desarrollo de aplicaciones en todo tipo de lenguajes de programación, pero que principalmente está orientado al desarrollo de aplicaciones con JavaScript. A diferencia de un Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*), los editores de código son programas mucho más sencillos, lo cual hace que ocupe un menor espacio en memoria y que su curva de aprendizaje sea mucho mayor. Esta herramienta fue seleccionada para realizar la construcción del código y la compilación para la aplicación móvil Unicauca Estéreo. A continuación se presentan sus principales características [49]:

- Permite depurar el código directamente desde el editor. Es posible iniciar o adjuntar aplicaciones o extensiones en ejecución y depurar con puntos de interrupción, pilas de llamadas y una consola interactiva.
- Es posible trabajar con Git y otros proveedores de SCM para revisión de diferencias de código, preparar archivos y hacer *Commits* directamente desde el editor. Ejecutar comandos *Push* y *Pull* de cualquier servicio SCM alojado.
- En el caso de necesitar más funcionalidades, VSCode permite instalar extensiones para añadir nuevos idiomas, temas, depuradores y para conectarse a servicios adicionales. Las extensiones se ejecutan en procesos separados, lo que garantiza que no afectará la ejecución del editor.
- Con Microsoft Azure es posible desplegar y alojar sitios de React, Angular, Vue,

Node, y Python, entre otros; almacenar y consultar datos relacionales y basados en documentos; y escalar con computación *serverless* con relativa facilidad.

#### 4.1.5 Drupal

Drupal es un sistema de gestión de contenidos gratuito usado generalmente para crear aplicaciones web de una manera interactiva, sencilla y sin necesariamente tener conocimientos en desarrollo de software [50]. La funcionalidad de Drupal se puede observar de manera simplificada en la Figura 33. Básicamente se basa en un sistema de gestión de módulos y plantillas, las cuales se integran sobre el núcleo o *core* de Drupal y se comunican con este mediante diferentes API. Estas últimas permiten implementar diferentes funcionalidades como crear formularios, desplegar menús, visualizar contenido multimedia o interactuar con migraciones desde otras plataformas. Se presentan a continuación algunas de las características más destacadas de Drupal:

- Alta escalabilidad.
- Desarrollo orientado a dispositivos móviles.
- Soporte en seguridad.
- Integración con otras plataformas.
- Edición dinámica de contenido.
- Arquitectura de contenido flexible.
- Cuenta con un sistema de soporte para múltiples idiomas.

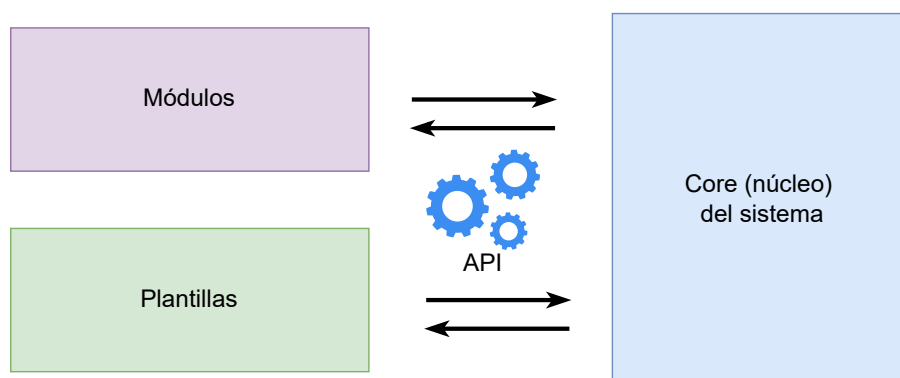


Figura 33: Arquitectura de construcción modular de Drupal

## 4.2 CONFIGURACIÓN SERVIDOR UNICAUCA ESTÉREO

Con el fin de realizar la conexión de la aplicación móvil en desarrollo con el servidor de la emisora, y que de esta manera sea posible consumir y posteriormente desplegar en el dispositivo móvil el mismo contenido del portal web de Unicauca Estéreo, se hace necesario realizar ciertas configuraciones en el panel de administración de Drupal, sobre el cual está construida la página web. Desde este panel es posible manejar los contenidos que se despliegan desde las diferentes interfaces de la página web, así como subir nuevos contenidos, instalar nuevos módulos o crear y configurar pantallas nuevas.

En la Figura 34 se presenta la disposición del panel de Drupal, y como se puede observar en la imagen, el sistema de información de la emisora Unicauca Estéreo se encuentra dividido en diferentes tipos de contenido que representan las diferentes secciones que existen en el portal web. De la lista de tipos de contenido, el objetivo es que la aplicación móvil pueda tener acceso a los contenidos creados para: calendario programación radial, eventos, noticias, la emisora, *player* en vivo, *podcast* y programa radial.

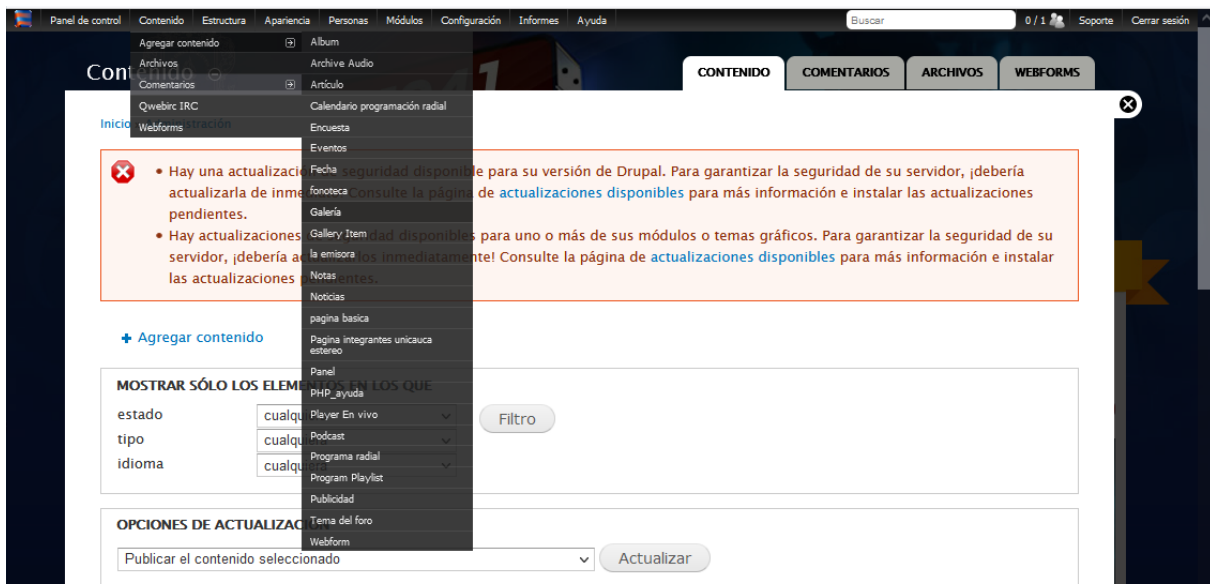


Figura 34: Portal de administración del portal web Unicauca Estéreo

Debido a lo anterior, fue necesario que el servidor de Drupal se comportara como un servidor *API REST* y para esto se instaló el módulo *Services* [51], el cual es compatible con la versión 7.21 de Drupal sobre la cual está construido el portal web y cuya configuración es sencilla y rápida. Para la instalación del módulo *Services* es necesario contar con dos módulos adicionales que complementan sus funcionalidades: *Tool Suite* y *Libraries* [52], [53]. El proceso de instalación consistió en la descarga desde la página oficial de Drupal,

de los archivos de instalación de los tres módulos en la versión necesaria e importarlos desde la consola de administración del portal web. Una vez finalizada la instalación se procedió con la configuración.

El proceso de configuración inició en la sección *Modules* del portal de administración, en la cual se activaron algunos submódulos de los módulos recientemente instalados: *Chaos tools* del módulo *Tool Suite*; *Libraries* del módulo *Libraries*; y finalmente *Services* y *REST Server* del módulo *services*. El paso siguiente del proceso fue la creación de la ruta a la cual la aplicación móvil debe realizar la peticiones. Esta configuración se realizó desde la sección *Structure/Services* del panel de administración donde, como se observa en la Figura 35, se deben agregar el nombre del servicio *REST* y la dirección de la ruta a la cual se apunta desde el cliente.

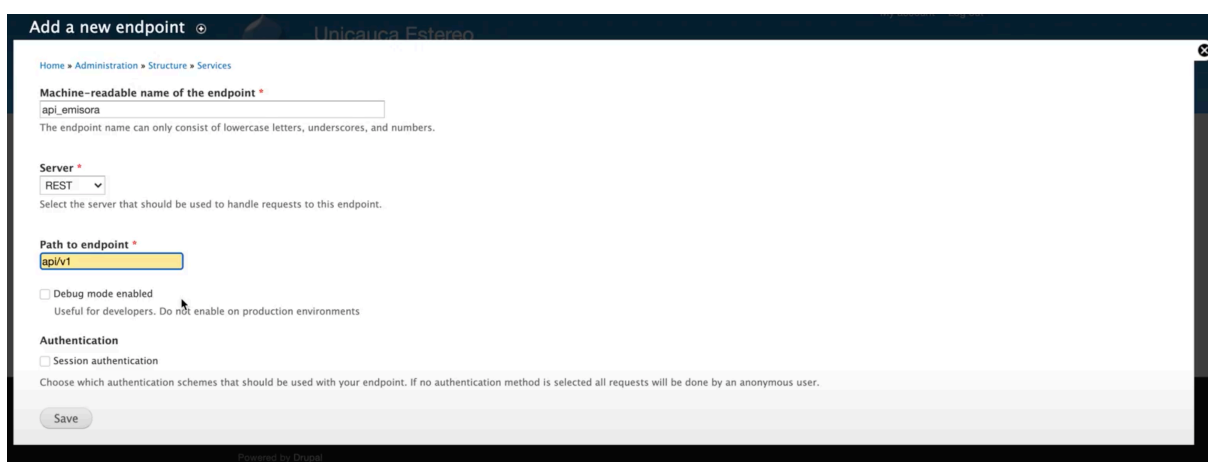


Figura 35: Panel de administración del módulo Services

Una vez finalizada la creación de la ruta para el servidor *REST*, el paso final del proceso de configuración fue activar los nodos que representan los tipos de contenidos a los cuales se requiere tener acceso por parte de la aplicación móvil, desde la lista como se presenta en la Figura 36, la cual se encuentra en el panel de configuración de la ruta creada anteriormente. De esta lista se deben seleccionar los siguientes nodos:

- *node-event*.
- *node-fonoteca*.
- *node-la-emisora*.
- *node-pagina-integrantes*.
- *node-player-en-vivo*.

- *node-podcast.*
  
- *node-publicidad.*
  
- *node-station-program.*

RESOURCE	OPCIONES	ALIAS
<input checked="" type="checkbox"/> ▶ node		<input type="text"/>
<input type="checkbox"/> ▶ node_article		<input type="text"/>
<input type="checkbox"/> ▶ node_date		<input type="text"/>
<input type="checkbox"/> ▶ node_event		<input type="text"/>
<input type="checkbox"/> ▶ node_field_noticias		<input type="text"/>
<input type="checkbox"/> ▶ node_fonoteca		<input type="text"/>
<input type="checkbox"/> ▶ node_forum		<input type="text"/>
<input type="checkbox"/> ▶ node_la_emisora		<input type="text"/>
<input type="checkbox"/> ▶ node_node_gallery_gallery		<input type="text"/>
<input type="checkbox"/> ▶ node_node_gallery_item		<input type="text"/>
<input type="checkbox"/> ▶ node_notas		<input type="text"/>
<input type="checkbox"/> ▶ node_page		<input type="text"/>
<input type="checkbox"/> ▶ node_pagina_integrantes		<input type="text"/>
<input type="checkbox"/> ▶ node_panel		<input type="text"/>
<input type="checkbox"/> ▶ node_php_ayuda		<input type="text"/>
<input type="checkbox"/> ▶ node_player_en_vivo		<input type="text"/>
<input type="checkbox"/> ▶ node_podcast		<input type="text"/>
<input type="checkbox"/> ▶ node_poll		<input type="text"/>
<input type="checkbox"/> ▶ node_publicidad		<input type="text"/>
<input type="checkbox"/> ▶ node_station_album		<input type="text"/>
<input type="checkbox"/> ▶ node_station_archive		<input type="text"/>
<input type="checkbox"/> ▶ node_station_playlist		<input type="text"/>
<input type="checkbox"/> ▶ node_station_program		<input type="text"/>
<input type="checkbox"/> ▶ node_station_schedule		<input type="text"/>
<input type="checkbox"/> ▶ node_webform		<input type="text"/>

Figura 36: Lista de recursos de la ruta de servidor API REST

## 4.3 IMPLEMENTACIÓN APLICACIÓN MÓVIL

### 4.3.1 Repositorio

Con el fin de almacenar el proyecto en la nube, llevar registro de los cambios realizados sobre los diferentes archivos durante el proceso de desarrollo y dar la posibilidad de que en trabajos futuros más desarrolladores tengan acceso al proyecto, se realizó la creación y configuración de un repositorio, en este caso se utilizó GitLab, el cual es un sistema basado en Git que permite llevar un control de versiones sobre un código fuente y alojado de manera remota en un servidor, además, permite ejecutar funciones que hacen parte del ciclo de *DevOps* (*Development Operations*) como Integración Continua (CI, *Continuous Integration*) y Despliegue Continuo (CD, *Continuous Delivery*).

En la Figura 37 se observa la interfaz de la consola de GitLab, desde la cual es posible clonar o descargar el proyecto y visualizar el historial de los cambios realizados a lo largo del proceso de desarrollo, por otra parte, se observa que el proyecto contó con una rama con nombre *develop* y otra con nombre *master*. En la primera se almacenaron todos los cambios realizados durante el desarrollo de la aplicación móvil, mientras que, en la segunda se almacenaron las versiones parciales o finales del sistema.

### 4.3.2 Instalación y Configuración del Entorno de Desarrollo

Para crear una aplicación móvil con React Native existen dos opciones principales: usar Expo<sup>5</sup> o crear un proyecto totalmente desde el inicio con el CLI que proporciona React Native. En este proyecto se definió implementar la segunda opción, debido a que un proyecto puro de React Native permitía muchas más libertades que usando Expo, y que en este último solo se puede implementar un número limitado de librerías o módulos secundarios, los cuales eran requeridos por la aplicación. Posteriormente, se realizó un proceso de instalación para cada uno de los sistemas operativos en los cuales se realizó el desarrollo, y para ello fue necesario en ambos casos tener previamente instalado el Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*)<sup>6</sup> para cada plataforma: Android Studio para Android y XCode para iOS, además, específicamente para Android se instaló y configuró el Java Development Kit, el cual funciona correctamente con Android Studio.

Para empezar con React Native, fue necesario instalar el editor de código Visual Studio Code, y luego Node JS, siendo este último un software de gestión de paquetes que permite al usuario descargar herramientas y librerías de desarrollo de software adicionales que

---

<sup>5</sup>Expo es una herramienta que se encarga de controlar la construcción de aplicaciones para iOS y Android programadas en React Native [54]

<sup>6</sup>Un IDE es una herramienta de software que proporciona un entorno de programación completo para los desarrolladores de software [55]

Name	Last commit	Last update
.vscode	Primer Commit	1 year ago
__tests__	Primer Commit	1 year ago
android	solucion animacion podcastDetail	6 months ago
ios	compartir en main player	9 months ago
src	base redux events, program grid y about us	6 months ago
.buckconfig	Primer Commit	1 year ago
.eslintrc.js	Primer Commit	1 year ago
.flowconfig	Primer Commit	1 year ago
.gitattributes	Primer Commit	1 year ago

Figura 37: Consola de administración de Gitlab

complementen las funcionalidades básicas que trae React Native por defecto. Una vez instalado Node JS, se navegó hacia la ruta en la cual se deseaba crear el proyecto desde una terminal de comandos y se ejecutó la siguiente instrucción `npm install -g create-react-native-app`. El comando anterior hace que se instale la librería `create-react-native-app` de manera global en el equipo. A continuación se creó el proyecto aplicación móvil Unicauca Estéreo en React Native ejecutando el siguiente comando:

*create-react-native-app <Nombre Del Proyecto>*

Esta librería se encarga de crear todos los archivos necesarios en la ruta especificada para que se pueda ejecutar de manera satisfactoria una aplicación móvil en las plataformas iOS y Android. Finalmente, se procede con la ejecución de la aplicación mediante los comandos `npx react-native run-android` y `npx react-native run-ios`.

La aplicación, con el objetivo de seccionar la lógica y funcionalidades, se divide en varias carpetas. La estructura completa de carpetas puede verse en la Figura 38. A

pesar de que los nombres de las carpetas pueden sugerir una descripción de su contenido, algunas de estas se describen a continuación, mencionando algunos de los archivos más relevantes en el proyecto. Las carpetas `android` e `ios` contienen los proyectos de cada una de estas plataformas que fueron creados de manera automática cuando se crea un nuevo proyecto de React Native. La carpeta `node_modules` contiene las dependencias instaladas por Node.js y React Native, esto incluye los módulos que se incluyen por defecto y las librerías adicionales que se instalan para implementar las diferentes funcionalidades de la aplicación.

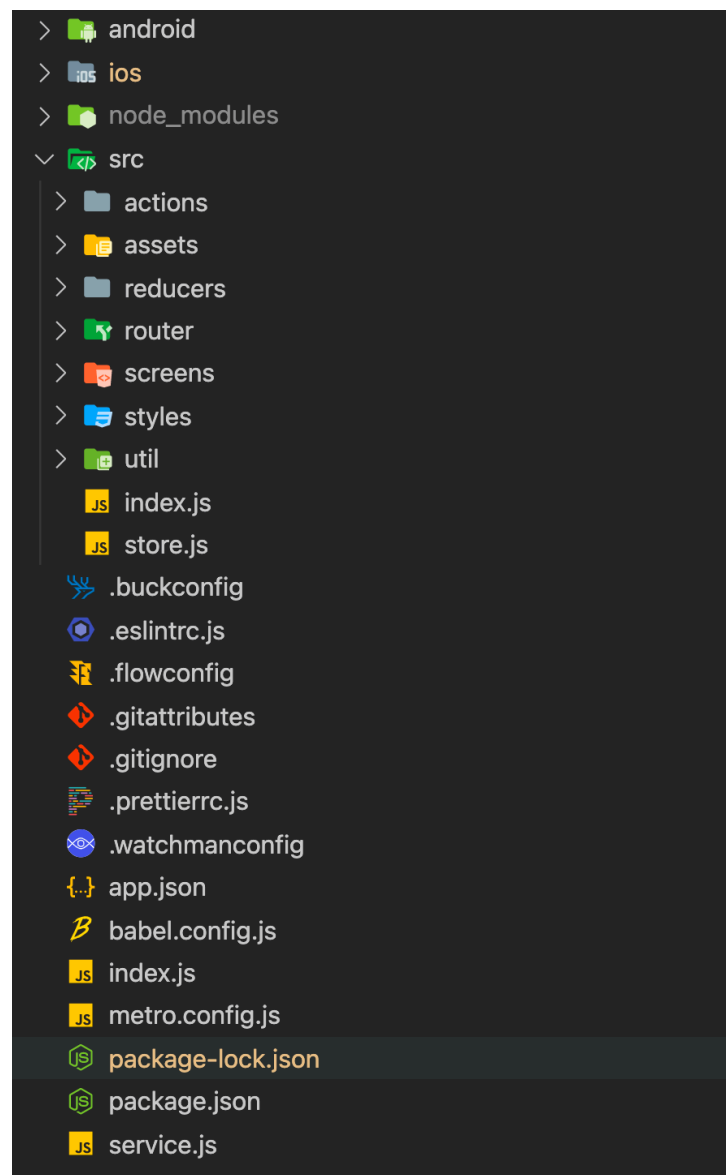


Figura 38: Estructura de archivos y carpetas del proyecto en React Native

La carpeta `src` contiene la gran mayoría de carpetas y archivos que le dan estructura



al proyecto. Inicialmente, la carpeta `screens` contiene todos los archivos de código creados para implementar las diferentes pantallas de la aplicación móvil; por otro lado, la carpeta `router` contiene todos los archivos en los cuales se crean los diferentes navegadores y su lógica de enrutamiento entre vistas; en las carpetas `actions` y `reducers` se encuentra toda la lógica de Redux que se implementó para que la aplicación cuente con una única fuente de almacenamiento para las diferentes variables y objetos, en este caso el `store`; finalmente, la carpeta `assets` contiene todos los archivos donde se declaran los recursos estáticos de la aplicación como caracteres, imágenes, iconos y constantes necesarios en la configuración de la aplicación.

La carpeta `util`, ubicada en el directorio raíz del proyecto, contiene todos los componentes que se repiten en diferentes partes de la aplicación o que son transversales para la misma como funciones para consumos de servicios proporcionados por el servidor de Unicauca Estéreo, componentes nativos que se usan en más de una pantalla o componentes que dan soporte al comportamiento de los diferentes navegadores de la aplicación.

`App.js` es el archivo que representa el punto de entrada de la aplicación; es la raíz del sistema de enrutamiento y gestiona los contenedores tanto para Redux como para la librería de navegación. El archivo `package.json` contiene varias informaciones sobre la aplicación, y una lista de dependencias que son necesarias para el correcto funcionamiento de la aplicación. Estas dependencias se instalan típicamente usando `npm` o `yarn`<sup>7</sup> mediante un simple comando como `npm install` o `yarn add` y se instalan en la carpeta `node_modules` mencionada anteriormente.

### 4.3.3 Módulo de Reproducción de Audio

Con el fin de cumplir con los requerimientos relacionados con el reproductor de audio, en la aplicación móvil fue necesario contar con una librería que soporte reproducción de archivos remotos de audio y reproducción de audio en *streaming*. Se escogió `react-native-track-player` para la implementación del reproductor de audio de la aplicación, el cual es un módulo de audio completo creado para aplicaciones de música y que cuenta con funcionalidades como reproducción de archivos de audio y *streaming*, controles de medios externos, modo de reproducción de fondo, entre otras características.

### 4.3.4 Manejo de Estados de Carga y Error

La aplicación móvil Unicauca Estéreo se conecta en modo cliente al servidor API REST de la emisora, esto con el fin de hacer peticiones de tipo HTTP a las diferentes rutas

---

<sup>7</sup>Yarn es un sistema de empaquetado de software desarrollado en 2016 por Facebook para el entorno de ejecución de JavaScript y Node JS [41].

configuradas, con el fin de acceder a los contenidos almacenados en la base de datos y posteriormente desplegarlos en las diferentes vistas de la aplicación, y de esta manera ofrecer los servicios requeridos. Realizar peticiones a un servidor es una acción síncrona, debido a que dependiendo de múltiples factores el tiempo de respuesta del servidor es variable, además de esto, se debe considerar que pueden existir errores durante el proceso de una petición, tanto del lado del servidor como del cliente. Debido a lo anterior, y teniendo como referencia las heurísticas H1 y H3 que expresan la necesidad de mostrar en todo momento al usuario el estado actual de la aplicación y brindar ayuda a los usuarios en caso de que se presente un error, se implementaron componentes que soportan tanto los estados de carga mientras se realizan peticiones al servidor, como los errores que se puedan presentar en estos procesos.

**4.3.4.1 Componente de carga** Una vez el usuario ingresa por primera vez a una vista de la aplicación, la aplicación realiza una petición al servidor para desplegar los componentes. Durante el tiempo que tarda el servidor en responder con la información solicitada, la aplicación muestra un componente tipo *loading* en el centro de la pantalla como se presenta en la Figura 39.

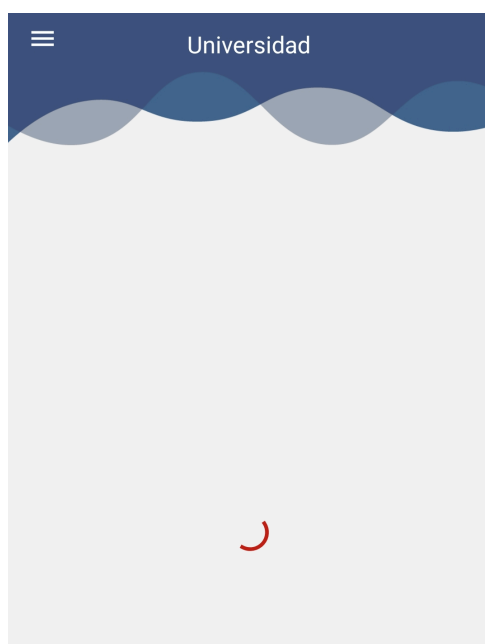


Figura 39: Vista con componente de carga.

Este comportamiento se puede presentar de igual manera durante el tiempo de carga para el reproductor de audio, tanto para la señal en vivo de la emisora, como para los *podcast*, tal como se presenta en la Figura 40.



Figura 40: Reproductor de audio con componente de carga.

**4.3.4.2 Componente de error** Al igual que con los estados de carga, para el caso en que se presente un error por parte del servidor o del cliente, o inclusive alguna excepción en alguna ejecución de alguna función relacionada con la conexión al portal web, se implementó un componente que se despliega en las secciones requeridas como se presenta en la Figura 41. En este componente se incluyó un botón que le permite al usuario volver a realizar la petición al servidor y recargar la página.

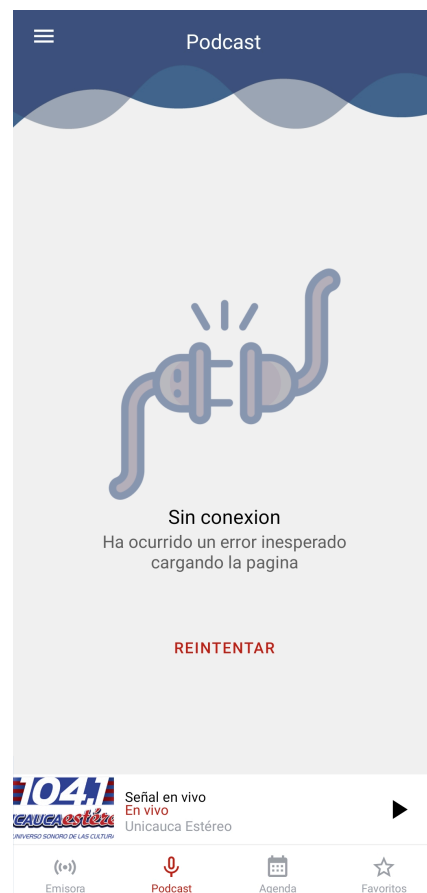


Figura 41: Vista con componente de error.

#### 4.3.5 Adaptación a Múltiples Tamaños de Pantalla

Los dispositivos móviles, específicamente los teléfonos inteligentes y dependiendo de la marca o modelo, cuentan con múltiples tamaños de pantalla. Con la intención de aplicar la heurística H9, la cual destaca la importancia de adaptación a múltiples entornos móviles, y en este caso específico los tamaños de pantalla, se hizo uso de la librería *react-native-extended-stylesheet*, librería que permitió extender las capacidades del módulo *stylesheet* que viene por defecto con React Native, y de esta manera aplicar diferentes estilos a los componentes. Una de las funcionalidades adicionales más importantes son los *rem* units, los cuales son similares a la unidad *rem* de CSS3, y permitió definir cualquier valor entero como relativo con base en un elemento raíz. En este caso, el valor raíz es una variable global denominada *rem* que puede configurarse mediante la función *EStyleSheet.build()* soportada por la librería, lo cual permite adaptar la aplicación a diferentes tamaños de pantalla y otras condiciones. El *rem* por defecto definido por *react-native-extended-stylesheet* es 16. Para definir el valor de la variable relativa *rem* en la aplicación móvil Unicauca Estéreo se aplicó la siguiente expresión matemática:

$$\text{rem} = \frac{\text{SW} \times 1.5}{\text{SH}}, \quad (1)$$

donde, *SW* representa el ancho total en píxeles de la pantalla del teléfono inteligente y *SH* representa el alto en píxeles de la pantalla. El valor de la variable *rem* sirve como un factor de escala para los diferentes tamaños de pantalla y los diferentes tamaños que se configuran para los componentes, textos, márgenes y espaciados a lo largo de la aplicación.

## 5 PRUEBAS

### 5.1 MODELO DE USABILIDAD

Partiendo de la definición descrita por la norma ISO 9241-11, la cual expresa que la usabilidad es “La medida en que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con eficacia, eficiencia y satisfacción en un contexto de uso específico” [25], es posible identificar tres atributos principales que deben ser tenidos en cuenta para la medición de la usabilidad de un sistema: eficacia, eficiencia y satisfacción, los cuales se presentan en la Figura 42.

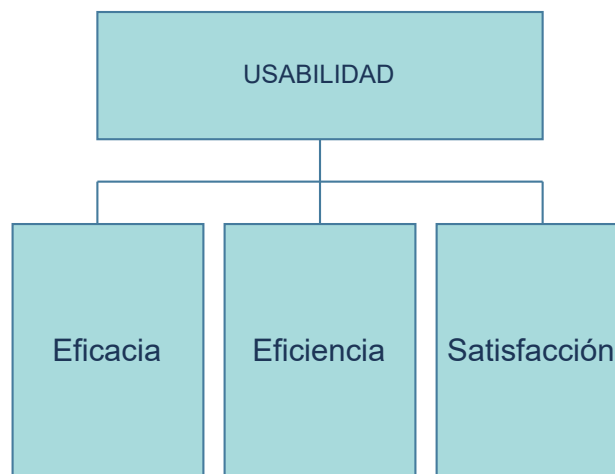


Figura 42: Modelo de usabilidad ISO 9241-11

- Eficacia: hace referencia a la precisión con la cual un usuario completa una tarea específica en el sistema.
- Eficiencia: hace referencia al número de recursos consumidos para realizar una tarea específica en el sistema.
- Satisfacción: hace referencia a las sensaciones que percibe el usuario al interactuar con el sistema al realizar una tarea específica.

Este modelo se enfoca en la definición de la usabilidad desde diferentes puntos de vista, en el cual la eficacia expresa el punto de vista de los procesos, la eficiencia expresa

los resultados y los recursos involucrados en cada interacción, y la satisfacción expresa la usabilidad desde el punto de vista del usuario [56]. Además de lo anterior, la ISO 9241-11 destaca tres factores claves que se deben tener en cuenta durante el proceso de medición de la usabilidad, los cuales son: el usuario (quien interactúa con la aplicación); el objetivo o resultado previsto; y el contexto de uso del sistema (tareas, herramientas empleadas, ambiente a nivel físico y social).

De este modo, se considera el modelo descrito por la ISO 9241-11, como el modelo para realizar la medición de la usabilidad de la aplicación móvil Unicauca Estéreo, con la medición de los tres parámetros descritos anteriormente.

## 5.2 ESTRATEGIA DE INVESTIGACIÓN

La estrategia de investigación seleccionada para la medición de la usabilidad de la aplicación móvil Unicauca Estéreo es la experimentación. La investigación experimental busca determinar con la mayor fiabilidad, posible relaciones de tipo causa y efecto por medio de una manipulación controlada de variables dentro de la realización de las pruebas [57]. Los experimentos tienen características específicas que aportan al proceso de medición de la usabilidad, por lo cual deben ser ejecutados de una manera sistemática y en un ambiente con condiciones controladas, además de esto se basa en la observación del objeto de estudio, con el fin de analizar su comportamiento en diferentes escenarios propiciados por el investigador.

Por otro lado, la investigación experimental puede ser categorizada en tres formas [58]:

- **Investigación pre-experimental:** Este tipo de investigación se realiza con el fin de tener un primer acercamiento al objetivo de estudio, y es aquí donde se realiza un análisis de una sola variable la cual no se va a modificar haciendo de la investigación un proceso más simple y sin la necesidad de un grupo de referencia.
- **Investigación experimental verdadera:** En este tipo de investigación se analiza una o más variables independientes en un ambiente controlado de pruebas. Es necesario contar con un grupo adicional que funcione como referencia o comparación.
- **Investigación cuasi-experimental:** Este tipo de investigación carece de un protocolo de control, es decir, que la selección del grupo de estudio no se hace de manera aleatoria y en cambio se aplican criterios convenientes que facilitan el flujo de los procesos de pruebas. Este tipo de investigación se aplica en procesos con carácter subjetivo.

Con base en las definiciones descritas anteriormente, se llegó a la conclusión de que la investigación pre-experimental posee las características más adecuadas para el presente

proyecto, ya que al ser una aproximación inicial de pruebas de usabilidad, la investigación requiere de un único grupo de estudio conformado por usuarios que interactúan con las funcionalidades dentro de un ambiente controlado [59]. Adicionalmente, se plantea la implementación de este proceso mediante pruebas de laboratorio, las cuales proveen tanto al investigador como al objeto de estudio con las herramientas y condiciones necesarias para ser ejecutadas de manera satisfactoria y libre de factores externos. Las pruebas de laboratorio permiten aplicar diferentes técnicas como la observación y “*think-aloud*”<sup>8</sup> [61], las cuales no podrían ser aplicadas usando otro tipo de métodos de investigación. Además de lo anterior, el método de investigación pre-experimental tiene la capacidad de revelar información relevante sobre la usabilidad de una aplicación, aún con un número reducido de participantes. Algunos estudios demuestran que hasta el 90% de los problemas relacionados con usabilidad pueden ser detectados realizando un proceso de pruebas con 10 participantes [27], [62], [63].

## 5.3 SELECCIÓN DE MÉTRICAS DE USABILIDAD

### 5.3.1 Medida de Eficacia

La eficacia hace referencia a la precisión con la cual los usuarios usan las funcionalidades de la aplicación. Este factor se puede medir a través de la tasa de finalización, o también llamada métrica fundamental de la usabilidad [64], la cual consiste en asignar un valor de “1” si el usuario logra completar una actividad en la aplicación y un “0” en el caso contrario. Por lo tanto, la eficacia de la aplicación móvil puede ser definida empleando la fórmula presentada a continuación [65].

$$\text{Eficacia} = \frac{\text{Número de tareas completadas exitosamente}}{\text{Número total de tareas}} \times 100\% \quad (2)$$

### 5.3.2 Medida de Eficiencia

En la medida de eficiencia es necesario tener en cuenta los recursos que el usuario gasta en su interacción con el sistema [25]. En este caso, se debe realizar una medición en función del tiempo que tarda un usuario en completar de manera satisfactoria una tarea de la aplicación. La eficiencia puede ser medida aplicando dos métodos: eficiencia basada en el tiempo y eficiencia relativa global. A continuación se presenta las fórmulas que permiten

---

<sup>8</sup>Método que consiste en solicitar a los potenciales usuarios la realización de una serie de tareas en un sistema, y la expresión verbal de sus pensamientos mientras se realizan las tareas [60].

obtener la medida de eficiencia [65]:

$$\text{Eficiencia basada en el tiempo} = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR} \times 100\% \quad (3)$$

y

$$\text{Eficiencia relativa global} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100\%, \quad (4)$$

donde:

- N: número total de tareas.
- R: número de usuarios.
- $n_{ij}$ : resultado de la tarea  $i$  por el usuario  $j$ . Si la tarea se completa con éxito, entonces  $n_{ij} = 1$ , en caso contrario,  $n_{ij} = 0$ .
- $t_{ij}$ : el tiempo empleado por el usuario  $j$  para completar la tarea  $i$ . Si la tarea no se completa con éxito, entonces se mide el tiempo hasta el momento en que el usuario abandona la tarea.

### 5.3.3 Medida de Satisfacción

La medida de satisfacción es el grado en que la interacción con un sistema satisfacen las necesidades del usuario [25]. Es necesario afirmar que es un factor subjetivo de la experiencia del usuario con el uso de la aplicación móvil. De esta manera, la satisfacción puede ser medida mediante diferentes cuestionarios estandarizados como: Cuestionario del Escenario Posterior (ASQ, *After Scenario Questionnaire*), Cuestionario de Esfuerzo Mental Subjetivo (SMEQ, *Subjective Mental Effort Questionnaire*), Estimación de Magnitud de Usabilidad (UME, *Usability Magnitude Estimation*) y Pregunta de Facilidad Única (SEQ, *Single Ease Question*) [65].

De acuerdo a [66], la pregunta de facilidad única (SEQ) se define como el cuestionario para la medición de la satisfacción de los usuarios, la cual exige poco tiempo para el usuario y respuestas simples. En este método se solicita al participante que califique que tan fácil o difícil fue realizar la tarea solicitada en una escala del 1 al 7, siendo 1 muy fácil y 7 muy difícil. El proceso anterior se realiza inmediatamente después de realizada una tarea en la aplicación.



## 5.4 DESCRIPCIÓN DEL AMBIENTE DE PRUEBAS

El proceso de pruebas de usabilidad se realizó en un ambiente de pruebas controlado. Se propició un espacio en el cual el participante se encontraba libre de distracciones que pudieran afectar su atención y concentración durante la interacción con la aplicación móvil. El ambiente de pruebas correspondió a una sala en la que el participante podía interactuar con la aplicación Unicauca Estéreo, mediante un dispositivo móvil Redmi Note 8, MIUI 12.0.2, y sistema operativo Android 10. El investigador realizó un proceso de observación de las acciones realizadas por los participantes mediante un computador MacBook Pro conectado de manera inalámbrica al dispositivo móvil, con el fin de afectar en la menor medida posible el proceso realizado por el participante [67]. La pantalla del dispositivo móvil fue grabada mediante el software AirDroid Cast [68] en ambos dispositivos (computador y teléfono móvil).

El proceso de pruebas se realizó con un total de diez participantes de manera individual y se realizó en cuatro fases:

- **Introducción:** En la primera fase, tanto los participantes como el investigador se presentan con el objetivo de establecer una interacción cómoda entre estos dos actores. Se solicitan datos generales al participante como el género, rango de edad y cargo u ocupación.
- **Contextualización:** Se describe de manera general al participante el objeto de estudio del proceso de pruebas, la situación actual del proyecto y datos generales de las funcionalidades de la aplicación móvil, sin que este haya interactuado en ningún momento con la interfaz gráfica.
- **Enfoque de estudio:** Se entrega al participante una lista con un total de 15 tareas, las cuales va a realizar dentro de la aplicación móvil mientras el investigador mide el tiempo que tarda en realizar cada una de las tareas. Al finalizar cada tarea se solicita al participante calificar su dificultad mediante un cuestionario tipo SEQ. En el anexo C se presenta el formato que contiene la lista de tareas entregado a los participantes para el proceso. Durante esta fase se motiva al participante a manifestar en voz alta todas las ideas y sensaciones que percibe al interactuar con la aplicación móvil y se registran.
- **Retroalimentación:** En la fase final del proceso se resuelven las dudas que pueda presentar el participante y se reciben todas las recomendaciones o sugerencias que quiera expresar sobre las diferentes funcionalidades y secciones de la aplicación móvil.

## 5.5 RESULTADOS

En esta sección se presentan los resultados obtenidos de las pruebas. Inicialmente, la Tabla 1 presenta información general de los usuarios: seis hombres y cuatro mujeres, quienes fueron parte del proceso de pruebas de usabilidad de la aplicación móvil Unicauca Estéreo. Cada participante realizó un total de quince tareas con el fin de explorar las diferentes funcionalidades desarrolladas.

En esta etapa se realizó una revisión de los formularios diligenciados durante las pruebas, así como una reproducción de los vídeos resultantes de la grabación de pantalla en las acciones llevadas a cabo por los participantes en los dispositivos móviles. Se realizó la debida valoración o puntuación de las actividades, en las cuales si el usuario completaba con éxito una tarea se le asignó un valor de 1 y en el caso contrario un 0. De manera simultanea, se midió el tiempo que tardaban los usuarios en completar cada tarea. Una vez la recolección de los datos fue finalizada, se procedió con la aplicación de las fórmulas para calcular las tres métricas de usabilidad definidos en el presente proyecto.

Usuarios	Género	Grupo de edad	Ocupación/Cargo
1	Femenino	35 a 44	Profesional universitario
2	Masculino	45 a 54	Comunicador social
3	Femenino	45 a 54	Empleada domestica
4	Masculino	25 a 34	Contratista de universidad pública
5	Masculino	18 a 24	Estudiante universitario
6	Femenino	18 a 24	Estudiante universitario
7	Masculino	25 a 34	Estudiante universitario
8	Masculino	25 a 34	Consultor técnico
9	Masculino	25 a 34	Desarrollador web
10	Femenino	25 a 34	Contadora pública

Tabla 1: Información de participantes

### 5.5.1 Eficacia

Si bien la tasa de finalización exitosa de cada una de las tareas es una medida importante, y su valor depende del contexto en el proceso de pruebas y del usuario que interactúa con la aplicación, es importante tener una medida de referencia que ayude a determinar si la eficacia del sistema se encuentra en un rango aceptable. El estudio realizado por Jeff

Sauro, en el cual se analizaron cerca de mil doscientas pruebas de usabilidad, demostró que el valor promedio para la medida de eficacia es del 75% [69]. Se espera de esta manera que los resultados de eficacia de la aplicación móvil Unicauca Estéreo se encuentren por encima de 75%.

La Figura 43 muestra la eficacia para cada una de las tareas ejecutadas por los usuarios. En general, se puede observar que la mayoría de las actividades obtuvieron un 100% de tasa de finalización, lo anterior significa que la totalidad de los participantes fueron capaces de realizar con éxito estas tareas (1-7, 11, 13-15). La tarea 10 fue la que obtuvo la menor puntuación entre las diferentes tareas con un 40%. Por otro lado, la tarea 8, la tarea 9 y la tarea 12, pese a que no fueron realizadas exitosamente por todos los participantes muestran una puntuación favorable de cara a la usabilidad del sistema. Finalmente, la efectividad general de la aplicación móvil tiene un valor de 92.66%, lo cual demuestra que esta por encima del valor promedio obtenido por Sauro en su estudio [69].



Figura 43: Tasa de finalización por tarea

### 5.5.2 Eficiencia Basada en el Tiempo

Para el caso de la eficiencia, un valor mayor indica que los usuarios son capaces de realizar un número mas alto de tareas por unidad de tiempo. La Figura 44 muestra la eficiencia para cada una de las tareas ejecutadas por los usuarios. La tarea 1 obtuvo la mayor puntuación de eficiencia basada en el tiempo entre las tareas con un valor de 0.404 tareas/s, seguido por la tarea 3 con una eficiencia de 0.242 tareas/s, y las tareas 14 y 15 con un valor de 0.182 tareas/s y 0.209 tareas/s, respectivamente. Las tareas que obtuvieron los valores mas bajos son 8, 2 y 10 con 0.037, 0.052 y 0.052, respectivamente, indicando que son tareas de mayor complejidad para los usuarios, ya que tardan más tiempo en ser finalizadas. Finalmente, la aplicación móvil Unicauca Estéreo, de manera general cuenta

con un valor de eficiencia de 0.147 tareas/s.

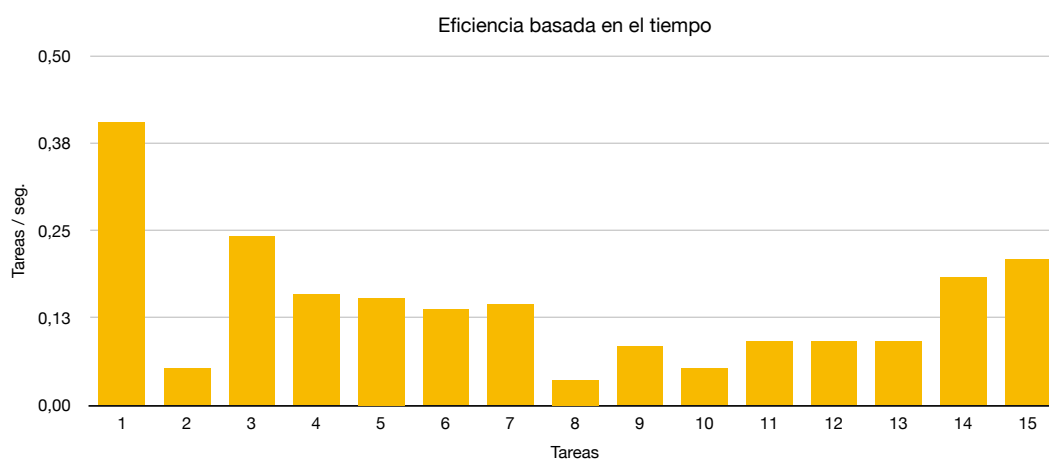


Figura 44: Eficiencia basada en el tiempo por tarea

### 5.5.3 Eficiencia Relativa Global

La eficacia relativa global nos permite medir el porcentaje de tiempo que gastan los usuarios cumpliendo una tarea de manera satisfactoria usando la aplicación en función del tiempo global gastado por los usuario, para la aplicación móvil Unicauca Estéreo la medida fue de un de un 94%. La Figura 45 muestra la eficacia relativa para cada una de las tareas, al igual que en el caso de la eficacia se puede observar que para la mayoría de las tareas todos los usuarios lograron completar con éxito todas la tareas, la tarea 10 obtiene el valor mas bajo con un 28% seguido por las tareas 8 y 9 con unos valores de 74% y 75%. Finalmente la tarea 12 obtuvo un valor de 88%.

### 5.5.4 Satisfacción

La Figura 46 muestra la puntuación media de satisfacción de cada usuario para todas las tareas realizadas, teniendo como referencia que en el formulario de satisfacción se le solicita a los participantes que califiquen entre 1 y 7 la facilidad de cada tarea, en la cual 1 representa "Muy fácil" y 7 representa "Muy difícil". El usuario 9 manifestó con su evaluación el mayor valor de dificultad al realizar las tareas con un valor promedio de 2.46, seguido por el usuario 2 y el usuario 5 que evaluaron la dificultad al utilizar la aplicación con 2.20 y 2.13 en promedio respectivamente. Por otro lado, el usuario 7 evaluó el uso de la aplicación con la puntuación más baja: 1.20. Lo anterior indica a nivel general que las funcionalidades de la aplicación móvil Unicauca Estéreo no representa una

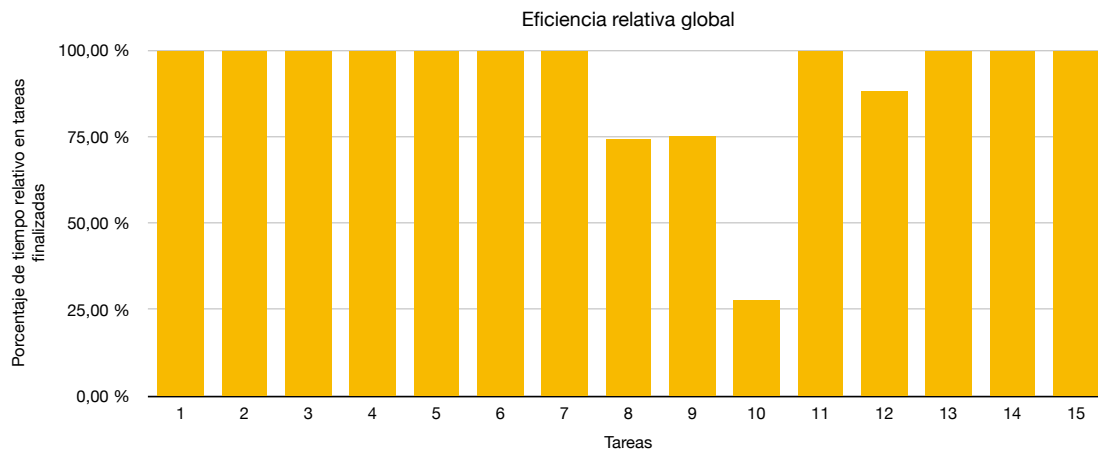


Figura 45: Eficiencia relativa global por tarea

mayor dificultad para los usuarios lo cual es un factor muy importante que contribuye a su satisfacción durante la interacción con las interfaces gráficas.

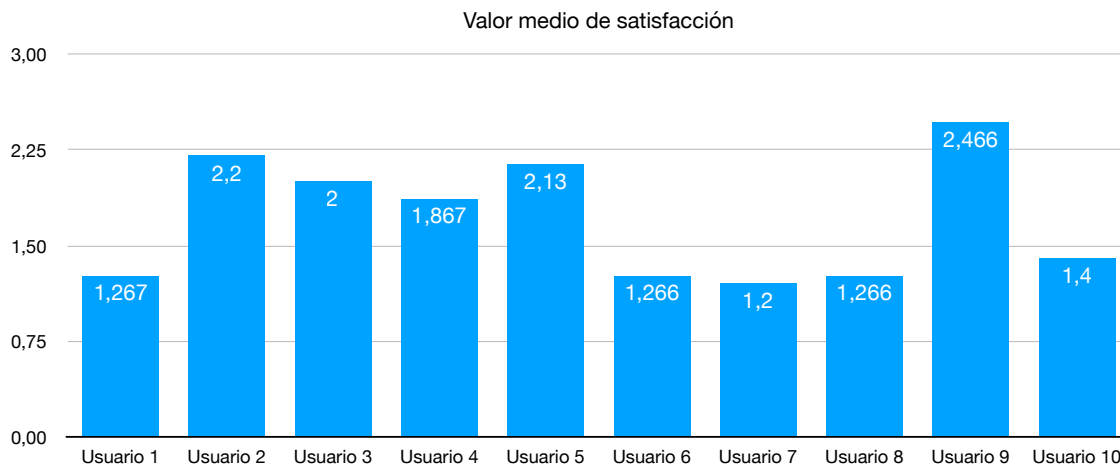


Figura 46: Puntuación media de satisfacción

## 6 CONCLUSIONES Y TRABAJOS FUTUROS

En el presente trabajo de grado, en modalidad de práctica profesional, se realizó el diseño, implementación y evaluación de una aplicación móvil para la emisora universitaria Unicauca Estéreo. Con el propósito de evaluar la usabilidad de la aplicación móvil desarrollada, se realizaron tanto pruebas objetivas como subjetivas con usuarios reales. Las primeras se aplicaron realizando la medición de parámetros como el tiempo de interacción de los usuarios con las funcionalidades de la aplicación, mientras que las pruebas subjetivas se realizaron con el propósito de medir la satisfacción de los usuarios mientras hacen uso de la aplicación móvil.

El presente capítulo presenta las conclusiones obtenidas a partir del desarrollo del trabajo de grado titulado: “Aplicación Móvil Unicauca Estéreo”, así como el planteamiento de algunos posibles trabajos de investigación y de práctica profesional a futuro que pueden realizarse a partir de los resultados obtenidos aquí.

### 6.1 CONCLUSIONES

Se cumplieron todos los objetivos planteados inicialmente y los requerimientos mencionados en el capítulo 2. El objetivo general fue crear una aplicación móvil de Unicauca Estéreo para los sistemas operativos Android y iOS, que permitiera reproducir la señal en vivo de la emisora, la reproducción de los *podcast* y el despliegue de los servicios de Unicauca Estéreo. Las siguientes son las conclusiones de este trabajo de grado en modalidad de práctica profesional:

- En el proceso de desarrollo de la aplicación móvil y del trabajo de grado, fue de suma importancia para el cumplimiento de los objetivos propuestos, el apoyo y retroalimentación de la dirección de Unicauca Estéreo y de la Comisión Técnica de la Emisora.
- Para la realización del diseño técnico y de componentes del sistema, se implementó el modelo 4+1 vistas como modelo de desarrollo, el cuál fue necesario y esencial en el proceso de abstracción del sistema, permitiendo realizar una descripción general del mismo que facilitó el desarrollo de la aplicación móvil. Por otra parte, el diseño de las interfaces gráficas se realizó aplicando diseño centrado en el usuario, para lo cual se tuvo acompañamiento y retroalimentación constante de la dirección de la emisora y de radioescuchas quienes permitieron probar y evaluar las funcionalidades diseñadas e implementadas de la aplicación móvil.

- La aplicación de criterios de usabilidad durante el proceso de diseño e implementación de la aplicación Unicauca Estéreo tuvo un alto impacto positivo en la calidad de la aplicación móvil resultante, ya que permitió contar con una guía que fue relevante al momento de tomar decisiones durante la creación de los componentes a nivel de diseño y de tecnología. Los criterios de usabilidad en este proyecto se expresaron en forma de un conjunto de heurísticas, las cuales demostraron ser una opción flexible debido a su posibilidad de adaptarse a las características específicas del proyecto y de la aplicación, y con ello evaluar la facilidad de su uso.
- La aplicación de React Native, como herramienta para la implementación de la aplicación móvil, permitió reducir significativamente el tiempo de desarrollo, debido a que con un único código fuente fue posible generar la aplicación móvil para los sistemas operativos iOS y Android.
- El que el servidor de la emisora estuviera soportado sobre una versión antigua de Drupal, limitó en gran parte las posibilidades al momento de elegir un módulo que hiciera posible convertir el portal en un servidor de tipo API REST, lo cual hubiese permitido cumplir de una manera más eficiente con las necesidades que surgieron durante el diseño de la aplicación, por lo anterior, fue necesario realizar un mayor procesamiento de los datos que se envían a la aplicación, pudiendo afectar el desempeño de la aplicación móvil.
- La implementación de pruebas de usabilidad mediante el cálculo de métricas tanto objetivas (eficacia y eficiencia) como subjetivas (satisfacción) representó una ayuda para la identificación de las secciones de la aplicación que presentaban problemas de usabilidad, lo que permitió plantear soluciones oportunas durante el proceso de desarrollo.
- Finalmente, es importante destacar que la inclusión de las nuevas tecnologías de la información y la comunicación en sistemas de radiodifusión, permiten que la emisora universitaria se adapte a las tendencias actuales del mercado, con nuevos canales de información que le permitirán llegar de una manera más efectiva a un número mayor de radioescuchas y a las nuevas generaciones.

## 6.2 TRABAJOS FUTUROS

Debido a que las aplicaciones móviles y en general el sector de las tecnologías digitales, de información y comunicación, hacen parte de un campo de trabajo con una alta proyección, es pertinente continuar realizando avances tomando como base el proyecto actual,

agregándole valor a la aplicación móvil desarrollada, y permitiéndole a Unicauca Estéreo acercarse a sus usuarios y brindar un mejor servicio, gracias a las nuevas tecnologías.

Teniendo en cuenta los resultados obtenidos en el presente trabajo de grado, se presentan a continuación las propuestas planteadas para trabajos futuros:

- Mejorar la estructura de los servicios del servidor API REST, ya que se realizó una adaptación en búsqueda de una solución eficaz, y es importante tener una estructura de datos que permita, primero *renderizar* los datos en las interfaces de manera sencilla y sin necesidad de procesamiento adicional o refinamiento de la información, y segundo realizar el menor número de peticiones posibles, con el fin de obtener un mejor desempeño a nivel de conexión con los servicios *backend*.
- Tanto Android como iOS son sistemas operativos que se actualizan continuamente, agregando nuevas funcionalidades en cada versión, lo anterior implica que es necesario un mantenimiento continuo y una actualización permanente de la aplicación móvil, de tal manera que las diferentes librerías instaladas y utilizadas se ajusten a futuras versiones de los sistemas operativos, evitando de este modo comportamientos erróneos de la aplicación.
- En la aplicación *backend* es necesario implementar una capa de seguridad y autenticación de las rutas que proveen los servicios a la aplicación móvil, para que estas no sean de acceso público, y de esta manera proteger los datos de navegación de los usuarios y la información del portal de la emisora Unicauca Estéreo.
- Implementar un sistema de registro y autenticación en el servidor para que los usuarios puedan crear una cuenta en la aplicación móvil, lo cual implica un desarrollo tanto en el lado de la aplicación *backend*, como en la aplicación móvil, y de esta manera llevar una trazabilidad de manera ordenada de la interacción y las preferencias de los usuarios del sistema.
- Añadir un servicio para soporte de idioma que permita a los usuarios visualizar la información de la aplicación móvil en inglés, tal como se puede realizar en la página de la Universidad del Cauca, lo cual permitiría a la emisora acercarse a un mayor número de usuarios.



## REFERENCIAS

- [1] E. Ngai and A. Gunasekaran, “A review for mobile commerce research and applications”, *Decision Support Systems*, vol. 43, no. 1, pp. 3–15, 2007, Mobile Commerce: Strategies, Technologies, and Applications, ISSN: 0167-9236. DOI: <https://doi.org/10.1016/j.dss.2005.05.003>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167923605000692>.
- [2] B. Still and K. Crane, *Fundamentals of User-Centered Design: A Practical Approach*. Boca Raton: CRC Press, Feb. 17, 2017, 347 pp., ISBN: 978-1-315-20092-7. DOI: 10.4324/9781315200927.
- [3] H. Hoehle, R. Aljafari, and V. Venkatesh, “Leveraging microsofts mobile usability guidelines: Conceptualizing and developing scales for mobile application usability”, *International Journal of Human-Computer Studies*, vol. 89, pp. 35–53, May 1, 2016, ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2016.02.001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1071581916000215> (visited on 08/31/2021).
- [4] (Jun. 27, 2012). La emisora, [Online]. Available: <http://www.unicauca.edu.co/sitioemisora/content/la-emisora> (visited on 08/31/2021).
- [5] (Jun. 27, 2012). Unicauca estéreo, [Online]. Available: <http://www.unicauca.edu.co/sitioemisora/content/programacion> (visited on 08/31/2021).
- [6] (Nov. 27, 2018). Esta es la historia de las aplicaciones móviles, Skyscanner España, [Online]. Available: <https://www.skyscanner.es/noticias/esta-es-la-historia-de-las-aplicaciones-moviles> (visited on 01/19/2022).
- [7] D. Amalfitano, A. R. Fasolino, P. Tramontana, and B. Robbins, “Chapter 1 - testing android mobile applications: Challenges, strategies, and approaches”, in *Advances in Computers*, A. Memon, Ed., vol. 89, Elsevier, Jan. 1, 2013, pp. 1–52. DOI: 10.1016/B978-0-12-408094-2.00001-1. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124080942000011> (visited on 01/23/2022).
- [8] (Jan. 9, 2007). Apple reinvents the phone with iPhone, Apple Newsroom, [Online]. Available: <https://www.apple.com/newsroom/2007/01/09Apple-Reinvents-the-Phone-with-iPhone/> (visited on 01/24/2022).
- [9] B. Fling, *Mobile design and development: Practical concepts and techniques for creating mobile sites and Web apps*. " O'Reilly Media, Inc.", 2009.

- [10] T. Warren. (May 23, 2016). Windows phone market share sinks below 1 percent, The Verge, [Online]. Available: <https://www.theverge.com/2016/5/23/11743594/microsoft-windows-phone-market-share-below-1-percent> (visited on 01/22/2022).
- [11] (2019). Develop applications - HUAWEI HarmonyOS APP, [Online]. Available: <https://developer.harmonyos.com/en/> (visited on 01/22/2022).
- [12] W. Jobe, “Native apps vs. mobile web apps”, *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 7, pp. 27–32, Oct. 1, 2013. DOI: 10.3991/ijim.v7i4.3226.
- [13] K. Buettner and A. M. Simmons, “Mobile web and native apps: How one team found a happy medium”, in *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, A. Marcus, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 549–554, ISBN: 978-3-642-21675-6.
- [14] A. Biørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, “Progressive web apps: The possible web-native unifier for mobile development:” in *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 344–351, ISBN: 978-989-758-246-2. DOI: 10.5220/0006353703440351. [Online]. Available: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006353703440351> (visited on 01/21/2022).
- [15] M. Panhale, *Beginning hybrid mobile application development*. Springer, 2016.
- [16] R. Raj and S. Tolety, “A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach”, Dec. 1, 2012, pp. 625–629, ISBN: 978-1-4673-2270-6. DOI: 10.1109/INDCON.2012.6420693.
- [17] M. Kuitunen, “Cross-platform mobile application development with react native”, 2019, Accepted: 2019-02-08T08:07:24Z. [Online]. Available: <https://trepo.tuni.fi/handle/123456789/27139> (visited on 01/24/2022).
- [18] (2022). Cross compilation, [Online]. Available: <https://mesonbuild.com/Cross-compilation.html> (visited on 01/23/2022).
- [19] A. Dix, *Human Computer Interaction*. Pearson Education, 2008, ISBN: 9788131717035. [Online]. Available: <https://books.google.com.co/books?id=00u06KszvIEC>.
- [20] R. M. Baecker, J. Grudin, W. Buxton, and S. Greenberg, *Readings in human-computer interaction : toward the year 2000*. Morgan Kaufmann Publishers, 1995, ISBN: 9781558602465. [Online]. Available: <https://archive.org/details/readingsinhuman/v6z1/mode/1up>.

- [21] (2021). About the ACM organization, [Online]. Available: <https://www.acm.org/about-acm/about-the-acm-organization> (visited on 08/17/2021).
- [22] S. ( : U.S.), Ed., *ACM SIGCHI curricula for human-computer interaction*, New York: Association for Computing Machinery, 1992, 162 pp., ISBN: 978-0-89791-474-1.
- [23] E. Eshet, M. d. Reuver, and H. Bouwman, “The role of organizational strategy in the user-centered design of mobile applications”, *Communications of the Association for Information Systems*, vol. 40, no. 1, Apr. 1, 2017, ISSN: 1529-3181. DOI: 10.17705/1CAIS.04014. [Online]. Available: <https://aisel.aisnet.org/cais/vol40/iss1/14>.
- [24] C. Abras, D. Maloney-Krichmar, and J. Preece, “User-centered design”, *User-Centered Design*, pp. 445–456, Jan. 1, 2004.
- [25] (2018). ISO 9241-11:2018(en), ergonomics of human-system interaction — part 11: Usability: Definitions and concepts, [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> (visited on 11/21/2021).
- [26] N. Bevana, J. Kirakowskib, and J. Maissela, “What is usability”, in *In Proceedings of the 4th International Conference on HCI*, 1991.
- [27] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, Oct. 7, 1994, 382 pp., Google-Books-ID: 95As2OF67f0C, ISBN: 978-0-12-518406-9.
- [28] A. Alshamrani and A. Bahattab, “A comparison between three sdlc models waterfall model, spiral model, and incremental/iterative model”, *International Journal of Computer Science Issues (IJCSI)*, vol. 12, no. 1, p. 106, 2015.
- [29] Y. Bassil, “A simulation model for the waterfall software development life cycle”, *arXiv preprint arXiv:1205.6904*, 2012.
- [30] R. D. Amlani, “Advantages and limitations of different sdlc models”, *International Journal of Computer Applications & Information Technology*, vol. 1, no. 3, pp. 6–11, 2012.
- [31] W. L. i. R.-B. U. Experience. (). 10 usability heuristics for user interface design, Nielsen Norman Group, [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/> (visited on 03/01/2022).
- [32] R. Baharuddin, D. Singh, and R. Razali, “Usability dimensions for mobile applications- a review”, *Res. J. Appl. Sci. Eng. Technol*, vol. 5, no. 6, pp. 2225–2231, 2013.

- [33] R. Bernhaupt, K. Mihalic, and M. Obrist, “Usability evaluation methods for mobile applications”, in *Handbook of research on user interface design and evaluation for mobile technology*, IGI Global, 2008, pp. 745–758.
- [34] G. Joyce and M. Lilley, “Towards the development of usability heuristics for native smartphone mobile applications”, in *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, A. Marcus, Ed., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 465–474, ISBN: 978-3-319-07668-3. DOI: 10.1007/978-3-319-07668-3\_45.
- [35] S. Po, S. Howard, F. Vetere, and M. Skov, “Heuristic evaluation and mobile usability: Bridging the realism gap”, Sep. 13, 2004, pp. 49–60, ISBN: 978-3-540-23086-1. DOI: 10.1007/978-3-540-28637-0\_5.
- [36] P. Ketola and M. Rykkee, “The three facets of usability in mobile handsets”, Apr. 1, 2001.
- [37] E. Bertini, S. Gabrielli, S. Kimani, T. Catarci, and G. Santucci, “Appropriating and assessing heuristics for mobile computing”, in *Proceedings of the working conference on Advanced visual interfaces*, ser. AVI '06, New York, NY, USA: Association for Computing Machinery, May 23, 2006, pp. 119–126, ISBN: 978-1-59593-353-9. DOI: 10.1145/1133265.1133291. [Online]. Available: <https://doi.org/10.1145/1133265.1133291> (visited on 03/12/2022).
- [38] R. Inostroza, C. Rusu, S. Roncagliolo, C. Jimenez, and V. Rusu, “Usability heuristics for touchscreen-based mobile devices”, in *2012 Ninth International Conference on Information Technology - New Generations*, Apr. 2012, pp. 662–667. DOI: 10.1109/ITNG.2012.134.
- [39] P. Kruchten, “Planos arquitectónicos: El modelo de 4+ 1 vistas de la arquitectura del software”, *IEEE Software*, vol. 12, no. 6, pp. 42–50, 1995.
- [40] (2022). Software de gráficos vectoriales líder del sector | Adobe Illustrator, [Online]. Available: <https://www.adobe.com/la/products/illustrator.html> (visited on 03/10/2022).
- [41] A. Boduch and R. Derks, *React and React Native: A complete hands-on guide to modern web and mobile development with React.js*. Packt Publishing Ltd, 2020.
- [42] (2017). Introducing JSX – react, [Online]. Available: <https://reactjs.org/docs/introducing-jsx.html> (visited on 03/28/2022).

- [43] A. Banks and E. Porcello, *Learning React: functional web development with React and Redux*. " O'Reilly Media, Inc.", 2017.
- [44] (2018). Introducing hot reloading · react native, [Online]. Available: <https://reactnative.dev/blog/2016/03/24/introducing-hot-reloading> (visited on 03/28/2022).
- [45] (2022). Redux - a predictable state container for JavaScript apps. | redux, [Online]. Available: <https://redux.js.org/> (visited on 03/28/2022).
- [46] (2022). The redux data flow - JavaScript by example [book]. ISBN: 9781788293969, [Online]. Available: <https://www.oreilly.com/library/view/javascript-by-example/9781788293969/1dfe8989-b67a-428f-bfcd-a667476f2149.xhtml> (visited on 03/28/2022).
- [47] (2005). About - git, [Online]. Available: <https://git-scm.com/about> (visited on 03/28/2022).
- [48] Atlassian. (2022). Flujo de trabajo de Gitflow | Atlassian Git Tutorial, Atlassian, [Online]. Available: <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow> (visited on 03/28/2022).
- [49] (2022). Visual studio code - code editing. redefined, [Online]. Available: <https://code.visualstudio.com/> (visited on 03/28/2022).
- [50] S. K. Patel, V. R. Rathod, and S. Parikh, "Joomla, drupal and wordpress-a statistical comparison of open source cms", in *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*, IEEE, 2011, pp. 182–187.
- [51] (Jan. 20, 2012). Services, Drupal.org, [Online]. Available: <https://www.drupal.org/project/services> (visited on 03/28/2022).
- [52] (Feb. 8, 2019). Ctools 7.x-1.15, Drupal.org, [Online]. Available: <https://www.drupal.org/project/ctools/releases/7.x-1.15> (visited on 03/28/2022).
- [53] (Oct. 5, 2018). Libraries 7.x-2.5, Drupal.org, [Online]. Available: <https://www.drupal.org/project/libraries/releases/7.x-2.5> (visited on 03/28/2022).
- [54] (2018). Expo, Expo, [Online]. Available: <https://expo.dev/> (visited on 03/29/2022).
- [55] S. Amann, S. Proksch, S. Nadi, and M. Mezini, "A study of visual studio usage in practice", in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, IEEE, vol. 1, 2016, pp. 124–134.
- [56] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, "Consolidating the iso usability models", in *Proceedings of 11th international software quality management conference*, vol. 2003, 2003, pp. 23–25.

- [57] J. Adams, H. T. Khan, and R. Raeside, *Research methods for business and social science students*. SAGE Publications India, 2014.
- [58] (2022). Investigación experimental, Significados, [Online]. Available: <https://www.significados.com/investigacion-experimental/> (visited on 03/17/2022).
- [59] M. A. Farooq, H. Nóvoa, A. Araújo, and S. M. Tavares, “An innovative approach for planning and execution of pre-experimental runs for design of experiments”, *European Research on Management and Business Economics*, vol. 22, no. 3, pp. 155–161, 2016.
- [60] M. Van Den Haak, M. De Jong, and P. Jan Schellens, “Retrospective vs. concurrent think-aloud protocols: Testing the usability of an online library catalogue”, *Behaviour & information technology*, vol. 22, no. 5, pp. 339–351, 2003.
- [61] E. Beck, M. Christiansen, J. Kjeldskov, N. Kolbe, and J. Stage, “Experimental evaluation of techniques for usability testing of mobile systems in a laboratory setting”, in *Proceedings of OzCHI 2003, Brisbane, Australia*, CHISIG, 2003.
- [62] J. Rubin and D. Chisnell, “How to plan, design, and conduct effective tests”, *Handbook of usability testing*, vol. 17, no. 2, p. 348, 2008.
- [63] J. S. Dumas, J. S. Dumas, and J. Redish, *A practical guide to usability testing*. Intellect books, 1999.
- [64] J. S. PhD. (Sep. 6, 2011). 10 things to know about completion rates – MeasuringU, [Online]. Available: <https://measuringu.com/completion-rates/> (visited on 03/17/2022).
- [65] (2015). Usability metrics - a guide to quantify the usability of any system, Usability Geek. Section: Terminology, [Online]. Available: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/> (visited on 03/17/2022).
- [66] J. Sauro and J. S. Dumas, “Comparison of three one-question, post-task usability questionnaires”, in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2009, pp. 1599–1608.
- [67] (2015). A guide to simple and painless mobile user testing, Smashing Magazine. Section: General, [Online]. Available: <https://www.smashingmagazine.com/2015/12/simple-and-painless-mobile-user-testing/> (visited on 03/19/2022).
- [68] (2022). Cast and control android and iOS screens from PC| AirDroid cast, AirDroid, [Online]. Available: <https://www.airdroid.com/cast/> (visited on 03/19/2022).

- [69] J. S. PhD. (Mar. 21, 2011). What is a good task-completion rate? – MeasuringU, [Online]. Available: <https://measuringu.com/task-completion/> (visited on 03/24/2022).

## ANEXOS

### A RESULTADOS ENCUESTA INICIAL

Con el propósito de evaluar el grupo de personas a la cual está dirigida la aplicación móvil y recolectar información relevante que pueda servir como insumo en el proceso de definición de los requerimientos y el alcance del proyecto, se realizó una encuesta libre a los miembros de la comunidad de la Universidad del Cauca. De acuerdo al tipo de preguntas que se realizaron, la encuesta se dividió en dos secciones: en la primera sección, se realizaron preguntas relacionadas con datos demográficos de los participantes y su uso de dispositivos móviles; mientras que en la segunda parte, se realizaron preguntas enfocadas al consumo de contenidos relacionado con la radio y noticias. A continuación, se presentan los resultados obtenidos a partir de una participación de 714 personas.

Dentro del grupo de encuestados se puede identificar una amplia diversidad en términos del cargo o profesión que cumplen dentro del ámbito universitario, siendo los estudiantes el grupo más numeroso con un 64.1%. La información detallada se puede observar en Figura 47. Por otra parte, en términos de género se evidencia una distribución equitativa con un 57.15% de hombres y un 42.9% de mujeres, tal y como se puede observar en la Figura 48.

#### Profesión

714 respuestas

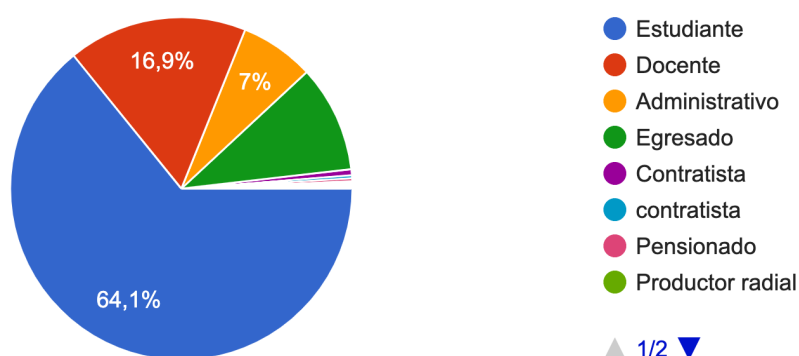


Figura 47: Profesión o cargo en la Universidad del Cauca



**Género**

714 respuestas

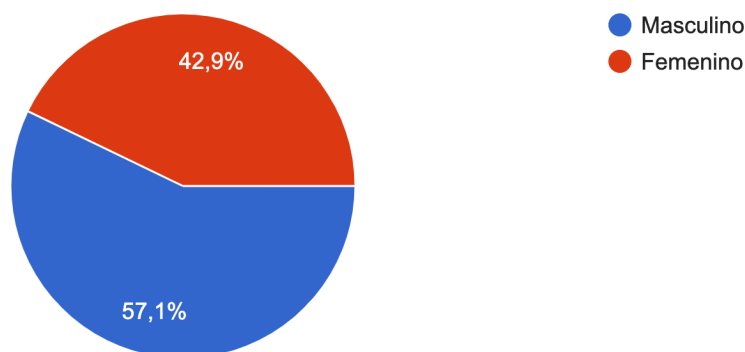


Figura 48: Género

Con respecto al uso de dispositivos móviles (Figura 49), se encontró que el 98,7% de los encuestados cuentan con un teléfono inteligente y que además la frecuencia de uso de estos dispositivos es muy alta, donde en una escala del 1 al 5 siendo 1 “poca frecuencia” y 5 “alta frecuencia” el 87,9% de los participantes seleccionaron un valor de 4 o 5 (Figura 50). Adicional a lo anterior, y de acuerdo a la Figura 51, del total de encuestados

**¿Cuenta usted con un smartphone (teléfono inteligente)?**

714 respuestas

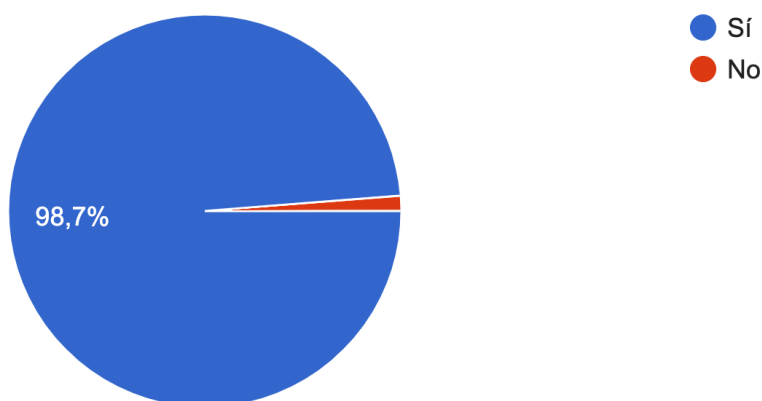


Figura 49: Uso de teléfono inteligente

¿Con que frecuencia considera usted que usa su teléfono móvil?

711 respuestas

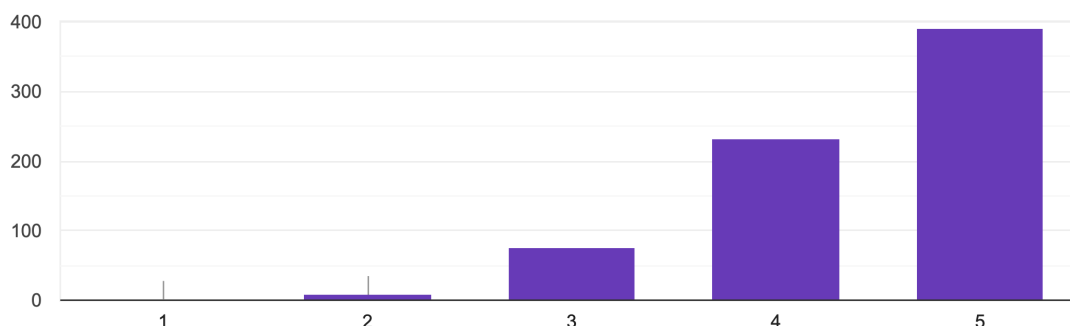


Figura 50: Frecuencia de uso de un teléfono inteligente

que cuentan con un teléfono inteligente, el 83.2% de los dispositivos usan Android como sistema operativo y el 14.6% iOS. En contraposición, sólo el 2% usa el nuevo sistema operativo de Huawei Harmony y ninguno de los dispositivos corre sobre sistema operativo Windows Phone. Lo anterior sirve como sustento para definir a iOS y Android como los sistemas operativos objetivos para los cuales se desarrolló la aplicación móvil de Unicauca Estéreo.

¿Que sistema operativo tiene su dispositivo?

704 respuestas

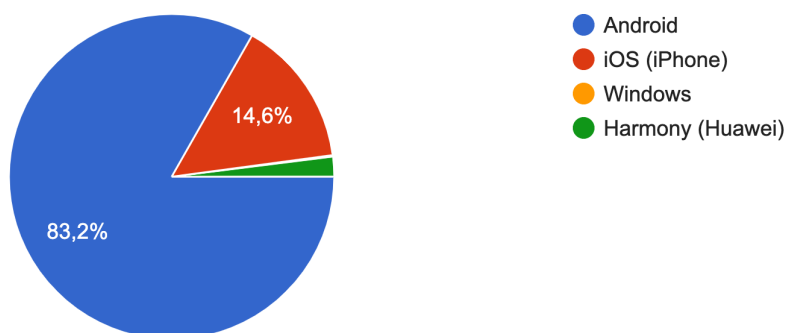


Figura 51: Género

Pasando a la segunda sección de la encuesta, se recolectó información sobre el consumo de los participantes en términos de contenido relacionado con radio y noticias. Como se observa en la Figura 52 y en la Figura 53, del total de encuestados, el 78.7% escucha de manera regular radio y el 77.3% consume o ha consumido programas en formato *podcast*,

respectivamente.

### ¿Escucha radio?

714 respuestas

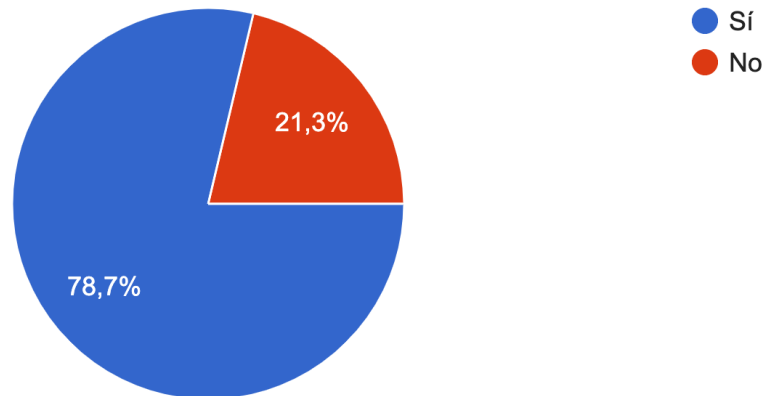


Figura 52: Consumo de radio

### ¿Escucha o ha escuchado un podcast?

714 respuestas

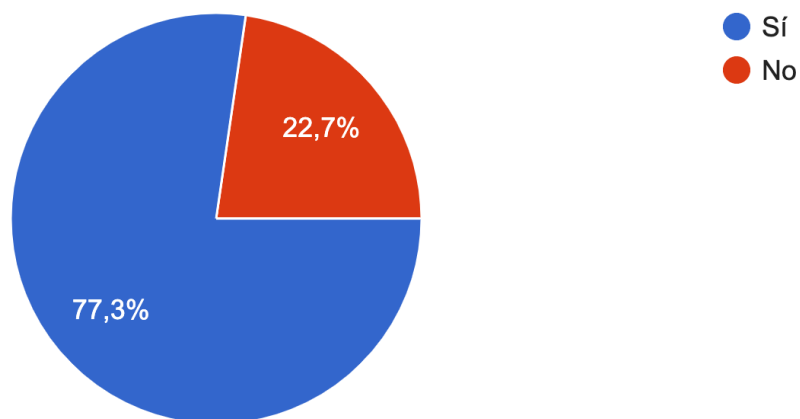


Figura 53: Consumo de programas en formato podcast

Un dato relevante de cara al impacto del proyecto dentro de la comunidad universitaria es, que de acuerdo a la Figura 54 de los diferentes medios usados para el acceso a la radio un total de 303 participantes (43.9%) manifestaron emplear aplicaciones móviles.

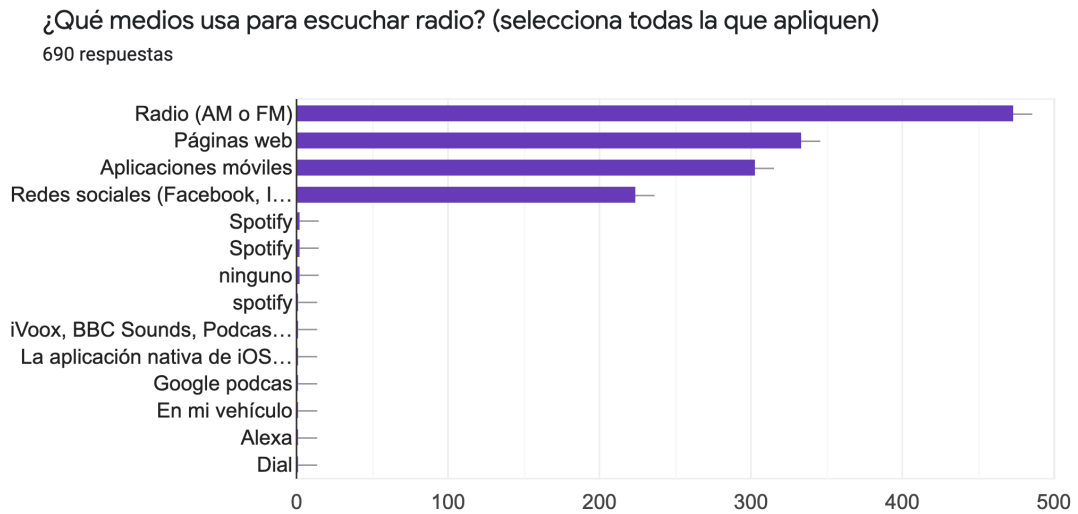


Figura 54: Medios usados para escuchar radio

Con respecto a las noticias, del total de encuestados, el 82.9% consume noticias de manera habitual (Figura 55).

### ¿Consume noticias?

714 respuestas

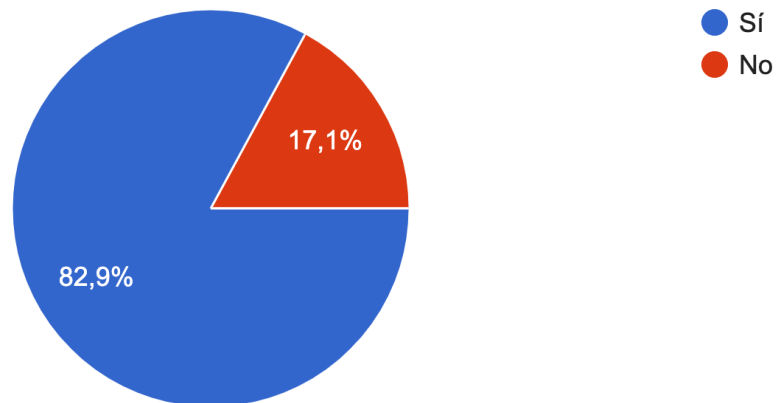


Figura 55: Consumo de noticias

## B MANUAL DE USUARIO APLICACIÓN MÓVIL UNICAUCA ESTÉREO

### B.1 VISTA SPLASH

Cuando se abre la aplicación, aparece de inmediato la pantalla de bienvenida que automáticamente dirige a la interfaz de inicio, como se presenta en la siguiente figura:



Figura 56: Vista splash

### B.2 VISTA DE INICIO

Después de la pantalla de bienvenida, el sistema se dirige automáticamente a la página de inicio, en la cual se encuentra: un menú lateral tipo hamburguesa; un *banner* en el cual se informa el programa que se está reproduciendo, y más abajo se encuentra una sección con los programas que se emitirán a continuación; posteriormente, hay una sección de noticias recientes y más abajo publicaciones de Twitter; en la parte inferior, se encuentra la barra de reproducción en vivo con la información del programa que se está escuchando; y finalmente, se encuentra la barra inferior con el menú principal de la aplicación. La vista de inicio se presenta en la siguiente figura:



Figura 57: Inicio de la aplicación móvil Unicauca Estéreo

### B.3 MENÚ PRINCIPAL

En este menú, ubicado en la barra inferior de la aplicación, al hacer clic sobre los iconos, se pueden encontrar los accesos rápidos a:

- La pagina principal de la aplicación.
- Listado de *podcast*.
- Agenda de eventos.
- Acceso a *podcast* y capítulos favoritos guardados.

La vista sobre el menú principal se presenta en la siguiente figura:

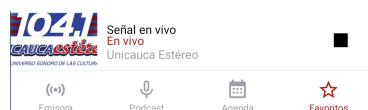


Figura 58: Menú de navegación principal

### B.4 MENÚ LATERAL

En la parte superior izquierda de la pagina principal de la aplicación y al hacer clic en el icono tipo hamburguesa o al deslizar el dedo de izquierda a derecha, aparecerá el menú lateral. En el menú lateral se encuentran las siguientes opciones:

- **Inicio:** al hacer clic repliega el menú lateral y presenta la pagina principal de la aplicación nuevamente. El menú lateral y la opción inicio se presenta en la siguiente figura:

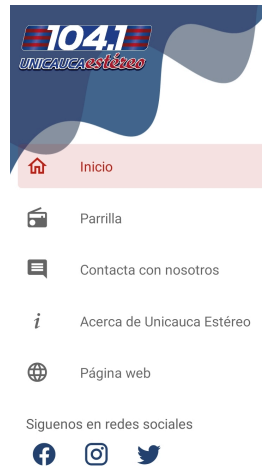


Figura 59: Menú de navegación lateral

- **Parrilla:** al hacer clic en esta opción identificada con el icono de un radio, se presenta los programas que se emiten por Unicauca Estéreo en cada uno de los días de la semana (parrilla de programación), identificando cada programa con una imagen representativa y el horario en el cual es transmitido, tal como se presenta en la figura siguiente:

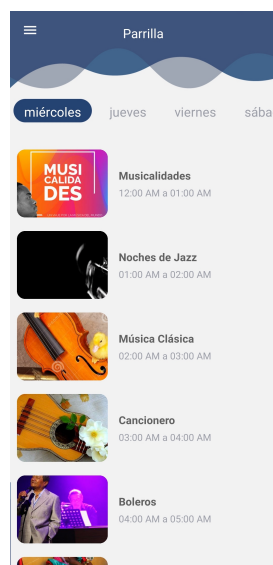


Figura 60: Parrilla de programación

Al hacer clic sobre cualquiera de los programas se presenta: el nombre del programa y su imagen representativa; todos los episodios disponibles listados por fecha, de la más reciente a la más antigua hacia abajo; la opción para reproducir o pausar en caso de estar siendo escuchado; y la posibilidad de ser agregado a la lista de favoritos. En la parte inferior se puede observar el reproductor en curso y la barra del menú principal, tal como se presenta en la figura siguiente:



Figura 61: Vista de detalle de un programa radial

## B.5 VISTA DE CONTACTO

Al hacer clic en la opción contacta con nosotros en el menú lateral. El contacto puede hacerse a través de correo electrónico o por medio de una llamada telefónica a una línea local de la ciudad de Popayán, dependiendo la opción que se elija al hacer clic redirige automáticamente a la opción de redactar correo en la cuenta que este activa en el equipo o directamente al teléfono para realizar la llamada. La vista de contacto se presenta en la siguiente figura:



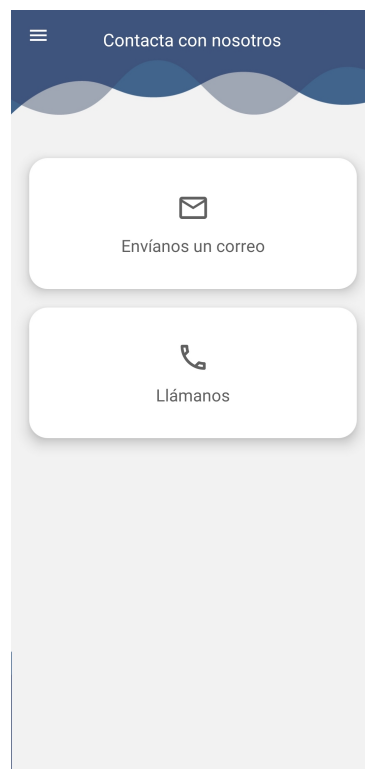


Figura 62: Vista de contacto con la emisora

## B.6 VISTA ACERCA DE UNICAUCA ESTÉREO

Al hacer clic en esta opción del menú lateral se desplegará una interfaz con la posibilidad de elegir una de tres opciones:

- ¿Quiénes somos?.
- Nuestra historia.
- Regionalización.

Al hacer clic en cualquiera de estas tres opciones se desplegara la información correspondiente, como se presenta en la figura siguiente:



Figura 63: Vista de acerca de la emisora

En este menú lateral también se encuentran los enlaces externos, los cuales al hacer clic sobre ellos permiten ir a la página web de la emisora y las redes sociales: Facebook, Instagram y Twitter, identificadas cada una con su respectivo logo.

## B.7 VISTA DE PODCAST

En el menú principal se encuentra la opción de acceder a los *podcast* disponibles en la aplicación. Al hacer clic en esta opción se encuentran todos los programas de la parrilla que tienen *podcast* y que pueden ser visualizados y escuchados en la aplicación. Al hacer clic en cualquiera de las opciones disponibles se puede seleccionar cualquiera de los capítulos que se desee escuchar. En esta sección se encuentra un listado de todos los episodios por fecha, el más reciente se encuentra en la parte superior y hacia abajo se pueden encontrar los más antiguos. Cada uno tiene la opción de reproducción o listas en la categoría de favoritos. La vista de *podcast* se presenta en la figura siguiente:

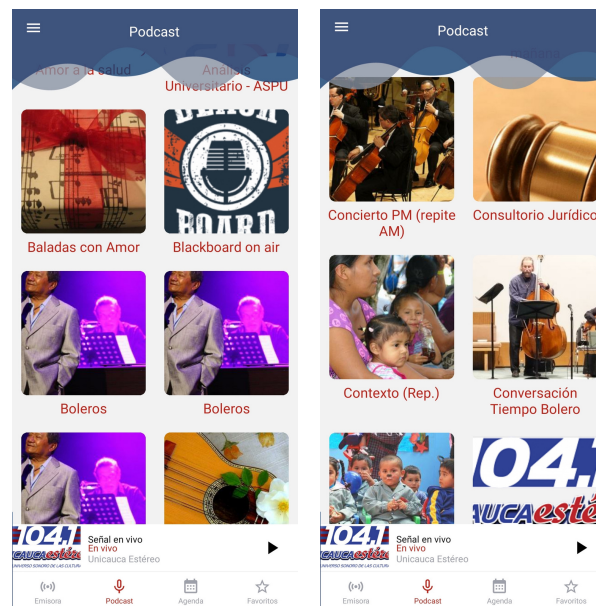


Figura 64: Vista de programas

## B.8 VISTA AGENDA

En esta sección del menú principal se encuentra un listado de los últimos eventos de interés, no solo de la emisora y sus diferentes programas, sino también de la comunidad universitaria en general. Cada evento tiene una imagen distintiva, título, fecha y hora de realización, como se presenta en la figura siguiente:

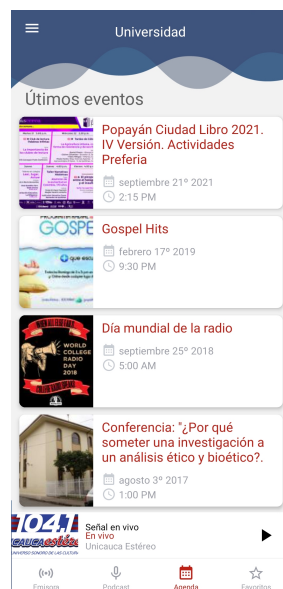


Figura 65: Vista de agenda universitaria

Al hacer clic en cualquiera de los eventos del listado de la agenda, aparece el título de evento como encabezado de la interfaz, la imagen publicitaria ampliada, la fecha y hora de realización y una descripción detallada con toda la información del evento, como se presenta en la figura siguiente:



Figura 66: Vista de detalle de un evento

## B.9 VISTA FAVORITOS

En el menú principal también se encuentra el icono de favoritos. Al hacer clic en esta opción se listan todos los programas y *podcast* que se han marcado como favoritos desde la parrilla y desde la sección *podcast*. Cada uno tiene su respectiva pestaña y se listan por separado, como se presenta en la figura siguiente:

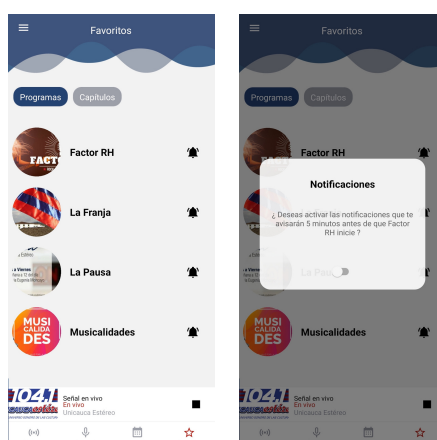


Figura 67: Vista de programas favoritos

En la sección de programas, adicionalmente, hay una opción para agregar notificaciones que avisan al usuario sobre el inicio del programa, y para ello solo basta hacer clic en la campana de activación.

En el caso de los *podcast*, con hacer clic en el icono de estrella, se puede eliminar de la lista de favoritos. Después de hacer clic en eliminar la alerta, el capítulo es retirado del listado de la aplicación, tal como se presenta en la figura siguiente:

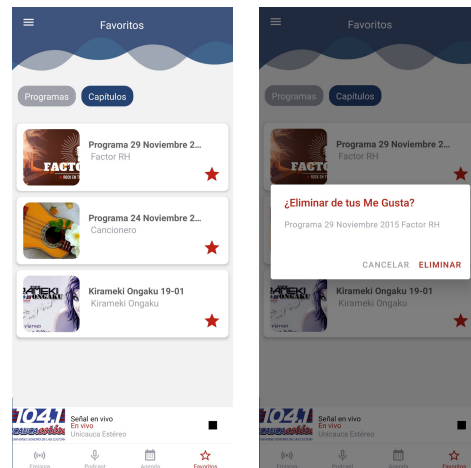


Figura 68: Vista de podcast favoritos

## B.10 REPRODUCTOR DE AUDIO

Al hacer clic en la barra de reproducción, se despliega una interfaz con la imagen del programa que se está escuchando, la información que es la señal en vivo y la opción de pausar o de reproducir el contenido, como se presenta en la siguiente figura:



Figura 69: Reproductor principal de la aplicación móvil

La aplicación Unicauca Estéreo se constituye en un referente a nivel nacional y en una opción relevante, moderna y pertinente para acercar a la comunidad universitaria a su emisora, que es un espacio abierto de conocimiento, expresión, información y entretenimiento.

## C FORMATO ENCUESTA PRUEBAS DE USABILIDAD

# Lista de Tareas App Móvil Unicauca Estéreo

(Medición de parámetros de usabilidad)

Por favor, luego de realizar cada tarea calificar qué tan fácil o difícil fue realizarla en la escala del 1 al 7 que se muestra inmediatamente después, en la escala 1 significa “Muy fácil” y 7 significa “Muy difícil”.

### C.1 PARAR LA REPRODUCCIÓN DE LA SEÑAL EN VIVO

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.2 REPRODUCIR UN PODCAST DEL PROGRAMA LLAMADO FACTOR RH

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.3 VISUALIZAR EL DETALLE DE UNA NOTICIA

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.4 VISUALIZAR EL DETALLE DE UN TWEET

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.5 VISUALIZAR EL DETALLE DE UN EVENTO DENTRO DE LA SECCIÓN “AGENDA”

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil



## C.6 VISUALIZAR EL DETALLE DE UN PROGRAMA DE LA SECCIÓN “PODCAST”

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

## C.7 VISUALIZAR LA PARRILLA DE PROGRAMACIÓN DE LA EMISORA DESDE EL MENÚ LATERAL

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

## C.8 AGREGAR EL PROGRAMA FACTOR RH A FAVORITOS DESDE EL DETALLE DE UN PROGRAMA

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.9 ACTIVAR NOTIFICACIONES PARA EL PROGRAMA FAVORITO AGREGADO ANTERIORMENTE

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.10 ELIMINAR EL PROGRAMA AGREGADO ANTERIORMENTE DE LA LISTA DE FAVORITOS DESDE LA SECCIÓN “FAVORITOS”

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.11 AGREGAR UN PODCAST DEL PROGRAMA FACTOR RH A FAVORITOS DESDE EL DETALLE DE UN PROGRAMA

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.12 ELIMINAR EL PODCAST AGREGADO ANTERIORMENTE DE LA LISTA DE FAVORITOS

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.13 VISUALIZAR LA HISTORIA E INFORMACIÓN DE LA EMISORA DESDE EL MENÚ LATERAL

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.14 VISUALIZAR LA PÁGINA WEB DE LA EMISORA DESDE EL MENÚ LATERAL

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

### C.15 VISUALIZAR EL INSTAGRAM DE LA EMISORA DESDE EL MENÚ LATERAL

¿Que tan fácil o difícil fue realizar la tarea?

	1	2	3	4	5	6	7	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil