



**UNIVERSIDAD
NACIONAL
DE LOJA**



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

**“IMPLEMENTACIÓN DEL SERVICIO DE
RADIO DE LA UNIVERSIDAD NACIONAL
DE LOJA PARA LOS MEDIOS DE
DIFUSIÓN DIGITAL Y DISPOSITIVOS
MÓVILES”**

Tesis previa a la
obtención del título de
Ingeniero en Sistemas

AUTOR:

Neira Balcázar Jonathan Ezequiel

DIRECTOR:

Ing. Cueva-Hurtado, Mario-Enrique Mg. Sc

LOJA-ECUADOR

2017

CERTIFICACIÓN DE DIRECTOR

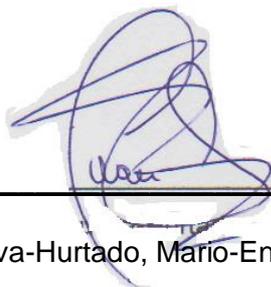
Ing. Cueva-Hurtado, Mario-Enrique, Mg. Sc

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DEL AREA DE LA ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Haber dirigido, revisado y corregido en todas sus partes el desarrollo del Trabajo de Titulación de Ingeniería en Sistemas titulado **“IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES”**, con autoría del egresado Jonathan Ezequiel Neira Balcázar. En razón de que el mismo reúne a satisfacción los requisitos de fondo y forma, exigidos para la investigación de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal designado para el efecto.

Loja, 24 de Noviembre 2016



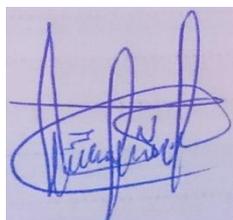
Ing. Cueva-Hurtado, Mario-Enrique, Mg. Sc

DIRECTOR DE TESIS

AUTORÍA

Yo **JONATHAN EZEQUIEL NEIRA BALCÁZAR**, declaro ser autor del presente trabajo de titulación previo a la obtención del título de Ingeniero en Sistemas denominado **“IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES”**, y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de titulación en el repositorio Institucional-Biblioteca Virtual.



Firma: _____

Cédula: 1104835945

Fecha: 13-11-2017

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

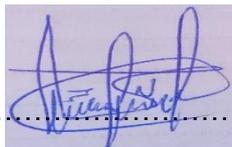
Yo **JONATHAN EZEQUIEL NEIRA BALCÁZAR**, declaro ser el autor de la tesis titulada: **IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES**, como requisito para optar el grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, publique al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Digital Institucional, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los trece días del mes noviembre del dos mil diecisiete.

Firma:



Autor: Jonathan Ezequiel Neira Balcázar

Cédula: 1104835945

Dirección: Loja (Cdla. Esteban Godoy, Oswaldo Flores y Galileo Galilei)

Correo electrónico: jeneirab@unl.edu.ec

Teléfono: 2545-940 **Celular:** 0997991524

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Cueva-Hurtado, Mario-Enrique, Mg. Sc

Tribunal de Grado: Ing. Jácome-Galarza, Luís-Roberto, Mg. Sc

Ing. Zúñiga-Tinizaray, Alfredo-Vinicio, Mg. Sc

Ing. Chamba-Romero, Gastón-Rene, Mg. Sc

DEDICATORIA

El presente trabajo de titulación está dedicado a Dios y a la Santísima Virgen del Cisne, por haberme dado la vida y permitirme llegar hasta este momento tan importante de mi formación profesional.

A mi Madre Dolores Balcázar y Tía Marcia Balcázar por el apoyo incondicional que me brindo en todo momento por sus sabios consejos que me han permitido ser una persona de bien, pero más que nada, por su amor, de igual a mi hermana, sobrinos, cuñado y amigos tanto del colegio como la universidad por brindarme su amistad y ser parte fundamental en toda mi formación.

El Autor

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por darme la fuerza necesaria para seguir adelante y guiarme en todo momento para cumplir cada una de mis metas.

Agradezco de manera especial a la Universidad Nacional de Loja, noble institución que nos ha alojado como nuestro segundo hogar y nos ha permitido tener el prestigio de estudiar en ella y formarnos como profesionales y mejores personas para bien de nuestro país y toda nuestra colectividad Lojaña.

Agradezco a mi Madre, Tía, hermana y demás familiares por darme constantemente su apoyo durante mi Facultad Universitaria.

De la misma manera agradezco a cada uno de los docentes que conforman la carrera de Ingeniería de Sistemas del Facultad de la Energía las Industrias y los Recursos Naturales no Renovables y, miembros de la Unidad de Telecomunicaciones e Información y Radio Universitaria 98.5FM quienes con su dedicación, paciencia y empeño supieron inculcar en mí el conocimiento necesario que hoy me permite culminar con nuestra formación académica e incluirnos en el mundo profesional.

De manera especial al docente Mario Cueva, por su apoyo brindado en la cristalización del presente trabajo de titulación y tan acertada dirección de la misma.

Jonathan Ezequiel Neira Balcázar.

Tabla de Contenidos

Índice General

CERTIFICACIÓN DE DIRECTOR	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
Tabla de Contenidos	VII
Índice General.....	VII
Índice de Figuras	XI
Índice de Tablas	XV
1. Título	- 1 -
2. Resumen	- 2 -
2.1. Abstract	- 3 -
3. Introducción	- 4 -
4. Revisión de literatura	- 6 -
4.1. Radio Universitaria	- 6 -
4.1.1. Antecedentes.....	- 6 -
4.1.2. Ubicación	- 6 -
4.1.3. Orgánico estructural de la universidad nacional de Loja.....	- 7 -
4.1.4. ¿Quién es la radio universitaria?	- 8 -
4.1.4.1. Misión	- 9 -
4.1.4.2. Visión	- 9 -
4.1.4.3. Valores	- 9 -
4.2. Internet.....	- 9 -
4.2.1. Desarrollo de internet.....	- 10 -
4.2.2. Característica esencial de internet	- 11 -
4.2.3. Principales servicios disponibles de internet.....	- 11 -
4.2.4. Aplicaciones Multimedia	- 14 -
4.2.5. Definición de streaming	- 16 -

4.3.	Medios digitales	- 27 -
4.3.1.	Características de los medios digitales.....	- 27 -
4.3.2.	Ventajas de los medios digitales	- 28 -
4.4.	Radio en internet.....	- 29 -
4.4.1.	Ventajas de radio en internet	- 29 -
4.4.2.	Problemas de radio en internet.....	- 29 -
4.4.3.	Radios universitarias en ecuador	- 30 -
4.5.	Aplicaciones Móviles	- 30 -
4.5.1.	Teléfono Móvil	- 31 -
4.5.2.	Smartphone	- 31 -
4.5.3.	Aplicación Móvil	- 31 -
4.5.4.	Tipos de aplicaciones Móviles	- 31 -
4.5.5.	Sistemas Operativos para dispositivos móviles.....	- 33 -
4.5.6.	Tecnologías y herramientas utilizadas en el desarrollo de la aplicación móvil	- 37 -
4.5.7.	Metodologías para aplicaciones móviles	- 39 -
4.5.8.	Metodología de desarrollo software	- 46 -
4.5.9.	Metodología para revisión sistemática.....	- 46 -
5.	MATERIALES Y MÉTODOS	- 48 -
5.1.	Materiales	- 48 -
5.2.	Métodos y técnicas.....	- 48 -
5.2.1.	Métodos	- 48 -
5.2.2.	Técnicas.....	- 49 -
5.2.3.	Metodología.....	- 49 -
6.	RESULTADOS	- 50 -
6.1.	OBJETIVO 1: ANALIZAR LA SITUACIÓN ACTUAL DEL SERVICIO DE LA RADIO UNIVERSITARIA EN LOS MEDIOS DE DIFUSIÓN DIGITAL.....	- 50 -
6.1.1.	Situación actual de la radio	- 50 -
6.1.2.	Datos generales.....	- 51 -
6.1.3.	Ubicación y alturas	- 51 -
6.1.4.	Características técnicas.....	- 51 -
6.1.5.	Parrilla de programación.....	- 53 -
6.1.6.	Infraestructura de la red de la Radio Universitaria	- 53 -
6.1.7.	Diagrama de servidores.....	- 55 -
6.1.8.	Equipos de producción y emisión de la radio	- 55 -

6.1.9. Índice de radioyentes	- 59 -
6.1.10. Viabilidad de implementación del streaming.....	- 59 -
6.2. OBJETIVO 2: DETERMINAR LOS REQUERIMIENTOS DE HARDWARE Y SOFTWARE PARA EL STREAMING.....	- 61 -
6.2.1. Selección de software:.....	- 61 -
6.2.2. Determinación de Hardware a emplear para streaming.....	- 74 -
6.2.3. Ancho de banda.....	- 75 -
6.3. OBJETIVO 3: IMPLEMENTAR EL STREAMING PARA LA RADIO UNIVERSITARIA.....	- 76 -
6.3.1. Instalación y configuración de los servicios para streaming.....	- 77 -
6.4. OBJETIVO 4: DESARROLLAR E IMPLEMENTAR UNA APLICACIÓN MÓVIL MULTIPLATAFORMA PARA ACCEDER A LA RADIO UNIVERSITARIA.	- 87 -
6.4.1. Primera Fase: Análisis	- 88 -
7. DISCUSIÓN.....	- 124 -
7.1. Desarrollo de la propuesta alternativa	- 124 -
7.2. Analizar la situación actual del servicio de la Radio Universitaria en los medios de difusión digital	- 124 -
7.3. Determinar los requerimientos de hardware y software para el streaming	- 124 -
-	
7.4. Implementar streaming en la Radio Universitaria	- 125 -
7.5. Desarrollar e implementar de una aplicación móvil multiplataforma para acceder a la Radio Universitaria.....	- 125 -
8. CONCLUSIONES	- 127 -
9. RECOMENDACIONES.....	- 129 -
10. BIBLIOGRAFÍA.....	- 130 -
11. ANEXOS	- 134 -
Requerimientos no Funcionales.....	- 195 -
Introducción	- 198 -
Propósito.....	- 198 -
Alcance	- 198 -
Definiciones, Siglas y Abreviatura.....	- 198 -
Referencias	- 198 -
Vista global.....	- 199 -
Representación de la Arquitectura	- 200 -
Objetivos y Restricciones de la Arquitectura.....	- 201 -

Vista de Casos de Uso.....	- 201 -
Modelo de casos de Uso	- 202 -
Especificación de casos de uso.	- 202 -
Vista Lógica	- 208 -
Diagrama de clases.....	- 208 -
Diagrama de secuencia	- 208 -
Vista de Procesos	- 210 -
Vista de Despliegue.....	- 215 -
Diagrama de componentes	- 215 -
Vista Física	- 216 -
Diagrama de Despliegue	- 216 -

Índice de Figuras

Figura 1. Ubicación de la Radio Universitaria.....	- 7 -
Figura 2. Orgánico Estructural de la Universidad Nacional de Loja [2].....	- 8 -
Figura 3. El modelo cliente / servidor[5].....	- 11 -
Figura 4. Conexión Telnet [6].....	- 12 -
Figura 5. Transferencia de archivos (FTP)[7].....	- 13 -
Figura 6. Arquitectura de las aplicaciones multimedia[4].....	- 14 -
Figura 7. Arquitectura streaming [8].....	- 18 -
Figura 8. Forma de Operación Unicast [13].....	- 22 -
Figura 9. Forma de operación Multicast[14].....	- 23 -
Figura 10. Arquitectura Android [10].....	- 36 -
Figura 11. Ciclo de vida de Scrum para móviles[21].....	- 41 -
Figura 12. Trasmisión de la Radio Universitaria.....	- 50 -
Figura 13. Backbone de la Universidad Nacional de Loja.....	- 54 -
Figura 14. Diagrama de servidores de la Universidad Nacional de Loja.....	- 55 -
Figura 15. Equipos de emisión y producción de la Radio Universitaria.....	- 56 -
Figura 16. Arquitectura del sistema streaming.....	- 76 -
Figura 17. Diagrama de equipos de Radio Universitaria con conexión de Cliente y Servidor streaming.....	- 76 -
Figura 18. Diagrama de Servidores de UNL integrado el servidor de Radio Universitaria.....	- 77 -
Figura 19 Código de acceso a streaming en plantilla web.....	- 79 -
Figura 20. Plantilla Web de la Radio Universitaria.....	- 79 -
Figura 21. Radio Universitaria en plataforma TuneIn.....	- 80 -
Figura 22. Componentes para evaluación del sistema.....	- 81 -
Figura 23. Monitor de Icecast2 con 943 usuarios.....	- 83 -
Figura 24. Rendimiento del sistema con relación al tiempo en Apache JMeter.....	- 83 -
Figura 25. 943 Conexiones concurrentes.....	- 84 -
Figura 26. Ancho servido en 943 conexiones concurrentes.....	- 85 -
Figura 27. Esquema de equipos para la prueba.....	- 85 -
Figura 28. Rendimiento del Servidor VS Número de conexiones.....	- 86 -
Figura 29. Porcentaje de memoria 0.90% y CPU 66.50% en 943 conexiones concurrentes.....	- 86 -
Figura 30. Diseño del Sistema.....	- 94 -
Figura 31. Prototipo: Pantalla Principal Radio Online.....	- 96 -
Figura 32. Pantalla: Login / Register.....	- 97 -
Figura 33. Pantalla: Navegación lateral.....	- 98 -
Figura 34. Pantalla: Navegación Lateral.....	- 99 -
Figura 35 Prototipo pantalla: Noticias.....	- 100 -
Figura 36 Prototipo de pantalla: Contenido de noticias.....	- 101 -
Figura 37 Prototipo de pantalla: Programación.....	- 102 -
Figura 38 Prototipo de pantalla: Programación diaria.....	- 103 -
Figura 39 Prototipo de pantalla: Programas grabados.....	- 104 -
Figura 40 Prototipo de pantalla: Quienes somos.....	- 105 -
Figura 41. Llamar la librería leanbackPlayer.....	- 107 -
Figura 42. Conexión con el flujo de audio emitido por el Servidor.....	- 107 -
Figura 43 Login/Register.....	- 107 -
Figura 44. Código para la generación del menú de navegación lateral.....	- 108 -

Figura 45	Formulario registrar nuevo usuario.....	108 -
Figura 46.	Código de la template “Quienes Somos”	109 -
Figura 47	Lista de noticias	109 -
Figura 48	Contenido de noticias de acuerdo al id recibido	109 -
Figura 49.	Lista de los días de la semana para la programación	110 -
Figura 50	Lista de programas de acuerdo al día seleccionado.....	110 -
Figura 51	Listar programas grabados.....	111 -
Figura 52.	PF: Acceder al streaming de la radio	116 -
Figura 53.	P.F: Visualización de información de “Quienes Somos”.....	117 -
Figura 54.	PF: Visualización de parrilla de programación	118 -
Figura 55.	PF: Visualización de información de redes sociales	119 -
Figura 56	PF: Acceder a contenido privado	120 -
Figura 57:	PF: Registrar nuevo usuario	122 -
Figura 58.	BackBone de la Universidad Nacional de Loja	135 -
Figura 59.	Primera Hoja de datos técnicos Estaciones de radiodifusión en FM	136 -
Figura 60.	Segunda Hoja de datos técnicos Estaciones de radiodifusión en FM...-	137 -
Figura 61.	Comando para acceder a la lista de repositorios.....	138 -
Figura 62.	Comando de instalación de Icecast2.....	138 -
Figura 63.	Ingreso de password e instalación de paquetes.	139 -
Figura 64.	Descarga e instalación de paquetes	139 -
Figura 65.	Comando para ingresar al archivo de configuración Icecast2.....	139 -
Figura 66.	Archivo de configuración de servidor Icecast2.....	140 -
Figura 67.	Línea de configuración de clientes	140 -
Figura 68.	Configuraciones de passwords para el servidor streaming Icecast2	140 -
Figura 69.	Configuración del hostname	141 -
Figura 70.	Configuración del hostname con el dominio	141 -
Figura 71.	Comando para abrir el archivo default de Icecast2.....	141 -
Figura 72.	Contenido de archivo default de Icecast2	141 -
Figura 73.	Archivo default modificado.....	142 -
Figura 74.	Comando para instalar Butt Broadcast	142 -
Figura 75.	Ventana de configuraciones de Butt Broadcast.....	143 -
Figura 76.	Ventana con datos de configuración de Butt Broadcast.....	144 -
Figura 77.	Datos de información para servidores.....	144 -
Figura 78.	Ventana de configuración de audio, códec y bitrate en Butt Broadcast -	145 -
Figura 79.	Comando para iniciar servicio de Icecast2.....	145 -
Figura 80.	Inicio satisfactorio de servicio de Icecast2	146 -
Figura 81.	Página de administración de Icecast2.....	146 -
Figura 82.	Conexión e inicio de streaming en Butt Broadcast	147 -
Figura 83.	Código para acceder streaming	148 -
Figura 84.	Sitio oficial de descargas de Apache JMeter	149 -
Figura 85.	Ubicación de archivo ApacheJMeter.jar	150 -
Figura 86.	Pantalla principal de Apache JMeter	150 -
Figura 87.	Agregar grupo de hilos (usuarios)	151 -
Figura 88.	Configuración de grupo de usuarios.....	151 -
Figura 89.	Agregar muestreador con petición http	151 -
Figura 90.	Peticiones HTTP	152 -
Figura 91.	Agregar receptores	152 -
Figura 92.	Orden jerárquico	153 -
Figura 93.	Página de descarga de Winbox.....	153 -

Figura 94. Pantalla principal de Winbox	- 154 -
Figura 95. Pantalla de TORCH (Running) Winbox	- 154 -
Figura 96. Inicio de escenario en Apache JMeter	- 155 -
Figura 97. Lista de interfaces en Winbox.....	- 155 -
Figura 98. Monitor de status del servidor Icecast2.....	- 156 -
Figura 99. Página Principal de Tuneln.....	- 157 -
Figura 100. Página de Tuneln opción (Radiodifusoras).....	- 157 -
Figura 101. Página de Tuneln Radiodifusoras opción (Add a station).....	- 158 -
Figura 102. Formulario para agregar una nueva estación	- 160 -
Figura 103. Formulario para agregar una nueva estación	- 160 -
Figura 104. Formulario para agregar una nueva estación	- 161 -
Figura 105. Envío del Formulario	- 161 -
Figura 106. Mensaje de finalización de registro de nueva estación	- 161 -
Figura 107. Correo electrónico de aceptación de la estación de radio en Tuneln ..	- 162 -
Figura 108. Radio Universitaria en plataforma Tunel.....	- 162 -
Figura 109. Máquina virtual Mac OS X10.11	- 163 -
Figura 110. Comando para emular en iOS	- 163 -
Figura 111. Inicio de emulación en iOS	- 164 -
Figura 112. Pantalla principal Online	- 164 -
Figura 113. Ejecución de streaming.....	- 165 -
Figura 114. Submenú de aplicación.....	- 165 -
Figura 115. Visualización de red social Facebook	- 166 -
Figura 116. Visualización de información Quienes Somos	- 166 -
Figura 117. Parrilla de programación.....	- 166 -
Figura 118. Programación diaria	- 167 -
Figura 119 Noticias.....	- 167 -
Figura 120 Login/Register	- 167 -
Figura 121 Registrar nuevo usuario.....	- 168 -
Figura 122 Logueo con red social Facebook.....	- 168 -
Figura 123. Registrar cuenta de desarrollador en Google Play	- 169 -
Figura 124. Finalización del registro en la consola de Google Play	- 170 -
Figura 125. Añadir nueva aplicación Android	- 170 -
Figura 126. APK para producción	- 170 -
Figura 127. Subir APK de producción.....	- 171 -
Figura 128. Ficha de Play Store.....	- 171 -
Figura 129. Aplicación publicada en Google Play	- 171 -
Figura 130 Escalamiento Horizontal	- 172 -
Figura 131 Escalamiento Horizontal	- 173 -
Figura 132 Modelo de casos de uso	- 202 -
Figura 133 Diagrama de secuencia Ingresar al sistema	- 209 -
Figura 134 Diagrama de secuencia Gestionar programas.....	- 209 -
Figura 135 Diagrama de secuencia Gestionar programas grabados	- 209 -
Figura 136 Proceso ingresar al sistema	- 211 -
Figura 137 Proceso agregar programa.....	- 212 -
Figura 138 Proceso agregar programa grabado	- 212 -
Figura 139 Proceso modificar programa.....	- 213 -
Figura 140 Proceso modificar programa grabado	- 213 -
Figura 141 Proceso eliminar programa.....	- 214 -
Figura 142 Proceso eliminar programa grabado	- 214 -

Figura 143 Diagrama de componentes.....	- 215 -
Figura 144 Diagrama de despliegue de SWAR.....	- 216 -

Índice de Tablas

TABLA I. CLASIFICACIÓN DE LAS APLICACIONES MULTIMEDIA.....	- 15 -
TABLA II. DATOS GENERALES DE LA RADIO UNIVERSITARIA	- 51 -
TABLA III. UBICACIÓN Y ALTURAS DE LOS ESTUDIOS Y TRASMISORES DE LA RADIO UNIVERSITARIA	- 51 -
TABLA IV. CARACTERÍSTICAS TÉCNICAS DE LA RADIO UNIVERSITARIA.....	- 51 -
TABLA V. CARACTERÍSTICAS TÉCNICAS DEL ENLACE ESTUDIO-TRASMISOR-	52 -
-	
TABLA VI. POLARIZACIÓN HORIZONTAL / VERTICAL	- 52 -
TABLA VII. FRANJAS HORARIAS	- 53 -
TABLA VIII. EQUIPOS DE RED DE LA RADIO UNIVERSITARIA	- 54 -
TABLA IX. INVENTARIO HARDWARE DE LA RADIO UNIVERSITARIA.....	- 57 -
TABLA X. INVENTARIO SOFTWARE DE LA RADIO UNIVERSITARIA	- 58 -
TABLA XI. NECESIDADES DE LA RADIO UNIVERSITARIA	- 59 -
TABLA XII. ICECAST2	- 61 -
TABLA XIII. VENTAJAS Y DESVENTAJAS DE SERVIDOR SHOUTCAST	- 62 -
TABLA XIV. VENTAJAS Y DESVENTAJAS DE SERVIDOR DARWIN STREAMING SERVER.....	- 63 -
TABLA XV.COMPARATIVA DE SERVIDORES STREAMING	- 64 -
TABLA XVI.VENTAJAS Y DESVENTAJAS DE CLIENTE BUTT BROADCAST.....	- 65 -
TABLA XVII. VENTAJAS Y DESVENTAJAS DE CLIENTE ICES2	- 67 -
TABLA XVIII. VENTAJAS Y DESVENTAJAS DE CLIENTE VLC	- 67 -
TABLA XIX. COMPARATIVA DE DSPS.....	- 68 -
TABLA XX. VENTAJAS Y DESVENTAJAS DE CÓDEC MP3.....	- 69 -
TABLA XXI. VENTAJAS Y DESVENTAJAS CÓDEC OGG VORBIS	- 70 -
TABLA XXII. COMPARATIVA DE CÓDECOS	- 71 -
TABLA XXIII. VENTAJAS, DESVENTAJAS Y REQUISITOS DEL SISTEMA DE LA PLATAFORMA DEBIAN.....	- 71 -
TABLA XXIV. VENTAJAS, DESVENTAJAS Y REQUISITOS DEL SISTEMA DE LA PLATAFORMA CENTOS	- 72 -
TABLA XXV. CARACTERÍSTICAS MÍNIMAS PARA SERVIDOR STREAMING	- 74 -
TABLA XXVI. CARACTERÍSTICAS MÍNIMAS PARA EL CLIENTE (PROCESADOR DE SEÑAL DIGITAL).....	- 74 -
TABLA XXVII. HARDWARE PARA SERVIDOR ICECAST2.....	- 75 -
TABLA XXVIII. HARDWARE PARA CLIENTE (DSP)	- 75 -
TABLA XXIX. EQUIPOS PARA EVALUACIÓN DEL SISTEMA.....	- 81 -
TABLA XXX. RESULTADOS PRUEBA NO. 1: VALORES PROMEDIOS EMULANDO 1000 USUARIOS.....	- 82 -
TABLA XXXI. RESULTADOS PRUEBA NO. 3: VALORES PROMEDIOS DEL REGISTRO	- 84 -
TABLA XXXII. EQUIPOS PARA PRUEBA DE RENDIMIENTO.....	- 85 -
TABLA XXXIII. COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE APLICACIONES MÓVILES.....	- 87 -
TABLA XXXIV. REQUERIMIENTOS FUNCIONALES	- 89 -
TABLA XXXV. REQUERIMIENTOS NO FUNCIONALES.....	- 90 -
TABLA XXXVI. MÓDULOS Y PROCESOS DE LA APLICACIÓN	- 90 -
TABLA XXXVII. PLANIFICACIÓN DE FASES	- 93 -
TABLA XXXVIII. STORYCARD DE LA PANTALLA PRINCIPAL.....	- 97 -

TABLA XXXIX. STORYCARD DE LOGIN/REGISTER	- 98 -
TABLA XL. STORYCARD DE REGISTRAR	- 99 -
TABLA XLI. STORYCARD DE NAVEGACIÓN LATERAL	- 100 -
TABLA XLIII. STORYCARD DE NAVEGACIÓN LATERAL	- 100 -
TABLA XLIII. STORYCARD DE NAVEGACIÓN LATERAL	- 101 -
TABLA XLIV. STORYCARD DE PROGRAMACIÓN	- 102 -
TABLA XLVI. STORYCARD DE PROGRAMACIÓN DIARIA.....	- 103 -
TABLA XLVII. STORYCARD DE PROGRAMAS GRABADOS.....	- 104 -
TABLA XLVIII. STORYCARD DE QUIENES SOMOS	- 105 -
TABLA XLVIII. PU: OBTENCIÓN DEL AUDIO STREAMING DE LA RADIO.....	- 111 -
TABLA XLIX. PU: OBTENCIÓN DE INFORMACIÓN QUIENES SOMOS LA RADIO UNIVERSITARIA	- 112 -
TABLA L. PU: LOGUEO DE USUARIO EN LA PANTALLA LOGIN/REGISTER ...	- 113 -
TABLA LI. PU: OBTENCIÓN DE LA INFORMACIÓN DE PÁGINA INSTITUCIONAL Y REDES SOCIALES	- 113 -
TABLA LII. VERIFICACIÓN DE PANTALLAS	- 114 -
TABLA LIII. VERIFICACIÓN DE REQUERIMIENTOS.....	- 115 -
TABLA LIV. PF: ACCEDER AL STREAMING DE LA RADIO.....	- 115 -
TABLA LV. PF: VISUALIZACIÓN DE INFORMACIÓN DE QUIENES SON LA RADIO UNIVERSITARIA	- 116 -
TABLA LVI. PF: VISUALIZACIÓN DE PARRILLA DE PROGRAMACIÓN.....	- 117 -
TABLA LVII. PF: VISUALIZACIÓN DE INFORMACIÓN DE REDES SOCIALES ..	- 118 -
TABLA LVIII. PF: VISUALIZACIÓN DE INFORMACIÓN DE REDES SOCIALES .-	- 119 -
TABLA LIX. PF: REGISTRAR NUEVO USUARIO	- 121 -
TABLA LX. CARACTERÍSTICAS MÍNIMAS PARA INSTALAR Y EJECUTAR LA APLICACIÓN	- 122 -
TABLA LXI. DATOS PARA HOSTNAME.....	- 141 -
TABLA LXII. DATOS DE CONFIGURACIÓN DE SERVIDOR EN BUTT BROADCAST - 143 -	
TABLA LXIII. DATOS DE CONFIGURACIÓN DE STEAMS INFO	- 144 -
TABLA LXIV. DATOS DE CONFIGURACIÓN DE AUDIO, CÓDEC Y BITRATE ...-	- 145 -
TABLA LXV. DATOS PARA AGREGAR RADIO UNIVERSITARIA A TUNEIN.....	- 158 -
TABLA LXVI. REQUISITO FUNCIONAL 001	- 193 -
TABLA LXVII. REQUISITO FUNCIONAL 002	- 193 -
TABLA LXVIII. REQUISITO FUNCIONAL 003	- 194 -
TABLA LXIX. REQUISITO FUNCIONAL 004	- 194 -
TABLA LXX. REFERENCIAS.....	- 198 -
TABLA LXXII. REPRESENTACION DE LA ARQUITECTURA	- 200 -
TABLA LXXIII. CASO DE USO AUTENTICACIÓN CON CORREO	- 202 -
Tabla LXXIV. CASO DE USO REESTABLECER CONTRASEÑA	- 203 -
TABLA LXXV. CASO DE USO GESTIONAR PERFIL DE USUARIO	- 204 -
TABLA LXXVI. CASO DE USO GESTIONAR PROGRAMAS	- 205 -
TABLA LXXVII. CASO DE USO GESTIONAR PROGRAMAS GRABADOS	- 206 -

1. Título

**“IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA
UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS
DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES”**

2. Resumen

En el presente Trabajo de Titulación es el resultado de la implementación de un sistema de transmisión de streaming para la Radio Universitaria, el cual permite la codificación y distribución de su contenido en audio digital en navegadores web, proporcionando mayor cobertura de transmisión, y radioescuchas conectados a la radio. Además, se desarrolla una aplicación móvil multiplataforma para acceder al streaming.

Para el análisis y selección de herramientas para la implementación de streaming, se usa la metodología de Barbara Kitchenham. Implementado el streaming, a través de las herramientas servidor Icecast2, cliente Butt Broadcast, códec MP3 y un ancho de banda mínimo de 256 kbps para la comunicación entre el cliente y servidor. Para el desarrollo de la aplicación móvil, se utiliza la metodología de desarrollo ágil Mobile-D, en sus cinco fases: exploración, inicialización, producción, estabilización y pruebas. En la codificación se emplea el framework Ionic, el cual utiliza HTML5, CSS y JavaScript.

2.1. Abstract

This present degree work is the result of the implementation of a streaming transmission system for the University Radio that allows the encoding and distribution of its content in audio digital in web browsers. It provides a greater coverage of transmission to the radio listeners. In addition, a multi-platform mobile application is developed to access the streaming.

The Barbara Kitchenham methodology is used for the analysis and selection of tools to the implementation of streaming, Once, it was Implemented through the server tools Icecast2, Butt Broadcast customer, MP3 codec and a minimum bandwidth of 256 kbps to develop the communication between the client and the server. To carry out the mobile application, the Mobile-D agile development methodology is used in its five phases: exploration, initialization, production, stabilization and testing. Also, the Ionic framework is used in the encoding which uses HTML5, CSS and JavaScript.

3. Introducción

Una radio en Internet dentro de una universidad permite amplias posibilidades de intercambio de información, no solo dando a conocer acontecimientos pertenecientes a la institución, sino generar un ambiente de colaboración donde se puedan compartir contenidos para fomentar la parte cultural y educativa, cumpliendo un rol fundamental en la difusión del conocimiento y de esta manera tener una relación continua con la colectividad en general.

Para llevar a cabo el presente trabajo de titulación se lo realiza en distintas fases, la primera fase es el análisis a la problemática en la radio de la Universidad Nacional de Loja (UNL), concerniente a la emisión de audio en un medio análogo, presentando el problema principal la cobertura limitada para ser escuchada por usuarios fuera de la ciudad de Loja.

En la segunda fase, se realiza una revisión sistemática para el análisis y selección de las herramientas para la implementación del streaming, para ello se utiliza la metodología de Barbara Kitchenham.

En la fase de implementación se utiliza el servidor Icecast2, cliente Butt Broadcast y códec MP3, los cuales son instalados y configurados en dos equipos, uno para el cliente y otro para el servidor, estos se ubican en las instalaciones de la radio y el departamento de la Unidad de Telecomunicaciones e Información (UTI) respectivamente. En la parte del servidor se configura el número de clientes, hostname, passwords de source, etc, este cumple la función de distribuir el streaming (audio) a los usuarios. En el cliente Butt Broadcast se configura el códec, entrada de audio, bitrate y conexión hacia el servidor, encargado de enviar el flujo hacia el servidor. En esta misma fase se realiza la evaluación y rendimiento del sistema, para ello se realizan las pruebas de funcionalidad y capacidad, donde se utiliza los recursos disponibles en la UTI y las herramientas de apoyo Apache JMeter y Winbox, las cuales proporcionan información detallada tanto del número de usuarios como el ancho de banda.

En la cuarta fase se desarrolla una aplicación móvil, donde se utiliza la metodología de desarrollo ágil Mobile-D, en sus cinco fases: exploración, inicialización, producción, estabilización y pruebas. En la fase de exploración se determina la especificación de requerimientos. En la fase de inicialización se realiza el diseño, prototipos y configuración del ambiente del desarrollo. En la fase de producción se elabora la codificación de las funcionalidades de la aplicación, para los sistemas operativos móviles Android e iOS, para ello se utiliza el framework Ionic, el cual se compone de

HTML5, CSS y JavaScript. En la fase de pruebas se determina si es una aplicación de calidad y que cumple con los requerimientos iniciales.

4. Revisión de literatura

4.1. Radio Universitaria

La Universidad Nacional de Loja, es una Institución de Educación Superior, laica, autónoma, de derecho público, con personería jurídica y sin fines de lucro, de alta calidad académica y humanística, que ofrece formación en los niveles: técnico y tecnológico superior; profesional o de tercer nivel; y, de postgrado o cuarto nivel; que realiza investigación científico-técnica sobre los problemas del entorno, con calidad, pertinencia y equidad, a fin de coadyuvar al desarrollo sustentable de la región y del país, interactuando con la comunidad, generando propuestas alternativas a los problemas nacionales, con responsabilidad social; reconociendo y promoviendo la diversidad cultural y étnica y la sabiduría popular, apoyándose en el avance científico y tecnológico, en procura de mejorar la calidad de vida del pueblo ecuatoriano.[1]

4.1.1. Antecedentes

La Radio Universitaria nace del proyecto “Implementación de medios para la Universidad Nacional de Loja”, dentro del centro Radio Cine y Televisión que tiempo atrás funcionaba el cual tenía dos brazos ejecutores tanto la Radio Universitaria como la imagen misma, la escritura de la radio fue entregada luego de los papeleos correspondientes el 27 de Julio del 2000 sin embargo no se inauguró de manera inmediata por cuestiones de requerimientos posteriores que exigía en ese entonces CONATEL, dicha entidad en la actualidad ya no existe, de esta manera la radio se inaugura formalmente el 4 de Diciembre del 2001, en la presencia del Lic. Milton Andrade Tapia director en ese momento, donde actualmente funcionan las instalaciones del Área Agropecuaria de la universidad antes mencionada.

4.1.2. Ubicación

La Radio Universitaria se encuentra funcionando actualmente en las instalaciones del ex Cater de la Universidad Nacional de Loja ubicada en la Av. Reinaldo Espinoza y la Av. Lateral de Paso Ángel F. Rojas frente al campus universitario.



Figura 1. Ubicación de la Radio Universitaria

4.1.3. Orgánico estructural de la universidad nacional de Loja

La Radio Universitaria cuenta con un representante legal que es el Rector de la Universidad Nacional de Loja, de la misma manera la radio se encuentra contemplada dentro del departamento de Comunicación e Imagen Institucional, cabe recalcar que esto aún no se oficializa en la institución por ende dentro de este organigrama no se encuentra representado en el departamento antes mencionado debido a que la radio no ha recibido la atención correspondiente por parte de las autoridades de la Universidad Nacional de Loja.

Organigrama estructural de la Universidad Nacional de Loja

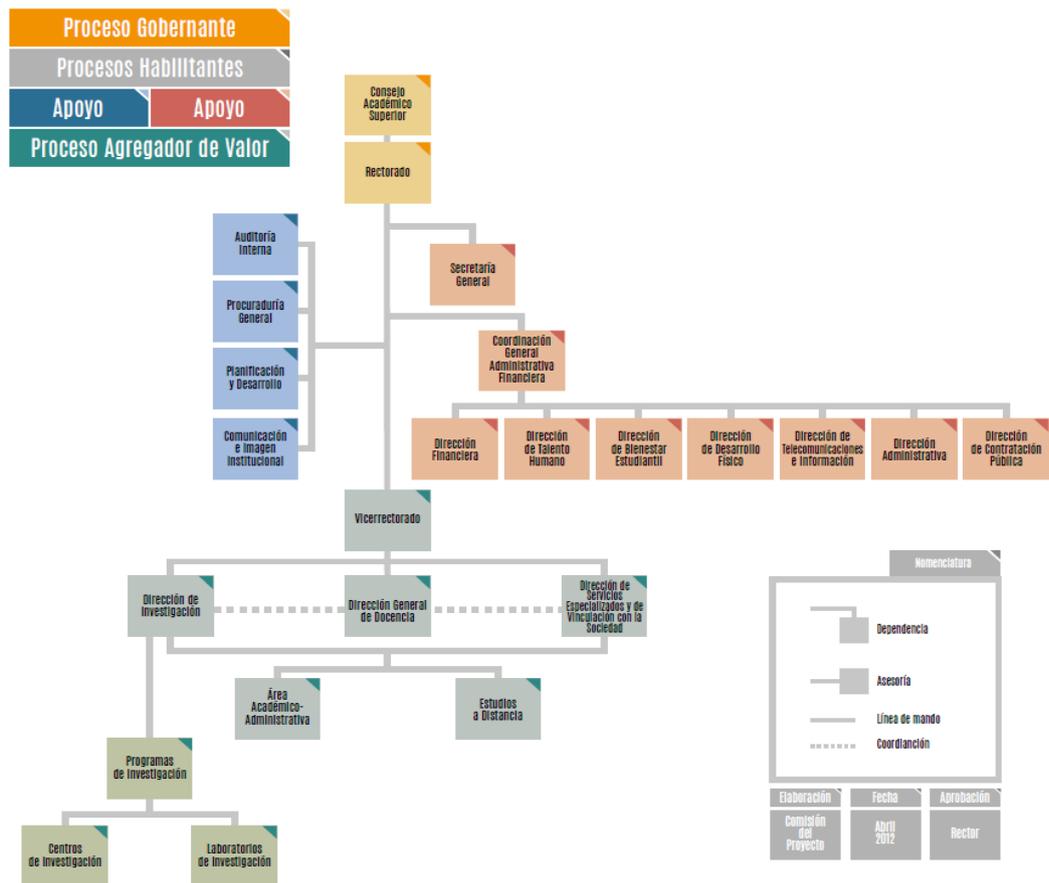


Figura 2. Orgánico Estructural de la Universidad Nacional de Loja [2]

4.1.4. ¿Quién es la radio universitaria?

Radio Universitaria 98.5 es un medio de comunicación público de la Universidad Nacional de Loja, para comunicarse con la comunidad a través de la coordinación, cooperación, consulta, intercambio y promoción del arte, la ciencia, la cultura y el desarrollo de Loja y la Región Sur.

Constituye en un laboratorio para las prácticas pre-profesionales de los estudiantes de la Carrera en Ciencias de la Comunicación Social; funciona las 24 horas al día, puede sintonizarse también en el portal de la Universidad Nacional de Loja, su concepción es de construcción de una red ciudadana en la que participan instituciones, organizaciones, y la sociedad en general, contribuyendo a reafirmar la identidad cultural y la diversidad social.

En nuestros estudios se realizan producciones radiofónicas como son: mensajes, cuñas, grabaciones de CDS, promocionales, documentales, reportaje, jingles, y otros productos comunicacionales con tecnología de punta.

La radio tiene equipos que permiten a la institución producir y emitir todo tipo de información enmarcada en programas científicos, académicos, de orientación, entretenimiento y otros, de acuerdo a las necesidades institucionales y sociales.

4.1.4.1. Misión

Somos un medio de comunicación público institucional, de difusión, y promoción de los saberes y culturas, constituyéndonos en un vínculo de opinión, reflexión análisis y formación del pensamiento universitario. Buscamos generar una sociedad mejor informada y educada partiendo de los principios y valores de nuestra Universidad Nacional de Loja. [3]

4.1.4.2. Visión

Aspiramos ser un medio de comunicación público institucional que muestre las realidades diferentes, garantice la opinión de todos/as y que sea un soporte en la vinculación de la Universidad Nacional de Loja con la colectividad, proponiendo una programación de alto contenido, con equipo técnico tecnológico; y profesional, eficiente y comprometido con los intereses institucionales para la generación de nuevas audiencias, con alta aceptación ciudadana. [3]

4.1.4.3. Valores

La Radio Universitaria en su quehacer cotidiano practica valores como:

- Excelencia
- Honradez
- Veracidad
- Respeto
- Responsabilidad

4.2. Internet

Internet ha evolucionado el mundo de la computación y las comunicaciones, además de ser un medio de difusión mundial de información, es un medio para la colaboración e interacción entre individuos sin importar su ubicación geográfica.

Desde un punto de vista técnico, se puede definir Internet como un inmenso conjunto de redes de computadoras que se encuentran interconectadas entre sí, dando lugar a la mayor red de redes de ámbito mundial. Los usuarios de cualquier computador

Desde un punto de vista más amplio, Internet constituye un fenómeno sociocultural de importancia creciente, una forma de entender las comunicaciones que está transformando el mundo, gracias a los millones de individuos que a diario tienen acceso a esta inagotable fuente de información, la mayor que jamás haya existido. [4]

4.2.1. Desarrollo de internet

Internet tiene su origen en un proyecto de investigación en redes realizado a finales de los años sesenta por DARPA (Defense Advance Research Projects Agency) del Departamento de Defensa de los EE.UU. el objetivo del proyecto era la construcción de un sistema de comunicación entre ordenadores altamente flexible y dinámico, que permitiera utilizar cualquier tipo de medio y tecnología de transmisión y que siguiera incluso ante la eventualidad de la destrucción de algunas partes de la red. DARPA desarrolló una nueva tecnología denominada conmutación de paquetes, cuya principal característica reside en fragmentar la información en porciones de una determinada longitud a las que se llama paquetes. [4]

En 1969 nació la red ARPANET, auténtica precursora de la posterior Internet, interconectando cuatro grandes ordenadores ubicados a lo largo de los Estados Unidos. ARPANET creció lentamente durante los años 70 hasta llegar a conectar unos 100 ordenadores a principios de los 80. Sin embargo, durante todos esos años, sirvió de banco de investigación, desarrollo y prueba de los pilares sobre los que se apoya hoy Internet. La adopción oficial de TCP/IP (Transmission Control Protocol/Internet Protocol) dentro de ARPANET en 1982 supuso un hito decisivo. Pronto surgieron otras redes independientes que también eligieron a TCP/IP para la comunicación entre sus equipos como CSNET (Computer Science NETWORK) y MILNET (red militar del Departamento de Defensa de EE.UU.). La interconexión entre ARPANET, MILNET y CSNET, que tuvo lugar en 1983. [4]

A mediados de los años 80 se produce un factor clave en el posterior despegue de Internet. Numerosos fabricantes empiezan a sacar al mercado equipos que se comunican mediante el protocolo TCP/IP, convirtiéndolo en el estándar de facto para la intercomunicación de ordenadores. [4]

Un último factor decisivo fue el nacimiento en 1986 de la red NSFnet. Con el objeto de facilitar el acceso a toda la comunidad científica americana a cinco grandes centros de supercomputadoras, la NSF (National Science Foundation), ante los impedimentos burocráticos para usar la red ARPANET, decidió crear una red propia que acabaría convirtiéndose en la auténtica espina dorsal de Internet. [4]

Desde finales de los 80, Internet ha experimentado un crecimiento exponencial en casi todos sus parámetros: número de computadores conectados, número de redes conectadas y tráfico mensual cursado. Este crecimiento no tiene precedentes en la historia de las comunicaciones, ni siquiera en la red telefónica. [4]

4.2.2. Característica esencial de internet

Una de las características esenciales de Internet es su descentralización: nadie gobierna Internet, cada red conectada conserva su independencia. Sin embargo, para que semejante anarquía funcione es necesaria la existencia de una serie de procedimientos y mecanismos de coordinación.

Es necesario tener claro que la conexión a Internet por parte de un usuario se efectúa siempre por medio de un proveedor de servicio de acceso a Internet. En función de las necesidades del usuario y de la oferta del proveedor las posibilidades de conexión son muy variadas, pudiendo ir desde una conexión permanente mediante una línea dedicada de gran capacidad, hasta un simple acceso de un modem.

4.2.3. Principales servicios disponibles de internet

Los servicios más conocidos proporcionados a través de Internet son, por una parte, las aplicaciones estándar de TCP/IP, como correo electrónico (SMTP), transferencia de archivos (FTP), conexión remota (TELNET), así como las nuevas aplicaciones que han incorporado a lo largo de los años, como el Sistema de Nombres de Dominio (DNS), y la World Wide Web (WWW).

4.2.3.1. Modelo cliente / servidor

Las aplicaciones del modelo de referencia TCP/IP utilizan un modelo cliente / servidor para comunicarse. Se denomina cliente al proceso que inicia el dialogo o solicita los recursos y servidor al proceso que responde a las solicitudes. Los principales componentes del esquema cliente / servidor son entonces los clientes, los servidores y la infraestructura de comunicaciones.

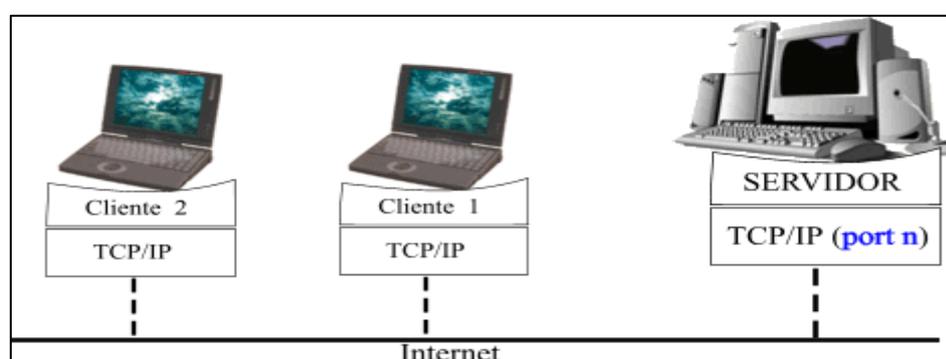


Figura 3. El modelo cliente / servidor[5]

En el modelo cliente / servidor, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece solo lo específico de cada usuario. Los usuarios interactúan con el servidor en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de

establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

4.2.3.2. Correo electrónico

El correo electrónico está basado en el protocolo SMTP (Simple Mail Transfer Protocol) definido en los RFCs 821 y 822. Este protocolo está dedicado a la transmisión de mensajes electrónicos sobre una conexión TCP. El protocolo especifica el formato de los mensajes definiendo la estructura de la información sobre el rendimiento, el destinatario, datos adicionales y naturalmente el cuerpo de los mensajes. Este protocolo no especifica como los mensajes deben ser editados. Es necesario tener un editor local o una aplicación nativa de correo electrónico. Una vez el mensaje es creado, el SMTP lo acepta y usa el protocolo TCP para enviarlo a un módulo SMTP de otra maquina implicadas. El módulo SMT "target" usará una aplicación de correo electrónico para almacenar los mensajes recibidos en el buzón.

4.2.3.3. Telnet

El protocolo Telnet permite a los usuarios conectarse a computadores remotos y utilizarlos desde el sistema local, mediante la emulación de terminal sobre una conexión TCP. Telnet interconecta el cliente local de una maquina con el servidor con el que se comunica. Los caracteres que se teclean en un cliente local son enviados por la red y procesados en el computador remoto, utilizando la información que ese computador contiene. El resultado de su ejecución se muestra en la pantalla del ordenador local.

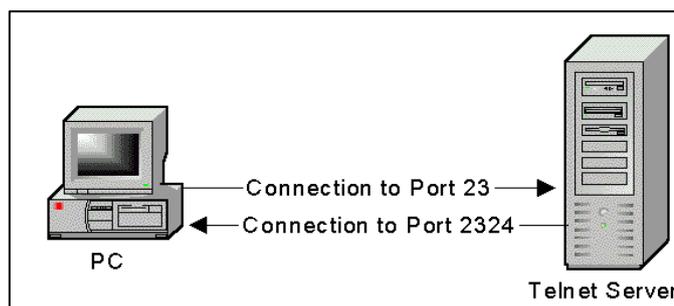


Figura 4. Conexión Telnet [6]

4.2.3.4. Transferencia de archivos

El protocolo FTP (File Transfer Protocol), permite la transferencia de archivos de texto o binarios desde un computador a otro sobre una conexión TCP. FTP implementa un sistema de restricciones basándose en propiedades y permisos sobre los archivos. Hay un control de acceso de los usuarios, y cuando un usuario quiere realizar la transferencia de un archivo, FTP establece una conexión TCP para el intercambio de mensajes de control. De esta manera se puede enviar el nombre de usuario, el password, los nombres de los archivos y las acciones que se requieren realizar. Una vez se ha

aceptado la transferencia de datos. El archivo se transfiere sobre la conexión de datos, sin la utilización de ninguna cabecera o información de control en la capa de aplicación. Cuando se completa la transferencia, la conexión de control es usada para transferir la señalización de que la transferencia se ha completado y para aceptar nuevos comandos de transferencia.

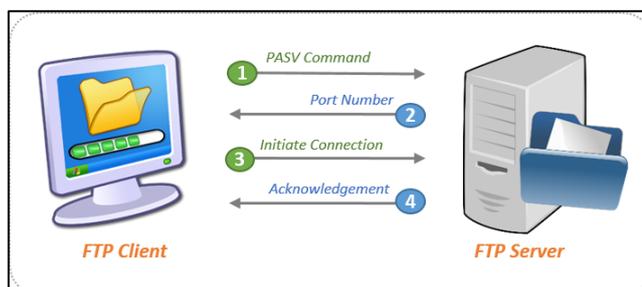


Figura 5. Transferencia de archivos (FTP)[7]

4.2.3.5. Sistema de Nombres de Dominio

El Sistema de Nombre de Dominio (DNS, Domain Name System) permite asignar nombres a equipos y servicios de red que se organizan en una jerarquía de dominios. La asignación de nombres DNS se utiliza en Internet para localizar equipos y servicios con nombres sencillos. Cuando un usuario escribe un nombre DNS en una aplicación, los servicios DNS pueden traducir el nombre a otra información asociada con el mismo, como una dirección IP.

La mayoría de los usuarios prefieren un nombre fácil de utilizar como google.com para localizar un equipo (como un servidor Web o de correo electrónico) en la red. Un nombre sencillo resulta más fácil de aprender y recordar. Sin embargo, los equipos se comunican a través de una red mediante direcciones numéricas. Para facilitar el uso de los recursos de red, DNS proporciona una forma de asignar estos nombres sencillos de los equipos o servicios a sus direcciones numéricas.

4.2.3.6. World Wide Web

World Wide Web (WWW) es uno de los servicios más importantes de Internet, permite a los usuarios acceder a documentos conocidos como páginas Web. Las páginas Web son documentos que pueden contener texto, audio, video y enlaces a otras páginas en la red. El Web se basa en los componentes hipertexto e hipermedia. El hipertexto es una cadena de texto resaltada en una página Web que permite el enlace hacia otro documento o página en la red. Hipermedia extiende el concepto de hipertexto y combina cadenas de texto resaltado, imágenes, iconos, etc. Que pueden poseer enlaces hacia otros documentos en la red.

4.2.4. Aplicaciones Multimedia

Las aplicaciones multimedia en su definición más general son aplicaciones que cambian diferentes tipos de medios como texto, imágenes, gráficos, video y audio entre otros y de los cuales al menos uno es en tiempo real. Debido a estos tipos de medios, éstas demandan altos recursos computacionales, red y almacenamiento entre otro, que hacen necesario configurar arquitecturas adecuadas para poder ejecutar eficientemente dichas aplicaciones.

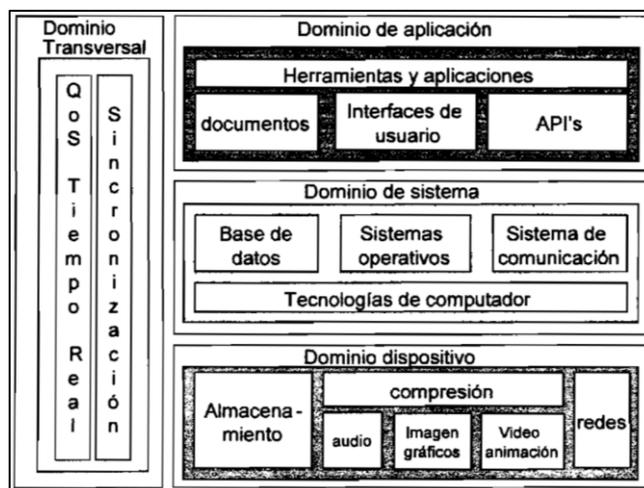


Figura 6. Arquitectura de las aplicaciones multimedia[4]

A pesar de todo el avance computacional y de telecomunicaciones, es necesario conformar arquitecturas multimedia que permitan ejecutar adecuadamente dichas aplicaciones. El modelo presentado en la figura 6, muestra los diferentes elementos y la relación entre ellos, que conforman un sistema multimedia. Sin embargo, hay algunos aspectos que se deben considerar críticos como las redes de telecomunicaciones, los protocolos multimedia, la provisión de calidad de servicio por parte de la red y las tecnologías de compresión multimedia.

4.2.4.1. Caracterización y clasificación de las aplicaciones multimedia

En un sistema multimedia, los datos de medios continuos y discretos deben ser transportados a través de redes de telecomunicaciones. Independiente del tipo de medio, los sistemas digitales de comunicaciones dividen la información en unidades discretas conocidas como paquetes, tramas, celdas, etc. Se define una secuencia de unidades transmitidas en una forma dependiente del tiempo como un flujo de datos o streams.

La trasmisión de flujos puede ser lograda con diferentes esquemas, distinguiéndose como trasmisión de datos asíncronos, síncronos e isócronos, las cuales están altamente

influenciados por las tecnologías de comunicaciones, protocolos y técnicas de comunicación.

- **Asíncrono:** Retraso de transmisión sin restricciones.
- **Síncrono:** El retraso de transmisión está acotado por cada mensaje.
- **Isócrono:** El retraso de transmisión es constante para cada mensaje.

La isocronía no tiene por qué mantenerse durante todo el camino de un mensaje, ya que puede ser recuperada en el destino mediante un almacenamiento para visualización.

De acuerdo a la direccionalidad de los flujos multimedia, se puede clasificar las aplicaciones en dos grandes grupos: Interactivas y Distribución (Streaming).

De acuerdo al tipo de flujos multimedia sobre interconexión de redes de las aplicaciones multimedia, se pueden agrupar de acuerdo a su sensibilidad al retardo y el número de nodos conectados simultáneamente, de aquí se extraen las siguientes conclusiones:

- Las aplicaciones que usan flujos multimedia almacenados no son sensibles al retardo.
- Las aplicaciones que involucran interactividad en tiempo real son sensibles al retardo de la red.
- Las aplicaciones punto a punto involucran comunicación entre un par de puntos de la red.
- Las aplicaciones multipunto típicamente involucran una difusión a muchos puntos de la red.

A continuación, se muestra la clasificación de las aplicaciones multimedia de acuerdo a la sensibilidad del tiempo.

TABLA I. CLASIFICACIÓN DE LAS APLICACIONES MULTIMEDIA

	Distribución / reproducción	Interactivas en tiempo real
Multipunto	Distribución de TV / Radio	Conferencia de escritorio. Ambientes multimedia colaborativos.
Punto a punto	Video/Audio bajo demanda	Entrenamiento. Conferencia de grupos o corporativa.

4.2.4.1.1. Aplicaciones multimedia en tiempo real interactivas

Son aplicaciones que contienen video y/o audio tomadas principalmente de fuentes vivas, en donde interactúan personas o aplicaciones simultáneamente en el sistema, esto implica retardos reducidos y en consecuencia altas restricciones en la red.

Las aplicaciones multimedia interactivas, se caracterizan porque los participantes (clientes) envían y reciben flujos multimedia. Generalmente se adquiere la información multimedia de fuentes vivas como cámaras y micrófonos. Se puede mencionar la videoconferencia y voz sobre IP como aplicaciones típicas de este grupo, y al ser de tiempo real, estas aplicaciones son muy exigentes respecto a parámetros críticos de la red como el ancho de banda, retardo y variación del retardo además de la necesidad de procesamiento en las funciones de captura, comprensión, descomprensión y reproducción en tiempo real y simétrica.

4.2.4.1.2. Aplicaciones multimedia de distribución (streaming)

Las aplicaciones multimedia de distribución (streaming) son aquellas en donde hay una o más estaciones que solo emiten flujos multimedia (servidor de streaming) y hay una o varias estaciones que solo reciben dichos flujos multimedia y su función principal es la reproducción mediante un player.

A su vez, las aplicaciones multimedia de distribución se dividen en:

- Tiempo real o sin control por parte del cliente.
- Bajo demanda o con control por el cliente.

La distribución en tiempo real se presenta cuando la señal original es tomada de una fuente viva o almacenada y es recibida "simultáneamente" por todos los clientes. Sigue el modelo convencional de distribución de Radio o Televisión comercial. La forma de distribución de los flujos multimedia es: unicast y multicast. El esquema unicast, implica que a cada cliente del sistema se le distribuye flujos multimedia independientemente. El esquema multicast implica que el servidor solo distribuye un único conjunto de flujos multimedia y todos los clientes reciben ese mismo conjunto.

4.2.5. Definición de streaming

La utilización de streaming permite a los usuarios el acceso a través de Internet a archivos de audio y video (deportes en vivo, música, noticias, entretenimiento y contenidos bajo demanda), utilizando su PC. Streaming permite distribuir contenido multimedia en Internet, con la característica de que cada usuario puede reproducir el

contenido tal como lo recibe a través de la red sin que sea necesario descargarlo por completo primero.

Streaming es el soporte principal para el transporte de multimedia en tiempo real (video, audio y datos asociados, etc.) entre, el cliente y los servidores de contenidos de Internet. El término “tiempo real” significa que el usuario recibe un flujo continuo (casi instantáneo) con mínimo retardo y que la duración del flujo transmitido y recibido es el mismo. [4]

Hay dos modos para transmitir multimedia sobre Internet: download (descarga) y streaming. En el modo download, el usuario no puede utilizar el archivo hasta que éste ha sido transferido completamente desde el servidor al disco duro de su computador. Los tiempos de transferencia dependen del tamaño del archivo y del ancho de banda de la conexión, resultando en algunos casos inaceptables. En el modo streaming, no es necesario que haya sido transferido el archivo completo para poder ser utilizado. El archivo es dividido en paquetes a los cuales se les añade la información de sincronización. Estos paquetes son enviados a través de la red y almacenados en el buffer del computador del usuario. Una vez que se ha almacenado los datos suficientes en el buffer del computador se inicia la reproducción y mientras se está reproduciendo estos datos, el buffer continuará recibiendo nuevos paquetes para su posterior reproducción.

4.2.5.1. Arquitectura de streaming

La arquitectura de los sistemas de streaming pueden ser dividida en seis áreas como se puede visualizar en la figura 7.

Compresión y codificación: para una adecuada transmisión el flujo de datos debe ser inferior a la velocidad de la conexión establecida. Para propósitos de transmisión de vídeo se llegan a utilizar relaciones de compresión muy altas. Los niveles de compresión más elevados pueden afectar a la calidad del contenido multimedia recibido. Generalmente, para niveles de compresión más elevados se obtiene una calidad subjetiva más baja. La calidad no sólo depende de la velocidad de transferencia, sino que también depende del contenido, de la resolución de visualización, composición de los paquetes y del algoritmo utilizado. Algunos compresores permiten además de acondicionar la duración y frecuencia de los paquetes la posibilidad de ajustar la profundidad del color, resolución de imagen y otras características adicionales.

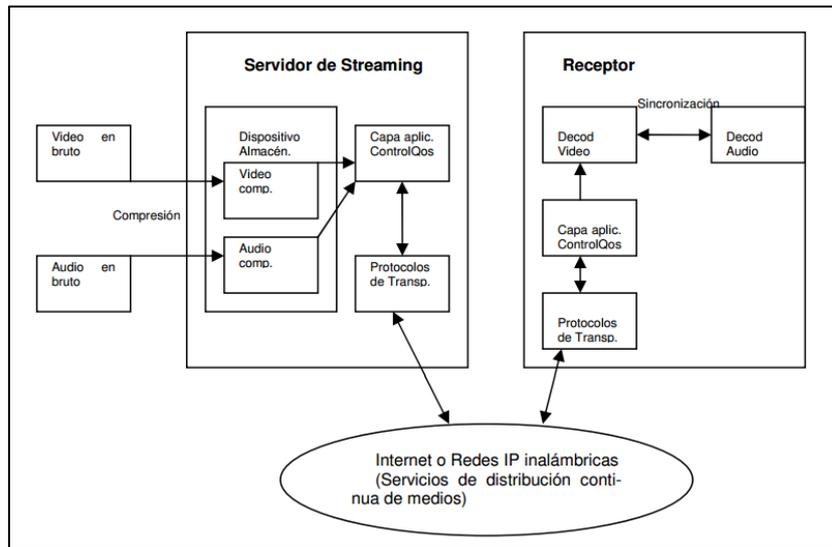


Figura 7. Arquitectura streaming [8]

Capa de aplicación y control de QoS: la capa de aplicación y control de calidad (QoS) incluye control de congestión y control de error, que se implementan en la aplicación (por delante de la capa de red). El control de congestión previene la pérdida de paquetes y reduce los tiempos de retardo. El control de error se utiliza para mejorar la calidad de la presentación en presencia de pérdida de paquetes. El control de error incluye mecanismos de corrección avanzada de errores FEC, y minimización de errores.

Servicios de distribución de audio y video: además de QoS y de adaptación de la red, utilizadas para reducir la pérdida de paquetes y los retardos, son necesarias otras aplicaciones como replicación y almacenamiento en caché para disminuir el tráfico en la red.

Servidor de Streaming: el servidor de streaming es el elemento principal de la cadena en cuanto a calidad del servicio se refiere. El servidor procesa los datos multimedia en cortos espacios de tiempo y soporta funciones de control interactivas siendo además el responsable de suministrar los servicios de audio y vídeo en modo sincronizado. El servidor de streaming está a la espera de la petición desde el usuario. Cuando recibe una petición, el servidor busca en el directorio apropiado el archivo solicitado. Si el nombre del archivo es el correcto, el servidor lo envía hacia el cliente.

Sincronización de audio y video en el receptor: en el lado del receptor se deberán reproducir los flujos de datos multimedia en el mismo orden en que fueron generados por el servidor de streaming. Esta funcionalidad es provista por el reproductor de contenidos multimedia que es un software capaz de procesar el flujo multimedia que

entra en la máquina destino. El reproductor contiene el decodificador que permitirá al usuario visualizar el contenido de manera adecuada y eficiente.

Protocolos utilizados en streaming: varios protocolos han sido normalizados para permitir la comunicación entre los servidores de streaming y los ordenadores cliente.

Los protocolos implementan las funcionalidades siguientes:

- Direccionamiento de red, para lo que se utiliza el Protocolo Internet (IP).
- Transporte, utilizando UDP o TCP
- Control de sesión, suministrado por protocolos como: Real Time Streaming
- Protocol (RTSP), Microsoft Media Service Protocol (MMS) o Hypertext Transfer Protocol (HTTP).

4.2.5.2. Servidores web y servidores de streaming

El estilo de difusión, utilizando un servidor Web, no es muy distinto al caso de descargar y reproducir comentado anteriormente. El contenido multimedia es comprimido en un archivo para ser transmitido a través de la red. Este archivo es puesto en un servidor Web, donde se encuentra la URL que activa el reproductor, por el lado del cliente, y posteriormente descarga el archivo.

A diferencia del caso de bajar o descargar el archivo multimedia completo para posteriormente reproducirlo, el cliente comienza a reproducir el archivo a medida que se descarga, segundos después de haber almacenado parte del archivo en un buffer. Con esta metodología, el cliente recibe datos tan rápido como el servidor le pueda enviar, sin importar la tasa de bits del archivo comprimido.

En el método de difusión que utiliza un servidor de streaming, el contenido multimedia es preparado, para luego ser copiado a un servidor. Después, una referencia dentro de una página Web, al archivo multimedia, es colocada en el servidor Web, para poder acceder a éste. Los datos enviados por el servidor de streaming son activa e inteligentemente transmitidos al cliente, es decir, el contenido es entregado a la tasa exacta asociada al archivo multimedia. El servidor y cliente están en contacto permanente durante el proceso, para poder responder ante cualquier urgencia del cliente; el servidor y el cliente intercambian mensajes de control. Debido al intercambio continuo de mensajes, los servidores de streaming pueden ajustar la transmisión de datos a los cambios en las condiciones de la red y proveer al cliente mayor calidad de reproducción. Los mensajes de control también incluyen funciones de usuario como parar, pausar o saltar a una parte particular en el archivo.

Aunque los servidores de streaming pueden hacer uso de protocolo TCP usado por los servidores Web, una de las ventajas de utilizar un servidor de streaming, es la capacidad de usar UDP. TCP añade mayores retardos a la transmisión de datos debido al reenvío de paquetes y subutiliza la capacidad de la red. Este no son características deseables en la transmisión de multimedia en tiempo real en donde se requiere que los paquetes lleguen al receptor los más rápidamente posible (los paquetes deben llegar a tiempo para ser reproducidos correctamente). UDP tiene precisamente esta característica siendo el protocolo más adecuado para la transmisión de multimedia en tiempo real.

4.2.5.2.1. Servidores de streaming

“El servidor streaming es el elemento principal de la cadena en cuanto a calidad del servicio se refiere. El servidor procesa los datos multimedia en cortos espacios de tiempo, además es el responsable de suministrar los servicios de audio y video en modo sincronizado.

El servidor de streaming está a la espera de la petición RTSP (Real Time Streaming Protocol) desde el usuario. Cuando recibe una petición, el servidor busca en el directorio apropiado el contenido media del nombre solicitado. Si el contenido media está en el directorio solicitado, el servidor hace streaming hacia el solicitante utilizando RTP (Real-time Transport Protocol).” [9]

4.2.5.3. Procesador de señal digital (Cliente)

Es un software que se encarga de recibir la señal de la tarjeta de audio y procesarla. Luego se conecta con el servidor por Internet y le va enviando nuestra voz y la música por pequeños paquetes digitales. El servidor recibe los datos y los va entregando a las personas que “sintonicen” nuestro programa. [10]

4.2.5.4. Códecs

El término códec viene de los términos codificar/decodificar y comprimir/descomprimir. Es una pieza de software capaz de transformar un flujo de datos o señal. Un códec codifica un flujo de datos o señal para su almacenaje o transmisión y lo codifica para su reproducción. [11], [12]

4.2.5.5. Adaptación a la congestión de la red

Como la congestión de la red depende del número de usuarios conectados, el ancho de banda disponible para streaming varía significativamente. Para la óptima visualización del archivo multimedia, el servidor de streaming deberá ajustarse dinámicamente para entregar en cualquier circunstancia la más elevada tasa de bits (bitrate) posible.

Durante la reproducción, el servidor y el cliente se comunican repetidamente para conmutar entre las diferentes opciones de bitrate, entregando en cualquier caso la

calidad más elevada que la conexión con el cliente puede soportar. Es importante indicar que la conmutación de bitrate de audio y vídeo es administrada independientemente.

4.2.5.6. Modalidades de streaming

En lo que se refiere a la transmisión de multimedia a los usuarios de la red se lo puede realizar de dos formas: transmisión bajo demanda y transmisión por difusión.

4.2.5.6.1. Transmisión bajo demanda

En una transmisión bajo demanda (a petición) el usuario establece una conexión activa con el servidor y selecciona el contenido de audio o video que quiere visualizar. Una vez que este contenido empieza a ser reproducido por el usuario, éste puede detener, retroceder, avanzar rápido o pausar la secuencia. Las conexiones bajo demanda ofrecen mayor control de la secuencia, pero pueden consumir rápidamente el ancho de banda de una red porque cada usuario tiene su propia conexión con el servidor.

4.2.5.6.2. Transmisión por difusión

En la transmisión por difusión el usuario recibe una secuencia de manera pasiva. Durante una transmisión por difusión, el usuario recibe la secuencia, pero no puede controlarla. Por ejemplo, el usuario no puede pausar ni avanzar o retroceder rápidamente la secuencia.

4.2.5.7. Formas de Operación

La distribución de audio y video en una red que utiliza TCP/IP, se la puede realizar de tres formas: unicast, broadcast y multicast. Cada forma de operación tiene sus ventajas y desventajas dependiendo del entorno en el que se la utilice.

4.2.5.7.1. Unicast

Esta forma de operación se caracteriza por el comportamiento del servidor respecto a la distribución de los flujos de audio y video a los usuarios. En unicast cada estación es identificada por su dirección IP, y es por tanto esta dirección la que se utiliza para encaminar paquetes de datos hacia su destino. Por cada receptor se emite un flujo independiente, por lo que el ancho de banda requerido por el servidor es directamente proporcional al número de usuarios simultáneos que tenga.

Unicast resulta adecuado para entregar audio y video bajo demanda en donde se requiere que cada usuario tenga su propia conexión con el servidor para intercambiar información de control sobre la secuencia que está recibiendo.

Unicast es utilizado por las estaciones de radio y televisión que operan en Internet. Cuando el número de usuarios es pequeño esta configuración no presenta mayores problemas, pero si trata de difundir la programación a miles de usuarios se debe considerar dos inconvenientes:

Demasiadas peticiones: Con el servicio unicast el servidor tiene que procesar cada solicitud y despacharla. Cada flujo toma una pequeña porción de la capacidad de procesamiento del servidor, si se tiene demasiadas solicitudes el servidor no podrá sostener la sobrecarga y muchas personas no podrán recibir la transmisión.

Demasiados paquetes: El segundo problema con unicast, y un gran número de solicitantes simultáneos, es que una serie separada de paquetes de datos debe ser enviada a cada persona. Incluso si el servidor pudiera hacer esta tarea, el número de paquetes de datos enviados inundarían el sistema entero haciendo que la transmisión se tome muy lenta, o hasta se detenga.

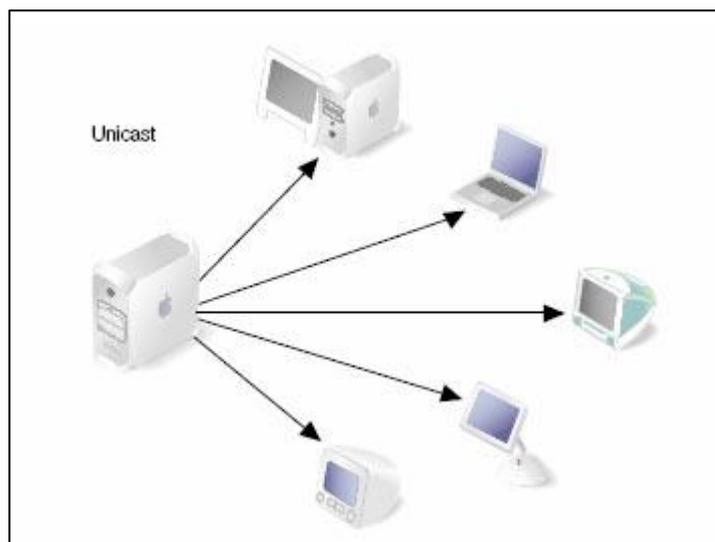


Figura 8. Forma de Operación Unicast [13]

4.2.5.7.2. Broadcast

Si se requiere transmitir multimedia en una red local de acceso compartido, se puede solucionar el problema anterior enviando un único flujo a dicha red local. Como toda la información que circula por una red local de acceso compartido es escuchada por todos sus miembros, para ser un oyente del sistema sólo se debería escuchar lo que circula por la red. Así se puede reducir drásticamente el tráfico generado, siempre que los destinatarios estén en la misma red local.

Este sistema tiene el inconveniente de que los interfaces de red de los host que no quieran escuchar la transmisión estarán aceptando un tráfico indeseado, a no ser que desactiven su función de escuchar el tráfico broadcast, con lo cual podrían perderse mensajes importantes.

4.2.5.7.3. Multicast

Multicast resuelve el problema de unicast y broadcast. Esta forma de operación se basa en utilizar un conjunto de protocolos que permiten crear direcciones IP multicast. De este modo, asociando una dirección IP multicast a un grupo, sólo es necesario enviar un único flujo a dicha dirección IP, y los routers se encargarán de hacer llegar la información a los host suscritos a dicha dirección IP multicast. La principal ventaja de utilizar multicast es la disminución del tráfico en la red; los datagramas que comparten un grupo de enlaces hasta sus destinos sólo precisan ser transmitidos una vez y sólo se replica el mensaje cuando es necesario, recayendo la responsabilidad de la duplicación de paquetes en los routers (para hacer llegar una copia a cada miembro del grupo), de modo que estos aseguren que los paquetes viajarán una vez por cada enlace como máximo (basta con asegurar que todos los miembros reciben una copia del mismo paquete).

El envío multicast no es orientado a la conexión; un datagrama se envía a los miembros del grupo de destino con la misma fiabilidad ("best-effort") que un datagrama unicast IP standard. Esto supone que los datagramas multicast no tienen garantizada la entrega a todos los miembros del grupo, ni la llegada en el mismo orden de transmisión de los paquetes.

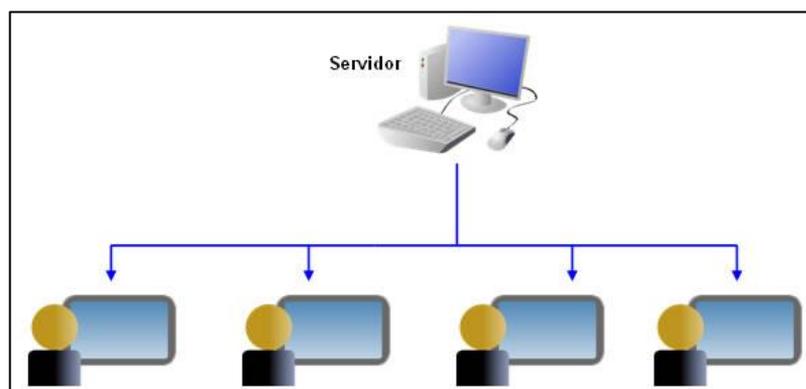


Figura 9. Forma de operación Multicast[14]

4.2.5.8. Protocolos Utilizados

La necesidad de transmitir información multimedia a través de redes de datos como Internet ha motivado la implementación de nuevos protocolos de transporte que permitan aplicaciones en tiempo real requeridas por el usuario de las redes.

La transmisión de información en tiempo real, no obstante, debe solventar las dificultades inherentes a las redes de paquetes, entre las que se encuentran: el retardo de los paquetes, la dispersión temporal (jitter), la pérdida de paquetes, la entrega desordenada y aparición de paquetes duplicados. Con la finalidad de solventar estos

problemas el IETF ha desarrollado una serie de protocolos que posibilitan la transmisión de contenido multimedia a través de redes de paquetes como Internet.

4.2.5.8.1. RTP (Real time transport protocol)

El protocolo RTP proporciona servicios de transporte extremo a extremo para aplicaciones que requieran transmisión en tiempo real, a través de redes de paquetes; puede ser utilizado para: video bajo demanda (VoD), control de maquinaria, telemedicina, videoconferencias, herramientas para monitoreo y administración de paquetes, servicios interactivos tales como telefonía IP y otros servicios. No proporciona ningún medio para asegurar temporización en la entrega de información, ni proporciona garantías de calidad de servicio sobre redes IP.

4.2.5.8.1.1. Funciones del protocolo RTP

La función básica de RTP es multiplexar varios flujos de datos en tiempo real en un solo flujo de paquetes UDP. El flujo UDP se puede enviar a un solo destino (unicast) o a múltiples destinos (multicast). Debido a que RTP utiliza UDP normal, sus paquetes no son tratados de manera especial por los enrutadores, a menos que se habiliten algunas características de calidad de servicio IP. Las funciones de RTP son:

- Identificar el tipo de información transportada.
- Añadir marcas temporales y números de secuencia, para poder reproducir los paquetes en el orden correcto.
- Sincronización de los paquetes de información.
- Detección de pérdidas, seguridad e identificación de participantes.
- Controlar la llegada de los paquetes.
- Supervisar la entrega de los datos.

4.2.5.8.2. RTCP (Real Time Control Protocol)

Es un protocolo basado en el intercambio periódico de paquetes de control entre los participantes de una sesión, diseñado para funcionar junto con RTP. La función principal es proporcionar mecanismos de realimentación para informar sobre la calidad de distribución de los datos, esta información se utiliza para controlar a los codificadores que sean adaptables o para diagnosticar fallos en la transmisión. La información de realimentación se envía mediante informes generados por el emisor o por el receptor.

4.2.5.8.2.1. Funciones del protocolo RTCP

- La identificación de participantes y sincronización entre distintas sesiones RTP. Los paquetes RTCP contienen un identificador textual, nombre canónico, único o CNAME para cada participante en una sesión multimedia.

- La realimentación proporciona información sobre la calidad de la distribución de los datos, esta función está relacionada con la función de control de congestión. Estos informes permiten evaluar el estado de la recepción de los paquetes en el receptor de los participantes y determinar si existen problemas en la transmisión.
- Estimación del número de participantes. El envío de paquetes de control RTCP por parte de todos los participantes en una sesión permite que cada uno de ellos pueda estimar de forma dinámica el número total de participantes, por lo que la tasa de envío de los paquetes RTP deber ser controlada. Así, pueden ajustar la frecuencia de transmisión de paquetes RTCP para mantener el límite de ancho de banda de control establecido en la sesión.
- Proporcionar información de control de la sesión, como nombre, dirección o teléfono de cada participante, esta función opcional se basa en la definición de unos pocos tipos de paquete RTCP, que proporciona la información necesaria para realizar un control mínimo de la sesión.

4.2.5.8.3. RSVP (Resource Reservation Protocol)

Este protocolo se encuentra normalizado por la IETF en el estándar RFC 2205, en principio solo era un protocolo de reserva de recursos y control de admisión. Este protocolo es usado para requerimientos específicos de servicio para una red en particular. RSVP es usado también por los routers para entregar calidad de servicio (QoS), dependiendo de los requerimientos de los nodos a lo largo de la trayectoria.

4.2.5.8.3.1. Funciones del protocolo RSVP

- Permitir a los emisores, receptores y routers de las sesiones de comunicación (tanto multicast como unicast) comunicarse con el resto para establecer una ruta que pueda soportar la calidad de servicio requerida por la aplicación.
- Identifica una sesión por medio de una dirección de destino, el protocolo de transporte y un número de puerto de destino.
- No es un protocolo de encaminamiento, se usa exclusivamente para reservar recursos a través de la ruta que se establezca por cualquiera de los protocolos de niveles inferiores.
- RSVP se utiliza para solicitar calidad de servicio en la red para una secuencia de datos, también se lo utiliza para entregar calidad de servicio a todos los nodos a lo largo de toda su trayectoria.

4.2.5.8.4. HTTP (Protocolo de transferencia de Hipertexto)

En 1991 HTTP fue implementado por Tim Berners-Lee para la WWW (World Wide Web) como un protocolo eficaz y sencillo que permite la transferencia de múltiples tipos de información de forma eficiente y rápida.

HTTP es un protocolo de nivel de aplicación para distribución de información hypermedia. La primera versión HTTP/0.9 fue implementada para la comunicación entre computadores mientras intercambiaban documentos, agregando conectividad e interfaces. Luego se creó la versión HTTP/1.0 definida en la RFC 1945 de la IETF que habilitaba la transmisión de un formato MIME con meta información acerca de los datos transferidos y modificaba la semántica de las solicitudes y respuestas. Sin embargo, en HTTP 1.0, una vez que se establecía una conexión TCP, se enviaba una solicitud y se obtenía una respuesta, luego se liberaba la conexión. Esto implicaba que se requería una conexión para transmitir cada objeto de una página Web. Debido a esta razón se diseñó HTTP 1.1, que soporta conexiones persistentes. Con ellas, es posible establecer una conexión TCP, enviar una solicitud y obtener una respuesta, y después enviar solicitudes adicionales y obtener respuestas adicionales. Las especificaciones del protocolo HTTP/1.1 definidas en la RFC 2616 incluyen además rigurosos requerimientos de seguridad y confiabilidad para las diferentes aplicaciones.

HTTP es usado como un protocolo genérico para la comunicación entre agentes, proxys, gateway y de otros sistemas de Internet, incluido soporte para protocolos SMTP, FTP, Gophery Wais. HTTP es un protocolo orientado a objetos, genérico, que puede usarse para muchas tareas extendiendo sus métodos. Una característica de HTTP es que permite que los sistemas se construyan independientemente de la información que se transfiere.

4.2.5.8.4.1. Características principales

- Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres comandos básicos que un cliente puede utilizar para dialogar con el servidor. GET, para recoger un objeto, POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto.
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto.
- Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.

4.2.5.8.5. RTSP (Real Time Streaming Protocol)

RTSP es un protocolo del nivel aplicación utilizado para controlar la distribución de los datos con propiedades de tiempo real. Está diseñado para funcionar con protocolos de bajo nivel como RTP y RSVP y se encuentra normalizado por la IETF en el estándar RFC 2326. RTSP permite controlar múltiples sesiones, y se puede escoger los protocolos de transporte a utilizar como UDP y TCP, y proporciona un medio para el seleccionar mecanismo de distribución basado en RTP.

4.2.5.8.5.1. Operaciones soportadas por RTSP

- Petición de medios al servidor. El cliente puede pedir una descripción de presentación vía HTTP u otro método. Si la presentación es multicast, la descripción contiene las direcciones multicast y puertos que pueden ser usados. Si la presentación sólo se va a enviar al cliente, el cliente proporcionará la dirección por motivos de seguridad.
- Invitación a un servidor de medios para una conferencia. Un servidor puede ser invitado a unirse a una conferencia existente, bien como participante o simplemente para grabar parte de la conferencia. Este modo es útil para las aplicaciones de enseñanza distribuida.
- Adición de medios a una presentación existente. Particularmente en presentaciones en directo es interesante que el servidor pueda avisar al cliente de que nuevos medios están disponibles.

El protocolo RTSP establece un control simple de sincronización para flujos continuos tales como audio y video. Típicamente entrega un flujo continuo, en otras palabras, RTSP actúa como un control remoto para los servicios multimedia.

4.3. Medios digitales

Los medios digitales hacen referencia al contenido de audio, video e imágenes que se ha codificado (comprimido digitalmente). La codificación de contenidos implica convertir la entrada de audio y video en un archivo de medio digital como, por ejemplo, un archivo de Windows Media. Una vez codificado el medio digital, se puede manipular, distribuir y representar (reproducir) fácilmente en otros equipos, así como transmitir a través de redes informáticas. [15]

Ejemplos de tipos de medios digitales: Windows Media Audio (WMA), Windows Media Video (WMV), MP3, JPEG y AVI.

4.3.1. Características de los medios digitales

Entre las principales características se tiene: [16]

- Hipertextualidad
- Multimedialidad
- Interactividad
- Actualización

Hipertextualidad: consiste en un conjunto de textos ligados entre sí por medio de conexiones, es decir es el conjunto de enlaces que llevan al usuario a más información relacionada con el tema que se está revisando.

Multimedialidad: se refiere a la implementación de la información diversos medios no solo texto, sino también videos, fotos e imágenes, de este modo se logra mejor comprensión.

Interactividad: está considerada una de las características más importantes presentes en estos medios porque nos da la oportunidad de dar nuestras opciones referentes a una noticia e información presentada en la red, mediante foros, o correos electrónicos inclusive salas de chat, obteniendo así una respuesta casi inmediata.

Actualización: la mayoría de sitios web ofrece actualización diaria, semanal y mensual de su información, con el fin de presentar continuidad y profundidad en la información presentada.

4.3.2. Ventajas de los medios digitales

Existen grandes ventajas aplicables en un sin número de campos, áreas y aplicaciones. Entre las principales ventajas que brinda a la sociedad tenemos:

- Permiten llevar al hombre a espacios de difícil acceso (e incluso inexistentes) y favorecen el desarrollo de artefactos que son una extensión de las capacidades de los sentidos ayudando a incrementar el conocimiento del mundo, no existe biblioteca alguna que se asemeje con la base de datos que nos brinda Internet, pues este es la suma de millones de computadoras alrededor del mundo con millones de sitios e información especializada.
- Comercio, muchas empresas ya tienen tiendas en línea, donde se puede comprar sin número de artículos, este mercado está en desarrollo en donde en un futuro sería la principal manera de compra.
- Las redes sociales, permiten crear vínculos de amistad entre otros, afianza la comunicación entre personas de diversos países, así como también facilita el intercambio de ideas y costumbres.
- En la educación sirven de manera interactiva para aclarar las dudas de los estudiantes sobre diversos temas, y no solo el Internet ya que también se

encuentran también videos interactivos en DVD, VCD, etc., así como también cintas de audio.

4.4. Radio en internet

El término “Radio en Internet” describe la conglomeración de streaming de audio que se encuentra disponible en el Internet que puede ser escuchado mediante un software reproductor o navegador que soporte streaming de audio. Una estación de radio en la red se puede acceder desde cualquier ordenador con acceso a Internet. Esta amplia capacidad de cobertura de emisión de radio por Internet es muy atractiva para los organismos de radiodifusión. [17]–[19]

4.4.1. Ventajas de radio en internet

Entre las principales ventajas de implementar una estación de radio en Internet se encuentran:

No se necesita solicitar una licencia especial al gobierno, ya que no se usa un espectro público de radiofrecuencia. La actual libertad que existe en Internet permite que incluso se pueda emitir sin ningún tipo de permiso, licencia o condición.

- Internet posibilita transmitir audio a cualquier parte del mundo a un costo muy bajo comparado con lo que costaría transmitir este contenido utilizando medios tradicionales.
- Se pueden implementar aplicaciones interactivas debido a la naturaleza bidireccional del Internet.
- En Internet no existen grillas de programación, ni horarios de emisión, ni la necesidad de que su programa tenga una duración específica, se puede elegir cuánto dura un audio, y por cuánto tiempo estará disponible.
- Es mucho más económico implementar un servicio de radio por Internet que crear una empresa y emitir de forma convencional, con la subsiguiente expedición de licencias apropiadas. Para implementar un sistema de transmisión de audio digital se requiere muy pocos recursos: lo mínimo es un computador con conexión a Internet y el software apropiado, que en muchos casos es libre y gratuito.

4.4.2. Problemas de radio en internet

- La calidad de audio recibido por el usuario está en función del tipo de conexión que se tenga.
- La mayor parte de Internet no admite aun multicast, por lo cual muchas estaciones de radio en Internet utilizan unicast para distribuir el contenido a los

usuarios. Debido a esto, el ancho de banda necesario para implementar una estación de radio es relativamente elevado.

- El ancho de banda de Internet no es confiable. El tráfico de Internet varía enormemente y el tipo de conexión que obtienen los usuarios puede ser distinto a cada conexión.

4.4.3. Radios universitarias en Ecuador

La Universidad Central del Ecuador fue la pionera en desarrollar un proyecto radial, fue presentado en 1973 y logro salir al aire por primera vez en marzo de 2011. La Universidad Católica Santiago de Guayaquil tuvo también una frecuencia desde marzo del 2011 hasta febrero del 2014. [20]

La red cibernética y el desarrollo acelerado de tecnologías de información han abierto grandes posibilidades para las universidades. En el Ecuador muchas comunidades académicas han creado sus radios online. Pero con esta nueva coyuntura política y educativa, las universidades tendrán mayor posibilidad de contar con un servicio radial.

La Red de Radios Universitarias del Ecuador (RRUE) la conforman 11 radios de diferentes universidades del país: Universidad de los Hemisferios, Universidad Católica de Cuenca, Universidad Católica Santiago de Guayaquil, Universidad de Cuenca, Universidad San Francisco, Universidad Central del Ecuador, Universidad Tecnológica Equinoccial y la Universidad del Azuay. Donde la Universidad Central del Ecuador tiene radio con frecuencia en el espectro radioeléctrico, las demás radios funcionan por Internet. [20]

La RRUE desde el mes de enero del 2014 forma parte de la Red de Radios Universitarias Latinoamericanas y del Caribe RRULAC, la misma que cuenta con aproximadamente 150 radios universidades alineadas a la red, el objetivo de la misma es dar la posibilidad de intercambio entre profesionales, docentes, alumnos y radialistas, los mismos que deben fomentar la creación de nuevas emisoras universitarias y generar una plataforma común donde se puedan compartir contenidos. [20]

4.5. Aplicaciones Móviles

En la actualidad, la tecnología ha visto un gran crecimiento que se refleja en la aparición de dispositivos móviles cada vez más llamativos. Ante esto los informáticos han creado una serie de aplicaciones móviles para el uso de los usuarios y de esta manera mejorando el concepto de acceder a recursos y procesos con mayor facilidad en relación con un ordenador personal; en este capítulo se aborda conceptos de acerca de dispositivos móviles, aplicativos móviles, tipos de aplicación que se puede desarrollar,

sistemas operativos para móviles, metodologías y frameworks para el desarrollo de las aplicaciones móviles.

4.5.1. Teléfono Móvil

“El teléfono móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente, intermitente o nula a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales.” [21]

4.5.2. Smartphone

“Los Smartphone o teléfonos inteligentes son teléfonos que soportan más funciones que un teléfono común como gestionar correo electrónico, ofrecer la funcionalidad completa de organizador personal, acceder de manera continua a Internet, la posibilidad de instalar programas adicionales, entre otros.” [21]

4.5.3. Aplicación Móvil

Una aplicación móvil es un programa software que puede ser descargada y a las que se puede acceder directamente desde un teléfono o desde algún otro dispositivo móvil, como por ejemplo tabletas o un reproductor de música como iPod. [21]

4.5.4. Tipos de aplicaciones Móviles

En la actualidad, el desarrollo de aplicaciones para los dispositivos móviles se puede desarrollar de tres formas: desarrollo web, entornos de desarrollos nativos y entornos de desarrollo multiplataforma. [21]

4.5.4.1. Desarrollo Web

El desarrollo web son aplicaciones basadas en lenguajes de marcas, añadiendo la facilidad de poder programar y probar sin necesidad de un emulador o un dispositivo real. A estas aplicaciones se accede mediante la red.

Para el desarrollo de estas aplicaciones se puede utilizar cualquier entorno de desarrollo conocido.

Las aplicaciones web móviles para adaptarlas a un dispositivo móvil se pueden utilizar lo siguiente: [21]

- Tener el conocimiento.
- Utilizar servidores que incorporen ese conocimiento y lo automaticen, de manera que lleguen a realizar transformaciones automatizadas.

Las pruebas se suelen empezar en navegadores de escritorio con soporte HTML5, pero luego se debe realizar en dispositivos reales para comprobar la compatibilidad de los componentes desarrollados. [22]

4.5.4.2. Entornos de desarrollo nativos

Las aplicaciones nativas son las que se realizan específicamente en las características del dispositivo o de la plataforma en la cual van a ser ejecutadas.

Para su desarrollo es necesario utilizar el entorno de desarrollo propio de la plataforma. Las pruebas se realizan en los emuladores que suelen ser incluidos dentro de cada IDE (Integrated Developer Environmet) o en dispositivos reales. [21]

Para la implementación, cada sistema utiliza su propio método y sus propios patrones, pero hay algunos puntos comunes:

- Existe un emulador con el que se prueban las aplicaciones. Pero, en ocasiones el emulador no permite emular todas las acciones de usuarios o la emulación no es lo suficientemente ágil, por lo que se necesita un dispositivo real. [22]
- Separación de presentación y lógica, de manera que se aprovecha al máximo los componentes. [22]
- Posibilidad de "debugar" la aplicación para poder tener mayor control. [22]
- Generalmente existen herramientas que facilitan la construcción de las interfaces gráficas o UI (user interface). [22]

En las pruebas, cada IDE (Integrated Developer Environmet) tiene sus herramientas, desde las típicas tecnologías de pruebas unitarias hasta sistemas más complejos, como el monkey runner de android. Sin duda, las posibles pruebas que se pueden realizar sobre las aplicaciones nativas son mucho más extensas y están más controladas que aquellas que se puedan realizar en otro tipo de aplicación, ya que se tienen las herramientas propias de la plataforma. [21]

4.5.4.3. Entornos de desarrollo multiplataforma

Las aplicaciones multiplataforma o aplicaciones híbridas son aquellas que desde una misma línea de código permite realizar aplicaciones nativas.

Para lograr este objetivo hay aplicaciones que realizan pre-procesamiento para acabar generando aplicaciones 100% nativas. Hay otras alternativas que proporcionan su propia arquitectura y sus propios lenguajes, y también mediante un sistema de compilación o ejecución vía máquina virtual consiguen tener aplicaciones nativas. [21]

Hay aspectos que estas aproximaciones no van a poder evitar fácilmente (a no ser que tengan código condicional específico para cada plataforma). Son los siguientes:

- **Pérdida de controles específicos de una plataforma:** Si se tiene un control de la User Interface o una funcionalidad concreta que solo existe en una plataforma, no se la podrá generar de manera única, por el desarrollo multiplataforma. [22]
- **Integración en el escritorio del dispositivo:** Según la plataforma, las posibilidades de añadir elementos en el escritorio de cada usuario varían. [22]
- **Gestión de la multitarea:** Debido a que se trata de conceptos de bajo nivel de cada plataforma, cada una la trata de manera diferente, con restricciones diferentes, por lo que no será fácil hacer código común para todas sin perder mucha potencia. [22]
- **Consumo de la batería:** Estas aproximaciones requieren de una capa de abstracción sobre el dispositivo, lo que provoca problemas como la multitarea. De la misma forma, el control sobre el consumo de batería se hace más difícil cuando no se tienen las capacidades concretas de la plataforma. [22]
- **Servicios de mensajería asíncrona o push services:** Sirven para implementar elementos como la mensajería instantánea, pero debido a que cada plataforma los implementa de una manera, se hace complicado atacarlos conjuntamente. [22]

Para el desarrollo, cada uno de los entornos proporciona su entorno de desarrollo completo y las aplicaciones se prueban en los emuladores de las plataformas nativas, de manera que hay que instalar el IDE propio del entorno de desarrollo y los emuladores o, en ocasiones, los SDK.

Durante la implementación, variará mucho en función de cada plataforma, pues hay algunas que podrán aprovechar todas las herramientas de desarrollo, otras que no lo necesitarán en exceso y algunas en las que el desarrollo será más difícil. En general, las herramientas facilitan el proceso de desarrollo, aunque sin llegar al nivel de las herramientas de desarrollo nativas. [21]

4.5.5. Sistemas Operativos para dispositivos móviles

El sistema operativo móvil es aquel que controla un dispositivo móvil, sirve de interface entre el hardware y el usuario; facilita al usuario o al programador las herramientas e interfaces adecuadas para el manejo del dispositivo. Este tipo de sistemas suelen ser más simples que los sistemas operativos de computadoras de escritorio o de las computadoras portátiles; suelen estar más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. [21]

4.5.5.1. Componentes de los sistemas operativos para dispositivos móviles

Los sistemas operativos para los dispositivos móviles constan de los siguientes componentes:

4.5.5.1.1. Capas

El sistema operativo que tienen los componentes informáticos más grandes son iguales al sistema operativo móvil que al igual que los otros está compuesto por varias capas. [23]

4.5.5.1.2. Kernel

Kernel o núcleo es la capa de software que permite el acceso a los diferentes elementos de hardware que conforman nuestro móvil. Asimismo, es el encargado de brindar diferentes servicios a las capas superiores como, los controladores de hardware, gestión de procesos, sistemas de archivos, además del acceso y administración de la memoria del sistema. [21]

Los sistemas operativos para móviles pueden basarse en núcleos Linux, tal como lo hace Android, o hasta inclusive IOS, el SO del iPhone que utiliza un kernel heredado de Unix; sin embargo, hay otros que prefieren usar sus kernels propios como Windows Mobile y RIM. [23]

4.5.5.1.3. Middleware

Middleware es el conjunto de módulos que permite que las aplicaciones diseñadas y escritas para tales plataformas puedan ser ejecutadas. Su funcionamiento es totalmente transparente para el usuario, no debiendo realizar ninguna acción ni configurar alguna para su correcto desenvolvimiento. [21]

Entre los servicios que presta están, los motores de comunicaciones y mensajería, funciones de seguridad, servicios para la gestión de diferentes aspectos del móvil, ofrece servicios claves como, el motor de mensajera y comunicaciones, codecs multimedia, intérpretes de páginas Web y servicios WAP, además de soporte para una gran variedad de servicios concernientes al apartado multimedia que es capaz de ejecutar el móvil. [23]

4.5.5.1.4. Entorno de ejecución de aplicaciones

Ejecución de aplicaciones, capa que provee de todos los elementos necesarios para la creación y desarrollo de software a los programadores. Entre los servicios que los programadores pueden encontrar, se destacan un gestor de aplicaciones y una serie de interfaces programables "Application Programming Interface" abiertas. [21]

4.5.5.1.5. Interfaz de usuario

Interfaz de usuario, es el elemento que permite la interacción del usuario con el dispositivo móvil, ésta presenta todos los elementos necesarios (botones, menús, pantallas y listas, entre otros) para facilitar cualquier tipo de tarea que se desee realizar en el terminal. [21]

4.5.5.2. Sistema Operativo iOS

El sistema operativo al principio se denominó iPhone OS, ya que se ejecutaba únicamente en los productos iPhone. El 7 de junio de 2010, durante la presentación del iPhone4, Steve Jobs anunció que iPhone OS pasaría a ser llamado iOS. iOS, por tanto, es el sistema operativo que ejecuta en dispositivo como iPhone, iPod, iPad, Apple TV. [21]

4.5.5.2.1. Arquitectura

En lo que compete a su kernel, iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix. iOS es un intermediario entre el hardware y las aplicaciones, cuenta con cuatro capas de abstracción: Cocoa Touch, Media, Core Services y Core OS.

La administración de procesos se realiza de acuerdo a una cola de prioridades pudiendo ser éstas normales, de prioridad alta, modo kernel o hilos de tiempo real. La administración de memoria la realiza de manera virtual. [21]

4.5.5.2.2. Características

- Está basado en tecnología multitáctil, empleándose gestos como deslizar o tocar para interactuar con las aplicaciones. [21]
- Su última versión es iOS 9. [21]
- Es soportado sólo por arquitecturas ARM. [21]
- Sólo funciona con dispositivos Apple. [21]
- La multitarea se da a partir de iOS4, pero sólo a través de siete APIS que proveen audio en segundo plano, voz IP, localización, notificaciones push, notificaciones locales, completado de tareas, cambio rápido de aplicaciones. [21]

4.5.5.3. Sistema Operativo Android

Android es un sistema operativo móvil basado en Linux enfocado para ser utilizado en dispositivos móviles como, teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, liderada por Google. [21]

4.5.5.3.1. Arquitectura

Los componentes principales del sistema operativo de Android se componen de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos

sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Para la gestión de procesos, Android utiliza el método CFS (Completely Fair Scheduler o Planificador Completamente Justo) que busca mantener el balance en el tiempo que se le otorga a cada proceso que busca mantener el balance en el tiempo que se le otorga a cada proceso; para lo cual guarda el tiempo de asignación llamado Virtual Runtime, el proceso con menor “Virtual Runtime” es el más próximo a ser ejecutado. [21]

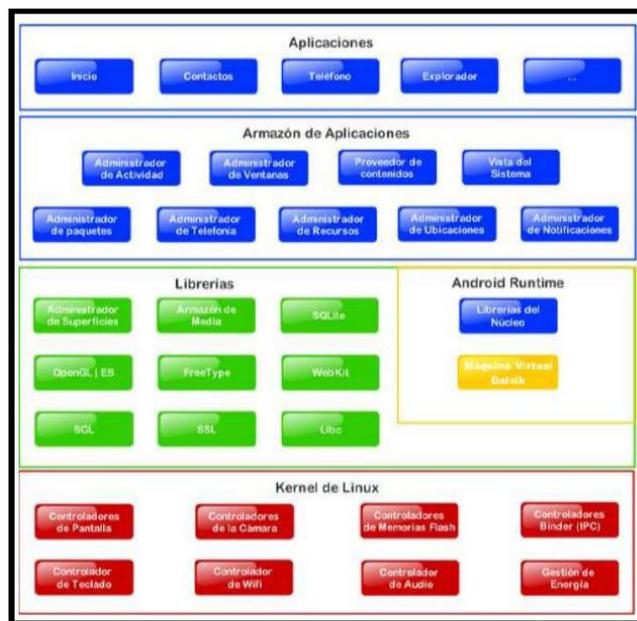


Figura 10. Arquitectura Android [10]

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), Existen otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C, C++ u otros lenguajes de programación.

4.5.5.3.2. Características

Cada aplicación de Android corre su propio proceso de Linux, y a su vez, cada uno de estos procesos corre su propia Máquina virtual Java. (Se aísla la ejecución entre aplicaciones).

Para facilitar la reutilización de código y agilizar el proceso de desarrollo, las aplicaciones Android se basan en componentes, ya sean actividades, servicios, receptores de eventos y proveedores de contenido.

Además, todas las aplicaciones Android deben tener un fichero AndroidManifest.xml donde se definan todos los componentes de la aplicación, así como los permisos que requiere, o los recursos y librerías que utiliza.

4.5.6. Tecnologías y herramientas utilizadas en el desarrollo de la aplicación móvil

Con el análisis realizado a algunas aplicaciones móviles para las Universidades de Educación Superior se pudo determinar que la mayoría de ellas son multiplataforma, lo cual permite que dichas aplicaciones puedan ser utilizadas por la mayoría de usuarios que cuenta con un dispositivo móvil.

A continuación, se explicará detalladamente como se utilizó cada una de las tecnologías y herramientas mencionadas.

4.5.6.1. Framework para el desarrollo de la aplicación móvil

Con el propósito de normalizar y estructurar el código del sistema, facilitando un esquema (un patrón, un esqueleto) para el desarrollo y/o la implementación de aplicaciones. El uso de frameworks para cualquier tipo de desarrollo reduce el tiempo de elaboración e implementación y ayuda a hacer un trabajo mantenible y escalable, según las características del mismo.

Un framework agrega funcionalidad extendida a un lenguaje de programación, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito, proporcionando una estructura al código, mejorándolo y haciéndolo más entendible y sostenible, y permite separar en capas la aplicación. En general, divide la aplicación en tres capas.

- La lógica de presentación que administra las interacciones entre el usuario y el software.
- La Lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.
- La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.

4.5.6.1.1. Ionic

Ionic es un marco de desarrollo en HTML5 dirigida a la creación de aplicaciones móviles híbridos, estas son esencialmente pequeños sitios web que se ejecutan en un navegador que tienen acceso a la capa de plataforma nativa. Las aplicaciones híbridas tienen muchas ventajas sobre las aplicaciones nativas puras, específicamente en términos de soporte de la plataforma, la velocidad de desarrollo, y el acceso al código.

4.5.6.1.2. Phonegap

PhoneGap se desarrolló originalmente por la empresa Nitobi en 2011 y fue adquirido posteriormente por Adobe, liberando el código bajo el proyecto Apache Cordova. En otras palabras, el producto que pertenece a Adobe se denomina actualmente PhoneGap y el proyecto de código abierto se denomina Cordova. Varios de los framework descritos en este documento utilizan Cordova. El objetivo de este framework es permitir desarrollar aplicaciones embebidas en código nativo utilizando exclusivamente código html5 y Javascript (Ghatol, 2012). Phone Gap Build ofrece varios tipos de plan pagos. Debido a que el código se compila a través de gitHub, el derecho a que el código sea privado depende del precio del plan. La versión gratuita permite una sola aplicación privada y la de paga va desde USD \$120 a USD \$900 por año, dependiendo del número de aplicaciones.

4.5.6.1.2.1. Prestaciones

Phonegap brinda las siguientes prestaciones:

- Aplicaciones nativas de dispositivos móviles para acceso a la información personal de la página web de los estudiantes de la Universidad Politécnica Salesiana.
- Soporta las plataformas Android, iOS, BlackBerry OS, Windows Phone, WebOS, Symbian y Bada.
- Está disponible en formato de plugin para diferentes IDE's como Eclipse o Xcode. No obstante, también ofrece la opción PhoneGap Build, un servicio gratuito en la "nube".
- Proporciona la API de JavaScript que permite tener acceso a las capacidades del dispositivo como cámara, GPS, información del dispositivo, etc. Además, posee un componente nativo, el cual trabaja de forma independiente y hace el trabajo en el dispositivo.
- Permite integrarse con frameworks de desarrollo web móvil como JQuery Mobile y Sencha Touch.
- El framework está disponible para ser instalado en las tres plataformas principales que son Windows, Linux y Mac

4.5.6.1.2.2. Ventajas

- Utiliza un lenguaje de programación fácil de aprender y de poca complejidad.
- No requiere instalación de aplicaciones complementarias
- El framework es de código abierto.
- Acceso a elementos y hardware del dispositivo mediante JavaScript, como cámara, GPS, acelerómetro, etc.

4.5.6.1.2.3. Desventajas

- Acceso a elementos y hardware del dispositivo mediante JavaScript, como cámara, GPS, acelerómetro, etc.
- No cuenta con su propio IDE.
- Acceso limitado a accesorios y hardware del dispositivo.
- Al no programarse en un lenguaje nativo del sistema operativo móvil, el rendimiento es afectado en un mínimo de porcentaje.

4.5.7. Metodologías para aplicaciones móviles

En el siguiente apartado se abordan las características principales de las metodologías para el desarrollo de aplicaciones móviles, las metodologías descritas son: Extreme Programming, Scrum para dispositivos móviles y Mobile-D.

4.5.7.1. Metodología Extreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen ambiente de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en la soluciones implementadas y coraje para enfrentar los cambios. XP se define como una metodología especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [24] [25] [26] [27]

4.5.7.1.1. Fases de la metodología extreme programming

La metodología extreme programming (XP) posee las fases: planificación, diseño, desarrollo y pruebas.

4.5.7.1.1.1. Planificación

- Se redactan las historias de usuarios.
- Se crea un plan de entregas.
- Se controla la velocidad del proyecto.
- Se divide el proyecto en iteraciones.
- Al comienzo de cada iteración se traza el plan de iteración.
- Se rota el personal.
- Cada día se concoca una reunión de seguimiento.
- Corregir la propia metodología XP cuando falla.

4.5.7.1.1.2. Diseño

- Realizar las cosas de la manera más sencilla.
- Coherencia de los nombres de todo lo que se va a implementar.

- Usar tarjetas CRC. En esta tarjeta se registra los nombres de las clases, sus responsabilidades y con qué otras clases colaboran.
- Soluciones puntuales para reducir riesgos.
- No se añadirá funcionalidad en las primeras etapas.
- Reaprovechar cuando sea posible.

4.5.7.1.1.3. Desarrollo

- El cliente está siempre disponible.
- Se debe escribir código de acuerdo a los estándares.
- Desarrollar la unidad de pruebas primero.
- Todo el código debe programarse por parejas.
- Solo una pareja se encargará de integrar el código.
- Actualizar las versiones de los módulos lo más rápido posible.
- Todo el código es común para todos.
- Dejar las optimizaciones para el final.

4.5.7.1.1.4. Pruebas

- Todo el código debe ir acompañado de su unidad de pruebas.
- Todo el código debe pasar las unidades de pruebas antes de ser implementado.
- Crear una unidad de pruebas para protegerse del mismo.
- Se deben ejecutar pruebas de aceptación a menudo y publicar los resultados.
- Las pruebas unitarias. [24]

4.5.7.2. Metodología Scrum para dispositivos móviles

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro, porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto, realizado por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80 y reestructurado por Jeff Sutherland. [25]

Scrum se básicamente se basa en la iteración y entrega de incrementales del desarrollo del producto software o servicio. [27]

4.5.7.2.1. Características de la metodología Scrum

La metodología Scrum posee las siguientes características:

- Su prioridad es la satisfacción del cliente; que se da con la continua interacción entre este y el equipo de desarrolladores.
- Se aceptan requisitos cambiantes.
- Enfocado a conseguir pequeños incrementos de software completamente funcionales.

- Es un modo de desarrollo adaptable, antes que predictivo.
- Orientado a las personas, más que a los procesos.
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.
- Alta flexibilidad.

4.5.7.2.2. Fases de la metodología Scrum para dispositivos móviles

Scrum posee las mismas fases que otras metodologías, caracterizándose por la realización de sprints con duración de dos a tres semanas.

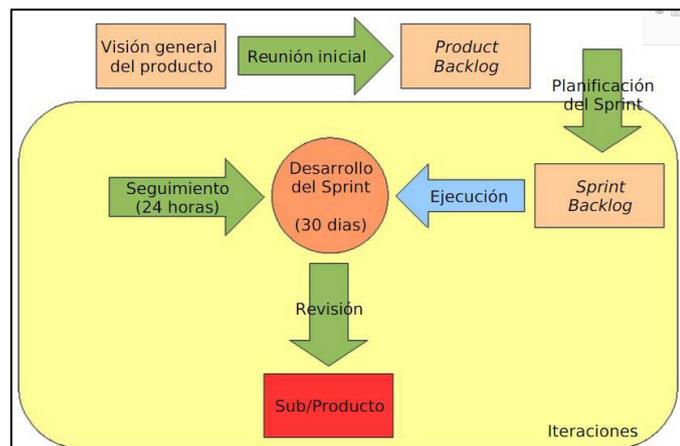


Figura 11. Ciclo de vida de Scrum para móviles[21]

4.5.7.2.3. Planificación Inicial

Esta reunión sucede una única vez al inicio del proyecto. Se reúne el grupo de trabajo con el representante del cliente. Se determinan los requisitos iniciales y la visión del producto desde el punto de vista del negocio. Se consolida esta información en el Product Backlog. [25]

4.5.7.2.4. Product Backlog

Su desarrollo parte de la visión del propietario del producto, la cual debe ser real, comprendida y compartida por parte de todo el equipo.

Es un inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones del desarrollo. En él se documenta todo lo que implique un trabajo. Nunca está completo, se considera en permanente evolución. [27]

4.5.7.2.5. Planificación del Sprint

Esta es gestionada por el Scrum Manager, realizándola en dos etapas, con una duración no mayor a un día de trabajo.

La primera etapa dura máximo 4 horas y en ella se definen cuales funcionalidades del Product Backlog se implementarán en la siguiente iteración. La segunda se descompone las funcionalidades elegidas en las tareas necesarias para implementar el incremento del producto.

Cada sprint, a excepción del primero, deberá generar un producto probado y funcional de interés para el cliente. Se auto-asignan las tareas a un integrante del equipo y se estiman los recursos necesarios para su desarrollo. El producto resultante de esta planificación es el Sprint Backlog. [21]

4.5.7.2.6. Sprint

Un Sprint en Scrum es el término que denomina a una iteración que está acotada en el tiempo, usualmente entre 2 y 4 semanas, durante la cual el Equipo trabaja.

4.5.7.2.7. Sprint Backlog

Como se mencionó en la sección de la Planificación del sprint, el Sprint Backlog incluye un listado de las funcionalidades que se van a desarrollar durante la siguiente iteración, desglosado en tareas con recursos definidos y asignadas a un integrante del grupo de trabajo.

Incluye un objetivo del sprint al rededor del cual giran las funcionalidades a implementarse. Sólo los miembros del equipo de trabajo pueden modificarlo durante el sprint. Es visible para todos los miembros del equipo. [27]

4.5.7.2.8. Desarrollo del Sprint

Después de realizado el Sprint Backlog el sprint puede iniciar su desarrollo. Su duración no deberá exceder los 30 días. Es desarrollado por el grupo de trabajo de acuerdo a la planeación realizada anteriormente. Durante su desarrollo el Product Backlog se congela y no se aceptan modificaciones. Su ejecución es continuamente monitoreada. [27]

4.5.7.2.9. Revisión del Sprint

Al finalizar cada sprint se realiza una reunión en la que el equipo de trabajo presenta al propietario del producto y demás involucrados el incremento construido durante el sprint. Su duración es de máximo 4 horas. Su preparación es de máximo 1 hora.

Este le permite al propietario del producto conocer de primera mano el estado actual del desarrollo del proyecto. El producto expuesto debe estar en su etapa final: terminado, probado y operando en el entorno del cliente (incremento). Puede incluir también documentación de usuario o técnica según se haya pactado.

Al final de la reunión se interroga individualmente a todos los asistentes para recabar impresiones, sugerencias de cambio y mejora, y su relevancia. El propietario del producto trata con los asistentes y con el equipo las posibles modificaciones en el Product Backlog. Se convoca para la reunión de planeación del próximo sprint. [21]

4.5.7.2.10. Retrospectiva del Sprint

La Retrospectiva del Sprint es una reunión facilitada por el Scrum Master en la cual los Miembros del equipo de Scrum discuten el Sprint que acaba de finalizar, y determinan qué podría cambiarse en el próximo Sprint para que sea más productivo y mejor. La Revisión del Sprint se focaliza en "Qué" construye equipo, mientras que la Retrospectiva se centra en "Cómo" están construyendo el sistema.

Cualquier cosa que afecte cómo el equipo construye software está abierto para discutirse. Esto puede incluir: procesos, prácticas, comunicación, entorno, artefactos y herramientas. [25]

4.5.7.3. Metodología Mobile-D

El Objetivo de esta metodología es conseguir ciclos de desarrollo muy rapidos en equipos muy pequeños. Fue creada en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: extreme programming, Crystal Methodologies y Rational Unified Process. [28]

4.5.7.3.1. Fases de la metodología Mobile-D

El ciclo del proyecto se divide en cinco fases: exploración, inicialización, productización, estabilización y prueba del sistema. En general, todas las fases (con la excepción de la primera fase exploratoria) contienen tres días de desarrollo distintos: planificación, trabajo y liberación. Se añadirán días para acciones adicionales en casos particulares (se necesitarán días para la preparación del proyecto en la fase de inicialización). [29]

4.5.7.3.2. Exploración

En esta fase se centra la atención en la planificación y a los conceptos básicos del proyecto. Aquí es donde hacemos una definición del alcance del proyecto y su establecimiento con las funcionalidades donde queremos llegar.

En la iniciación configuramos el proyecto identificando y preparando todos los recursos necesarios como hemos comentado anteriormente en esta fase la dedicaremos un día a la planificación y el resto al trabajo y publicación.

Subfases:

- Establecimiento de los stakeholders
- Definición del alcance

- Establecimiento del proyecto

Establecimiento de las partes interesadas: es una etapa en la que todos los grupos de interés relevantes excluyendo el equipo del proyecto en sí se necesita en el establecimiento, así como en las diferentes tareas del proyecto incipiente se definen con las funciones y los recursos pertinentes. Además del grupo de clientes (que se define en el patrón de tareas Establecimiento del cliente), los grupos de interés en Mobile-D pueden incluir, por ejemplo, del grupo de supervisión, gestión de proyectos, grupos de la arquitectura, y el proceso de los especialistas. Todos estos actores juegan un papel vital en las tareas posteriores de Explora fase y en la ejecución del proyecto.

Definición del Alcance: es una etapa en la que los objetivos y el alcance del incipiente proyecto de desarrollo de software se definen y acordado por las partes interesadas grupos. Esto incluye temas como los requisitos (iniciales) para el producto y la línea de tiempo del proyecto.

Establecimiento del Proyecto: es una etapa de acordar las cuestiones ambientales del proyecto (físicas y técnicas), así como el personal necesario en el desarrollo de software (promotores y de apoyo). Además, las cuestiones de proceso se definen en esta etapa.

Salidas:

- Los requisitos iniciales documentan en que se han definido los requisitos iniciales,
- Plan del proyecto que incluye la línea de tiempo, el ritmo, las terminaciones, los recursos proyecciones / grupos de interés y sus responsabilidades,
- Base descripción del proceso que incluye el proceso de la línea de base, actividades documentación, puntos de integración (con, por ejemplo, salidas proyecciones hardware), y
- Plan de Medida,
- Plan de capacitación y
- Descripción de línea Architecture.

4.5.7.3.3. Inicialización

Esta fase tiene el propósito de asegurar el éxito de las próximas fases del proyecto mediante la preparación y verificación de todas las cuestiones fundamentales del desarrollo a fin de que todo esté en plena disposición para la aplicación de los requisitos seleccionado por el cliente.

- Entre los documentos obtenidos en esta etapa están:

- El plan de proyecto actualizado
- La primera versión de la Arquitectura del Software y la descripción del Diseño.
- Primera versión de Product Backlog.
- Documento de requisitos iniciales actualizado
- Notas de Desarrollo e Interfaz de usuario

4.5.7.3.4. Producción

El objetivo es la implementación de las funcionalidades requeridas del producto mediante la aplicación de un ciclo de desarrollo iterativo e incremental. Se repiten iterativamente las subfases. Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase podemos decir que se lleva a acabo toda la implementación.

Entre los documentos obtenidos se encuentran los siguientes:

- Funcionalidades implementadas
- Notas de Desarrollo
- Ilustraciones acerca de la Interfaz de Usuario
- StoryBoards y StoryCards
- Documento de Requisitos actualizado

4.5.7.3.5. Estabilización

El propósito de esta fase es el aseguramiento de la calidad de la implementación del proyecto. Esta etapa es en la que se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación.

Luego de culminada su duración se obtiene:

- La funcionalidad implementada de todo el software de proyecto
- La documentación del producto finalizado.

4.5.7.3.6. Pruebas

Una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

Una vez acabada todas las fases se debería tener una aplicación publicable y entregable al cliente. [30]

4.5.8. Metodología de desarrollo software

4.5.8.1. Modelo Arquitectura 4+1

El modelo 4+1 describe la arquitectura del software usando cinco vistas concurrentes. Cada vista se refiere a un conjunto de intereses de diferentes stakeholders del sistema: Los usuarios finales, los desarrolladores, entre otros; y de esta manera manejar por separado los requerimientos funcionales y no funcionales.

El modelo propone las siguientes perspectivas o vistas: [31], [32]

- Vista lógica: Ofrece soporte a los requerimientos funcionales, lo que el sistema debe proveer en términos de servicios a sus usuarios. en la vista lógica se mostrará la división del sistema en subsistemas y paquetes.
- Vista de procesos: La vista de procesos permite describir los procesos del sistema y como estos se comunican. Esta vista toma en cuenta algunos requerimientos no-funcionales, como lo son disponibilidad, desempeño y tolerancia a fallos; tomando en cuenta la distribución, integridad y concurrencia del sistema.
- La vista física: describe el mapeo del software en el hardware y refleja los aspectos de distribución.
- Vista de desarrollo o de implementación: Esta vista se concentra en la organización en módulos del software.
- Vista de casos de uso: La vista de casos de uso consolida las vistas anteriores, donde los escenarios se convierten en una abstracción de los requerimientos más importantes.

4.5.9. Metodología para revisión sistemática

4.5.9.1. Metodología propuesta de Barbara Kitchenham

Barbara Kitchenham presenta un método para la realización de revisiones sistemáticas en el contexto de Ingeniería del Software, antes de dicha propuesta no se disponía de ninguna guía o método eficiente para realizar estudios exhaustivos en tal contexto. Para la realización de una revisión sistemática se involucran diferentes actividades independientes, para lo cual el método propone tres fases fundamentales.[33]

- La primera fase es la planificación de la revisión que consta de las actividades de:
 - Identificación de la necesidad de la revisión y
 - Desarrollo de un protocolo de revisión.
- La segunda fase es el desarrollo de la revisión que consta de las siguientes actividades:
 - Identificación de la investigación

- Selección de los estudios primarios
 - Evaluación de la calidad del estudio
 - Extracción y monitoreo de datos y
 - Síntesis de datos
- La tercera fase es la publicación de los resultados de la revisión.

5. MATERIALES Y MÉTODOS

Para la realización del presente trabajo de titulación titulado “IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES”, se ha creído conveniente utilizar materiales y métodos descritos a continuación.

5.1. Materiales

Los materiales y métodos para el desarrollo del presente trabajo han sido seleccionados bajo las comparativas realizadas, es así que se ha usado herramientas Open Source como la plataforma Debian, servidor Icecast2, cliente Butt Broadcast para el desarrollo del sistema de audio streaming y para la evaluación se utiliza Winbox y Apache JMeter.

De la misma manera para el desarrollo de la aplicación móvil se utiliza el Framework Ionic, el editor de texto Sublime Text, DIA, Lucichart, los cuales sirven para desarrollar diagramas, prototipos, y la codificación correspondiente a la aplicación.

5.2. Métodos y técnicas

Para el desarrollo del presente proyecto se utilizó diferentes métodos y técnicas que ayuden a lograr cumplir con los objetivos planteados.

5.2.1. Métodos

Método Deductivo: se utilizó para analizar las particularidades de la situación actual en la forma emisión de audio por parte de la Radio Universitaria; con el propósito de obtener un diagnóstico de la problemática para extraer las causas y características. Asimismo, gracias a la generalización de los datos recolectados en la etapa de análisis se logró implementar el streaming de audio en la radio, para cumplir con los objetivos planteados del presente trabajo de titulación, y realizar las conclusiones y recomendaciones.

Método inductivo: este método se utilizó para extraer los problemas principales y generales de la falta de un sistema de audio streaming para la Radio Universitaria, a partir de los cuales se determinó la problemática en forma general que se presenta en la cobertura de emisión de la radio. A si mismo se logró obtener, clasificar y deducir por medio del razonamiento lógico, la información y documentación para obtener un conocimiento verdadero que fortalezca el desarrollo del proyecto.

5.2.2. Técnicas

Entrevista: Con esta técnica se obtuvo información relevante como la tecnología que cuenta y utiliza la Radio Universitaria, y poder determinar las necesidades existentes en la misma.

Cuestionario: Con esta técnica se obtuvo información relevante y necesaria referente a requerimientos por parte de las partes interesadas para el desarrollo del proyecto y la misma que ayudara a sustentarlo y justificarlo.

Investigación Bibliográfica: Esta técnica se sustenta la base teórica del presente trabajo de titulación, mediante consultas a fuentes bibliográficas confiables, libros, revistas indexadas, artículos científicos, base de datos científicas entre otras.

5.2.3. Metodología

Se planteó la utilización de dos metodologías en donde, la metodología de Barbara Kitchenham se emplea para realizar el análisis y selección de las herramientas para la implementación del streaming y la metodología de desarrollo ágil Mobile-D, la cual se utiliza para el desarrollo de la aplicación móvil, permitiéndonos avanzar de forma rápida en cada una de sus cinco fases y tener que elaborar un nivel medio de documentación, únicamente lo necesario para la guía y desarrollo de la aplicación.

6. RESULTADOS

6.1. OBJETIVO 1: ANALIZAR LA SITUACIÓN ACTUAL DEL SERVICIO DE LA RADIO UNIVERSITARIA EN LOS MEDIOS DE DIFUSIÓN DIGITAL

6.1.1. Situación actual de la radio

La Radio Universitaria funciona en la frecuencia 98.5 FM transmitiendo su programación analógicamente las 24 horas al día los 7 días a la semana siendo dicha radio de carácter académica, la cobertura de la radio es de carácter local perteneciendo dicha localidad a la ciudad de Loja y sus alrededores, en esta radio laboran dos personas encargadas de la programación y administración de la radio la Ing. Paulina Jara y el Sr. Stewart Naranjo algunas funciones que cumplen son: periodismo, oficinista, pauta de cuñas, presentadores de noticias, monitoreo de la radio y asistencia técnica. De la misma manera se pone a consideración que la radio actualmente no cuenta con el servicio de streaming, esto acontece desde el 4 de febrero de 2011 hasta la actualidad y como medios de difusión digital para dar a conocer su imagen se utiliza la página web de la Universidad Nacional de Loja en su sección de servicios, así mismo se utilizan las redes sociales como Facebook y Twitter contando con sus cuentas oficiales de las mismas.

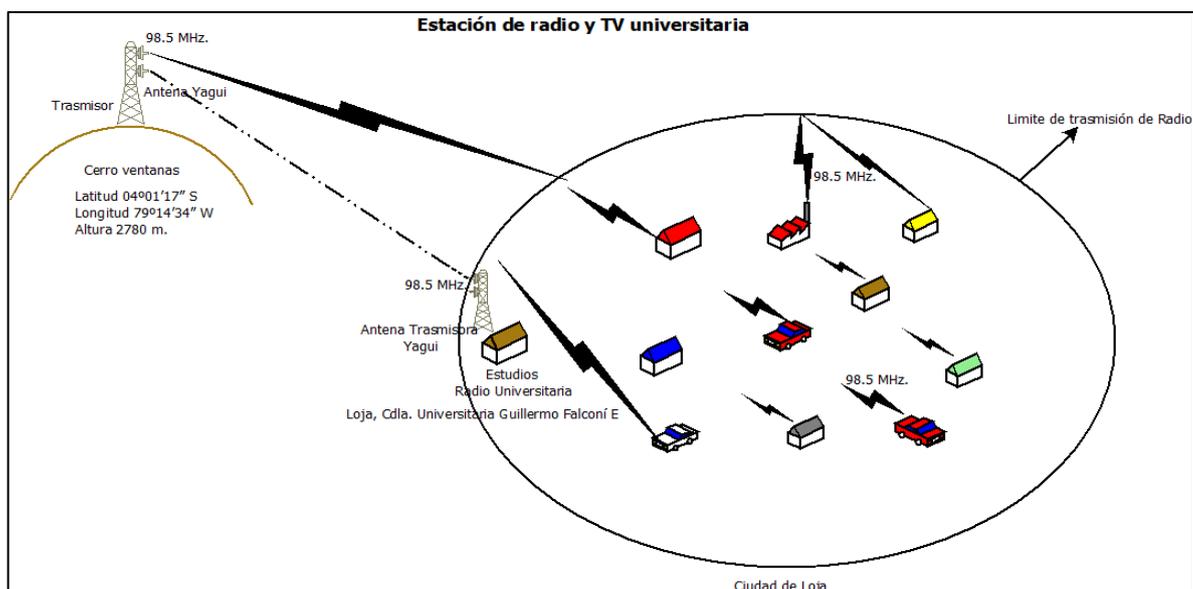


Figura 12. Trasmisión de la Radio Universitaria

La Figura 12 muestra la transmisión analógica que está llevando la radio hacia sus radioyentes contando para esta con antenas transmisoras una en las instalaciones de la Radio Universitaria y la segunda en el Cerro Ventanas abarcando la transmisión en la ciudad de Loja y sus alrededores.

Los Datos Técnicos de Estaciones de Radiodifusión en FM, se representan en las tablas de la II hasta la VI, cuya información se obtiene de los documentos en el Anexo 1.

6.1.2. Datos generales

TABLA II. DATOS GENERALES DE LA RADIO UNIVERSITARIA

Nombre de la estación	Estación de radio y TV universitaria
Concesionario	Universidad Nacional de Loja
Representante Legal	Dr. Reinaldo Valarezo García
Categoría	Servicio Publico
Área de Servicio	Loja y sus alrededores para una intensidad de campo de 500 uV/m.
Provincia	Loja
Indicativo	HC UL3

6.1.3. Ubicación y alturas

TABLA III. UBICACIÓN Y ALTURAS DE LOS ESTUDIOS Y TRASMISORES DE LA RADIO UNIVERSITARIA

Nombre	Ubicación	Altura
Estudios	Loja, Cdla. Universitaria Guillermo Falconí E.	Latitud 03°59'44" S Longitud 79°12'00" W Altura 2100 m.
Trasmisor	Cerro Ventanas	Latitud 04°01'17" S Longitud 79°14'34" W Altura 2780 m.

6.1.4. Características técnicas

TABLA IV. CARACTERÍSTICAS TÉCNICAS DE LA RADIO UNIVERSITARIA

Frecuencia de Operación	98.5 MHz
Tipo de emisión	200KF8EHN

Potencia de Trasmisión	1000 w
Tipo y forma de antena	Arreglo de 4 radiadores
Polarización	Circular
Angulo de azimut de máxima radiación	23°
Angulo de elevación	-6.2°
Marca y tipo de transmisor	ITELCO
Excursión máxima de frecuencia	+ - 75KHz
Tipo de enlace	Radioeléctrico
Área de cobertura técnica	La aprobada por la Superintendencia de Telecomunicaciones de acuerdo al estudio técnico presentado, para una intensidad de campo eléctrico de 500 uV/m

TABLA V. CARACTERÍSTICAS TÉCNICAS DEL ENLACE ESTUDIO-TRASMISOR

Características Técnicas del enlace Estudio-Trasmisor	
Nombre de la estación	Estación de radio y TV universitaria
Frecuencia	943.25 MHz
Tipo de emisión	200KFEHN
Potencia de Salida	5 w
Antena trasmisora	Yagi

TABLA VI. POLARIZACIÓN HORIZONTAL / VERTICAL

Polarización Horizontal/Vertical	
Azimut de Tx	237°
Equipo de Tx	OMB Modelo MT
Antena Receptora	Yagi
Polarización	Horizontal/Vertical
Azimut de RX	57°
Equipo de RX	OMB Modelo MR

6.1.5. Parrilla de programación

La programación de la Radio Universitaria al ser 24 horas al día cuenta con su respectiva parrilla de programación la misma que contiene: anuncios publicitarios, avances institucionales, cuñas de servicio social, conservación del medio ambiente, programas especiales, etc.

De la misma manera para poder generar dichas programaciones y su contenido se debe de tomar en cuenta las franjas horarias: [3]

TABLA VII. FRANJAS HORARIAS

Franja	Descripción	Horario
Familiar	Incluye a todos los miembros de la familia. Esta franja solo se podrá difundir programación de clasificación "A": Apta para todo público.	Desde las 06h00 a las 18h00.
Responsabilidad Compartida	Dirigida a personas de entre 12 y 18 años con supervisión de personas adultas. Esta franja se podrá difundir información de clasificación "A" y "B": Apta para todo público con vigilancia de una persona adulta	Desde las 18h00 a las 22h00.
Adultos	Dirigida a personas mayores de edad (18 años). En esta franja se podrá difundir programación clasificada con "A", "B", "C": Apta solo para personas adultas.	Desde las 22h00 a las 06h00

6.1.6. Infraestructura de la red de la Radio Universitaria

La Universidad cuenta con un modelo jerárquico de 3 capas: Capa de núcleo, capa de distribución y capa de acceso de la misma manera cuenta con 300Mbps de Internet comercial y 1Gbps de Internet avanzado, este ancho de banda se encuentra distribuido en todo el campus Universitario utilizando para ello el enrutamiento OSPF y configuraciones VTP, VLAN, DHCP, ACL, SNMP, Spanning tree, Port security.

La Radio Universitaria se encuentra conectada al switch L3 del bloque 10 del Área Jurídica a su vez perteneciendo a la subred 10.10.128.0/30. La radio para acceder a la

red de datos cuenta con equipos que operan en la capa de acceso del modelo jerárquico, dichos equipos son (ver tabla VIII):

TABLA VIII. EQUIPOS DE RED DE LA RADIO UNIVERSITARIA

Descripción	Marca	Modelo	Características Técnicas
Switch	Cisco	C2960X	24 PS-L
Router	Linksys	EA6900 AC1900	Switch con 4 puertos LAN y 1 puerto WAN, todos ellos son Gigabit Ethernet.

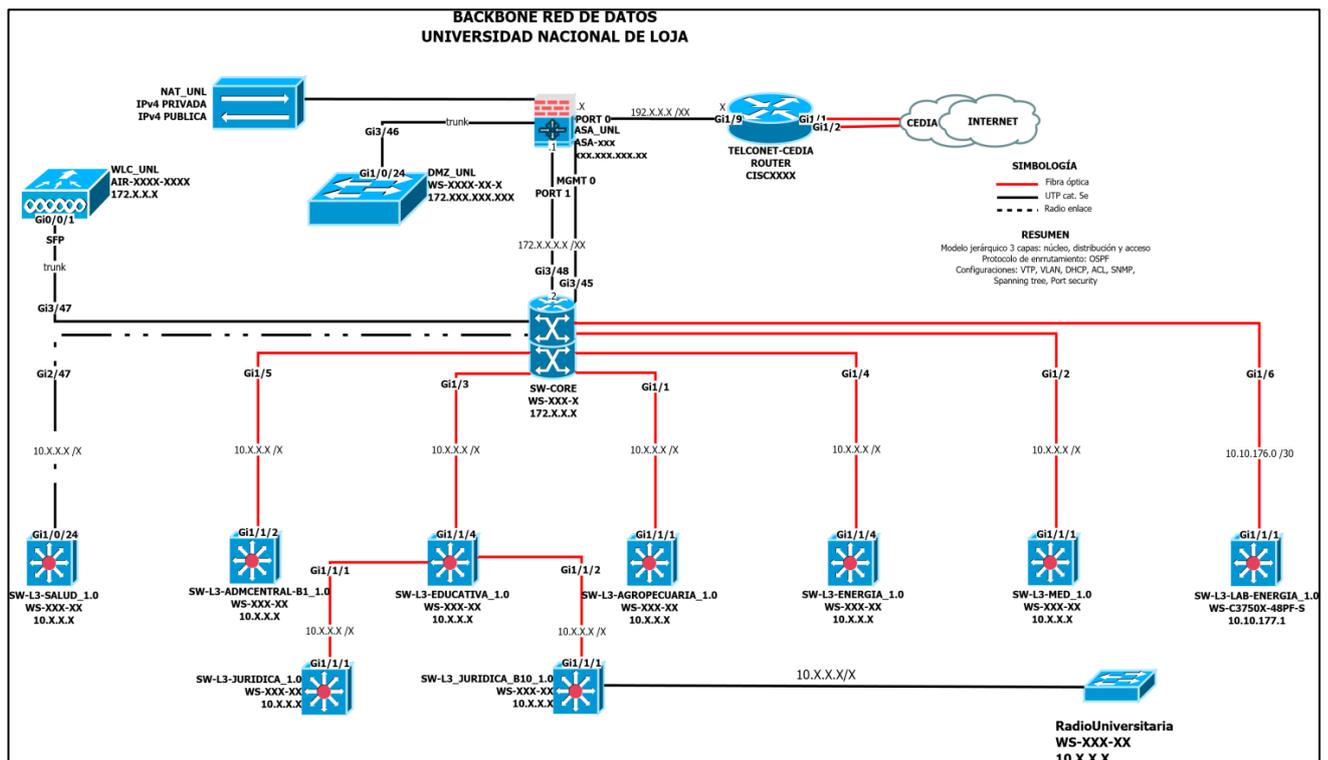


Figura 13. Backbone de la Universidad Nacional de Loja

Fuente: Documentos del Departamento de la UTI

La Figura 13 representa la conexión hacia el Core de la Universidad Nacional de Loja, la cual va desde la Radio Universitaria, esta se encuentra en la VLAN 10 con la dirección IP 10.1x.x.x/xx, conectándose por cable UTP desde el switch de la radio hacia el switch del área jurídica y este a su vez al switch del área educativa mediante una conexión de fibra óptica, este último se conecta al switch core por medio de fibra óptica, el ancho de banda que existe en la radio es de un aproximado de 10Mbps/s ya que toda la Institución

Educativa no cuenta con segmentación de ancho de banda para hacer uso de la red de datos de la misma.

6.1.7. Diagrama de servidores

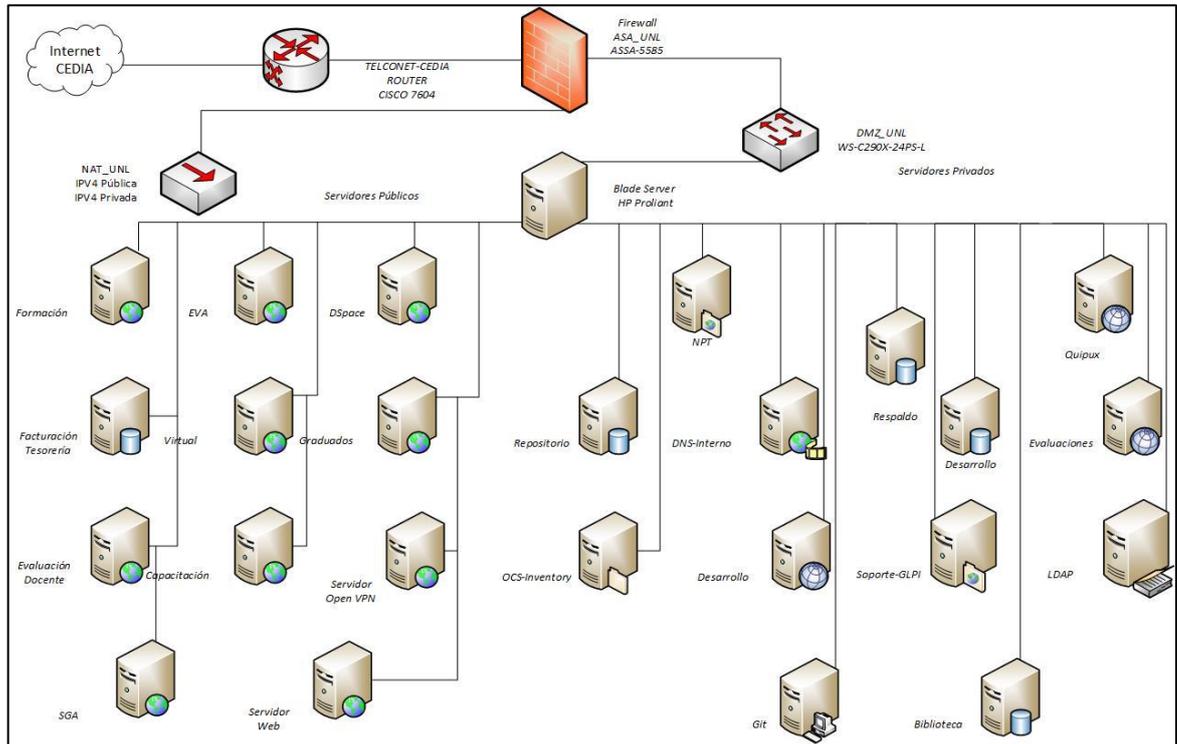


Figura 14. Diagrama de servidores de la Universidad Nacional de Loja

La Figura 14 nos muestra los servidores con que cuenta la Universidad Nacional de Loja a la parte izquierda se encuentran los once servidores públicos y a la parte derecha se encuentran los trece servidores privados, entre los servidores públicos se encuentra lo que es el EVA, SGA, seguimiento a graduados, DSpace, etc, los mismos que prestan servicios a los usuarios públicos. Y en los servidores privados se encuentran servidores de repositorios, respaldos, evaluaciones, etc, los mismos que son de servicio privado y ciertos usuarios tienen acceso a los mismos. Todos estos servidores se encuentran conectados al Blade Server HP Proliant y este a su vez conectado a la DMZ de la universidad.

6.1.8. Equipos de producción y emisión de la radio

La Radio Universitaria cuenta con equipos de producción y emisión los mismos que se representan a continuación. (Ver figura 15)

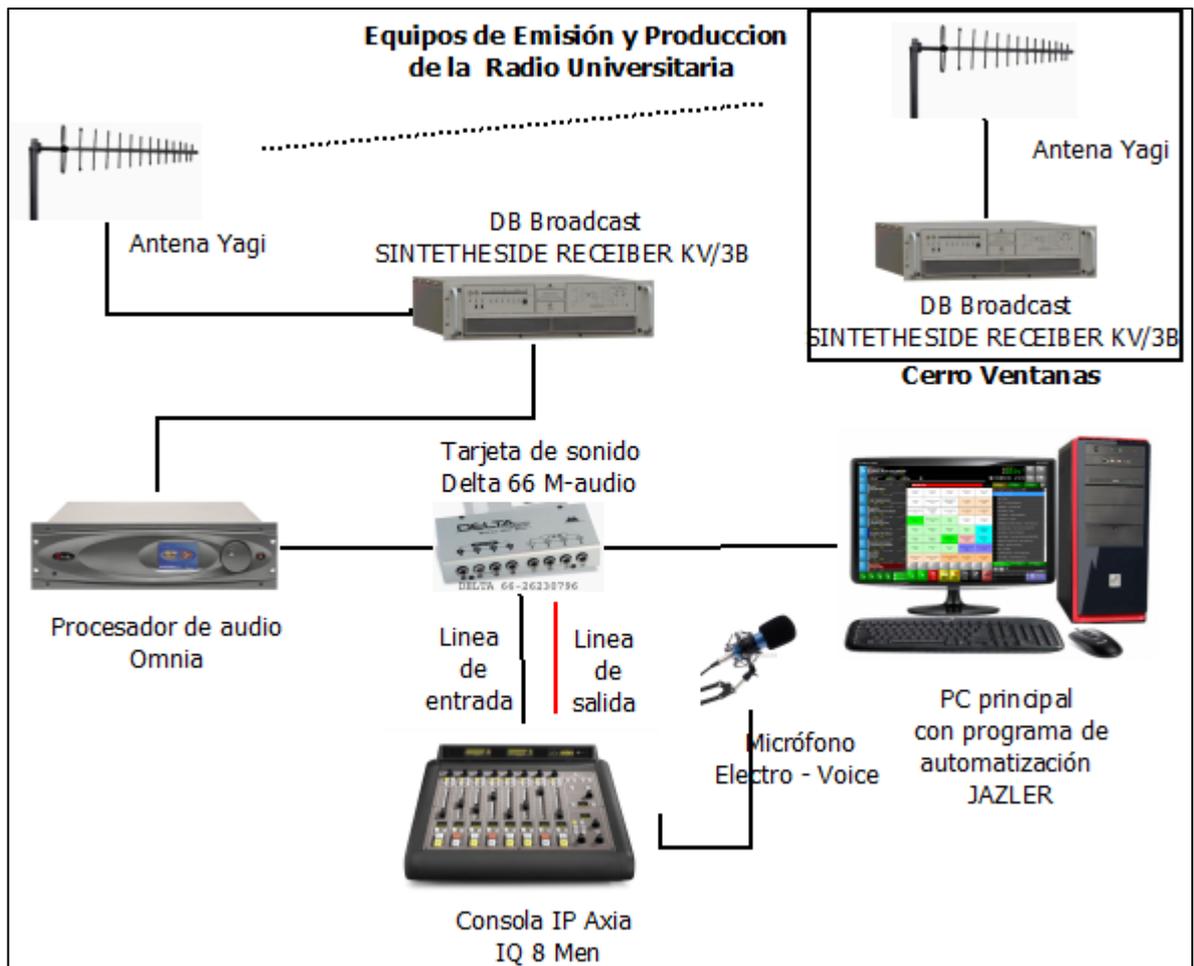


Figura 15. Equipos de emisión y producción de la Radio Universitaria

La Figura 15 nos muestra un esquema básico del funcionamiento interno de los equipos de la Radio Universitaria el cual cuentan con una PC principal en el cual se encuentra ejecutado el programa de automatización JAZLER el mismo, que es uno de los elementos principales para el funcionamiento 24 horas al día los 7 días a la semana, esta PC se encuentra conectada a la tarjeta de sonido y posterior a la consola IP la misma que permite su administración mediante conexión remota, además de abrir o cerrar canales de audio y dar paso a uno u otro micrófono, está nuevamente se conecta a la tarjeta de audio y posteriormente al procesador de audio mismo que permite mejorar todo el sonido que se requiere transmitir hacia los radioyentes, una vez procesado este se conecta a un DB broadcast para ser transmitido hacia su receptor que se encuentra en el cerro ventanas, para ello se utiliza las antenas yagi para conectar dichas señales de trasmisión y recepción y de esta manera emitir su señal de radio.

TABLA IX. INVENTARIO HARDWARE DE LA RADIO UNIVERSITARIA

Inventario de Hardware			
DESCRIPCIÓN	MARCA	MODELO	CARACTERISTICAS TÉCNICAS
2 Computador	Mac Apple	1419	Sistema Operativo Mac OS X, Core i7
2 Monitor	HP COMPAQ	LEI711	15" color
Monitor	SAMSUNG	SYNC master 750 ST	17" color SVGA
CPU	HP	-----	Sistema Operativo Windows 7, 4Gb de RAM, Procesador Intel Core 2 Duo, Disco Duro 1TB+ 810Gb
CPU	COMPAQ	-----	Sistema Operativo GNU/EterTICs, 1Gb de RAM, Procesador Intel Pentium 4, Disco duro 250GB
Consola ip	Axia	iQ 8 MEN	8 canales, 22 entradas, capacidad de administración remota
Audífonos	Stilo	SENNHEISER/ HD280 PRO	Frecuencia: 8Hz a 25Khz; Nivel de presión sonora: 102 dB (IEC 268-7); Distorsión total de armónicos: 0,1%; acoplamiento al oído
4 Micrófonos Profesionales	Electro - Voice	RE-20	Para la locución y entrevistas
Antena para transmisiones en vivo	Yagi	MARTI/ YC-450	Frecuencias 475 Mhz, 10dBd de ganancia.
Procesador de audio	Omnia	4.5 estéreo	Cumple con el estándar de 50 o 75 microsegundo, norma dentro de $\pm 0,50$ dB, 30 Hz a 15 kHz, salidas y entradas analógicas y digitales,

			sistema de separación de estéreos.
Tarjeta de sonido	Delta	66 M-audio	4 entradas, 4 salidas, entrada/salida S/PDFI digital
Equipo de enlace para transmisión	DB Broadcast	SINTETHESIDE RECEIBER KV/3B	Transmisor de la serie PM / excitadores operan en el rango de frecuencias de FM (87,5 - 108 MHz) La potencia de salida RF es continuamente ajustable de 0 a 300 W (PM 300), 500W (PM 500) o 1000W (PM 1000)
UPS	TRIPP LITE	SU3000RTXL3U	22VA (16600 WATIOS)
Consola de audio	AEQ S	BC-300	132 Canales sistema hibrido de teléfono

TABLA X. INVENTARIO SOFTWARE DE LA RADIO UNIVERSITARIA

Inventario Software		
Nombre	Descripción	Licencia
JAZLER	Software completamente integrado de automatización para radio, puede operar una radio automáticamente las 24horas incluyendo programas con playlists, locuciones, jingles, tandas musicales, etc.	Propietario
Pro Tools Express	SOFTWARE de edición y procesamiento de audio para la producción de audio. Permite grabar, componer, editar y mezclar con gran rapidez y facilidad. Permite Crear música y sonidos con una enorme colección de más de 75 instrumentos virtuales, efectos de procesamiento de sonido, y la utilidad	Propietario

	plug-ins, además de una enorme biblioteca de 8 GB de bucles de audio de alta calidad.	
--	---	--

Con estos equipos hardware y software la Radio Universitaria puede realizar su programación y transmitir a todos sus radioescuchas.

6.1.9. Índice de radioyentes

De acuerdo con el estudio realizado en el año del 2015 en la población Urbana de la Ciudad de Loja, la Radio Universitaria 98.5 FM se encuentra en el puesto 11 del ranking de sintonía, la misma que ocupa este puesto por los programas más escuchados como son: emociones, charlas narradas, universidad al día-(noticiero) y transmisiones deportivas. Es así que se llega a un porcentaje de 2.70% de radioyentes con respecto a las demás radios en la Ciudad de Loja. [34]

6.1.10. Viabilidad de implementación del streaming

Durante los últimos diez años, la tecnología de audio de codificación ha hecho enormes progresos. Se han desarrollado muchos sistemas de codificación muy avanzados utilizados con éxito en la radiodifusión y el Internet. Por ello muchas estaciones de radio están en una fase de transición al entrar la radio en la era digital DAB (Digital Audio Broadcasting), donde la tecnología Streaming es aquella que reduce el tiempo de espera en la transmisión de la información ya que se lo realiza en vivo; su importancia radica en que permite la transmisión en tiempo real ya sea de audio o video, es así que al realizar el análisis de la situación actual de la radio se presenta varias necesidades las mismas que se puede resumir en la siguiente tabla.

TABLA XI. NECESIDADES DE LA RADIO UNIVERSITARIA

Necesidad	Problema
Integrar la Radio Universitaria a medios de difusión digital.	La radio solamente se encuentra disponible en un medio analógico.
Integrar el servicio de radio on-line.	La radio no cuenta con una amplia cobertura de transmisión.
Integrar el streaming a la página oficial de la Radio Universitaria.	La radio cuenta con una página oficial dentro de la Universidad donde no se cuenta con un reproductor-streaming) para poder escuchar la transmisión en vivo que esta emite.

<p>Desarrollar una aplicación para dispositivos móviles para acceder al streaming mediante estos dispositivos.</p>	<p>La radio al estar en un medio analógico y al estar en una evolución tecnológica con el uso dispositivos móviles como Smartphone, tabletas se ve en necesidad de crear un acceso más fácil hacia la radio por medio de estos dispositivos.</p>
--	--

Se determina que la implementación del streaming es posible, pero se toma en consideración la limitante en la infraestructura que es en el ancho de banda, debido a que el switch permite únicamente consumir hasta un máximo de 100Mbps, lo cual genera una limitante en conexiones de usuarios hacia el streaming.

6.2. OBJETIVO 2: DETERMINAR LOS REQUERIMIENTOS DE HARDWARE Y SOFTWARE PARA EL STREAMING

6.2.1. Selección de software:

6.2.1.1. Servidores Streaming

“El servidor streaming es el elemento principal de la cadena en cuanto a calidad del servicio se refiere. El servidor procesa los datos multimedia en cortos espacios de tiempo, además es el responsable de suministrar los servicios de audio y video en modo sincronizado.

El servidor de streaming está a la espera de la petición RTSP (Real Time Streaming Protocol) desde el usuario. Cuando recibe una petición, el servidor busca en el directorio apropiado el contenido media del nombre solicitado. Si el contenido media está en el directorio solicitado, el servidor hace streaming hacia el solicitante utilizando RTP (Real-time Transport Protocol).” [9]

6.2.1.1.1. Icecast2

TABLA XII. ICECAST2

Nombre	Descripción	Ventajas	Desventajas
Icecast2	Icecast2 es un servidor de streaming de medios (audio / vídeo), que actualmente soporta Ogg (Vorbis y Theora), Opus, WebM y MP3 corrientes.	Icecast2 se distribuye bajo la GNU GPL, versión 2. Soporta los códec Ogg (Vorbis y Theora), Opus, WebM y MP3 Tiene puntos de montaje (mountpoints). Transfiere oyentes del Auto DJ para transmisión en vivo y viceversa sin necesidad de activar manualmente.	No tiene panel de administración para echar oyentes. No cuenta con un directorio tan popular como es el de shoutcast por lo que su emisora tal vez no logre tanta visibilidad como una emisora configurada con shoutcast.

		<p>Mayor compatibilidad con móviles con enlaces (iPhone y Android)</p> <p>Multiplataforma.</p> <p>Es flexible, permite diferentes orígenes del sonido a reproducirse como fuentes de reproducción (playlists) predefinidas.</p> <p>Compatible con los estándares abiertos de comunicación e interacción</p>	
--	--	---	--

6.2.1.1.2. ShoutCast

TABLA XIII. VENTAJAS Y DESVENTAJAS DE SERVIDOR SHOUTCAST

Nombre	Descripción	Ventajas	Desventajas
ShoutCast	Es un servidor streaming y un es el último directorio de estaciones de radio en línea. Con más de 50.000 música, entrevistas, deportes y estaciones de radio	<p>Cuenta con panel de administración para echar a oyentes o dj.</p> <p>Es más compatibles con los PC.</p> <p>Cuenta con directorio gratuito el cual permite listar su emisora de radio, dando más visibilidad a su emisora y poder atraer más oyentes.</p>	<p>Puede transmitir AutoDJ o vivo, pero no simultaneo.</p> <p>Versión Free limitada</p> <p>Actualmente ShoutCast es propietario.</p>

	comunitarias de todo el mundo.	Soporta codificación MP3, AAC, AAC+ y NSV. Multiplataforma. El plug-in de Shoutcast (Gratuito) Sencillo de configurar.	
--	--------------------------------	---	--

6.2.1.1.3. Darwin Streaming Server

TABLA XIV. VENTAJAS Y DESVENTAJAS DE SERVIDOR DARWIN STREAMING SERVER

Nombre	Descripción	Ventajas	Desventajas
Darwin Streaming Server	Es un servidor de código abierto que le permite enviar la transmisión de medios a los clientes a través de Internet utilizando el estándar de la industria RTP y RTSP.	Libre distribución. Multiplataforma. Emite cualquier contenido mediante RTP/RTSP. Distribuye archivos en formato MPEG-4. Trasmite archivos en formato MP3. Permite crear listas de reproducción para difundir contenido pregrabado. Permite utilizar un servidor como retransmisor de contenido multimedia de otro servidor.	Tiende a desbordamiento de búfer en el módulo QTSSReflector, cuando la solicitud ANNOUNCE manipulada con un valor muy grande es enviada, provocando un "Integer Overflow". Requiere aplicaciones de terceros para la codificación. No tiene las herramientas y extensiones adicionales que son

		<p>Aumento de calidad de stream, presenta una colección de servicios de calidad.</p> <p>Soporta transmisiones multicast y broadcast</p> <p>Permite modificar, manipular y adaptar el código fuente del servidor.</p>	<p>exclusivas de versión comercial.</p> <p>No posee en ningún momento soporte técnico de Apple.</p>
--	--	--	---

6.2.1.1.4. Comparativa de Servidores Streaming

TABLA XV.COMPARATIVA DE SERVIDORES STREAMING

Nombre	Icecast2	ShoutCast	Darwin Streaming Server
Licencia libre.	SI	Versión libre limitada	SI
Multiplataforma.	SI	SI	SI
Formatos de difusión de audio.	SI	SI	SI
Necesidad de terceras aplicaciones para codificación.	NO	NO	SI

Transferencia de oyentes del Auto DJ para transmisión en vivo y viceversa sin necesidad de activar manualmente.	SI	NO	NO
Soporta transmisiones multicast y broadcast	SI	SI	SI
Permite crear listas de reproducción para difundir contenido pregrabado.	SI	SI	SI

6.2.1.2. Procesador de Señal Digital

Es un software que se encarga de recibir la señal de la tarjeta de audio y procesarla. Luego se conecta con el servidor por Internet y el cual envía la voz y música por pequeños paquetes digitales. El servidor recibe los datos y los va entregando a las personas que “sintonicen” el programa.[10]

6.2.1.2.1. Butt Broadcast

TABLA XVI. VENTAJAS Y DESVENTAJAS DE CLIENTE BUTT BROADCAST

Nombre	Descripción	Ventajas	Desventajas
Butt Broadcast	Es una herramienta fácil de usar, multi sistema operativo para enviar flujos	Multiplataforma. Compatible con ShoutCast e Icecast2.	No está diseñada para ser un servidor por sí mismo o

	<p>de audio a un servidor streaming. El objetivo principal de Butt es para transmitir datos de audio en directo desde su computador micrófono o la entrada de línea a un servidor Shoutcast o Icecast2.</p>	<p>Los archivos de configuración pueden ser importados y exportados</p> <p>Se ejecuta en los tres principales sistemas operativos. Mac OS X, Linux y Windows 8.1</p> <p>Soporta AAC +, MP3, OGG / Vorbis y Ogg / Opus para los flujos.</p> <p>Soporta AAC +, MP3, OGG / Vorbis, Ogg / opus, FLAC y WAV para la grabación.</p> <p>Es capaz de iniciar una grabación después de conectarse a un servidor de forma automática.</p> <p>Reconecta automáticamente en caso de interrupción de la conexión.</p>	<p>automáticamente transmitir un conjunto de archivos de audio.</p>
--	---	--	---

6.2.1.2.2. Ices2

TABLA XVII. VENTAJAS Y DESVENTAJAS DE CLIENTE ICES2

Nombre	Descripción	Ventajas	Desventajas
Ices2	Es un cliente de código abierto para el servidor de streaming Icecast2. El propósito de este cliente es el de proporcionar un flujo de audio de Icecast2, de modo que uno o más oyentes pueden acceder a la transmisión.	Admite el envío de un flujo Ogg Vorbis a un servidor Icecast2. Compatible con la mayoría de versiones Unix (incluyendo Redhat y Debian) Tiene soporte por las personas que están involucradas con Icecast2.	No trabaja con diferentes códecs. Para codificar utiliza una utilidad como aumix / alsamixer para ver la configuración y cambiar el dispositivo de captura o grabación.

6.2.1.2.3. VLC

TABLA XVIII. VENTAJAS Y DESVENTAJAS DE CLIENTE VLC

Nombre	Descripción	Ventajas	Desventajas
VLC	VLC es un reproductor multimedia libre y de código abierto multiplataforma y un framework que reproduce la mayoría de archivos multimedia, así como DVD, Audio CD, VCD y	Reproduce todo archivo, discos, cámaras web y flujos. Reproduce la mayoría de códecs sin paquetes de códecs necesarios - MPEG-2, MPEG-4, H.264, MKV, WebM, WMV, MP3. Ejecutable en todas las plataformas: Windows,	No se reconecta si existe problemas de conexión de red. No está preparado para recibir dos emisiones diferentes en la misma dirección multicast.

	diversos protocolos de transmisión.	Linux, Mac OS X, Unix, iOS, Android. Completamente gratis sin software espía, sin anuncios y sin seguimiento de usuario. Compatible con el servidor Icecast2 y ShoutCast.	
--	-------------------------------------	---	--

6.2.1.2.4. Comparativa de DSP (Procesador de Señal Digital)

TABLA XIX. COMPARATIVA DE DSPS

Nombre	Plataformas	Códecs	Compatibilidad de servidores	Reconexión automática
Butt Broadcast	Linux, Windows, Mac OS X	AAC +, MP3, OGG / Vorbis y Ogg / Opus	Icecast2 y ShoutCast	SI
Ices2	Linux	Ogg Vorbis	Icecast2	NO
VLC	Linux, Windows, Mac OS X, Unix, iOS y Android	MPEG-2, MPEG-4, H.264, MKV, WebM, WMV, MP3.	Icecast2 ShoutCast	NO

6.2.1.3. Códecs

El término códec viene de los términos codificar/decodificar y comprimir/descomprimir. Es una pieza de software capaz de transformar un flujo de datos o señal. Un códec codifica un flujo de dato o señal para su almacenaje o transmisión y lo codifica para su reproducción. [11], [12]

De acuerdo a los resultados de la revisión sistemática se analizará los dos códecs más utilizados en implementaciones de streaming los cuales son MP3 y Ogg Vorbis.

6.2.1.3.1. MP3

TABLA XX. VENTAJAS Y DESVENTAJAS DE CÓDEC MP3

Nombre	Descripción	Ventajas	Desventajas
MP3	MP3, o llamado exactamente MPEG-1 Audio Layer 3, es una forma de codificar audio usando un algoritmo de compresión que genera pérdida de datos reduciendo la cantidad de información requerida para representar una grabación de audio, pero de modo en que a su vez sea muy similar a la calidad del archivo original para la mayoría de los oyentes.	<p>Tasa de bits de 8 kb/s hasta 320 kb/s.</p> <p>Alta velocidad de codificación.</p> <p>Permite reproducción sin pausas por medio de LAME-Headers (solo para decodificadores compatibles).</p> <p>Posee licencia GPL a una LGPL, permitiendo su uso en aplicaciones comerciales.</p> <p>Es adaptable para la mayoría de los dispositivos, incluyendo todos los navegadores web.</p>	<p>Es un códec con pérdidas.</p> <p>La calidad de sonido no es muy buena</p> <p>Se pierde los derechos de autor.</p> <p>Es patentado.</p>

		<p>Es muy accesible.</p> <p>Permite seleccionar en tres tipos de tasa de bits: CBR, VBR, ABR.</p> <p>Compresión: 11 a 1</p>	
--	--	---	--

6.2.1.3.2. Ogg Vorbis

TABLA XXI. VENTAJAS Y DESVENTAJAS CÓDEC OGG VORBIS

Nombre	Descripción	Ventajas	Desventajas
Ogg Vorbis	Es un códec de audio libre de compresión con pérdida desarrollado por Xiph.org. Forma parte del proyecto Ogg y entonces es llamado Ogg Vorbis y también solo Ogg por ser el códec más comúnmente encontrado en el contenedor Ogg.	<p>Soporta tasas de bit de 32 kb/s hasta 500 kb/s</p> <p>Tasas de muestreo de 8 kHz a 48 kHz</p> <p>Soporte de tasas de bits constantes (CBR) y variables (VBR)</p> <p>Mejor calidad que MP3</p> <p>Permite realizar streaming fácilmente.</p> <p>No es patentado</p>	<p>Es un códec con pérdidas</p> <p>No tiene mucha compatibilidad como MP3</p>

6.2.1.3.3. Comparativa de códecs

TABLA XXII. COMPARATIVA DE CÓDECS

Nombre	Tasas de bits	Compatibilidad con todos los dispositivos	Patentado
MP3	8 kb/s hasta 320 kb/s	Compatibilidad con todos los dispositivos, incluyendo todos los navegadores web.	Sí, pero posee licencia GPL a una LGPL, permitiendo su uso en aplicaciones comerciales.
Ogg Vorbis	32 kb/s hasta 500 kb/s	Compatible solo con algunos dispositivos	No, este códec es de uso libre

6.2.1.4. Plataforma

Para la selección de la plataforma se tomó como referencia las dos principales plataformas que utiliza la Unidad de Telecomunicaciones para los diferentes servidores y ejecución de aplicaciones dichas plataformas son Linux Debian y Centos.

6.2.1.4.1. Debian

TABLA XXIII. VENTAJAS, DESVENTAJAS Y REQUISITOS DEL SISTEMA DE LA PLATAFORMA DEBIAN

Nombre	Descripción	Ventajas	Desventajas	Requisitos del sistema
Debian	El Proyecto Debian es una asociación	Respaldado por el trabajo constante y la participación de desarrolladores.[35]	No todo el hardware esta soportado.	CPU A 1GHz como mínimo. 64MB de RM (256MB

	<p>de personas que han hecho causa común para crear un sistema operativo (SO) libre.</p> <p>Los sistemas Debian actualmente usan el núcleo de Linux o de FreeBSD.</p>	<p>Instalación sencilla e incluye casi 43.000 elementos software.[35]</p> <p>Integración de los paquetes y código fuente.</p> <p>Estabilidad.</p> <p>Rápido y ligero en memoria.</p> <p>Buena seguridad del sistema.</p>	<p>Falta de software comercial popular.[35]</p>	<p>recomendados) y 1GB de espacio de disco para la instalación sin escritorio.</p> <p>128MB de RAM (512 recomendado) y 5GB de espacio en disco para la instalación con escritorio.</p> <p>Tarjeta VGA</p> <p>Lector de CD, puerto USB o conexión a red Ethernet.</p>
--	---	--	---	--

6.2.1.4.2. Centos

TABLA XXIV. VENTAJAS, DESVENTAJAS Y REQUISITOS DEL SISTEMA DE LA PLATAFORMA CENTOS

Nombre	Descripción	Ventajas	Desventajas	Requisitos del sistema
Centos	La distribución CentOS Linux es una plataforma estable, predecible, manejable y reproducible	Se basa en Red Hat Enterprise y compatible con sus productos.	Limitaciones para audio y video sincronizado. Si falla el servidor falla todo.	CPU a 1GHz como mínimo. 128MB de memoria RAM, 512 recomendados.

	<p>derivado de las fuentes de Red Hat Enterprise Linux (RHEL). Para las comunidades de código abierto, ofrece una base sólida y previsible para construir sobre, sus amplios recursos además la liberación, y mantener su código.</p>	<p>Es un sistema estable y eficaz en el consumo de recursos.</p> <p>Esta optimizado para correr Apache, PHP, MySQL.</p>	<p>Para escalar hay que añadir servidores.</p>	<p>1,2GB de espacio en disco, recomendados 2GB.</p> <p>Tarjeta gráfica VGA y monitor capaz de soportar una resolución de 1024x768.</p> <p>Lector de CD-ROM o puerto USB.</p>
--	---	---	--	--

Por medio de la “Revisión sistemática de herramientas para la implementación de streaming en la Radio Universitaria basado en software libre”, se concluye que las herramientas idóneas son: Icecast2, Butt Broadcast, instalados sobre una plataforma libre en este caso Debian. Es así que se elige Icecast2 como servidor de streaming, por ser multiplataforma, poseer una licencia GPL, compatibilidad con la mayoría de dispositivos, entre ellos los dispositivos móviles, además soporta varios códecs. De la misma manera se elige Butt Broadcast como Procesador de Señal Digital (DSP), el cual enviara el flujo de audio hacia el servidor de streaming, elegido por su compatibilidad con el servidor, ser multiplataforma, soportar varios códecs y reconectarse de forma automática tras una interrupción en la conexión en internet. Finalmente se selecciona la plataforma Debian, para la cual se toma en cuenta los requerimientos mínimos que se necesitan para su instalación y la facilidad de instalar el servidor Icecast2. [36]

Se elige MP3 como códec para él envió de flujo de audio, por su gran compatibilidad con todos los dispositivos, entre ellos navegadores web que permiten su reproducción,

siendo su compresión de 11 a 1 y permite su distribución bajo una licencia LGPL la cual admite su uso en aplicaciones comerciales. [36]

6.2.2. Determinación de Hardware a emplear para streaming

Para la determinar el hardware a emplear en el proyecto se lo realiza en base a que el software a emplear para el streaming es liviano y el sistema operativo Debian sin escritorio (sin entorno gráfico), se necesita 512Mb de RAM, 2GB de espacio en disco duro y con escritorio (con entorno gráfico), se necesita de 1GB de RAM, 10GB de disco duro. Con estos datos se dispone a describir en dos tablas los requerimientos mínimos, tanto para el servidor streaming como para el cliente (Procesador de señal digital).

TABLA XXV. CARACTERÍSTICAS MÍNIMAS PARA SERVIDOR STREAMING

Hardware Mínimo para servidor streaming	
Procesador	Intel o AMD
CPU	1 GHz
Memoria RAM	512 MB o más
Sistema Operativo	Linux Debian
Tamaño de disco	10 GB

TABLA XXVI. CARACTERÍSTICAS MÍNIMAS PARA EL CLIENTE
(PROCESADOR DE SEÑAL DIGITAL)

Hardware Mínimo para cliente	
Procesador	Intel o AMD
CPU	1GHz
Memoria RAM	1 GB o más
Sistema Operativo	Linux
Tamaño de Disco	10 GB

Una vez revisado el hardware mínimo y tomado en consideración los equipos que posee y proporciona la Unidad de Telecomunicaciones e Información, se detallan los dos equipos que se usan para la implementación. En el primer equipo se instala el servidor Icecast2 y en el segundo equipo se instala el cliente Butt Broadcast.

TABLA XXVII. HARDWARE PARA SERVIDOR ICECAST2

Servidor Virtual sobre plataforma KVM (QEMU VIRTUAL CPU)	
Procesador	GenuineIntel 1 GHz
Memoria	1 GB
Adaptador de Red	Realtek RTL-8100/8101L/8139 PCI Fast Ethernet Adapter
Tamaño en disco	10 GB
Sistema Operativo	GNU LINUX DEBIAN 8.2
Nombre de la Máquina	radio.unl.edu.ec

TABLA XXVIII. HARDWARE PARA CLIENTE (DSP)

Procesador	Intel (R) Xeon(RM) CPU 3.40GHz
Memoria	2GB
Adaptador de Red	Intel Corporation 82541GI
Tamaño en disco	80 GB
Sistema Operativo	GNU ETHERTICS
Nombre de la Máquina	RadioUnl

6.2.3. Ancho de banda

De acuerdo a los resultados de la revisión sistemática se determina, que el ancho de banda debe ser 256 kbps o más para una comunicación entre el cliente y el servidor, teniendo una compresión de audio mínima de 64 kbps para tener un audio de calidad y mayor compresión a medio- altos bitrates. Estos valores pueden ser actualizados posteriormente al realizar las pruebas del sistema. [36]

6.3. OBJETIVO 3: IMPLEMENTAR EL STREAMING PARA LA RADIO UNIVERSITARIA.

Los equipos utilizados en la implementación se ubicarán en dos edificios, el equipo para el servidor se instala en el Edificio 2 de Administración Central en las oficinas de la Unidad de Telecomunicaciones e Información (UTI), y cliente (DSP) se ubica en las instalaciones de la Radio Universitaria.

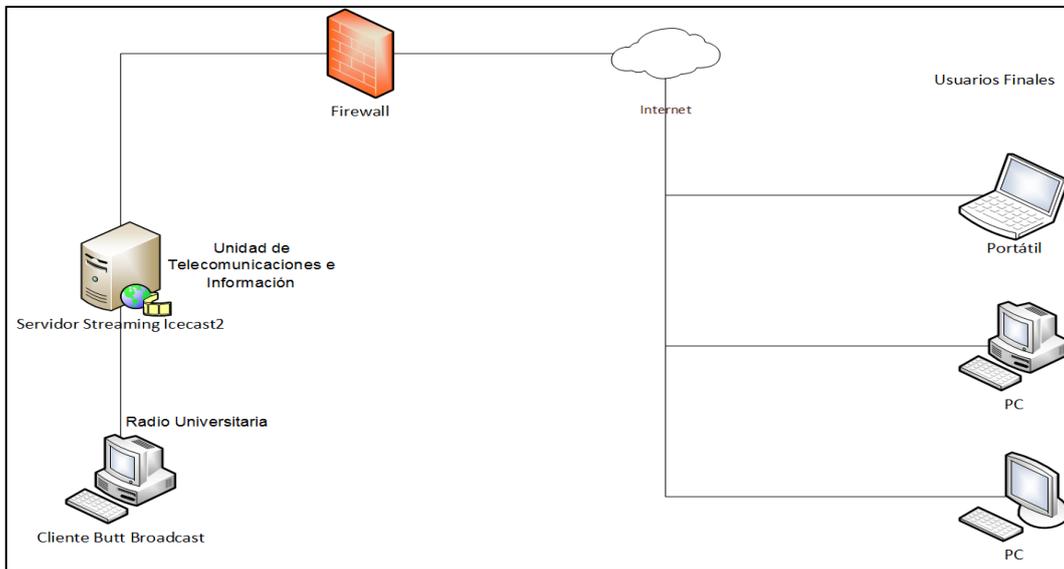


Figura 16. Arquitectura del sistema streaming

En la figura 16 se muestra la arquitectura del sistema streaming, donde se visualiza la conexión desde el cliente Butt Broadcast hasta el usuario final, el cual accede al streaming de la radio.



Figura 17. Diagrama de equipos de Radio Universitaria con conexión de Cliente y Servidor streaming

La figura 17 se puede visualizar la conexión de todos los equipos de la radio incluyendo el servidor Icecast2 y el cliente Butt Broadcast.

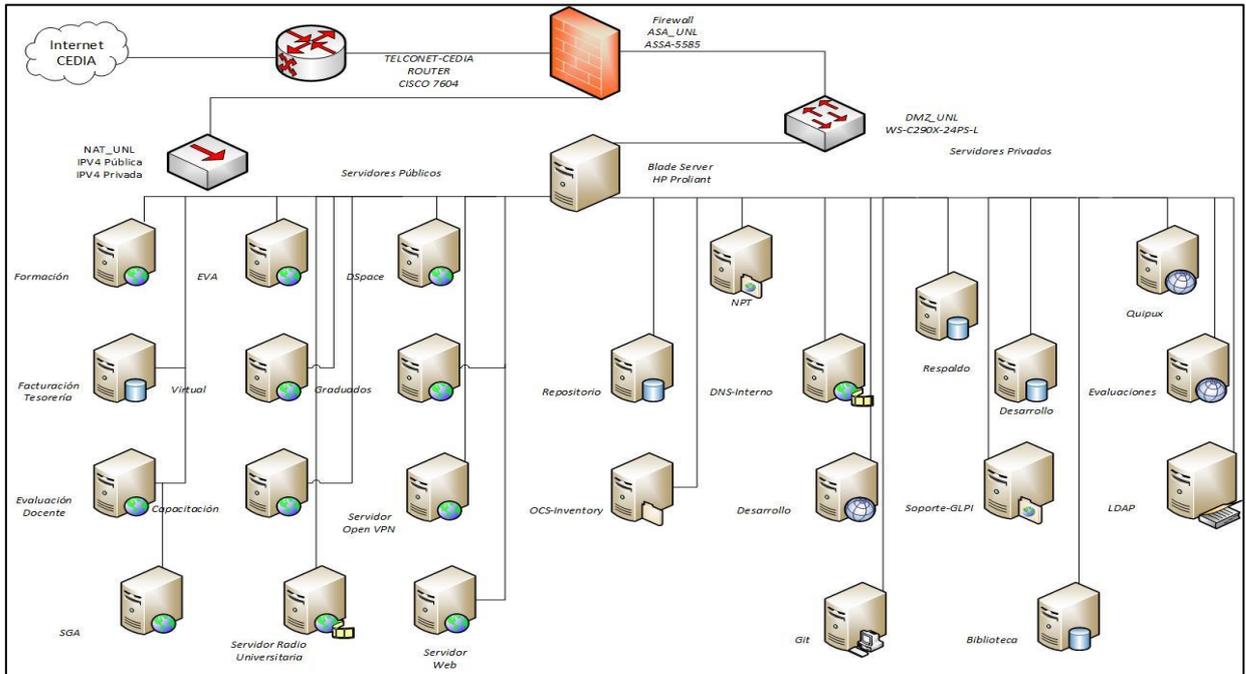


Figura 18. Diagrama de Servidores de UNL integrado el servidor de Radio Universitaria

6.3.1. Instalación y configuración de los servicios para streaming

6.3.1.1. Pseudocódigo de proceso de instalación de servidor streaming Icecast2

1. Preparar los repositorios en Linux Debian 8 Jessie
2. Instalar el servidor Icecast2
3. Entrar a configuraciones del Servidor
4. Configurar número de clientes
5. Configurar password en el source, relay,
6. Configurar usuario y password de servidor streaming Icecast2.
7. Configurar hostname
8. Configurar archivo Default
9. Guardar configuraciones
10. Inicializar el servicio de Icecast2

Para revisar todo el proceso de instalación y configuración revisar en los Anexos.

6.3.1.2. Pseudocódigo de proceso de instalación del Cliente Butt Broadcast

1. Descargar el instalador Butt Broadcast para Linux
2. Desempaquetar e instalar Butt Broadcast
3. Actualizar el sistema
4. Entrar a configuraciones de Butt Broadcast
5. Configurar hostname, Source y Password para servidor Icecast2
6. Configurar Punto de Montaje
7. Configurar información de streaming
8. Guardar configuraciones

Para revisar todo el proceso de instalación y configuración revisar en los Anexos.

6.3.1.3. Pseudocódigo de proceso de inicialización de streaming

1. Abrir Butt Broadcast
2. Inicializar streaming
3. Minimizar Butt Broadcast

Para revisar todo el proceso inicialización revisar en los Anexos.

6.3.1.4. Integrar streaming a plantilla web

Para acceder al streaming se desarrolla una plantilla web en la cual conste la dirección o punto de montaje del streaming de la radio para poder acceder al contenido.

Para ello se realiza los siguientes pasos:

1. Instalar Apache
2. Instalar MySQL
3. Instalar PHP
4. Configurar Apache
5. Editar página Índice de Apache
6. Integrar streaming a página índice de Apache.
7. Reiniciar servicios de Apache.

Para ver el proceso de instalación de Apache, MySQL y PHP visitar la siguiente dirección URL <https://www.digitalocean.com/community/tutorials/como-instalar-linux-apache-mysql-php-lamp-en-ubuntu-14-04-es>.

Para revisar el proceso de integrar el streaming en la página de Índice de Apache revisar en los Anexos.

```
<h1>Radio Universitaria 98.5 FM</h1>
<p>Universidad Nacional de Loja</p>
<audio controls autoplay >
<source src="http://radio.unl.edu.ec:8080/stream" type="audio/mpeg">
</audio>
```

Figura 19 Código de acceso a streaming en plantilla web

La figura 19, se presenta la porción de código HTML donde se integra la dirección o punto de montaje a la plantilla web, la cual es visualizada por el usuario y podrá acceder y escuchar el streaming en vivo que emite la radio.



Figura 20. Plantilla Web de la Radio Universitaria

6.3.1.5. Integrar streaming a plataforma TuneIn

Se realiza la integración del streaming de la radio a la plataforma TuneIn (Ver Anexo 7), la cual es una de las plataformas más utilizadas de radio online a nivel mundial y cuyo propósito es de tener a su disposición streaming de radios de todo el mundo en una sola plataforma disponible tanto en navegadores como en dispositivos móviles mediante su aplicación del mismo nombre.

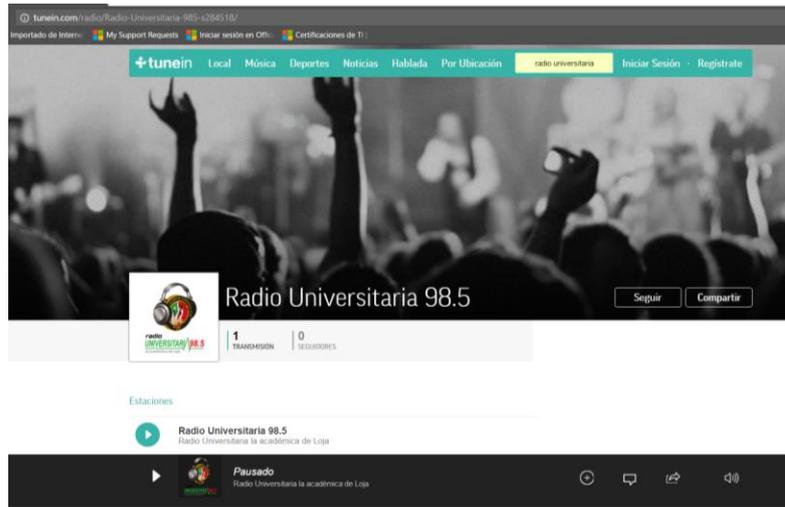


Figura 21. Radio Universitaria en plataforma Tuneln

6.3.1.6. Evaluación del sistema

Para la evaluación se realiza un conjunto de pruebas en las cuales se simula las situaciones más comunes de funcionamiento del sistema streaming. El objetivo de estas pruebas es determinar la capacidad y rendimiento del sistema streaming.

A continuación, en la figura 22 se representa todos los componentes conectados para realizar la evaluación del sistema, donde se encuentra conectado un portátil el cual tiene están instaladas y configuradas las herramientas Winbox y Apache JMeter, de la misma manera se visualiza la conexión al servidor y cliente Butt Broadcast.

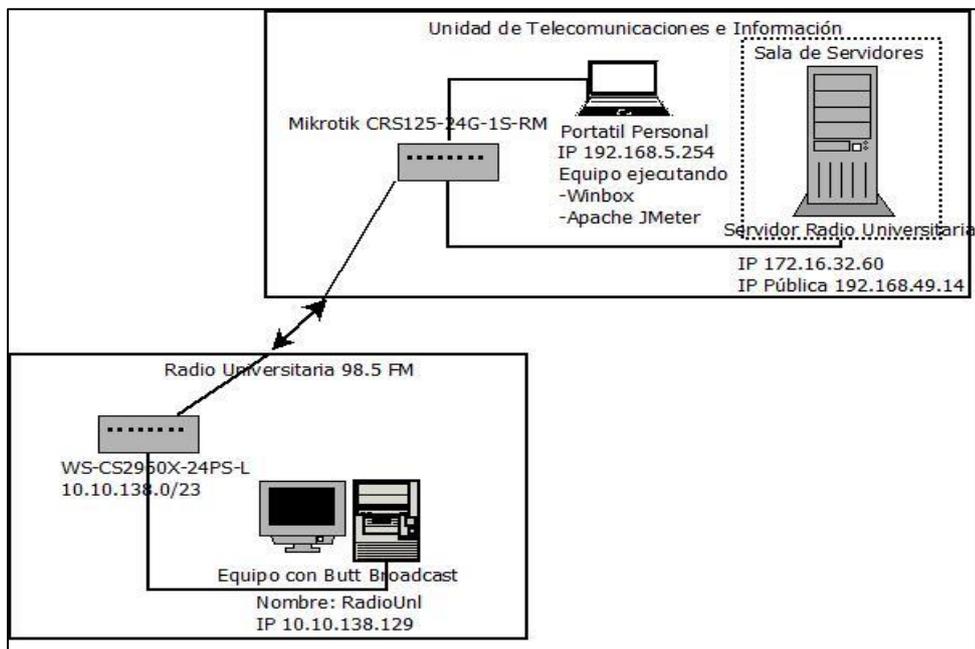


Figura 22. Componentes para evaluación del sistema

6.3.1.6.1. Herramientas y equipos para evaluación del sistema

Las herramientas que se utiliza para evaluar el rendimiento del sistema son: los registros que se da mediante el comando TOP aplicado en la consola de comandos de Linux, monitoreando de esta manera el funcionamiento del sistema, así mismo se cuenta con el software Apache JMeter, el cual nos permite emular a los usuarios que se conectaran al streaming y Winbox, donde por medio de su interfaces se puede visualizar el ancho de banda servido durante las conexiones que se realizan mediante Apache JMeter.

TABLA XXIX. EQUIPOS PARA EVALUACIÓN DEL SISTEMA

Equipo	Nombre	Software instalado
Servidor	radio.unl.edu.ec IP 172.16.32.60 IP Pública 192.168.49.14	Icecast2
Cliente	Computador Personal IP 192.168.5.254	Apache JMeter Winbox
Mikrotik	CRS125-24G-1S-RM	RouterOS
Encoder	RadioUnl IP 10.10.138.129	Butt Broadcast

6.3.1.6.2. Configuraciones de escenario de pruebas

Partiendo del esquema de la figura 22, se lista el proceso de configuración del escenario y como se lleva a cabo la evaluación del sistema.

1. Descargar, ejecutar y configurar Apache JMeter
2. Descargar, ejecutar y configurar Winbox
3. Emular los usuarios que se conectan al streaming en Apache JMeter
4. Visualizar los resultados en la pantalla de Apache JMeter
5. Visualizar el ancho de banda consumido en las interfaces de Winbox
6. Visualizar el monitor de servidor Icecast2 mediante un navegador web

Para más detalles de instalación y configuración de las herramientas, y escenario ver en los Anexos.

6.3.1.6.3. Pruebas de Capacidad

El objetivo de las pruebas de capacidad es determinar el máximo de tráfico multimedia que puede soportar la red de la Universidad Nacional de Loja, sin que afecte el funcionamiento de otras aplicaciones en la red. Se realiza dos pruebas en días distintos (Prueba No. 1) y (Prueba No.2). Donde se utiliza como clientes los hilos (grupos de usuarios) emulados por Apache JMeter.

6.3.1.6.3.1. Prueba N° 1

Para revisar el tráfico de red que existe al conectar 1000 usuarios, se configura el grupo de usuarios (hilos) dentro Apache JMeter, para que este emule las 1000 peticiones de conexión al sistema streaming.

Para más detalles de configuración del grupo de usuarios revisar el Anexo 6.

6.3.1.6.3.1.1. Resultados

Del registro global tanto de Apache JMeter y Winbox se obtiene los valores promedios representados en la siguiente tabla:

TABLA XXX. RESULTADOS PRUEBA NO. 1: VALORES PROMEDIOS EMULANDO 1000 USUARIOS.

Parámetro	Valor Promedio
Ancho de banda	92.3Mbps enviados, 3.9Mbps recibidos
Nº de paquetes enviados por el servidor	7468
Número de usuarios concurrentes	943
Pico de usuarios	944

De los monitores de Apache JMeter y Winbox se extrae: el ancho de banda servido, número de usuarios conectados (concurrentes) y pico de usuarios.

Donde el número de usuarios (concurrentes) es de 943, visualizado en el monitor de Iccast2 y un pico de usuarios de 944 (ver Figura 23), donde el ancho de banda servido en las 943 conexiones es de 92.3 Mbps, además se determina que un usuario consume aproximadamente un ancho de banda de 97.87 kbps.

Global Server Stats	
admin	soporte.uti@unl.edu.ec
client_connections	7297
clients	944
connections	7566
file_connections	296
host	radio.unl.edu.ec
listener_connections	6617
listeners	943
location	Unidad de Telecomunicaciones e Información
server_id	Icecast 2.4.0
server_start	Thu, 01 Dec 2016 16:07:29 -0500
server_start_iso8601	2016-12-01T16:07:29-0500

Figura 23. Monitor de Icecast2 con 943 usuarios.

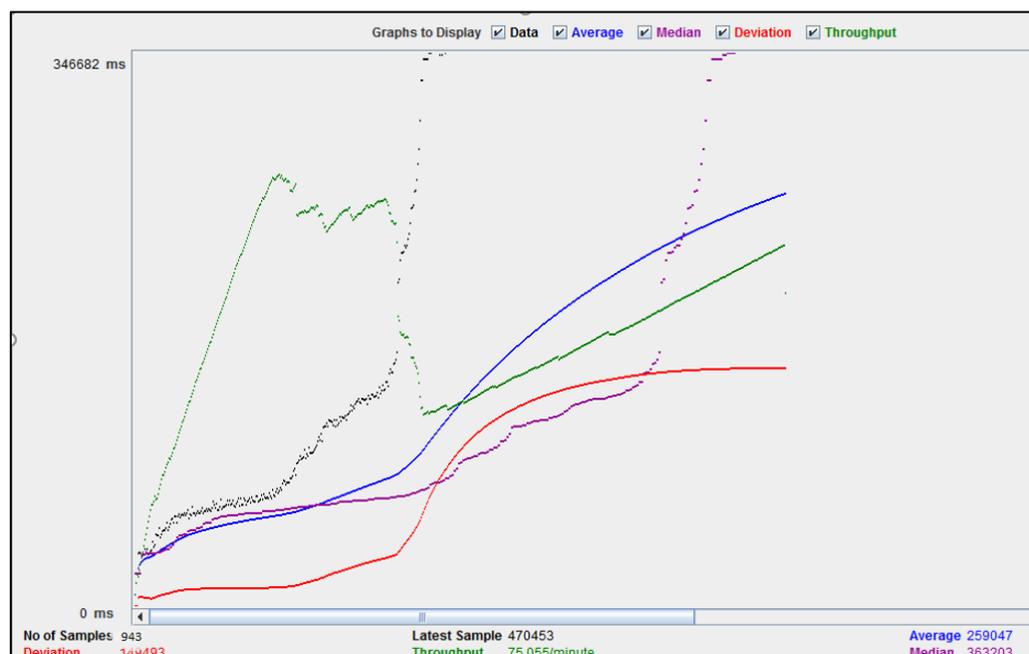


Figura 24. Rendimiento del sistema con relación al tiempo en Apache JMeter.

En grafica en la figura 25 se puede observar el rendimiento del sistema con relación al tiempo, en el eje de las X se representa el número de conexiones y el eje de las Y, se representa el tiempo de las conexiones en ms, donde la línea de color verde refleja el desempeño del sistema y la línea de color rojo es la probabilidad de error durante una conexión. Es así que en 943 conexiones el tiempo de respuesta del sistema es de 75,055/minutos, es decir cada petición se la realiza en un tiempo estimado de 4775,50/ms o lo que es igual 4,77 segundos.

6.3.1.6.3.2. Prueba No. 2

Esta prueba se la realiza el día 13 de diciembre de 2016, donde se vuelve a repetir con los valores de la Prueba No. 1, para de esta manera obtener alguna variación en el rendimiento del sistema de streaming.

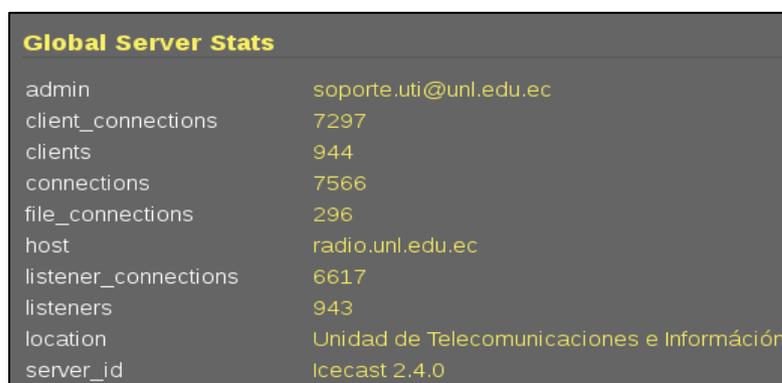
6.3.1.6.3.2.1. Resultados

Del registro global de las herramientas Apache JMeter, Winbox y el monitor del servidor Icecast2, se captura los siguientes valores:

TABLA XXXI. RESULTADOS PRUEBA NO. 3: VALORES PROMEDIOS DEL REGISTRO

Parámetro	Valor Promedio
Ancho de banda	92.3Mbps enviados, 3.9Mbps recibidos
Nº de paquetes enviados por el servidor	7468
Número de usuarios concurrentes	943
Pico de usuarios	944

Donde se constata que en 943 conexiones concurrentes se consume un ancho de banda de 92.3 Mbps al igual que en la Prueba No. 1, este ancho de banda se lo ve reflejado en el monitor de interfaces de Winbox (Ver figura 26).



Global Server Stats	
admin	soporte.uti@unl.edu.ec
client_connections	7297
clients	944
connections	7566
file_connections	296
host	radio.unl.edu.ec
listener_connections	6617
listeners	943
location	Unidad de Telecomunicaciones e Información
server_id	Icecast 2.4.0

Figura 25. 943 Conexiones concurrentes

Interface List										
Interface: Ethernet EoIP Tunnel IP Tunnel GRE Tunnel VLAN VRRP Bonding LTE										
Name	Type	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)				
R	br_vlanDOCE...	Bridge	1584	3.2 kbps	0 bps	3	0			
R	br_vlanIMPRES...	Bridge	1584	3.3 kbps	0 bps	3	0			
R	ether1-Link	Ethernet	1588	3.8 Mbps	92.4 Mbps	7 312	7 912			
R	ether2-master1...	Ethernet	1588	0 bps	0 bps	0	0			
RS	ether3-slave-lo...	Ethernet	1588	51.7 kbps	14.2 kbps	40	17			
R	ether4-slave-lo...	Ethernet	1588	92.3 Mbps	3.9 Mbps	7 900	7 425			
S	ether5-slave-lo...	Ethernet	1588	0 bps	0 bps	0	0			

Figura 26. Ancho servido en 943 conexiones concurrentes

Esta prueba confirma la funcionalidad y la capacidad del sistema streaming, el cual soporta un límite de 943 conexiones concurrentes, donde se consume un ancho de banda de 92.3Mbps.

6.3.1.6.4. Pruebas de rendimiento del servidor Radio Universitaria

El objetivo de esta prueba es determinar la influencia de las conexiones unicast en el rendimiento del hardware especificado en la tabla XXXII. Con este fin se procede abrir la consola de comandos en el equipo donde se realizaron las pruebas de capacidad del sistema, pero en este caso se evalúa el uso del CPU (Unidad Central de Procesamiento) y Memoria RAM (Memoria de Acceso Aleatorio) a continuación se presenta el esquema para evaluar el rendimiento del servidor.

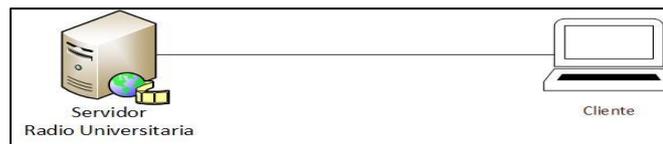


Figura 27. Esquema de equipos para la prueba

TABLA XXXII. EQUIPOS PARA PRUEBA DE RENDIMIENTO

Equipo	Nombre	Software instalado	Procesador	Memoria RAM	Disco Duro
Servidor	radio.unl.edu.ec IP 172.16.32.60 IP Pública 192.168.49.14	Debian 8.2 Iccast2	1 GHz	1 GB	10 GB
Cliente	Computador Personal IP 192.168.5.254	Debian 8 Jessie Apache JMeter Winbox	2.00 GHz	16 GB	1024 GB

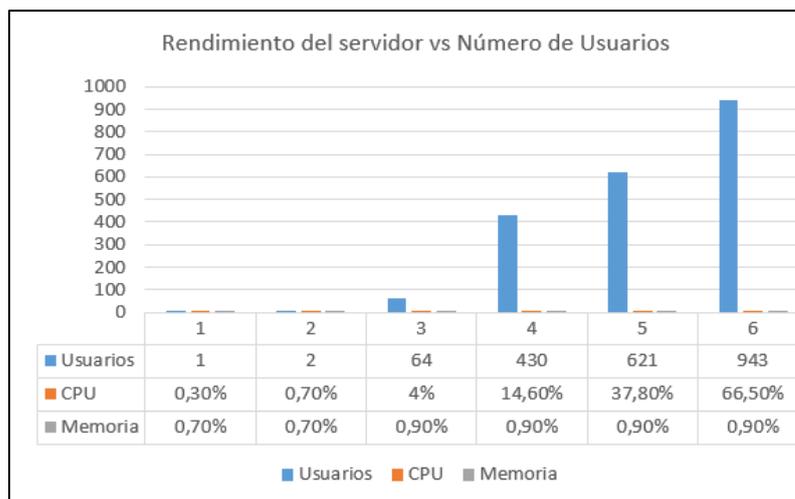


Figura 28. Rendimiento del Servidor VS Número de conexiones

La figura 28 representa el rendimiento del servidor vs el número de conexiones, donde en el eje de las Y se presenta el número de usuarios y en el eje de las X se evalúa el consumo de memoria y CPU. Estos porcentajes se los obtiene ejecutando el comando **top** en la consola de comandos de Linux (Ver figura 29).

```

Archivo Editar Ver Buscar Terminal Ayuda
top - 10:26:09 up 11 days, 18:18, 1 user, load average: 0,63, 0,16, 0,09
Tasks: 71 total, 2 running, 69 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,9 us, 19,4 sy, 0,0 ni, 1,0 id, 0,0 wa, 0,0 hi, 77,4 si, 0,3 st
KiB Mem: 1024448 total, 297732 used, 726716 free, 42700 buffers
KiB Swap: 975868 total, 0 used, 975868 free. 102364 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 671 icecast2 20   0 421088  8720  5112 S  66,5  0,9  44:46.34 icecast2

```

Figura 29. Porcentaje de memoria 0.90% y CPU 66.50% en 943 conexiones concurrentes

Resultado:

Se determina que, en una conexión, el porcentaje de uso de CPU y memoria es 0.30% y 0.70% respectivamente, y en 943 conexiones el porcentaje de CPU es de 66.50% y 0.90% de memoria RAM.

6.4. OBJETIVO 4: DESARROLLAR E IMPLEMENTAR UNA APLICACIÓN MÓVIL MULTIPLATAFORMA PARA ACCEDER A LA RADIO UNIVERSITARIA.

TABLA XXXIII. COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE APLICACIONES MÓVILES

Metodología	Extreme Programing	Scrum para móviles	Mobile-D
Fases	Planificación, diseño, desarrollo y pruebas.	Esta se define por sprints de dos semanas aproximadamente y de un producto entregable.	Exploración, inicialización, producción, estabilización y pruebas del sistema
Comunicación con el cliente	Basada en la retroalimentación continua entre el equipo de desarrollo y el cliente.	El product owner maneja la comunicación con el cliente.	La comunicación con el cliente es menor que en Scrum.
Programación	La programación es en parejas, en jornadas largas; revisando el código mutuamente.	El tiempo de programación se determina de acuerdo a la puntuación dada a cada tarea.	La programación es la parte más importante, y que mayor tiempo conlleva junto con las pruebas.
Documentación	Las historias de usuario son las más importantes y las tarjetas CRC.	Lo principal lo constituyen las historias de usuario.	Lo principal es el modelo de dominio, storycards y storyboards.

Pruebas	Se realizan pruebas unitarias, pruebas de aceptación.	Se realizan pruebas unitarias, pruebas de integración y pruebas de aceptación.	Se realizan pruebas unitarias, pruebas de interfaz, pruebas de datos integrados, y pruebas funcionales.
----------------	---	--	---

Al finalizar la comparativa de las metodologías para el desarrollo de aplicaciones móviles se determina que las fases de la metodología XP y Mobile-D tienen similitud a las fases de análisis, diseño, codificación y pruebas; en cambio Scrum se caracteriza por ser iterativo, se podría decir que realiza las fases de análisis, diseño, codificación y pruebas por cada iteración que esta tenga.

En lo referente a la documentación la metodología XP y Scrum tienen un nivel bajo en comparación con la metodología Mobile-D. sin embargo Mobile-D no exige una gran cantidad de documentación, lo necesario para guiarse en el desarrollo.

6.4.1. Primera Fase: Análisis

Con esta fase se busca realizar el análisis pertinente de la problemática y obtener los requerimientos de la aplicación a desarrollar.

6.4.1.1. Exploración

En esta etapa se definen los requerimientos y alcance de la investigación, bases fundamentales para un adecuado desarrollo y exitoso producto.

6.4.1.1.1. Establecimiento de los Grupos de Interés o Stakeholders

Los grupos de personas interesadas en la realización de la presente investigación son los siguientes:

- **Desarrollador:** Es la persona encargada de análisis, desarrollo y pruebas de la aplicación; en este caso el tesista.
- **Usuarios:** Son todas las personas que poseen dispositivo móvil con Sistema Operativo Android o iOS.

1. **Personal de la UTI y Radio Universitaria:** Son las personas, quienes ofrecerán la información que alimente el desarrollo de la aplicación.

6.4.1.1.2. Requerimientos Iniciales

Se pretende realizar una aplicación móvil para los sistemas operativos Android e iOS que permita escuchar la programación en tiempo real que está emitiendo la Radio Universitaria 98.5 FM, registro de usuarios, así como también visualizar la parrilla de programación e información de contactos de la radio, visualizar noticias de la institución universitaria, escuchar audios de entrevistas grabadas y tener acceso directo a la información de las redes sociales de la Radio Universitaria. En base a ello se ha recolectado la información que permita conocer las necesidades de la Radio Universitaria.

6.4.1.1.3. Requerimientos Funcionales

Los requerimientos que la aplicación permitirá se encuentra en la siguiente tabla.

TABLA XXXIV. REQUERIMIENTOS FUNCIONALES

Código	Descripción	Categoría
RF001	Al usuario registrarse y loguearse para acceder al contenido de la aplicación.	EVIDENTE
RF002	Al usuario acceder al streaming de la radio.	EVIDENTE
RF003	Al usuario visualizar noticias de la institución universitaria.	EVIDENTE
RF004	Al usuario escuchar audios de entrevistas o programas grabados.	EVIDENTE
RF005	Al usuario visualizar la parrilla de programación e información de contactos de la radio universitaria.	EVIDENTE
RF006	Al usuario visualizar la información de las redes sociales asociadas a la radio.	EVIDENTE

6.4.1.1.4. Requerimientos no Funcionales

Los requerimientos no funcionales para la aplicación a realizar son descritos en la siguiente tabla.

TABLA XXXV. REQUERIMIENTOS NO FUNCIONALES

Código	Descripción
RNF01	La aplicación será desarrollada para la plataforma Android e iOS.
RNF02	La aplicación utilizara una interfaz amigable.
RNF03	La aplicación permitirá el acceso a los usuarios luego de loguearse.
RNF04	La aplicación ofrecerá respuesta al usuario en tiempo real de noticias, redes sociales y el streaming trasmitido por la radio.
RNF05	La aplicación debe tener conexión a internet para escuchar el streaming, visualización de contenido dinámico y de redes sociales de la aplicación.

6.4.1.1.5. Análisis de los Requerimientos

En base a los Requerimientos establecidos se ha podido determinar los procesos a realizar en la siguiente tabla.

TABLA XXXVI. MÓDULOS Y PROCESOS DE LA APLICACIÓN

Módulo	Código	Proceso	Requerimientos
Módulo de acceso a la aplicación	P001	Registrarse y loguearse para acceder al contenido de la aplicación.	RF001
Módulo de acceso al streaming	P002	Acceder al streaming en directo que emite la radio universitaria.	RF002
Módulo de acceso a audios y	P003	Escuchar audios de entrevistas o	RF004

programas grabados.		programas grabados.	
Módulo de Visualización de contenidos dentro de la aplicación.	P004	Visualizar noticias de la institución universitaria.	RF003
	P005	Visualizar la parrilla de programación e información de contactos de la radio universitaria.	RF005
	P006	Visualizar la información de las redes sociales asociadas a la radio universitaria.	RF006

6.4.1.2. Segunda Fase: Diseño

A través del diseño se desea obtener los diagramas que representa el funcionamiento de la aplicación, los procesos que va a ejecutar con el fin de cumplir los requerimientos definidos en la primera fase; además la descripción de cada uno de ellos y las configuraciones necesarias para desarrollar una aplicación móvil.

6.4.1.2.1. Inicialización

En esta etapa se realizan las actividades relacionadas a la configuración del ambiente de desarrollo y al diseño de la aplicación.

6.4.1.2.1.1. Configuración del ambiente de desarrollo

En esta actividad se debe realizar la configuración de los elementos físicos y técnicos, en este caso se realizó el entorno de desarrollo para aplicaciones híbridas.

6.4.1.2.1.2. Configuración para aplicaciones móviles híbridas

- **Framework:** Ionic, con ayuda de su complemento Ionic Creator
- **Configuraciones:** inclusión de Cordova y la librería Leanbackplayer para realizar conexiones de audio streaming.

6.4.1.2.1.3. Planificación Inicial

Como resultado de esta actividad se busca establecer los pre-requisitos de cada proceso

6.4.1.2.1.4. Análisis de Procesos y Pre-requisitos

Para la realización de los procesos es necesario cumplir con ciertos pre-requisitos con el fin de implementar la funcionalidad del proceso.

P001: Registrarse y loguearse para acceder al contenido de la aplicación.

- Abrir la aplicación loguearse o registrarse sino se encuentra registrado.
- Establecer conexión con la base de datos de la aplicación para poder validar las credenciales del usuario.
- Una vez validada las credenciales permitir el acceso al contenido de la aplicación.

P002: Acceder al streaming en directo que emite la Radio Universitaria

- Establecer conexión con el streaming emitido por el servidor Iccast2.
- Escuchar streaming de la radio.

P003: Escuchar audios de entrevistas o programas grabados.

- Establecer conexión con el servidor que almacena los audios de entrevistas o programas grabados.
- Reproducir el o los audios de entrevista o programa.

P004: Visualizar noticias de la institución universitaria.

- Establecer conexión con el servidor que almacena las noticias.
- Visualizar la lista de noticias y contenido de cada noticia.

P005: Visualizar la parrilla de programación e información de contactos de la radio universitaria.

- Establecer conexión con el servidor que almacena la parrilla de programación por día y la información de contactos de la radio.
- Consultar programación por día y visualizar su contenido.
- Visualizar la información de contactos de la radio.

P004: Visualizar la información de las redes sociales asociadas a la radio universitaria

- Establecer conexión con las redes sociales de la Radio Universitaria (Facebook, Twitter) y con la página oficial de la radio.
- Visualizar la información de las redes sociales y pagina institucional.

6.4.1.2.1.5. Planificación de fases

Las fases y las iteraciones que se darán en el desarrollo de la aplicación móvil se reflejan en la tabla XXVIII, donde se explica las iteraciones que se realizan para el módulo de acceso a streaming y módulo de visualización de información de la radio en la etapa de producción y estabilización.

TABLA XXXVII. PLANIFICACIÓN DE FASES

Fase	Iteración	Descripción
Exploración	Iteración 0	Establecimiento de stakeholders, requerimientos iniciales, definición del alcance, establecimiento del proyecto.
Inicialización	Iteración 0	Configuración del ambiente de desarrollo, Planificación inicial, planificación de interfaces.
Producción	Iteración Módulo de acceso a la aplicación	Implementación del Módulo de Acceso a la aplicación, Refinamiento y actualización de storycards, Refinamiento de interfaces.
	Iteración Módulo de acceso a streaming	Implementación del Módulo de Acceso a Streaming, Refinamiento y actualización de storycards, Refinamiento de interfaces.
	Iteración Módulo de acceso a audios y programas grabados.	Implementación del Módulo de Acceso a audios y programas grabados, Refinamiento y actualización de storycards, Refinamiento de interfaces.
	Iteración Módulo de Visualización de contenidos dentro de la aplicación.	Implementación del Módulo de visualización de contenidos dentro de la aplicación, Refinamiento de storycards, Refinamiento de Interfaces.
	Iteración Módulo de acceso a la aplicación	Implementación del Módulo de Acceso a la aplicación, Refinamiento y actualización de storycards, Refinamiento de interfaces.
	Iteración Módulo de acceso a streaming	Implementación del Módulo de Acceso a Streaming, Refinamiento y

Estabilización		actualización de storycards, Refinamiento de interfaces.
	Iteración Módulo de acceso a audios y programas grabados.	Implementación del Módulo de Acceso a audios y programas grabados, Refinamiento y actualización de storycards, Refinamiento de interfaces.
	Iteración Módulo de Visualización de contenidos dentro de la aplicación.	Implementación del Módulo de visualización de contenidos dentro de la aplicación, Refinamiento de storycards, Refinamiento de Interfaces.
Pruebas del Sistema	Iteración Pruebas del Sistema	Evaluación de Pruebas y Análisis de Resultados.

6.4.1.2.2. Diseño del Sistema

El diseño general del sistema está compuesto por la conexión al servidor streaming, interfaces y el usuario móvil.

El servidor Icecast2 se encuentra ubicado en las instalaciones de la UTI el cual proporciona el flujo de audio streaming hacia los usuarios. (Ver figura 30)

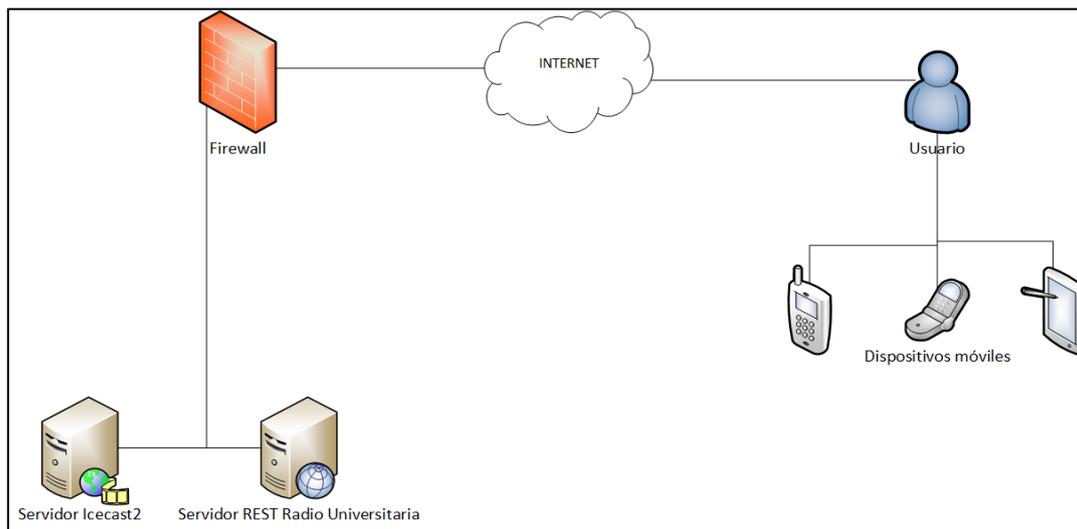


Figura 30. Diseño del Sistema

6.4.1.2.3. Modelo de Dominio

A continuación, se detallan las clases que componen el Sistema de Radio Universitaria Online que sirve de base para el desarrollo de la presente aplicación.

Al modelo de dominio se lo realiza basándose en las tres interfaces, las cuales se presentarán al usuario interfaz Online, Quienes Somos y Programación que

corresponden al streaming de la radio, información de la Radio Universitaria y la Programación Semanal, las cuales se representan en el Diagrama 1.

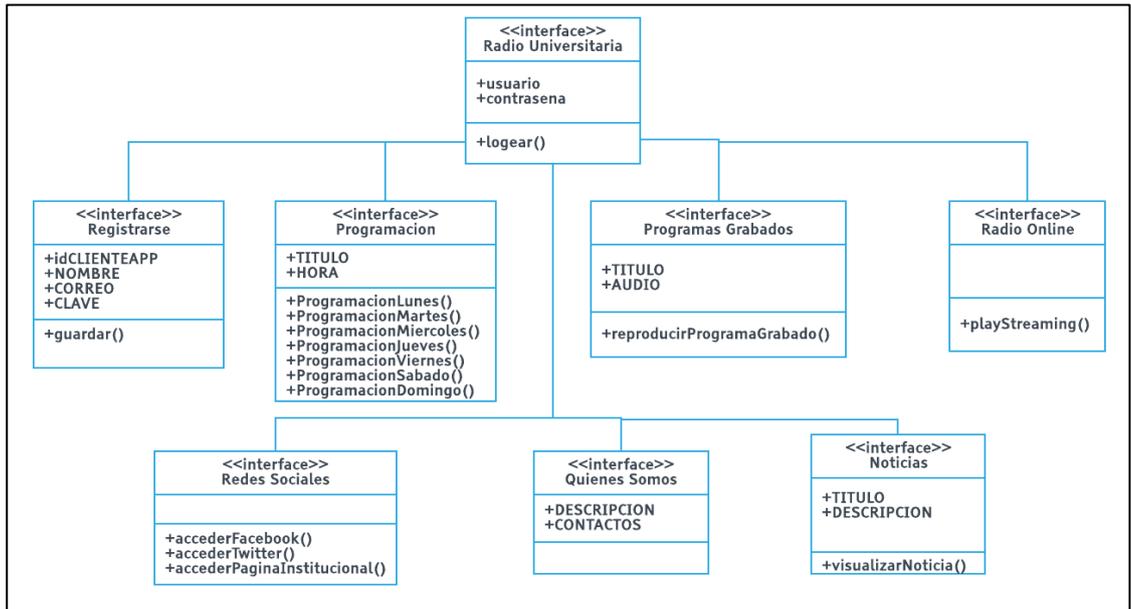


Diagrama 1 Modelo de Dominio

6.4.1.2.4. Descripción de la Interfaz de Usuario

A continuación, se describe el esquema de navegabilidad del sistema o storyboard cuyo objetivo es describir la navegabilidad y conexiones entre las principales pantallas de la aplicación desarrollada.

La primera pantalla a visualizar es el splash con el logo de la aplicación, que dará paso a la pantalla principal donde el usuario accederá directamente al streaming de la radio y al menú de navegación lateral con las opciones de Pagina Institucional, Facebook, Twitter y Salir. (Ver diagrama 2)

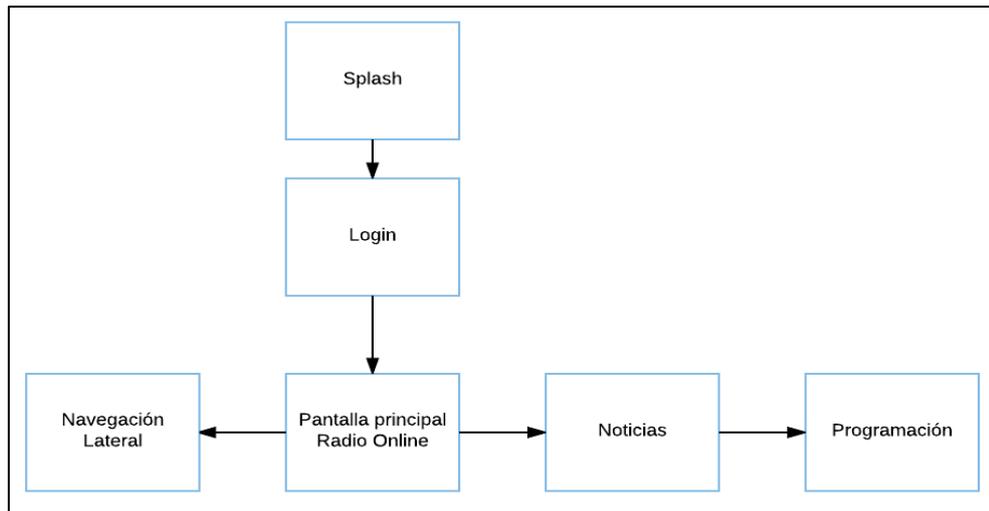


Diagrama 2 Storyboard de la Aplicación

- **Pantalla Principal Radio Online**

La pantalla principal está compuesta por el logo de la aplicación y reproductor de streaming (Ver Figura 31).

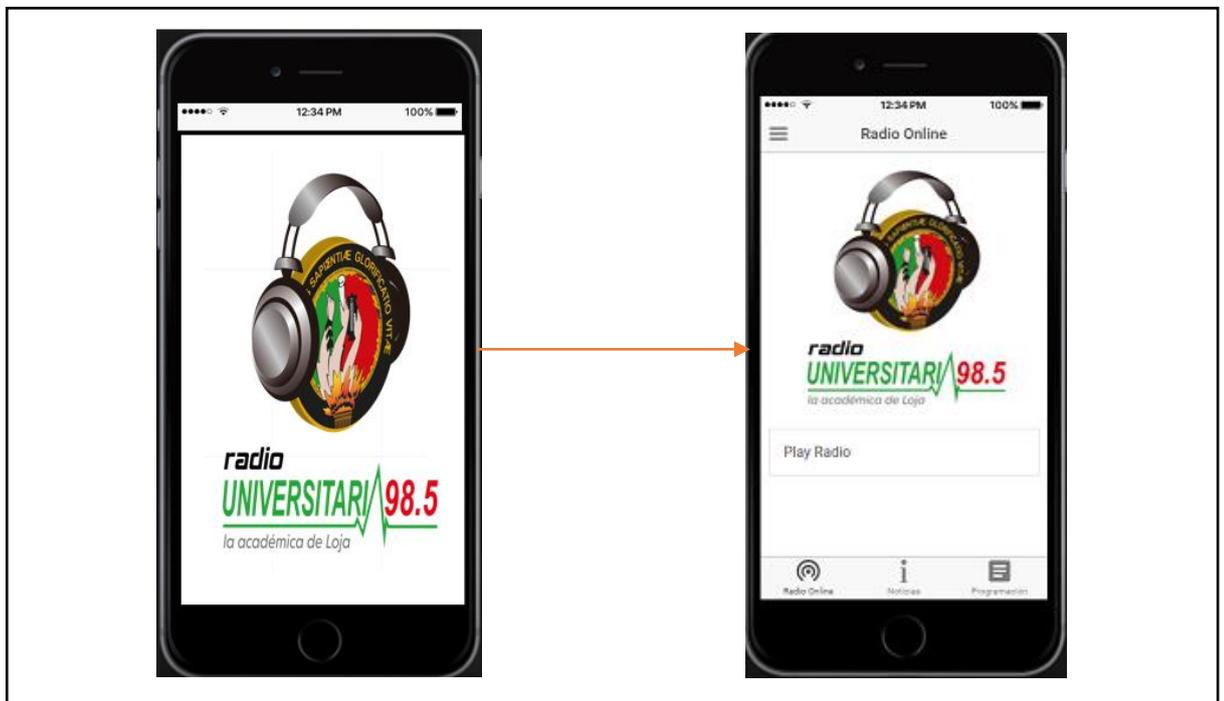


Figura 31. Prototipo: Pantalla Principal Radio Online

A continuación, se describe el StoryCard de Pantalla Principal Radio Online en la tabla XXIX. El storycard muestra una descripción del funcionamiento de la Pantalla Principal y de las fechas de su desarrollo.

TABLA XXXVIII. STORYCARD DE LA PANTALLA PRINCIPAL

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
01	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Cuando el usuario ingresa a la aplicación se muestra el logo de la aplicación y el reproductor de streaming de la radio.							
Excepciones							
Si el dispositivo no cuenta con conexión a internet el reproductor se deshabilitará poniéndose en un color transparente. En caso de que exista un problema con el streaming igualmente el reproductor se deshabilitara imposibilitando escuchar el streaming de audio.							

- **Login / Register**

Esta pantalla está formada por un formulario que permitirá ingresar usuario y contraseña o loguearse con la red social Facebook para ingresar al contenido privado de la aplicación (Ver figura 32).

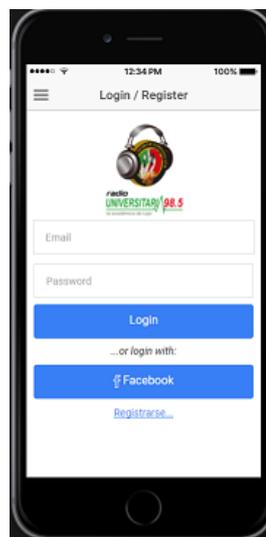


Figura 32. Pantalla: Login / Register

A continuación, se describe el StoryCard de la Pantalla: Registrar en la tabla XXX.

TABLA XXXIX. STORYCARD DE LOGIN/REGISTER

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
02	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta aplicación tendrá un formulario donde se ingresa el usuario y contraseña para ingresar al contenido privado de la aplicación o redirigirse al formulario de nuevo registro de usuario. De la misma manera el usuario tendrá la posibilidad de loguearse mediante la red social Facebook.							
Excepción							
Si el usuario no está registrado o no tiene cuenta en la red social Facebook no podrá ingresar al contenido privado de la aplicación.							

- **Registrar**

La página de registro muestra un formulario con los campos de nombre, correo y contraseña, donde se realizará varias validaciones en los campos para poder completar el registro.

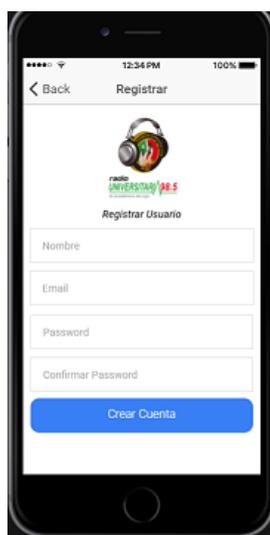


Figura 33. Pantalla: Navegación lateral

A continuación, se describe el StoryCard del Prototipado de Pantalla: Registrar en la tabla XXXI.

TABLA XL. STORYCARD DE REGISTRAR

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
03	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta pantalla contiene el formulario para el registro de la aplicación con los campos de nombre, correo y contraseña, donde se realizará varias validaciones en los campos para poder completar el registro.							
Excepción							
Si el usuario no ingresa bien los datos la aplicación no permitirá el registro de nuevo usuario.							

- **Navegación lateral**

El Menú de Navegación Lateral, muestra una lista con las opciones: Inicio, Contactos, Pagina Institucional, Facebook, Twitter, programas grabados y Salir.

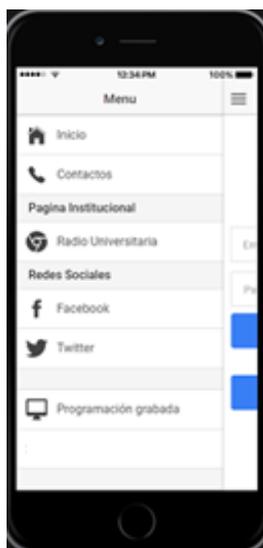


Figura 34. Pantalla: Navegación Lateral

A continuación, se describe el StoryCard del Prototipado de Pantalla: Navegación Lateral en la tabla XXXII.

TABLA XLI. STORYCARD DE NAVEGACIÓN LATERAL

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
04	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta pantalla, muestra una lista con las opciones: Inicio, Contactos, Pagina Institucional, Facebook, Twitter, programas grabados y Salir, donde el usuario podrá acceder a las redes sociales o programas grabados de la aplicación.							

- **Noticias**

Noticias muestra un banner y la lista de las noticias de la institución en orden cronológico de la más nueva a la mas antigua.



Figura 35 Prototipo pantalla: Noticias

A continuación, se describe el StoryCard del Prototipado de Pantalla: Noticias en la siguiente tabla.

TABLA XLIII. STORYCARD DE NAVEGACIÓN LATERAL

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
05	Nuevo Fijo	Fácil Moderado	Fácil Moderado			Baja Media	

	Mejora	Duro	Duro			Alta	
Descripción							
En esta pantalla se muestra un banner y a continuación se muestra la lista de noticias donde el usuario visualiza las noticias en orden cronológico de la más nueva a la más antigua. El usuario visualiza estas noticias con una imagen, título y una descripción breve de la misma.							
Excepción							
Si el dispositivo no cuenta con una conexión a internet no se mostrara la información tanto del banner como de las noticias.							

- **Contenido de Noticias**

Contenido de noticias cuenta con el título de la noticia, una foto y la descripción de la noticia.



Figura 36 Prototipo de pantalla: Contenido de noticias

A continuación, se describe el StoryCard del Prototipado de Pantalla: Contenido de Noticias en la tabla.

TABLA XLIII. STORYCARD DE NAVEGACIÓN LATERAL

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
06	Nuevo	Fácil	Fácil			Baja	
	Fijo	Moderado	Moderado			Media	

	Mejora	Duro	Duro			Alta	
Descripción							
Esta pantalla que será visualizada por el usuario cuenta con el título, foto y descripción de la noticia seleccionada.							
Excepción							
Si el dispositivo no cuenta con una conexión a internet no se mostrará la información tanto del banner como de las noticias.							

- **Programación**

Esta pantalla presenta una lista las cuales reflejan los días de la semana conteniendo en su interior la programación de la radio de cada uno de sus días.

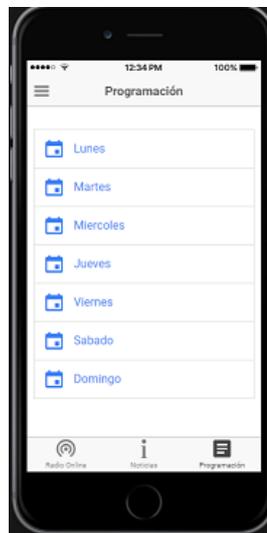


Figura 37 Prototipo de pantalla: Programación

A continuación, se describe el StoryCard del Prototipado de Pantalla: Programación en la tabla.

TABLA XLVI. STORYCARD DE PROGRAMACIÓN

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
07	Nuevo	Fácil	Fácil			Baja	
	Fijo	Moderado	Moderado			Media	
	Mejora	Duro	Duro			Alta	

Descripción
Esta pantalla que será visualizada por el usuario cuenta con una lista con los días de la semana donde el usuario podrá escoger cualquier día para posterior visualizar su contenido.

- **Programación diaria**

Esta pantalla presenta una lista con la lista de programación diaria de acuerdo al día seleccionado anteriormente, esta lista contiene la hora y título de programa.

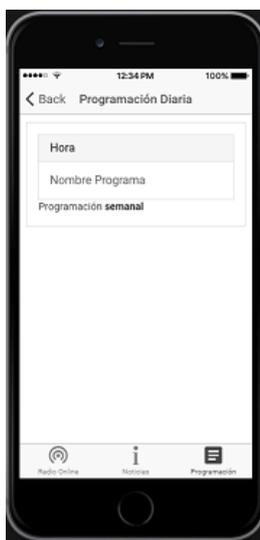


Figura 38 Prototipo de pantalla: Programación diaria

A continuación, se describe el StoryCard del Prototipado de Pantalla: Programación diaria en la tabla.

TABLA XLVI. STORYCARD DE PROGRAMACIÓN DIARIA

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
08	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta pantalla que será visualizada por el usuario cuenta con la lista de la programación diaria de acuerdo al día de la semana la misma cuenta con la hora y el título del programa.							
Excepción							
Si el dispositivo no cuenta con una conexión a internet no se mostrará la información de la lista de programación.							

- **Programas grabados**

Esta pantalla contiene una lista de programa grabados ordenados cronológicamente desde la más nueva a la más antigua. Esta lista contiene el título y audio del programa, entrevista o noticiero grabado en formato mp3.

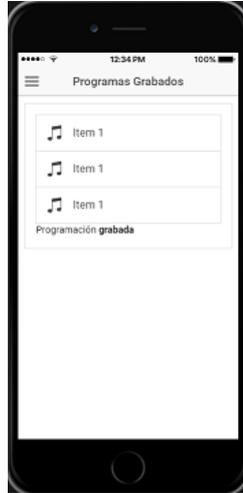


Figura 39 Prototipo de pantalla: Programas grabados

A continuación, se describe el StoryCard del Prototipado de Pantalla: Programas grabados en la tabla.

TABLA XLVII. STORYCARD DE PROGRAMAS GRABADOS

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
09	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta pantalla que será visualizada por el usuario, cuenta con la lista de los programas grabados ordenados cronológicamente de la más nueva a la más antigua, esta lista contiene el título y audio del programa, entrevista o noticiero grabado.							
Excepción							
Si el usuario no se loguea ya sea con una cuenta anteriormente registrada en la aplicación o loguearse con la red social Facebook no podrá acceder al contenido de esta pantalla debido a que es privado.							
Si el dispositivo no cuenta con una conexión a internet no se mostrará la información de la lista de programación.							

- **Quienes somos**

Esta pantalla muestra información de la radio quienes son, contactos tanto correos electrónicos como número de tele-fax.



Figura 40 Prototipo de pantalla: Quienes somos

A continuación, se describe el StoryCard del Prototipado de Pantalla: Quienes somos en la tabla.

TABLA XLVIII. STORYCARD DE QUIENES SOMOS

Numero /Id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
10	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta pantalla que será visualizada por el usuario, cuenta con información de la radio como la descripción de la radio y contactos tanto correos electrónicos como tele-fax							

6.4.1.3. Tercera Fase: Codificación

En esta fase se realiza el desarrollo de las funcionalidades planteadas en el diseño mediante un lenguaje de programación, para aplicaciones móviles híbridas con el uso de HTML5, JavaScript, CSS; además se plantean los estándares de codificación y la estructuración de la aplicación.

6.4.1.3.1. Estándares de Codificación

Paquetes: Los paquetes están escritos en minúsculas.

Clases: Las clases están escritas con la primera letra en mayúscula, si se compone de dos o más palabras la primera letra de cada una de ellas está en mayúscula.

Variables: Las variables tienen un nombre relacionado a su valor, están escritas con la primera letra en minúscula, si se compone de dos o más palabras la primera letra de ellas está en mayúscula, a excepción de la primera palabra.

Métodos: Los métodos llevan un nombre relacionado al proceso que ejecutan, están escritos con la primera letra en minúscula, si se compone de dos o más palabras la primera letra de ellos está en mayúscula, a excepción de la primera palabra.

Layouts: Los layouts tienen todo su nombre en minúsculas, si poseen más de dos palabras tendrán un sub-bajo para separar cada una de ellas.

6.4.1.3.2. Codificación

La codificación se realiza basándose en cada uno de los storycards realizados en la fase de inicialización.

6.4.1.3.2.1. Pantalla Principal Online

Para la presentación de la pantalla principal se realiza un conjunto de procesos como la construcción del splash, y configuración de la librería JavaScript leanbackPlayer para la conexión con el flujo de audio que emite el servidor streaming.

- **Construcción del Splash**

Para la creación del Splash se ejecuta comandos en la consola donde se encuentra ubicado nuestro proyecto estos comandos son los siguientes:

\$ ionic resources --splash: el cual al ejecutar lo agregara en los diversos tamaños que necesitaran los distintos dispositivos donde se instale la aplicación.

- **Configuración de la librería JavaScript leanbackPlayer**

Para poder agregarla primeramente debemos descargarla, descomprimirla y pegar el archivo en la carpeta contenedora de todos los JavaScript, así mismo con su respectivo CSS en la carpeta de CSS de nuestro proyecto.

La configuración comienza con el llamado de la librería en la sección de la página index.html de nuestro proyecto tanto el JavaScript y CSS.

```
<script type="js/leanbackPlayer.js"></script><!--se agrega el script-->
```

Figura 41. Llamar la librería leanbackPlayer

Para la conexión del flujo de audio nos dirigimos hacia la template online para agregar el código con el cual realiza la conexión al flujo de audio emitido por el servidor.

```
<div class="leanback-player-audio" width="350px" align="center">
  <audio preload="metadata" controls>
    <!-- HTML5 <audio> sources -->
    <source src="http://radio.unl.edu.ec:8080/stream" type='audio/mpeg'/><!--Conexion con el
    flujo de audio
    por parte del servidor-->
  </audio>
</div>
```

Figura 42. Conexión con el flujo de audio emitido por el Servidor

6.4.1.3.2.2. Login/Register

Para el Login/Register se lo realizo un formulario donde podrá loguearse el usuario sí anteriormente se registró o con la red social Facebook.

```
<ion-view title="Login / Register" id="page6">
  <ion-content padding="true" scroll="false" class="has-header">
    <div>
      
    </div>
    <div id="loginRegister-container1">
      <form id="loginRegister-form1" class="list">
        <label class="item item-input" id="loginRegister-input1" name="email">
          <input type="email" placeholder="Email" ng-model="data.email">
        </label>
        <div class="spacer" style="width: 300px; height: 13px;"></div>
      </form>
    </div>
    <form id="loginRegister-form2" class="list">
      <label class="item item-input" id="loginRegister-input2" name="password">
        <input type="password" placeholder="Password" ng-model="data.clave">
      </label>
      <a ng-click="enviar(data)" id="loginRegister-button1" class="button button-positive button-block">Login</a>
      <h5 id="loginRegister-heading1" style="color: #000000;font-weight:400;font-style:italic;text-align:center;">...or login with:</h5>
      <button class="button button-block button-positive" ng-click="loginFacebook()">
        <b>Facebook</b>
      </button>
      <center>
        <a ui-sref="registrar">Registrarse...</a>
      </center>
    </form>
  </ion-content>
</ion-view>
```

Figura 43 Login/Register

6.4.1.3.2.3. Menú de Navegación Lateral

Para el menú de navegación lateral se lo realizo en el index.html con las etiquetas <ion-side-menus>, el cual dentro de este tiene una lista de ítems contenidos en la etiqueta <ion-list>.

```

<ion-content ng-controller="menuCtrl" padding="false" class="side-menu-left has-header">
<ion-list id="menu-list2" data-componentid="list2">
  <ion-item class="item-icon-left" id="menu-list-item22" ui-sref="tabsController.radioOnline()" menu-close="" data-componentid="list-item22">
    <i class="icon ion-ios-home"></i>Inicio</ion-item>
  <ion-item class="item-icon-left" id="menu-list-item11" ui-sref="quienesSomos()" menu-close="" data-componentid="list-item11">
    <i class="icon ion-ios-telephone"></i>Contactos</ion-item>
  <ion-item class="item-divider" id="menu-list-item-divider175" data-componentid="list-item-divider175"> Pagina Institucional</ion-item>
  <ion-item class="item-icon-left" id="menu-list-item5" data-componentid="list-item5" onclick="window.open('http://unl.edu.ec/radio/inicio')">
    <i class="icon ion-social-chrome"></i>Radio Universitaria</ion-item>
  <ion-item class="item-divider" id="menu-list-item-divider1" data-componentid="list-item-divider1"> Redes Sociales</ion-item>
  <ion-item class="item-icon-left" id="menu-list-item2" data-componentid="list-item2" onclick="window.open('https://www.facebook.com/universidadnacional')">
    <i class="icon ion-social-facebook"></i>Facebook</ion-item>
  <ion-item class="item-icon-left" id="menu-list-item3" data-componentid="list-item3" onclick="window.open('https://twitter.com/radiounl')">
    <i class="icon ion-social-twitter"></i>Twitter</ion-item>
  <ion-item class="item-divider" id="menu-list-item-divider3" data-componentid="list-item-divider3"> </ion-item>
  <ion-item class="item-icon-left" id="menu-list-item10" ui-sref="programasGrabados()" menu-close="" data-componentid="list-item10">
    <i class="icon ion-android-desktop"></i>Programación grabada</ion-item>
  <ion-item class="item-divider" id="menu-list-item-divider172" data-componentid="list-item-divider172"> </ion-item>
  <ion-item class="item-icon-left" id="menu-list-item4" data-componentid="list-item4" onclick="cerrar()">
    <i class="icon ion-close-circled"></i>Salir</ion-item>
</ion-list>
</ion-content>

```

Figura 44. Código para la generación del menú de navegación lateral

6.4.1.3.2.4. Registrar

Para realizar el registro dentro de la aplicación se lo debe hacer dentro de un formulario que posteriormente pasara al respectivo controlador y enviado al servidor de la aplicación para su respectivo almacenamiento y dar por concluido el registro.

```

<ion-view title="Registrar" id="page7">
  <ion-content padding="true" scroll="false" class="has-header">
    <div>
      
    </div>
    <h5 id="registrar-heading3" style="color: #000000;font-style:italic;text-align:center;">Registrar Usuario</h5>
    <div id="registrar-container2">
      <form id="registrar-form3" class="list" name="formregistrar" novalidate>
        <label class="item item-input" id="registrar-input5" name="nombre">
          <input type="text" placeholder="Nombre" ng-model="user.nombre" name="nombre" required>
          <span style="color: red; font-weight: bold; font-size: small; margin-left: 5px;">Nombre Requerido</span>
        </label>
        <div class="spacer" style="width: 300px; height: 13px;"></div>
        <label class="item item-input" id="registrar-input3" name="email">
          <input type="email" placeholder="Email" ng-model="user.email" name="email" required>
          <span style="color: red; font-weight: bold; font-size: small; margin-left: 5px;">Email requerido</span>
        </label>
        <div class="spacer" style="width: 300px; height: 13px;"></div>
        <label class="item item-input" id="registrar-input4" name="password">
          <input type="password" placeholder="Password" ng-model="user.clave" name="password" ng-minlength="5" required>
          <span style="color: red; font-weight: bold; font-size: small; margin-left: 5px;">Mínimo 5 caracteres</span>
        </label>
        <div class="spacer" style="width: 300px; height: 13px;"></div>
        <label class="item item-input" id="registrar-input6" name="confpassword">
          <input type="password" placeholder="Confirmar Password" ng-model="user.repclave" name="password2" required>
          <span style="color: red; font-weight: bold; font-size: small; margin-left: 5px;">Repita contraseña</span>
        </label>
        <a ng-disabled="formregistrar.$invalid" ng-click="enviar(user)" id="registrar-button3" style="font-weight: bold; border-radius: 12px 12px 12px 12px; padding: 5px 10px; width: 100%;">Registrar</a>
      </form>
    </div>
  </ion-content>
</ion-view>

```

Figura 45 Formulario registrar nuevo usuario

6.4.1.3.2.5. Quienes Somos

Para la generación del contenido de esta pantalla se lo realiza en la template que lleva el mismo nombre "Quienes Somos", en el cual se agrega la imagen y el texto a presentar en la aplicación.

```

<ion-view title="Quienes Somos" id="page3">
  <ion-content padding="true" scroll="false" class="has-header" ng-repeat="info in informacion">
    <div>
      
    </div>
    <div id="quienesSomos-markdown1" style="text-align: justify;" class="show-list-numbers-and-dots">
      <p style="color: #000000;">{{info.DESCRIPCION}}</p>
    </div>
    <div class="spacer" style="width: 300px; height: 81px;"></div>
    <div id="quienesSomos-container5">
      <h5 id="quienesSomos-heading4" style="color: #000000; font-weight: 400; font-style: italic;">Contactos:</h5>
      <div id="quienesSomos-markdown14" class="show-list-numbers-and-dots">
        <p style="color: #000000;">
          <strong>Email: </strong>&nbsp;&nbsp;{{info.CORREO}}</p>
        <p style="color: #000000;">
          <strong>Teléfono - Fax: </strong>&nbsp;&nbsp;{{info.TELEFONO}}</p>
        </div>
      </div>
    </ion-content>
  </ion-view>

```

Figura 46. Código de la template "Quienes Somos"

6.4.1.3.2.6. Noticias

Para la generación de la pantalla que contendrá la lista de noticias, se realizó una lista la misma que se irá incrementando conforme se agreguen nuevas noticias desde el servidor.

```

<ion-view title="Noticias" id="page8">
  <ion-content padding="true" class="has-header">
    <div id="noticias-container3">
      <ion-slides disable-side-menu-drag="" options="{ 'loop': true }" slider="slider2" delegate-handle="slider2" id="noticias-slider2" style="width: 100%; height: 100px;">
        <div>
          
        </div>
      </ion-slides>
    </div>
    <div class="spacer" style="width: 300px; height: 10px;"></div>
    <ion-refresher pulling-text="Pull to refresh..." on-refresh="doRefresh()">
      </ion-refresher>
    <ion-list id="noticias-list103" ng-repeat="noticia in noticias">
      <ion-item class="item-thumbnail-left" id="noticias-list-item9" ui-sref="tabsController.tituloDeNoticia({id:'{{noticia.IDNOTICIA}}'})">
        
        <h2>{{noticia.TITULO}} </h2>
        <span>{{noticia.DESCRIPCION}}</span><br>
        <span>{{noticia.FECHA}}</span>
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>

```

Figura 47 Lista de noticias

6.4.1.3.2.7. Contenido Noticias

Esta pantalla está codificada para obtener el id de la noticia seleccionada y visualizar toda la información de la noticia y presentado los datos luego de haber obtenido la información en formato .json

```

<ion-view id="page9">
  <ion-nav-title>{{noticia.TITULO}}</ion-nav-title>
  <ion-content padding="true" class="has-header">
    <div>
      
    </div>
    <div id="tituloDeNoticia-container6">
      <p style="text-align: justify;">{{noticia.DESCRIPCION}}</p>
      <p><strong>FECHA: </strong>{{noticia.FECHA}}</p>
    </div>
  </ion-content>
</ion-view>

```

Figura 48 Contenido de noticias de acuerdo al id recibido

6.4.1.3.2.8. Programación

Para la generación de la template “programación” donde contiene la información de la parrilla de programación, se la realiza con una lista, dentro de la cual esta dirige al contenido del día elegido

```
<ion-view title="Programación" id="page4" style="background-color:#9AB1B8;">
<ion-content padding="true" class="has-header">
<ion-list id="programaciN-list20">
<ion-item class="item-icon-left positive" id="programaciN-list-item135" ui-sref="tabsController.programaciNLunes">
<i class="icon ion-android-calendar"></i>Lunes</ion-item>
<ion-item class="item-icon-left positive" id="programaciN-list-item136" ui-sref="tabsController.programaciNMartes">
<i class="icon ion-android-calendar"></i>Martes</ion-item>
<ion-item class="item-icon-left positive" id="programaciN-list-item137" ui-sref="tabsController.programaciNMiércoles">
<i class="icon ion-android-calendar"></i>Miércoles </ion-item>
<ion-item class="item-icon-left positive" id="programaciN-list-item138" ui-sref="tabsController.programaciNJueves">
<i class="icon ion-android-calendar"></i>Jueves</ion-item>
<ion-item class="item-icon-left positive" id="programaciN-list-item139" ui-sref="tabsController.programaciNViernes">
<i class="icon ion-android-calendar"></i>Viernes</ion-item>
<ion-item class="item-icon-left positive" id="programaciN-list-item140" ui-sref="tabsController.programaciNSabado">
<i class="icon ion-android-calendar"></i>Sabado</ion-item>
<ion-item class="item-icon-left positive" id="programaciN-list-item141" ui-sref="tabsController.programaciNDomingo">
<i class="icon ion-android-calendar"></i>Domingo </ion-item>
</ion-list>
</ion-content>
</ion-view>
```

Figura 49. Lista de los días de la semana para la programación

6.4.1.3.2.9. Programación diaria

Para la programación diaria se realizó una lista la cual recibirá como id el día seleccionado y de acuerdo al día seleccionado mostrará la programación del día.

```
<ion-view title="Programación Diaria" id="page5">
<ion-content padding="true" class="has-header" >
<ion-list id="programaciNDiaria-list107">
<div class="item item-body" id="programaciNDiaria-list-item-container1">
<ion-list id="programaciNDiaria-list99" ng-repeat="programa in programas">
<ion-item class="item-divider" id="programaciNDiaria-list-item-divider159">{{programa.HORA}}</ion-item>
<ion-item id="programaciNDiaria-list-item8">{{programa.TITULO}}</ion-item>
</ion-list>
<div id="programaciNDiaria-markdown3" class="show-list-numbers-and-dots">
<p style="margin-top:0px;color:□ #000000;">Programación
<strong>semanal</strong>
</p>
</div>
</div>
</ion-list>
</ion-content>
</ion-view>
```

Figura 50 Lista de programas de acuerdo al día seleccionado

6.4.1.3.2.10. Programas Grabados

Para la generación de la template programas grabados se la realizó como una lista la cual listara los programas que se agregaron o se irán agregando visualizados en forma cronológica de la más nueva a la más antigua.

```

<ion-content padding="true" class="has-header">
  <div id="programasGrabados-container4">
    <ion-list id="programasGrabados-list1">
      <div class="item item-body" id="programasGrabados-list-item-container8">
        <ion-list id="programasGrabados-list106"ng-repeat="prograb in programasgrabados">
          <ion-item class="item-divider" ><i class="icon ion-ios-musical-notes">&nbsp;{{prograb.TITULO}}</i></ion-item>
          <ion-item id="programaciNDiaria-list-item8">
            <audio src="{{prograb.AUDIO}}!" audioplayer controls</audio>
          </ion-item>
        </ion-list>
      </div>
      <div id="programasGrabados-markdown12" class="show-list-numbers-and-dots">
        <p style="margin-top:0px;color:□#000000;">Programación
          <strong>grabada</strong>
        </p>
      </div>
    </ion-list>
  </div>
</ion-content>
</ion-view>

```

Figura 51 Listar programas grabados

6.4.1.4. Cuarta Fase: Pruebas

Luego de terminar con la codificación, se realiza las pruebas que comprueben que la aplicación es de calidad y que cumple con todos los requerimientos.

6.4.1.4.1. Pruebas del Sistema y arreglos

Las pruebas que la metodología Mobile-D establece para una aplicación son la comprobación de requerimientos, las pruebas unitarias, además de una lista de dispositivos en donde se ha comprobado el adecuado funcionamiento.

6.4.1.4.2. Pruebas Unitarias

Las pruebas unitarias se realizan sobre un conjunto de clases y métodos que tienen funciones específicas como la conexión del flujo de audio emitida por el servidor streaming.

Caso de prueba 1: Obtención del audio streaming de la radio:

Esta prueba se realiza para comprobar la correcta conexión con el flujo de audio emitida por el servidor streaming y el correcto funcionamiento en nuestra aplicación.

TABLA XLVIII. PU: OBTENCIÓN DEL AUDIO STREAMING DE LA RADIO

Código	Nombre
PU001	Obtención del audio streaming de la radio
Objetivo	Obtener el audio en directo que emite la Radio Universitaria
Pasos	<ol style="list-style-type: none"> 1. Iniciar la actividad 2. Realizar clic en el botón Play y así invocar al método de la clase leanback-player-audio el cual a su vez contiene la dirección del streaming en directo.

	3. Comprobar que el audio que se está emitiendo no tiene interrupciones y se escucha todo su contenido.
Resultados Esperados	El audio en vivo que emite la Radio Universitaria debe estar activo. Que el reproductor no esté deshabilitado.
Resultados Obtenidos	El audio en vivo que emite la Radio Universitaria. El reproductor no está deshabilitado.

Caso de prueba 2: Obtención de Información de quienes son la Radio Universitaria:

Esta prueba se realiza para comprobar que la aplicación permite visualizar la información correcta sin sobreponer palabras o la imagen.

TABLA XLIX. PU: OBTENCIÓN DE INFORMACIÓN QUIENES SOMOS LA RADIO UNIVERSITARIA

Código	Nombre
PU002	Obtención de Información de quienes son la Radio Universitaria
Objetivo	Visualizar la información correcta de quienes son la Radio Universitaria.
Pasos	<ol style="list-style-type: none"> 1. Iniciar la actividad 2. Realizar clic en el submenú opción Quienes Somos y así re direccionará a la página correcta.
Resultados Esperados	La redirección a la página correcta y visualización de la correcta información.
Resultados Obtenidos	La redirección a la página correcta y se visualiza la información correcta.

Caso de Prueba 3: Acceder y visualizar a lista de programas grabados

Esta prueba se realiza con el fin de validar si el usuario está registrado pueda loguearse con sus credenciales o proceder a registrarse como nuevo usuario o simplemente realice el logueo con la red social Facebook con el fin de acceder y visualizar la lista de programas grabados.

TABLA L. PU: LOGUEO DE USUARIO EN LA PANTALLA
LOGIN/REGISTER

Código	Nombre
PU003	Acceder y visualizar a lista de programas grabados
Objetivo	Permitir el logueo, registro del usuario y acceso a la lista de programas grabados.
Pasos	<ol style="list-style-type: none"> 1. Iniciar la actividad 2. Ingresar a menú de navegación lateral. 3. Programas grabados 4. Ingresar las credenciales para el logueo. <ol style="list-style-type: none"> 4.5. Si el usuario cuenta con las credenciales permitir acceso 4.6. Sino permitir al usuario registrarse 4.7. Alternativa permitir logueo con Facebook 5. Visualizar lista de programas grabados
Resultados Esperados	Acceso valido al contenido privado de la aplicación, permitir registro de nuevo usuario y logueo alternativo con la red social Facebook, visualizar la lista de programas grabados.
Resultados Obtenidos	El acceso con credenciales antes registradas fue correcto, si el usuario no estaba registrado se permitió correctamente para ser ingresado, se le proporciono el logueo alternativo con la red social Facebook, se pudo visualizar la lista de programas grabados.

Caso de Prueba 3: Obtención de información de la página institucional y redes sociales:

Esta prueba se realiza para comprobar la conexión con la página institucional y las redes sociales de la radio, las cuales deben permitir la visualización del contenido de las mismas.

TABLA LI. PU: OBTENCIÓN DE LA INFORMACIÓN DE PÁGINA
INSTITUCIONAL Y REDES SOCIALES

Código	Nombre
PU003	Obtención de información de la página institucional y redes sociales
Objetivo	Conectar y visualizar la información de la página institucional y redes sociales de la radio universitaria.
Pasos	<ol style="list-style-type: none"> 1. Iniciar la actividad

	<ol style="list-style-type: none"> 2. Ingresar a menú de navegación lateral. 3. Realizar clic en una de las opciones página institucional, Facebook o Twitter según la información que desee visualizar. 4. Comprobar que este re direccionando hacia la información correcta.
Resultados Esperados	La redirección a la página correcta y visualización de la correcta información.
Resultados Obtenidos	La redirección a la página correcta y se visualiza la información correcta.

6.4.1.4.3. Pruebas de Interfaz de Usuario

Este tipo de pruebas tienen por objetivo comprobar que el diseño y conexión de los prototipos de pantallas es igual al obtenido en la aplicación desarrollada.

Verificación de Pantallas

A continuación, se describe una tabla en la que se puede verificar que la aplicación posee las mismas pantallas que el prototipo de pantallas.

TABLA LII. VERIFICACIÓN DE PANTALLAS

Prototipos de Pantalla	Pantallas de Aplicación	Cumplimiento
Splash	Splash	✓
Pantalla Principal Online	Pantalla Principal Online	✓
Navegación Lateral	Navegación Lateral	✓
Quienes Somos	Quienes Somos	✓
Programación	Programación	✓
Programación diaria	Programación diaria	✓
Login/Register	Login/Register	✓
Registrar	Registrar	✓
Programas Grabados	Programas grabados	✓

6.4.1.4.4. Pruebas Funcionales

En base a los requerimientos planteados en la etapa de Análisis (Ver Tabla XXVI) se procede a verificar el cumplimiento de cada uno de ellos, de acuerdo al proceso al que pertenecen.

La comprobación de estos requerimientos se realiza de acuerdo a los procesos especificados en la siguiente tabla.

TABLA LIII. VERIFICACIÓN DE REQUERIMIENTOS

Código	Descripción	Categoría	Cumplimiento
RF001	Al usuario registrarse y loguearse para acceder al contenido privado de la aplicación.	EVIDENTE	✓
RF002	Al usuario acceder al streaming de la radio.	EVIDENTE	✓
RF003	Al usuario visualizar noticias de la institución universitaria.	EVIDENTE	✓
RF004	Al usuario escuchar audios de entrevistas o programas grabados.	EVIDENTE	✓
RF005	Al usuario visualizar la parrilla de programación e información de contactos de la radio universitaria.	EVIDENTE	✓
RF006	Al usuario visualizar la información de las redes sociales asociadas a la radio.	EVIDENTE	✓

PF01: Acceder al streaming de la radio

TABLA LIV. PF: ACCEDER AL STREAMING DE LA RADIO

Código	Nombre
PF001	Prueba funcional del proceso
Objetivo	Comprobar que el usuario puede acceder al streaming de audio
Pasos	1. El usuario ingresa a la aplicación

	<ol style="list-style-type: none"> 2. El usuario realiza clic en el botón Play 3. El usuario escucha el streaming de audio es decir la Radio.
Resultados Esperados	El usuario podrá escuchar la radio por medio de la aplicación tan solo presionando play en la aplicación.
Resultados Obtenidos	El usuario puede escuchar la radio mediante la aplicación.

El resultado de la ejecución se visualiza en la siguiente figura, donde el usuario accede mediante la aplicación al streaming de la radio.



Figura 52. PF: Acceder al streaming de la radio

PF02: Visualización de información de quienes son la Radio Universitaria

TABLA LV. PF: VISUALIZACIÓN DE INFORMACIÓN DE QUIENES SON LA RADIO UNIVERSITARIA

Código	Nombre
PF002	Prueba funcional del proceso
Objetivo	Comprobar que el usuario puede visualizar la información correcta de la radio
Pasos	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación 2. El usuario realiza clic en Quienes Somos en el submenú ya sea en la parte superior si es iOS y en la parte inferior si es Android. 3. El usuario visualiza la información.
Resultados Esperados	El usuario podrá visualizar la información de quienes son la Radio Universitaria.

Resultados Obtenidos	El usuario puede visualizar la información correcta de quienes son la Radio Universitaria.
----------------------	--

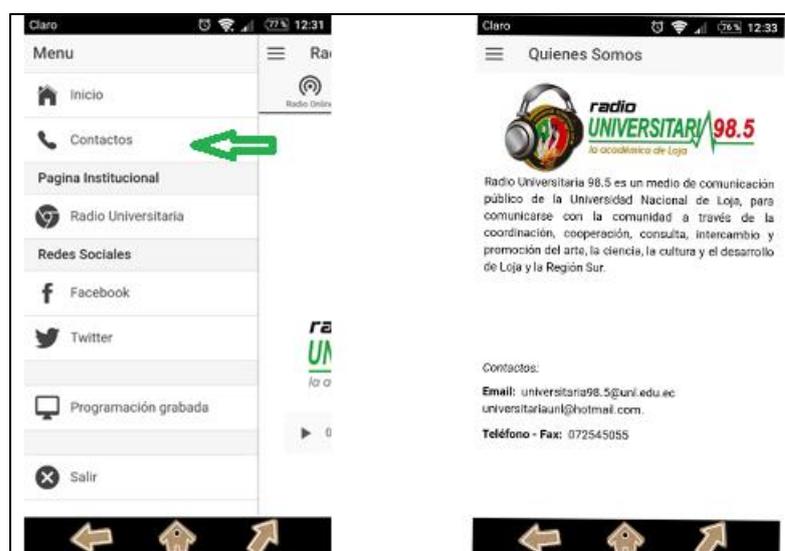


Figura 53. P.F: Visualización de información de “Quienes Somos”

PF03: Visualización de parrilla de programación

TABLA LVI. PF: VISUALIZACIÓN DE PARRILLA DE PROGRAMACIÓN

Código	Nombre
PF003	Prueba funcional del proceso
Objetivo	Comprobar que el usuario puede visualizar la información de la parrilla de programación de la radio
Pasos	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación 2. El usuario realiza clic en Programación en el submenú ya sea en la parte superior si es iOS y en la parte inferior si es Android. 3. El usuario accede a la pantalla con la lista de los días de la semana 4. El usuario selecciona el día del cual desea visualizar su programación 5. El usuario visualiza la información de la parrilla de programación de acuerdo al día seleccionado.
Resultados Esperados	El usuario podrá visualizar la información de la parrilla de programación de acuerdo al día seleccionado.
Resultados Obtenidos	El usuario puede visualizar la información correcta de la parrilla de programación de acuerdo al día seleccionado.

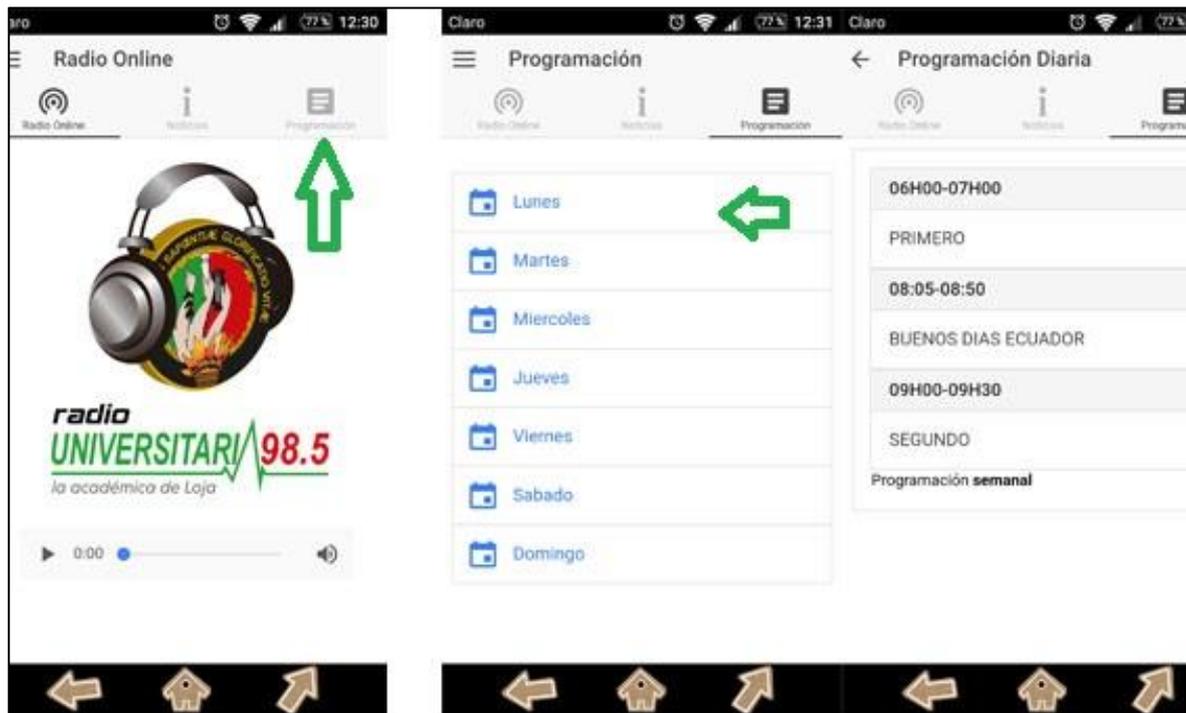


Figura 54. PF: Visualización de parrilla de programación

PF04: Visualización de información de redes sociales

TABLA LVII. PF: VISUALIZACIÓN DE INFORMACIÓN DE REDES SOCIALES

Código	Nombre
PF004	Prueba funcional del proceso
Objetivo	Comprobar que el usuario puede visualizar la información de las redes sociales como Facebook, Twitter y pagina institucional de la radio
Pasos	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación 2. El usuario realiza clic en el menú o deslizar el dedo de izquierda a derecha 3. El usuario selecciona un ítem del menú de acuerdo que red social desea acceder 4. El usuario visualiza la información de la red social seleccionada o página institucional de la radio.
Resultados Esperados	El usuario podrá visualizar la información de las redes sociales de Facebook, Twitter o página institucional de la radio.

Resultados Obtenidos	El usuario puede visualizar la información correcta de las redes sociales de Facebook, Twitter o página institucional de la radio.
----------------------	--

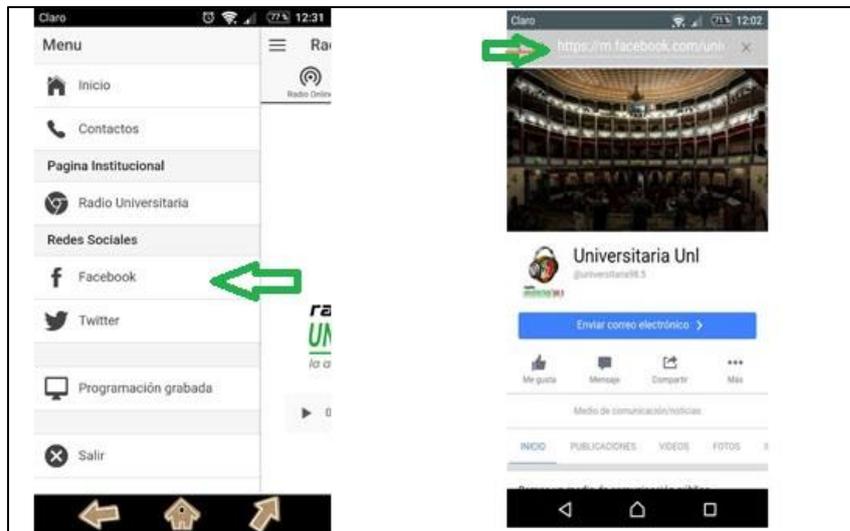


Figura 55. PF: Visualización de información de redes sociales

La presente figura es el resultado de acceder a la red social de Facebook perteneciente a la Radio Universitaria, donde es visualizada dentro del propio navegador de la aplicación desarrollada.

PF05: Validación de logueo con credenciales o red social Facebook para acceder a contenido privado programas grabados.

TABLA LVIII. PF: VISUALIZACIÓN DE INFORMACIÓN DE REDES SOCIALES

Código	Nombre
PF005	Prueba funcional del proceso
Objetivo	Comprobar que el usuario puede loguearse con sus credenciales o con la red social Facebook para acceder a contenido privado programas grabados.
Pasos	<ol style="list-style-type: none"> 1. Iniciar la actividad 2. Ingresar a menú de navegación lateral. 3. Programas grabados 4. Ingresar las credenciales para el logueo. <ol style="list-style-type: none"> a. Si el usuario cuenta con las credenciales permitir acceso b. Sino permitir logueo con Facebook 5. Visualizar lista de programas grabados

Resultados Esperados	El usuario puede ingresar sus credenciales para loguearse o como alternativa loguearse con la red social Facebook y acceder a contenido privado programas grabados.
Resultados Obtenidos	El usuario puede ingresar sus credenciales para loguearse o como alternativa loguearse con la red social Facebook y acceder a contenido privado programas grabados.

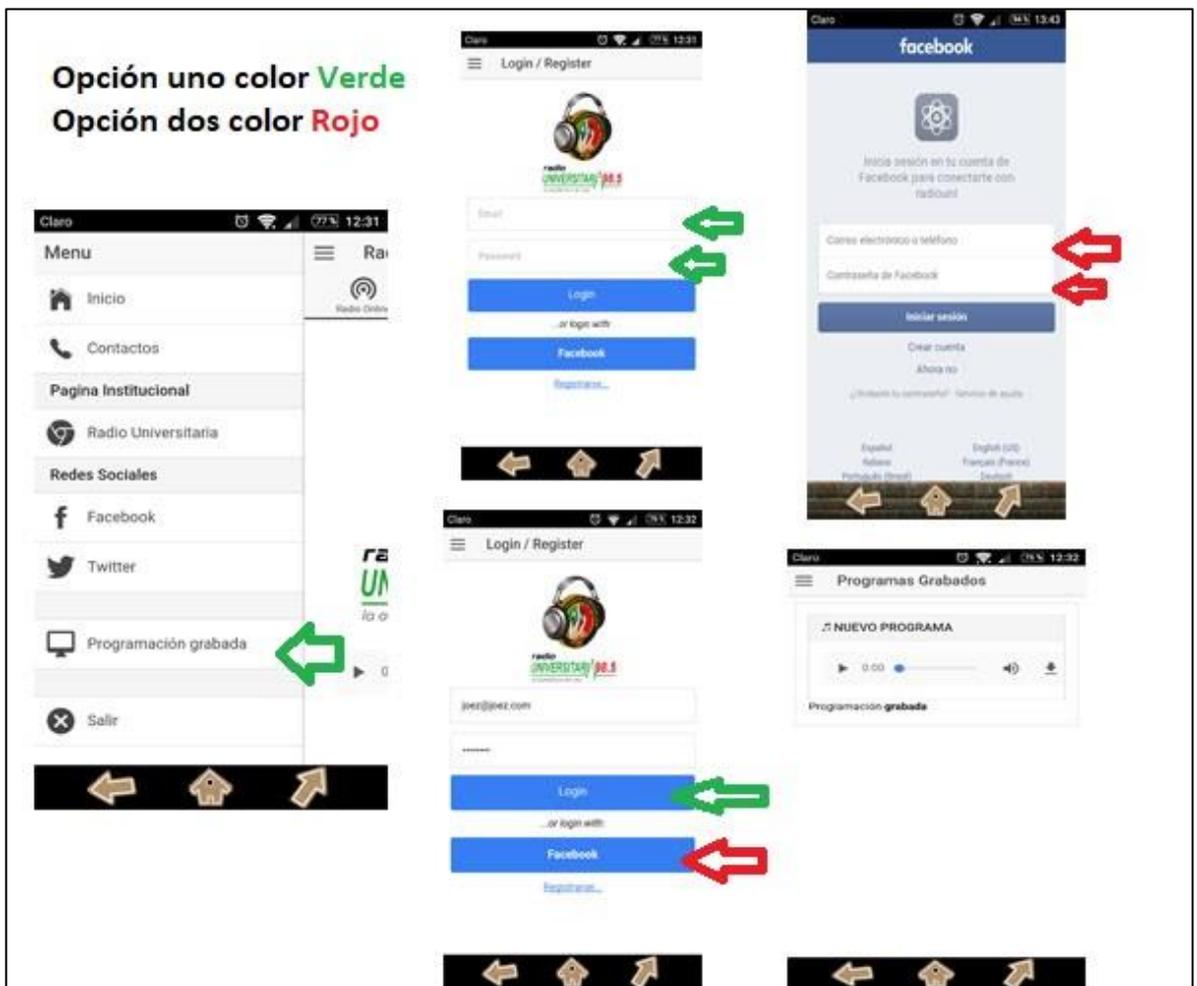


Figura 56 PF: Acceder a contenido privado

La presente figura es el resultado de acceder al contenido privado programas grabados para lo cual el usuario tiene dos opciones ingresar con credenciales usuario y contraseña o acceder con la red social de Facebook.

PF05: Registrar nuevo usuario

TABLA LIX. PF: REGISTRAR NUEVO USUARIO

Código	Nombre
PF005	Prueba funcional del proceso
Objetivo	Comprobar que el usuario puede registrarse dentro de la aplicación.
Pasos	<ol style="list-style-type: none"> 1. Iniciar la actividad 2. Ingresar a menú de navegación lateral. 3. Programas grabados 4. Registrarse. <ol style="list-style-type: none"> a. Ingresar nombre, correo y contraseña b. Si contraseña coinciden prosigue <ol style="list-style-type: none"> i. Verifica que correo no esté registrado anteriormente. ii. Si correo no está registrado prosigue <ol style="list-style-type: none"> 1. Registra usuario iii. Sino mostrar mensaje de error c. Sino mostrar mensaje de alerta e ingresar nuevamente contraseña
Resultados Esperados	El usuario puede registrarse correctamente.
Resultados Obtenidos	El usuario puedo registrarse correctamente.

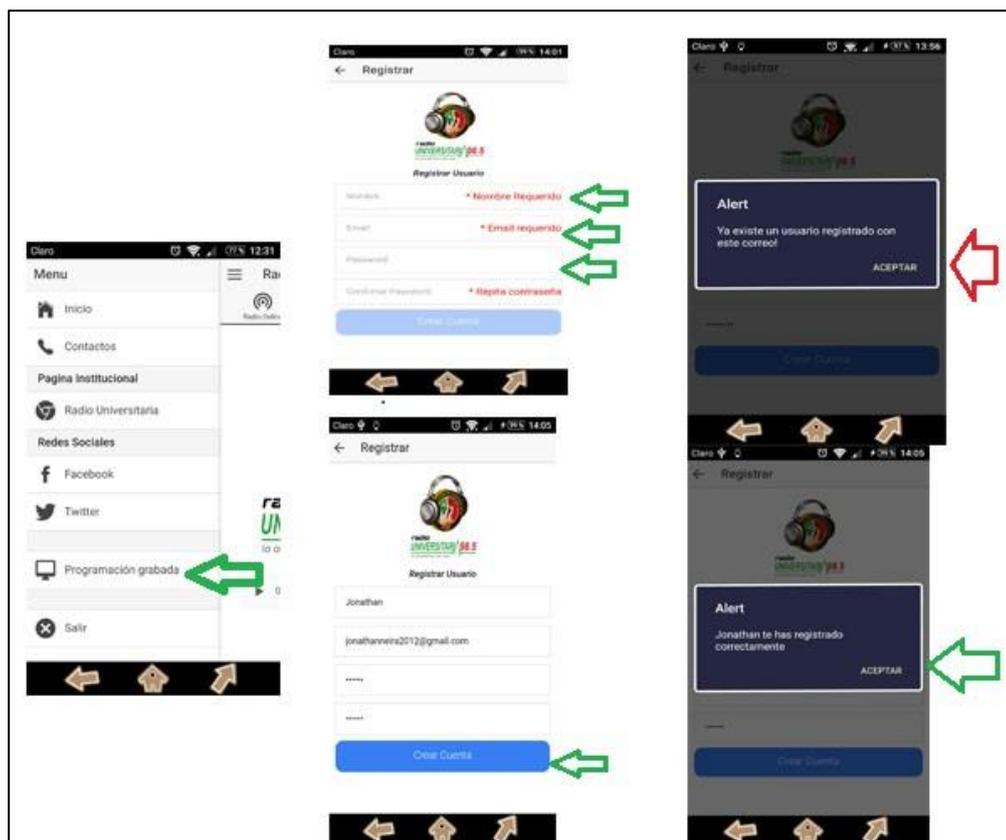


Figura 57: PF: Registrar nuevo usuario

La presente figura es el resultado de registrar un nuevo usuario y validar tanto el formulario como también si el correo ya fue registrado anteriormente.

Las pruebas también se realizaron en el sistema operativo iOS, para el cual se emula un dispositivo iPhone, para ver más detalles de la emulación revisar el Anexo 8.

Una vez realizadas las pruebas se determina las características mínimas para instalar y ejecutar aplicación en los dispositivos móviles.

TABLA LX. CARACTERÍSTICAS MÍNIMAS PARA INSTALAR Y EJECUTAR LA APLICACIÓN

Nombre de Sistema Operativo	Versión de software	RAM	Almacenamiento
Android	5.1 o superior	512 MB	10 MB
iOS	9 o superior	512 MB	10 MB

Posterior a las pruebas se procede a subir la aplicación a la tienda de Google Play, para que usuarios con sistema operativo Android puedan descargarla directamente a sus Smartphone.

Para más detalles del proceso de subir la aplicación a Google Play revisar la sección de Anexos.

7. DISCUSIÓN

7.1. Desarrollo de la propuesta alternativa

El desarrollo del presente trabajo de titulación consiste en 4 fases, de las cuales cada una corresponde a un objetivo específico planteado para dar cumplimiento al objetivo general del proyecto. Primera fase, se realiza el análisis de la situación actual del servicio de la Radio Universitaria, segunda fase, se realiza la determinación de los requerimientos hardware y software para la realización del streaming, tercera fase, se realiza la implementación del streaming para la Radio Universitaria, finalmente en la cuarta fase, se desarrolló una aplicación móvil para acceder a la radio la misma que contaba con sus fases de análisis, diseño, implementación y pruebas para las sistemas operativos (Android e iOS).

A continuación, se describe las actividades realizadas en cada uno de los objetivos, así como los resultados obtenidos.

7.2. Analizar la situación actual del servicio de la Radio Universitaria en los medios de difusión digital

En el presente objetivo se realizó el análisis de la situación actual del servicio de radio en la Universidad Nacional de Loja, donde determinamos con que tecnología cuenta para realizar su programación y salida al aire, además porque medio se puede acceder al contenido de la radio. Para ello se elaboró un inventario de los equipos tecnológicos, se elaboró los diagramas de red, diagrama de conexión con sus componentes tecnológicos y recolección de datos técnicos de las estaciones de radiodifusión con los datos referentes a la Radio Universitaria 98.5 FM. Para finalizar con esta fase, se determinó las necesidades que se requiere dentro de la radio, para posteriormente dar solución a las mismas.

7.3. Determinar los requerimientos de hardware y software para el streaming

En este objetivo se centró principalmente en determinar que recursos tanto hardware y software se necesitan para implementar el streaming de la Radio Universitaria, para lo cual se realizó una revisión sistemática aplicando la metodología de Barbara Kitchenham. Esto nos ayudó a determinar el servidor, códec y cliente que se debe contar y ajustar al marco de la radio universitaria con el uso de tecnología Open Source.

Luego de haber realizado el análisis de tecnología software, se seleccionó el servidor de streaming Icecast2, como cliente al software Butt Broadcast, códec MP3 y el ancho de banda mínimo de 256 kbps, con una compresión del audio de mínima de 64 kbps, de la misma manera se estableció el hardware necesario para poder instalar todos los componentes software (Ver sección de Resultados, fase 2, subsección 2.2).

7.4. Implementar streaming en la Radio Universitaria

En este objetivo se realizó todo el proceso de implementación del streaming, para ello se cumplió con varias actividades, en primer lugar, la instalación y configuración del servidor Icecast2, seguidamente la instalación y configuración del cliente Butt Broadcast, para posterior se realizar el proceso de inicialización del streaming.

De la misma manera en este objetivo se realizó una plantilla web para que el usuario pueda acceder y escuchar la radio mediante un navegador, donde se integra la dirección URL del streaming de audio. Posterior a esto se realizó la integración del audio streaming en la plataforma TuneIn, la cual permite ser accedida desde el navegador y en la aplicación TuneIn para usuarios con dispositivos móviles, para lo cual deben tener instalado dicha plataforma en sus Smartphone.

Finalmente, para el cumplimiento del objetivo se realizaron las pruebas del sistema, para determinar el correcto funcionamiento del streaming de la radio, en esta subsección de pruebas se realizó: pruebas de capacidad, la cual tuvo como objetivo determinar el máximo de tráfico multimedia que puede soportar la red de la Universidad Nacional de Loja sin que afecte el funcionamiento de otras aplicaciones en esta red, soportando 943 conexiones concurrentes donde obtuvo un ancho de banda de 92.3 Mbps consumidos, con un pico de usuarios de 944, y el porcentaje de error de 0.50% en una muestra de 600 conexiones, con un rendimiento del sistema de 1.0/s, el cual determina que no afectara a terceras aplicaciones en la red de la Universidad y las pruebas de rendimiento, donde se determinó que el servidor y todo el sistema podía soportar 943 conexiones concurrentes sin afectar a los componentes hardware.

7.5. Desarrollar e implementar de una aplicación móvil multiplataforma para acceder a la Radio Universitaria

En este último objetivo se realizó el desarrollo e implementación de la aplicación multiplataforma, la cual permite acceder a la Radio Universitaria desde los dispositivos

móviles. Para poder realizar esta aplicación se utilizó la metodología de desarrollo Mobile-D, la cual se adapta al ciclo de desarrollo software en sus fases de análisis, diseño, codificación, y pruebas.

El desarrollo de la aplicación empezó con el análisis, donde se realiza el estudio de la problemática, obteniendo de esta manera los requerimientos funcionales y no funcionales de la aplicación a desarrollar.

Posteriormente se continuo con la fase de diseño, cuyo objetivo fue obtener los diagramas que representan el funcionamiento de la aplicación y procesos que ejecuta.

A continuación, se procedió con la fase de codificación, en la cual se realizó el desarrollo de las funcionalidades planteadas en la fase de diseño, las cuales fueron codificadas en el lenguaje de programación para aplicaciones híbridas o multiplataforma, con el uso de framework Ionic el cual utiliza HTML5, JavaScript, Apache Córdova y CSS.

Para la culminación se realiza las pruebas, las mismas que determinan que la aplicación es de calidad y que cumple con los requerimientos para la que fue creada, dando un resultado positivo, ya que la aplicación es 100% funcional y cumple con todos los requerimientos propuestos en primera instancia.

Luego de haber realizado las pruebas y comprobar la funcionalidad de la aplicación se procedió a publicarla en la tienda de Google Play, permitiendo a los usuarios poderla descargar e instalar en sus Smartphones.

8. CONCLUSIONES

Luego de haber realizado el presente trabajo de titulación se ha llegado a las siguientes conclusiones:

- Dentro de las tecnologías open-source actuales para implementar una radio online en un ambiente universitario, se encontró los servidores streaming ShoutCast, Icecast, VLCKern, Windows Media Server, Darwing Server, donde el servidor Icecast2 fue el mejor para la implementación, gracias a su capacidad de activar paralelamente varios stream (mountpoint) a través de una misma IP, además ser de fácil instalación, configuración, administración y brindar la posibilidad del envío de archivos multimedia desde una base de datos.
- Para la codificación se eligió el códec MP3, por soportar tasas de bits desde los 8 kbps hasta los 320kbps en CBR, VBR y ABR, además de contar con una alta velocidad de codificación, de la misma manera consigue mayor compresión, alta compatibilidad con los servidores, reproductores, navegadores y dispositivos móviles que facilitan la decodificación en el cliente final.
- Se seleccionó Butt Broadcast como cliente, por ser multiplataforma, tener compatibilidad con el códec MP3, facilitar la conexión con el servidor y reconectarse de forma automática tras una interrupción en la conexión en internet.
- En la implementación del streaming, se determinó que el ancho de banda consumido por un usuario es de 97.87 kbps, de la misma manera se comprobó que el sistema tiene un buen rendimiento, soportando conexiones concurrentes de 943 usuarios, consumiendo un ancho de banda de 92.3 Mbps, para el cual utiliza un 66.50% del CPU y 0.90% de memoria RAM. Además, se determina que 256 kbps es el ancho de banda mínimo para la comunicación entre el cliente y el servidor streaming.
- Se pudo determinar que Mobile-D, permitió desarrollar la aplicación móvil en un menor tiempo, debido a que esta metodología no exigió demasiada documentación, únicamente lo necesario para guiarse en la elaboración de la aplicación. En la producción, se utilizó el framework Ionic, el cual permitió realizar una sola codificación para las plataformas (Android e iOS).
- Para la instalación y ejecución de la aplicación se necesita la versión de software 5.1 o superior en dispositivos Android y la versión 9 o superior en dispositivos iOS.
- Al realizar una comparación de costos entre un servicio alquilado y equipos físicos dentro de la institución, para transferencia de datos para streaming, se determina que resulta más económico comprar un equipo físico que alquilar un servicio en la

nube, esto se da debido a la gran cantidad de datos que se transfiere en un sistema de streaming.

9. RECOMENDACIONES

- Al departamento de la Unidad de Telecomunicaciones e Información, agregar el código del streaming embebido a la página oficial de la Radio Universitaria.
- Crear una VLAN y aumentar el ancho de banda para el sistema de streaming y con ello soportar más conexiones concurrentes a las actuales.
- Agregar y mejorar funcionalidades a la aplicación móvil como: suscripción a programas de la radio, notificaciones de acuerdo a las suscripciones de los programas experimentado así un mejor servicio para los usuarios.
- Se sugiere al personal de la Radio Universitaria realizar uso de la administración de la aplicación la cual le permite actualizar información dinámica dentro de la aplicación como son: noticias, parrilla de programación y programas grabados.
- Se sugiere eliminar contenido de programas grabados que presenten una antigüedad de una semana, con el fin de liberar espacio en el servidor de administración.
- Realizar un estudio de herramientas adicionales, en las cuales se pueda implementar a futuro el servicio de TV-Online, partiendo de la tecnología y herramientas con las que se desarrolló el presente proyecto.
- Finalmente se puede decir que el presente proyecto brinda una proyección inicial del uso de la tecnología del streaming y también puede servir como punto de referencia a aquellas instituciones que quieran incursionar en esta área.

10. BIBLIOGRAFÍA

- [1] “Universidad Nacional de Loja”. [En línea]. Disponible en: <http://unl.edu.ec/>. [Consultado: 19-ene-2016].
- [2] “Organigrama_unl.png (PNG Image, 925 x 890 pixels)”. [En línea]. Disponible en: http://unl.edu.ec/sites/default/files/organigrama/organigrama_unl.png. [Consultado: 25-ene-2017].
- [3] “Radio Universitaria - Nosotros | Universidad Nacional de Loja”. [En línea]. Disponible en: <http://unl.edu.ec/radio/radio-universitaria-nosotros>. [Consultado: 16-may-2016].
- [4] V. Monge, P. Javier, G. Requelme, y E. Reinaldo, “Diseño e implementación de un sistema de transmisión de audio y video por internet para la ex-Facultad de Ingeniería Eléctrica”, Thesis, Quito/EPN, 2004.
- [5] “Cliente-servidor”. [En línea]. Disponible en: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>. [Consultado: 25-ene-2017].
- [6] “Administrar Ubuntu Server mediante telnet – The Bit Developer”. [En línea]. Disponible en: <https://www.thebitdeveloper.com/2013/05/29/administrar-ubuntu-server-mediante-telnet/>. [Consultado: 25-ene-2017].
- [7] “How to Set up FTP Server - Windows FTP| Serv-U”. [En línea]. Disponible en: <http://www.serv-u.com/ftp-server-setup-windows>. [Consultado: 25-ene-2017].
- [8] “iSCSI en una Red SAN basada en TCP/IP para transmisiones de video streaming: Streaming”. [En línea]. Disponible en: http://rdsaltavelocidadrdsantxvsunexpo.blogspot.com/2012/03/streaming_19.html. [Consultado: 25-ene-2017].
- [9] “E. A. M. P. Sebastian Manuel Cajamarca Sacta, «Estudio tecnico sobre la implementacion de una estacion de radiodifusion por internet para la universidad politecnica salesiana sede cuenca,» Cuenca , 2011”.
- [10] R. Libres, «Radios Libres tutorial_4_-_radioslibres_-_radio_en_linea,» [En línea]. Available: https://radioslibres.net/media/uploads/documentos/tutorial_4_-_radioslibres_-_radio_en_linea.pdf.
- [11] “Introducción a los códecs de archivos sonoros y audiovisuales y 10 recomendaciones para seleccionar y gestionar códecs - PrimerForCodecs_spanish.pdf”. .
- [12] “Formatos de audio y vídeo: códecs - FormacionBasica9-Codecs.pdf”. .

- [13] “Revista betsime-la revista del empresario cubano”. [En línea]. Disponible en: http://www.betsime.disaic.cu/secciones/tec_ma_06.htm. [Consultado: 25-ene-2017].
- [14] “Monográfico: introducción al streaming | Observatorio Tecnológico”. [En línea]. Disponible en: <http://recursostic.educacion.es/observatorio/web/fr/component/content/article/567-monografico-introduccion-al-streaming?showall=1>. [Consultado: 25-ene-2017].
- [15] “¿Qué son los medios digitales?” [En línea]. Disponible en: [https://technet.microsoft.com/es-es/library/what-is-digital-media-2\(v=ws.11\).aspx](https://technet.microsoft.com/es-es/library/what-is-digital-media-2(v=ws.11).aspx). [Consultado: 12-ene-2017].
- [16] H. Riofrío y J. Tatiana, “Aplicación móvil para consultas académicas en la Universidad Nacional de Loja.”, 2014.
- [17] U. Goel, G. Aamir, K. Shah, y M. A. Qadeer, “E-radio: The electronic music service”, en *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, 2011, pp. 612–616.
- [18] V. Krishnan y S. G. Chang, “Customized Internet radio”, *Comput. Netw.*, vol. 33, núm. 1–6, pp. 609–618, jun. 2000.
- [19] M. Y. Kang y J. S. Nam, “The study of overlay multicast based Internet broadcasting in parallel media stream server”, en *2009 International Conference on Ultra Modern Telecommunications Workshops*, 2009, pp. 1–3.
- [20] V. Moscoso Moreno, “Propuesta de una radio universitaria comunitaria en la Universidad del Azuay”, 2014.
- [21] C. Cuenca y M. Juliana, “Desarrollo de una aplicación móvil android para la búsqueda de plazas disponibles en un parqueadero.”, 2016.
- [22] “Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf”. [En línea]. Disponible en: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf). [Consultado: 17-ene-2017].
- [23] “Sistemas_Operativos_en_Dispositivos_Moviles.pdf”. [En línea]. Disponible en: http://exa.unne.edu.ar/informatica/SO/Sistemas_Operativos_en_Dispositivos_Moviles.pdf. [Consultado: 17-ene-2017].
- [24] Kent Beck, Cynthia Andres, «Extreme Programming Explained,» de *Extreme Programming Explained*, Boston, Pearson Education, 2012.
- [25] P. Letelier y M. C. Penadés, “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)”, www.cyta.com.ar/ta0502/v5n2a1.htm, 15-abr-2006.

- [En línea]. Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. [Consultado: 17-ene-2017].
- [26] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2000.
- [27] G. Ávila y A. Alejandro, “Desarrollo de un sistema de puntos de ventas para micromercados, utilizando la metodología EXTREME PROGRAMMING”, may 2007.
- [28] T. Rodríguez, “Métodos aplicables para el desarrollo de aplicaciones móviles”, *Genbeta Dev*, 29-sep-2011. [En línea]. Disponible en: <https://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>. [Consultado: 18-ene-2017].
- [29] Paco Blanco, Julio Camarero, Antonio Fumero, Adam Warterski, Pedro Rodríguez, «Metodología de desarrollo ágil para sistemas móviles,» ETSIT Universidad Politécnica de Madrid, 2009.
- [30] “Metodología Mobile-D: Para desarrollos de aplicaciones móviles - Blog de Manuel Guerrero”. [En línea]. Disponible en: <http://manuelguerrero.blogspot.es/1446543763/metodologia-mobile-d-para-desarrollos-de-aplicaciones-moviles/>. [Consultado: 18-ene-2017].
- [31] J. A. O. Henández, «Desarrollo de proyectos de software,» La Paz, Estado de México, 2011.
- [32] L. F. B. León, «Diseño de arquitectura del software,» ZUGYM Pontifica Universidad Javeriana Bogota Colombia, Bogota, 2011.
- [33] B. Kitchenham, «Guidelines for performing Systematic Literature Reviews in Software Engineering,» Department of Computer Science, Durham, 2007.
- [34] P. L. J. Delgado, «Análisis de audiencias de la parrilla de programación de la estación de radio y television universitaria 98.5, y su entorno local,» LOJA, 2015.
- [35] “Debian -- Razones para escoger Debian”. [En línea]. Disponible en: https://www.debian.org/intro/why_debian.es.html. [Consultado: 10-nov-2016].
- [36] J. N. Balcázar, «Revisión Sistemática de herramientas para la implementación de streaming en la Radio Universitaria basado en software libre,» 2016.
- [37] “Conceptos sobre la escalabilidad | Marco de Desarrollo de la Junta de Andalucía”. [En línea]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/220>. [Consultado: 09-jun-2017].
- [38] “Escalabilidad horizontal”. [En línea]. Disponible en: <https://access.redhat.com/documentation/es->

ES/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/intro-horizontal.html. [Consultado: 08-jun-2017].

11. ANEXOS

ANEXO 1

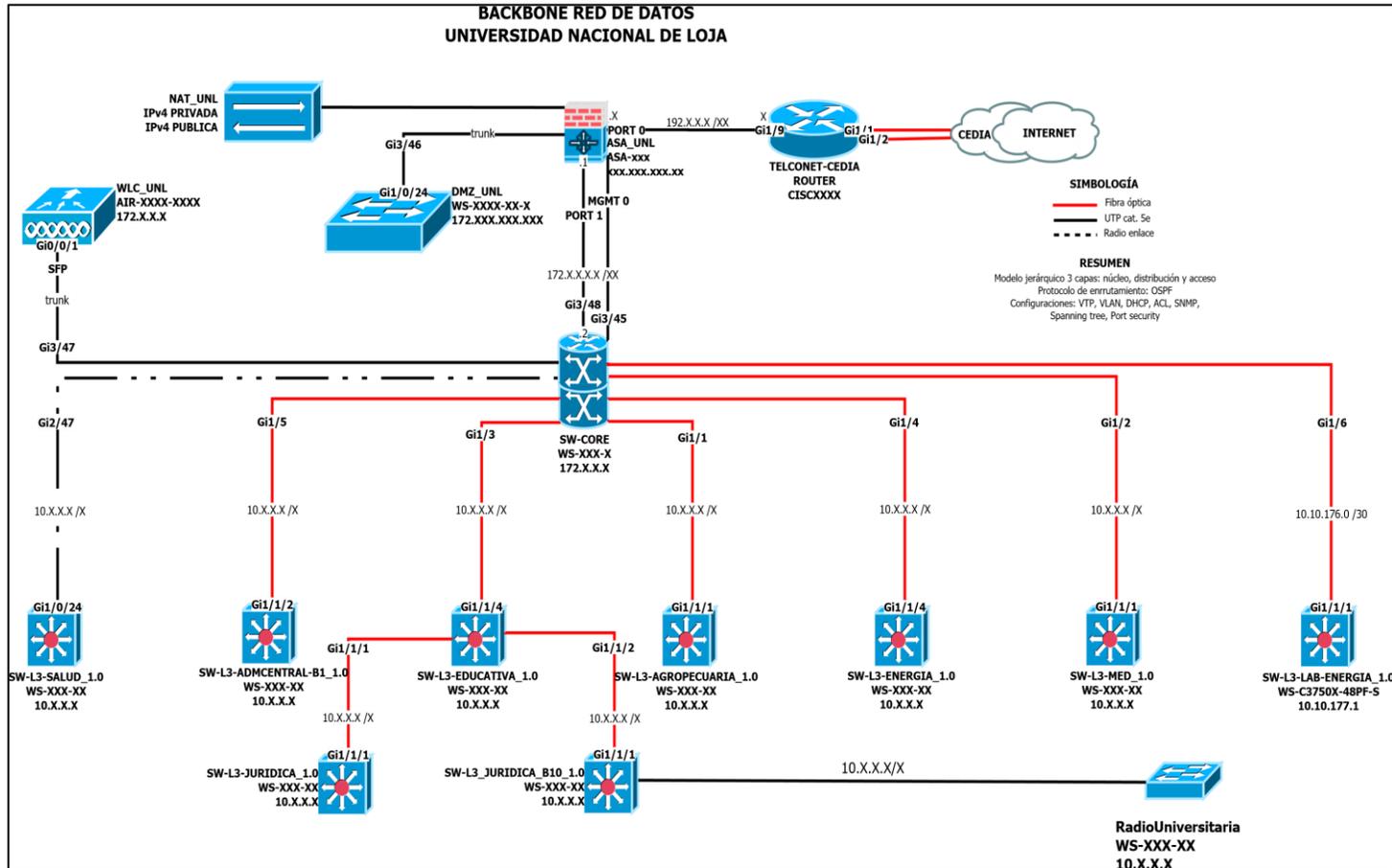


Figura 58. BackBone de la Universidad Nacional de Loja

Fuente: Diagramas UTI

AZIMUT DE TX: 237°
EQUIPO DE TX: OMB MODELO MT

ANTENA RECEPTORA: YAGUI
POLARIZACION: HORIZONTAL/VERTICAL
AZIMUT DE RX: 57°
EQUIPO DE RX: OMB MODELO MR

Si por la asignación de la frecuencia principal y de radioenlace se producen interferencias a otros sistemas de radiocomunicaciones, será de entera responsabilidad del concesionario el solucionarlas, debiendo acatar cualquier disposición que emita la Superintendencia de Telecomunicaciones para el efecto, sin opción a reclamo alguno.

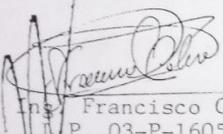
Las emisiones de la estación radiodifusora no pueden llegar a zonas no autorizadas con una intensidad de campo eléctrico mayor a 30 dB(uV/m). Si esto sucede la Superintendencia de Telecomunicaciones puede disponer que se realicen las modificaciones necesarias al sistema de transmisión para evitar que esto ocurra.

Cualquier cambio técnico que desee efectuar en las instalaciones de la estación, debe ser autorizado por el CONARTEL, para lo cual el concesionario lo solicitará con la presentación de los requisitos y estudios que se requieran.

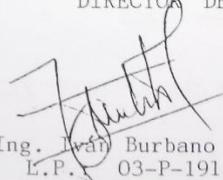
Antes de proceder a autorizar su funcionamiento, se debe notificar a la Superintendencia de Telecomunicaciones, para que ésta proceda a hacer un examen de las emisiones, a fin de verificar que no se produzcan espúreas que causen interferencias, luego de lo cual podrá darse el visto bueno para su funcionamiento.

El transmisor necesariamente debe estar provisto de los filtros que atenúen las emisiones de la 2ª armónica y de las frecuencias no deseadas.

Area a proteger: la limitada por un nivel de intensidad de campo eléctrico de 500 uV/m para las emisiones estereofónicas.


Ing. Francisco Calero
L.P. 03-P-1605
UNIDAD DE ADMINISTRACION
DE CONTRATOS


Ing. Efrén Díaz Villacís
L.P: 03-P-1115
DIRECTOR DE AREA DE SERVICIOS DE DIFUSION


Ing. Iván Burbano R.
L.P. 03-P-191
DIRECTOR GENERAL DE SERVICIOS DE DIFUSION

FC/GCH
C.C.DSD-CONARTEL-JURIDICA-RECER-REG.SUR
19-04-00

Figura 60. Segunda Hoja de datos técnicos Estaciones de radiodifusión en FM

ANEXO 3

INSTALACIÓN Y CONFIGURACIÓN DE LOS SERVICIOS PARA STREAMING

Antes de proceder a la instalación de los servicios a utilizar se tomó en cuenta la preparación de los repositorios en Linux Debian 8 Jessie para posteriormente iniciar con la instalación de los servicios a utilizar.

Paso 1: Preparación de repositorios en Linux Debian 8 Jessie

Antes de iniciar con las instalaciones de nuestros servicios verificamos que los repositorios contienen la siguiente lista de repositorios:

```
deb http://security.debian.org/ jessie/updates main contrib
```

```
deb-src http://security.debian.org/ jessie/updates main contrib
```

```
deb http://http.us.debian.org/debian jessie main contrib non-free
```

Para poder acceder a este archivo de configuración procedemos abrir una consola e ingresamos como superusuario y digitamos el siguiente comando **nano /etc/apt/sources.list** si no contiene alguno de los repositorios colocarlo.



```
# nano /etc/apt/sources.list
```

Figura 61. Comando para acceder a la lista de repositorios

Seguidamente procedemos a guardar los cambios presionando **Ctrl+O**, salimos del archivo de configuración presionando **Ctrl+X** y procedemos a actualizar nuestro sistema con los comandos **apt-get update** y **apt-get upgrade**.

Nota: Cabe mencionar que no se necesita ningún servicio adicional levantado.

INSTALACIÓN DE SERVIDOR ICECAST2 EN DEBIAN 8 JESSIE

Paso 2: Instalación del servidor Icecast2

Para instalar el servidor de streaming Icecast2 se procede abrir una terminal, seguidamente se digita el siguiente comando sudo **apt-get install icecast2**-



```
jeneirab@radio:~$ sudo apt-get install icecast2
```

Figura 62. Comando de instalación de Icecast2

Una vez que se ha digitado el comando nos procederá a pedir el password seguidamente pondremos **S** para aceptar instalar los paquetes y presionamos **Intro** para continuar con la instalación-

```

jeneirab@radio: ~
Archivo Editar Ver Buscar Terminal Ayuda
jeneirab@radio:~$ sudo apt-get install icecast2

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for jeneirab:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libogg0 libspeex1 libtheora0 libvorbis0a libxslt1.1
Paquetes sugeridos:
  ices2 speex
Se instalarán los siguientes paquetes NUEVOS:
  icecast2 libogg0 libspeex1 libtheora0 libvorbis0a libxslt1.1
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 848 kB de archivos.
Se utilizarán 2.394 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] 

```

Figura 63. Ingreso de password e instalación de paquetes.

Y de esta manera procede la descarga e instalación de Icecast2-

```

jeneirab@radio:~$ sudo apt-get install icecast2

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for jeneirab:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libogg0 libspeex1 libtheora0 libvorbis0a libxslt1.1
Paquetes sugeridos:
  ices2 speex
Se instalarán los siguientes paquetes NUEVOS:
  icecast2 libogg0 libspeex1 libtheora0 libvorbis0a libxslt1.1
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 848 kB de archivos.
Se utilizarán 2.394 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://http.us.debian.org/debian/ jessie/main libogg0 amd64 1.3.2-1 [19,9 kB]
Des:2 http://security.debian.org/ jessie/updates/main icecast2 amd64 2.4.0-1.1+deb8u1 [277 kB]
4% [2 icecast2 13,3 kB/277 kB 5%]

```

Figura 64. Descarga e instalación de paquetes

CONFIGURACIÓN DEL SERVIDOR ICECAST2

Paso 3: Configuración del servidor Icecast2

Se procede abrir una terminal e ingresamos en modo superusuario y a continuación ingresamos el siguiente comando `nano /etc/icecast2/icecast.xml`.

```

root@radio:/home/jeneirab# nano /etc/icecast2/icecast.xml

```

Figura 65. Comando para ingresar al archivo de configuración Icecast2

Inmediatamente se procede abrir nuestro archivo de configuración (ver Figura 53)

```

GNU nano 2.2.6                               Fichero: /etc/icecast2/icecast.xml
icecast>
<!-- location and admin are two arbitrary strings that are e.g. visible
on the server info page of the icecast web interface
(server_version.xml). -->
<location>Earth</location>
<admin>icemaster@localhost</admin>

<limits>
  <clients>100</clients>
  <sources>2</sources>
  <threadpool>5</threadpool>
  <queue-size>524288</queue-size>
  <client-timeout>30</client-timeout>
  <header-timeout>15</header-timeout>
  <source-timeout>10</source-timeout>
  <!-- If enabled, this will provide a burst of data when a client
first connects, thereby significantly reducing the startup
time for listeners that do substantial buffering. However,
it also significantly increases latency between the source
client and listening client. For low-latency setups, you
might want to disable this. -->
  <burst-on-connect>1</burst-on-connect>
  <!-- same as burst-on-connect, but this allows for being more
specific on how much to burst. Most people won't need to
change from the default 64k. Applies to all mountpoints -->
  <burst-size>65535</burst-size>
</limits>

<authentication>
  <!-- Sources log in with username 'source' -->
  <source-password>hackme</source-password>
  <!-- Relays log in username 'relay' -->
  <relay-password>hackme</relay-password>

  <!-- Admin logs in with the username given below -->
  <admin-user>admin</admin-user>
  <admin-password>hackme</admin-password>
</authentication>

<!-- get the mountpoint for a shoutcast source to use, the default if not
specified is /stream but you can change it here if an alternative is
wanted or an extension is required
-->
<shoutcast-mount>/live.nsv</shoutcast-mount>
-->

<!-- Uncomment this if you want directory listings -->
<!--
<directory>
  <yp-url-timeout>15</yp-url-timeout>

```

Figura 66. Archivo de configuración de servidor Icecast2

Paso 3.1: Configuración de clientes

A continuación, se procede a configurar el número de clientes para nuestro servidor de Icecast2 que escucharan nuestro streaming, en este caso se ubicara el valor de 1000 clientes.

```
<clients>100</clients>
```

Figura 67. Línea de configuración de clientes

Paso 3.2: Configuración de password en el source, relay, usuario y password de servidor streaming Icecast2.

```

<authentication>
  <!-- Sources log in with username 'source' -->
  <source-password>hackme</source-password>
  <!-- Relays log in username 'relay' -->
  <relay-password>hackme</relay-password>

  <!-- Admin logs in with the username given below -->
  <admin-user>admin</admin-user>
  <admin-password>hackme</admin-password>
</authentication>

```

Figura 68. Configuraciones de passwords para el servidor streaming Icecast2

Paso 3.3: Configuración de hostname para streaming

Para la configuración del hostname tenemos que proporcionar una dirección IP o dirección de dominio que utilizaremos.

Nos dirigimos hacia la línea de configuración **hostname** y procedemos a ingresar nuestra dirección IP o dominio que utilizaremos para la difusión del streaming.

```
<hostname>localhost</hostname>
```

Figura 69. Configuración del hostname

TABLA LXI. DATOS PARA HOSTNAME

Hostname	radio.unl.edu.ec
Dirección IP	192.168.49.14

En este caso se procedió a configurar con el dominio **radio.unl.edu.ec** como hostname quedando de la siguiente manera.

```
<hostname>radio.unl.edu.ec</hostname>
```

Figura 70. Configuración del hostname con el dominio

CONFIGURACIÓN DE ARCHIVO DEFAULT DE ICECAST2

Paso 4: Configuración de archivo default de Icecast2

Procedimos abrir una terminal, ingresamos como superusuario y a continuación ingresamos el comando **nano /etc/default/icecast2**.

```
root@radio:/home/jeneirab# nano /etc/default/icecast2
```

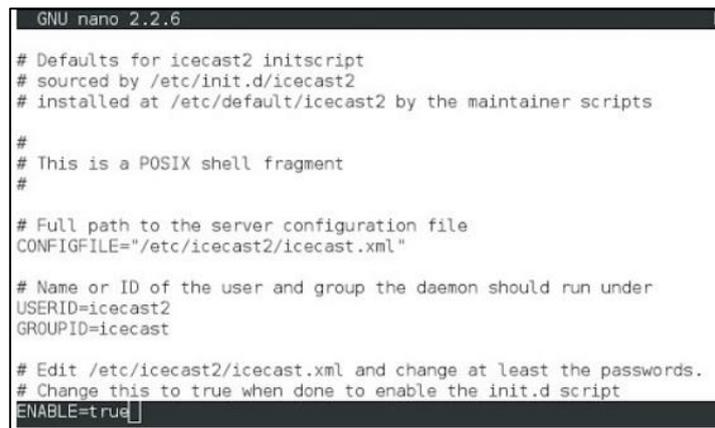
Figura 71. Comando para abrir el archivo default de Icecast2

De manera inmediata se procederá abrir el archivo de configuración el cual vamos a editar.

```
GNU nano 2.2.6                               File
# Defaults for icecast2 initscript
# sourced by /etc/init.d/icecast2
# installed at /etc/default/icecast2 by the maintainer scripts
#
# This is a POSIX shell fragment
#
# Full path to the server configuration file
CONFIGFILE="/etc/icecast2/icecast.xml"
# Name or ID of the user and group the daemon should run under
USERID=icecast2
GROUPID=icecast
# Edit /etc/icecast2/icecast.xml and change at least the passwords.
# Change this to true when done to enable the init.d script
ENABLE=false
```

Figura 72. Contenido de archivo default de Icecast2

En este punto observamos que el archivo de configuración tenemos la variable **ENABLE** con un valor de **false** por lo que se procedió a cambiar al valor de **true** como se muestra en la figura 73, posteriormente guardamos y salimos del archivo de configuración.



```
GNU nano 2.2.6 F
# Defaults for icecast2 initscript
# sourced by /etc/init.d/icecast2
# installed at /etc/default/icecast2 by the maintainer scripts
#
# This is a POSIX shell fragment
#
# Full path to the server configuration file
CONFIGFILE="/etc/icecast2/icecast.xml"
# Name or ID of the user and group the daemon should run under
USERID=icecast2
GROUPID=icecast
# Edit /etc/icecast2/icecast.xml and change at least the passwords.
# Change this to true when done to enable the init.d script
ENABLE=true
```

Figura 73. Archivo default modificado

INSTALACIÓN DE BUTT BROADCAST EN LINUX DEBIAN

Paso 1: Descarga de instalador Butt Broadcast para Linux

Antes de realizar la instalación de Butt Broadcast, primero se procederá a descargar el archivo de instalación para Linux ya sea para distribuciones de 32 bits o 64 bits en este caso se descargó el de 64 bits del siguiente enlace <http://gnetertics.org/article/butt-streaming-facil/> el archivo que se descarga tiene una extensión **.deb**

Paso 2: Instalación de Butt Broadcast

Se procede abrir una consola, accedemos a la carpeta que se descargó el archivo **.deb** e insertamos el siguiente comando **dpkg --install butt_0.1.14_amd64**.



```
/Descargas$ dpkg --install butt_0.1.14-1_amd64.deb
```

Figura 74. Comando para instalar Butt Broadcast

Posteriormente se ejecuta los comandos **apt-get update** luego, **apt-get upgrade** y finalmente ejecutar **apt-get -f install**.

CONFIGURACIÓN DE BUTT BROADCAST

Paso 1: Abrir el programa Butt Broadcast

Para poder abrir nuestro programa Butt Broadcast se la puede realizar mediante el modo gráfico y búsqueda por su nombre o abrir una terminal e ingresar el siguiente comando **/usr/bin/butt**.

Paso 2: Configuración de Butt Broadcast

Una vez abierto el programa procedemos a abrir la ventana de configuración para ello nos dirigimos al botón **settings** y realizamos un clic el cual desplegará la ventana de configuraciones.

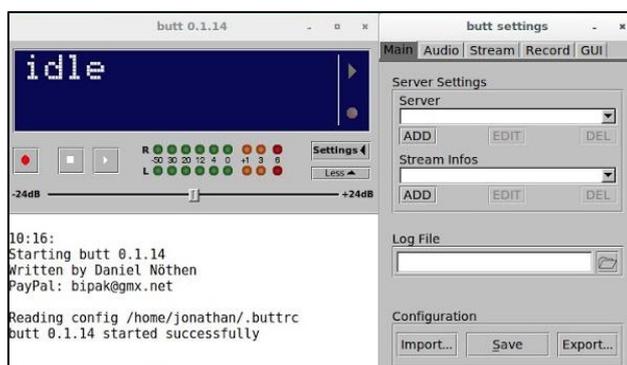


Figura 75. Ventana de configuraciones de Buttt Broadcast

Paso 2.1: Configuración de Servidor streaming

Para agregar un servidor streaming nos dirigimos a **Server Settings**, presionamos **ADD** y elegimos el servidor que se utilizará en este caso **Icecast** y se procede a llenar la información de nuestro servidor: la dirección de nuestro servidor ya sea el dominio o dirección IP, el puerto, el password de nuestro source, el punto de montaje donde se reproducirá el streaming (ver Tabla XLV) y finalmente guardamos los cambios dando clic en el botón **save**.

TABLA LXII. DATOS DE CONFIGURACIÓN DE SERVIDOR EN BUTT BROADCAST

Datos de configuración	
Name	Radio Universitaria 98.5
Address	Radio.unl.edu.ec
Port	8080
Password	*****
Icecast mountpoint (Punto de Montaje)	Stream
Icecast User	source

Figura 76. Ventana con datos de configuración de Butt Broadcast

Paso 2.2: Configuración de información de streaming

Nos dirigimos a la ubicación de **Stream Info** y realizamos clic en el botón **ADD** y procedemos agregar la información de nuestro servidor como: el nombre, la descripción, el género y la dirección URL de nuestra página Web.

TABLA LXIII. DATOS DE CONFIGURACIÓN DE STEAMS INFO

Datos a ingresar	
Name (Nombre)	Radio Universitaria
Genre (Género)	Académico
URL	http://unl.edu.ec/radio/inicio-radio-universitaria

Figura 77. Datos de información para servidores

Paso 2.3: Configuración de entrada de audio, códec y bitrate

Para realizar las configuraciones de entrada de audio, códec y bitrate se procede hacer clic en la pestaña de **Audio** y nos situamos en **Audio Device** y configuramos en la opción Default la cual evalúa la tarjeta de sonido y captura su audio de la misma manera se configuro el códec en el formato establecido anteriormente que es MP3 y el bitrate en 128k para una calidad de sonido a continuación se procede a guardar realizando clic en **Save**.

TABLA LXIV. DATOS DE CONFIGURACIÓN DE AUDIO, CÓDEC Y BITRATE

Datos a configurar	
Audio Device	Default PCM device (default)
Códec	MP3
Bitrate	128k

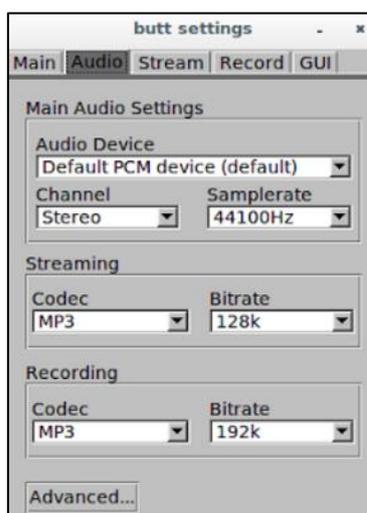


Figura 78. Ventana de configuración de audio, códec y bitrate en Butt Broadcast

INICIALIZACIÓN DE SERVICIOS

INICIALIZACIÓN DEL SERVICIO DE ICECAST2

Paso 1: Inicialización del servidor Icecast2

Se procede abrir una terminal e ingresamos el siguiente comando **sudo /etc/init.d/icecast2 start**

```
# sudo /etc/init.d/icecast2 start
```

Figura 79. Comando para iniciar servicio de Icecast2

Seguidamente nos aparecerá en la terminal que nuestro servicio se inició correctamente.

```
root@radio:/home/jeneirab# sudo /etc/init.d/icecast2 start  
[ ok ] Starting icecast2 (via systemctl): icecast2.service.
```

Figura 80. Inicio satisfactorio de servicio de Icecast2

Paso 2: Confirmación de servicio Icecast2 iniciado

Para comprobar y verificar el inicio del servicio de Icecast2 se procede abrir el navegador de e introducir en la barra de dirección de nuestro **hostname** seguido del puerto que se configuro para nuestro servidor, quedándonos de la siguiente manera **radio.unl.edu.ec:8080** si todo va bien se debe visualizar la página de administración de Icecast2

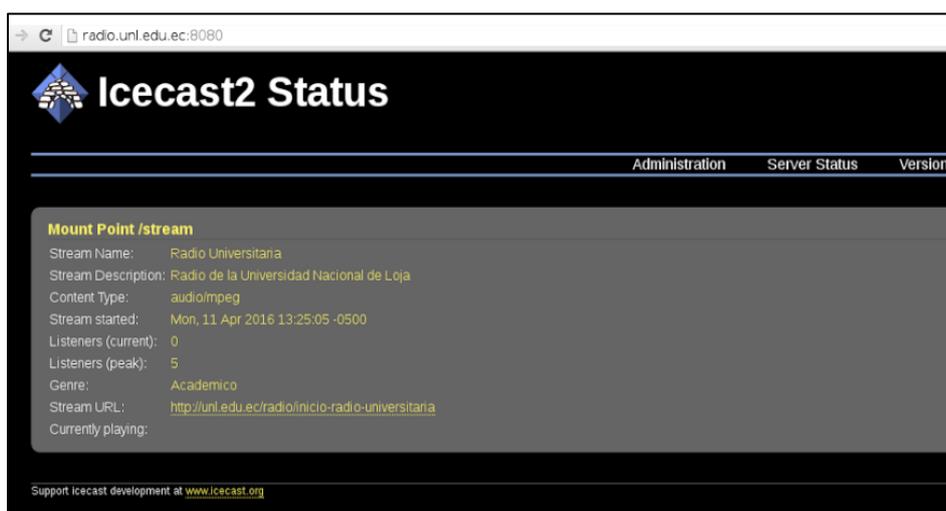


Figura 81. Página de administración de Icecast2

INICIALIZACIÓN DE STREAMING

Paso 1: Iniciar la aplicación Butt Broadcast

Para la inicialización del streaming se procede abrir la aplicación Butt Broadcast (ver subsección 4.2.1 Paso 1)

Paso 2: Inicio de streaming

Una vez abierto Butt Broadcast y realizado las configuraciones antes mencionadas, se procede a realizar clic en el botón **Play**  y de esta manera se inicia el streaming.

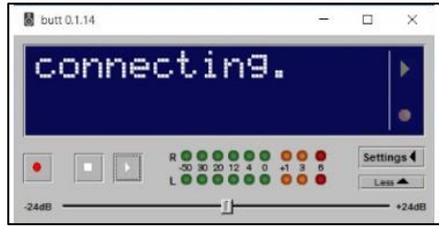


Figura 82. Conexión e inicio de streaming en Buttt Broadcast

ANEXO 4

INTEGRACIÓN DE STREAMING A PLANTILLA WEB EN APACHE

Paso 1: Ingresar al archivo índice de Apache

Para ello digitamos el siguiente comando `nano /var/www/html/index.html` esto nos permite abrir con el editor propiamente de Linux ya que no se cuenta con una interfaz gráfica en el servidor.

Paso 2: Integrar streaming a plantilla web

Una vez abierto el editor procedemos a pegar el código que nos permite acceder y escuchar el streaming en la plantilla web:

```
<h1>Radio Universitaria 98.5 FM</h1>
<p>Universidad Nacional de Loja</p>
<!--Reproductor Radio-->
<audio controls autoplay >
  <source src="http://radio.unl.edu.ec:8080/stream" type="audio/mpeg">
</audio>
```

Figura 83. Código para acceder streaming

Paso 3: Guardar cambios y cerrar

Para guardar los cambios se lo realiza mediante las teclas **Ctrl+O** y **enter**, posterior a esto procedemos a cerrar el editor usando **Ctrl+X**

De esta manera tenemos integrado el streaming a la plantilla web, el cual permite ser accedido por los usuarios mediante un navegador web.

ANEXO 5

INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTAS PARA EVALUACIÓN DEL SISTEMA STREAMING

Paso 1: Descargar Apache JMeter

Para poder utilizar Apache JMeter, primero se descarga el ejecutable desde su sitio oficial, para ello nos dirigimos a la siguiente dirección http://jmeter.apache.org/download_jmeter.cgi y descargamos el archivo zip o tgz de la opción Source.

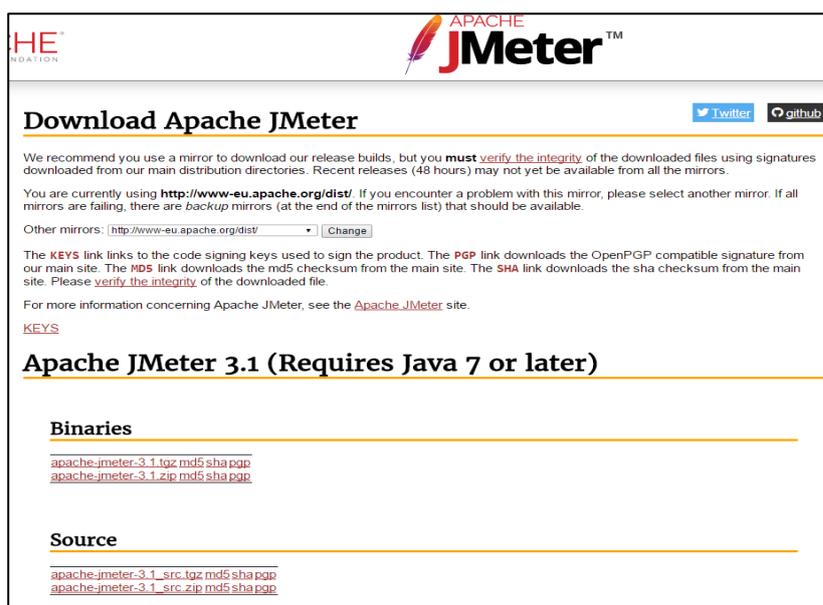


Figura 84. Sitio oficial de descargas de Apache JMeter

Nota: Para poder utilizar esta herramienta se requiere tener instalado Java 7 o superior.

Paso 2: Ejecutar Apache JMeter

Una vez descargado el archivo zip o tgz procedemos a descomprimirlo, posteriormente nos dirigimos a la carpeta descomprimida y navegamos hacia la carpeta donde tiene el archivo ApacheJMeter.jar, ubicada en la siguiente ruta `apache-jmeter-3.1\bin\ApacheJMeter.jar` y procedemos a ejecutarlo.

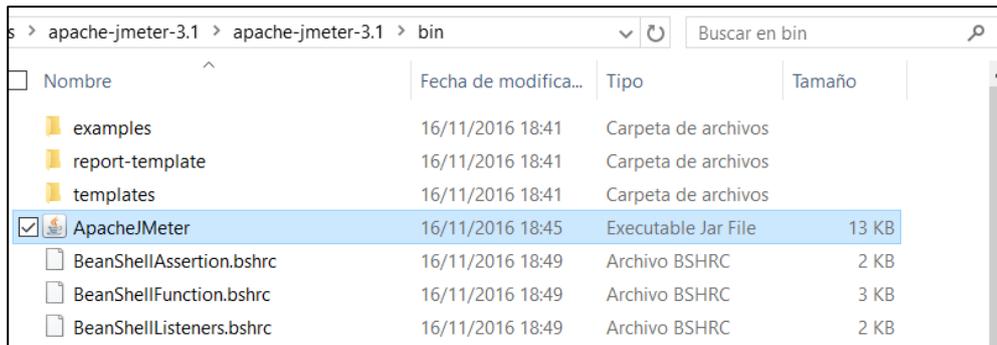


Figura 85. Ubicación de archivo ApacheJMeter.jar

Una vez ejecutado se visualiza la pantalla principal de Apache JMeter, donde se procede con la configuración del escenario para las respectivas pruebas.

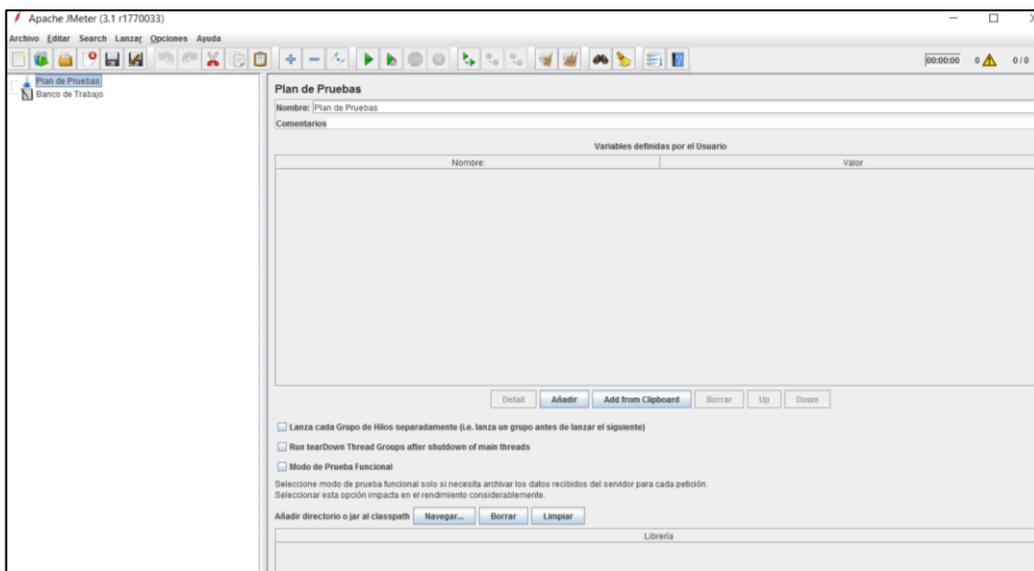


Figura 86. Pantalla principal de Apache JMeter

Paso 3: Configuración de escenario de pruebas en Apache JMeter

En este paso se prepara el escenario de la prueba, donde se emula a los usuarios que realizan las peticiones de conexión hacia el streaming.

Primero se debe ingresar el nombre al plan de pruebas, en este caso se lo nombro como **Pruebas Radio Universitaria**, a continuación, procedemos agregar el grupo de usuarios, para ello damos clic derecho abrir, hilos (Usuarios), Grupo de usuarios como se muestra en la siguiente figura.

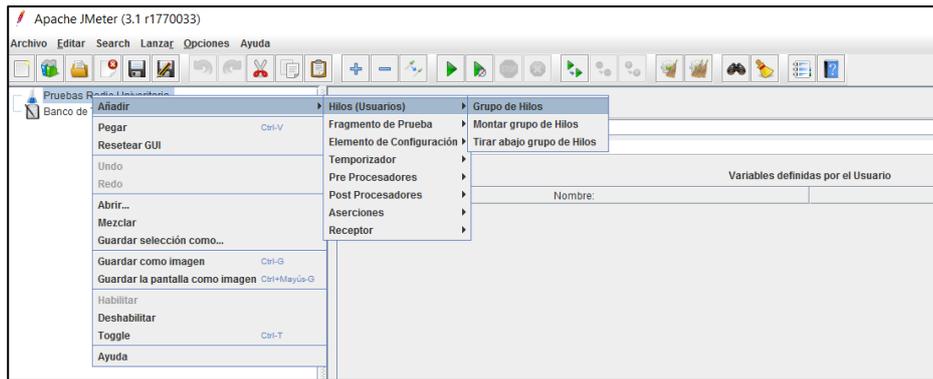


Figura 87. Agregar grupo de hilos (usuarios)

A continuación, se agrega el nombre del grupo de hilos (usuarios), el número de hilos a emular y cada qué periodo se conecta un nuevo usuario (ver Figura 75), de la misma manera se agrega el muestreador donde se realiza la petición HTTP, para ello se realiza clic derecho añadir, muestreador, petición HTTP como se visualiza en la figura 76.

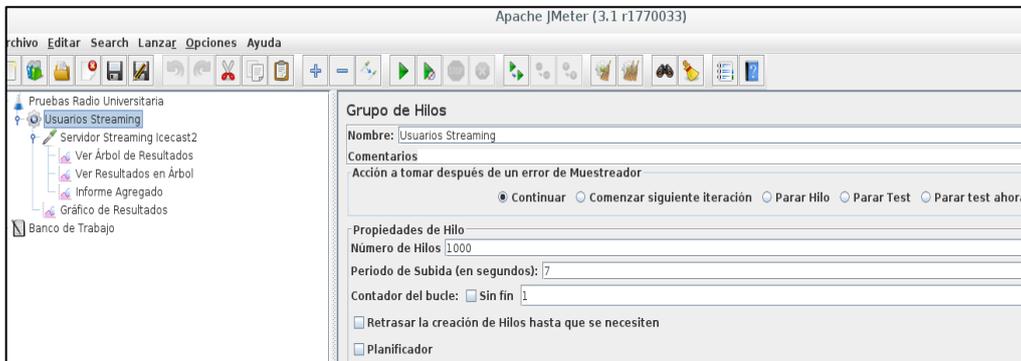


Figura 88. Configuración de grupo de usuarios

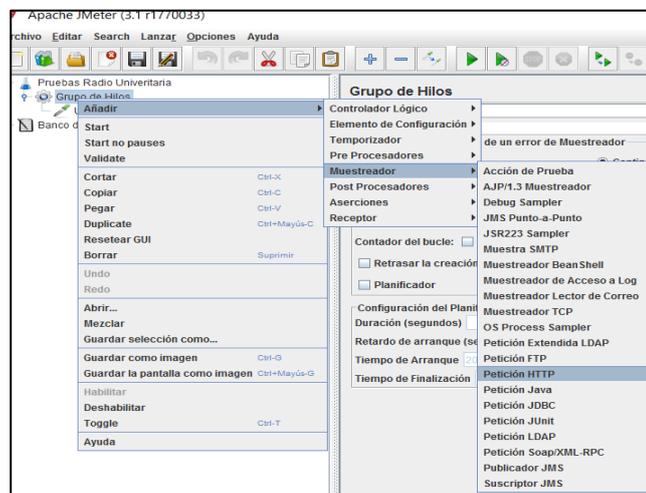


Figura 89. Agregar muestreador con petición http

A hora se procede a configurar la petición http, donde colocamos el nombre de la petición, en este caso se lo nombro **Servidor Streaming Icecast2**, seguidamente configuramos el nombre del servidor o dirección IP **radio.unl.edu.ec**, seguidamente colocamos el puerto que utiliza el sistema streaming **8080** y para finalizar la configuración de la petición http se coloca el punto de montaje donde se reproduce el streaming, para ello nos ubicamos en Ruta y colocamos **/stream** de esta manera emulamos la petición que se realiza en un navegador normal.

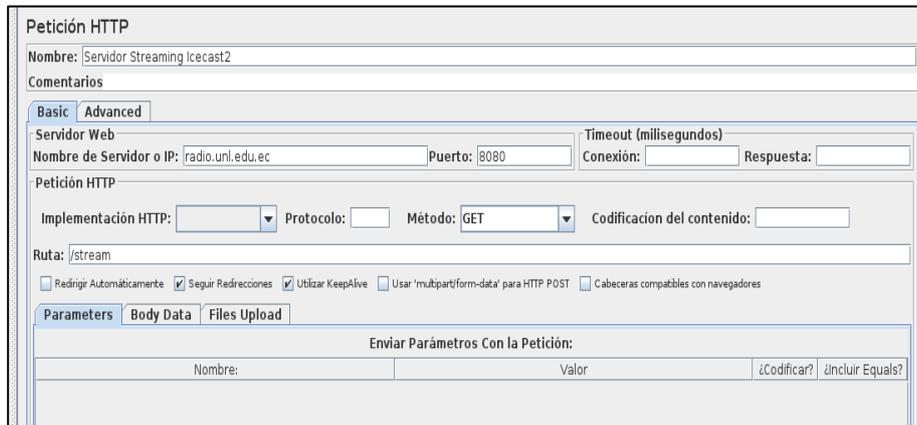


Figura 90. Peticiones HTTP

A continuación, necesitamos configurar receptores donde se pueda visualizar los resultados, para ello se configura 4 receptores: ver árbol de resultados, ver resultados en árbol, informe agregado y la gráfica de resultados. Para poder agregar estos receptores, se realiza clic derecho sobre la petición http añadir, receptor, y agregamos cada uno de los receptores (ver árbol de resultados, ver resultados en árbol, informe agregado y la gráfica de resultados).

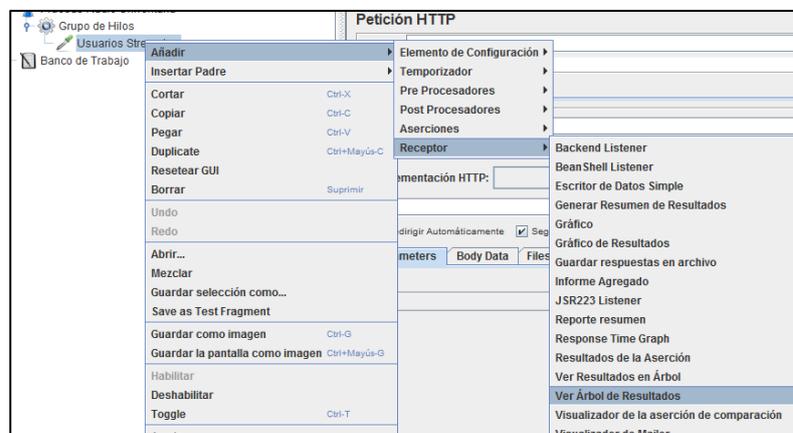


Figura 91. Agregar receptores

Una vez agregado los receptores, nuestro escenario de prueba en Apache JMeter nos debe quedar en el orden jerárquico como se muestra en la figura 79.



Figura 92. Orden jerárquico

Paso 4: Descargar Winbox

Para descargar la herramienta Winbox procedemos a ingresar a la URL http://wirelessconnect.eu/articles/winbox_download, presionamos el botón descargar, con lo cual procede a descargarnos un archivo .rar.

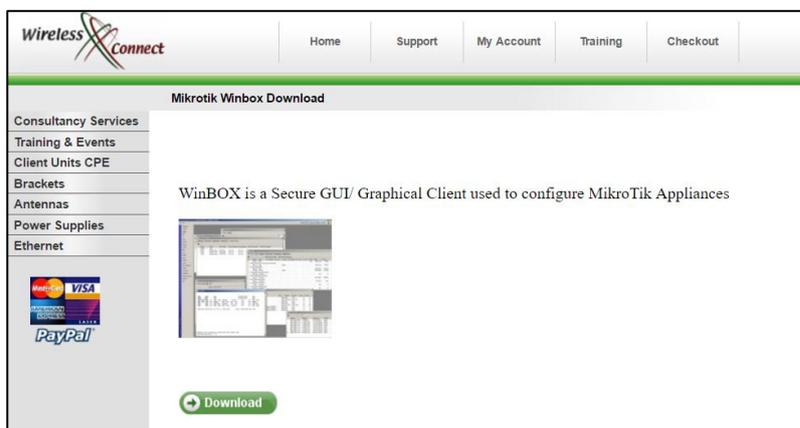


Figura 93. Página de descarga de Winbox

Paso 5: Ejecutar Winbox

Para ejecutar Winbox, se procede a descomprimir el archivo .rar que se descargó con anterioridad, a continuación, ejecutamos el archivo winbox.exe, donde presenta la pantalla principal (ver figura 81).

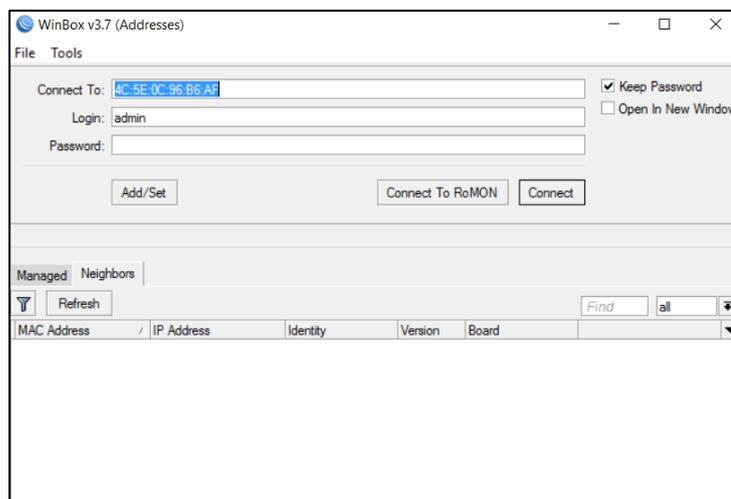


Figura 94. Pantalla principal de Winbox

Paso 5: Conexión y configuración con equipo Mikrotik

Antes de proceder a configurar Winbox se realiza la conexión con el equipo Mikrotik, para realizar esta conexión se lo hace mediante la MAC del equipo Mikrotik, se presiona el botón **Connect**, de esta manera quedara conectado y podemos acceder hacia la configuración del equipo Mikrotik.

Posterior a la conexión, se realiza un **TORCH** el cual nos ayuda a monitorear el (detalle de consumo de usuarios), para realizar el **TORCH** accedemos a la opción **TOOLS -> TORCH**, donde nos presenta la siguiente pantalla:

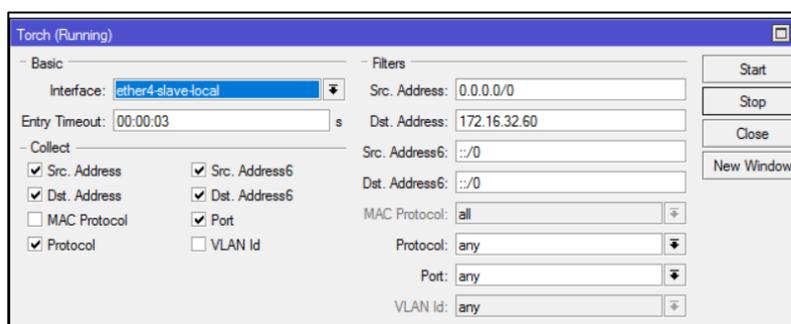


Figura 95. Pantalla de TORCH (Running) Winbox

A hora se procede a configurar las opciones para el **TORCH**, donde seleccionaremos la **interface 4**, la cual se elige por que el equipo de pruebas se encuentra conectado en el puerto de red número 4 del equipo Mikrotik, a continuación, se ingresa la dirección IP de destino **172.16.32.60**, se selecciona las opciones Src. Address/(6), Dst. Address/(6), Protocol y Port, los cuales permiten visualizar las conexiones que se realizan hacia el servidor.

Paso 6: Iniciar el escenario de pruebas en Apache JMeter y Winbox

Para iniciar el escenario en Apache JMeter, se hace clic en el botón play, de esta manera queda iniciada la emulación de los usuarios, los cuales se configuraron en pasos anteriores

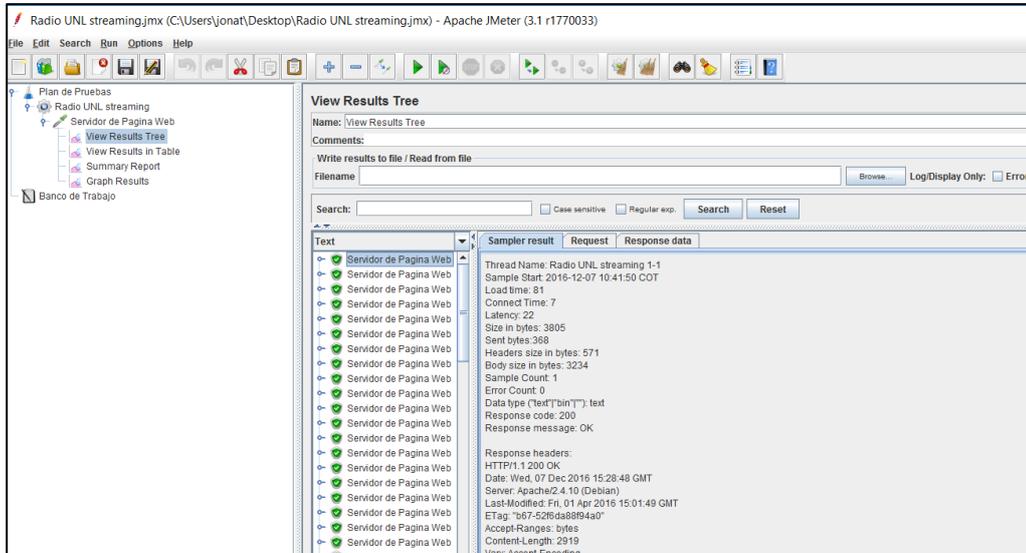


Figura 96. Inicio de escenario en Apache JMeter

Al iniciar el escenario en Apache JMeter, se debe iniciar también el TORCH de Winbox, para ello presionamos el botón **start**, el cual nos permite visualizar el detalle de consumo por los usuarios emulados en Apache JMeter.

Paso 7: Visualizar las pantallas de resultados

Para visualizar las pantallas de resultados nos dirigimos a la pantalla de Apache JMeter y damos clic en los receptores que deseamos visualizar ya sea resultados en árbol, la tabla de resultados en árbol, informe agregado y la gráfica de resultados.

Para visualizar el ancho de banda a través de Winbox nos ubicamos en la pantalla de interfaces y prestamos atención a la interfaz número 4 la cual es la que se encuentra conectada al equipo de las pruebas.

Interface List									
Interface Ethernet EoIP Tunnel IP Tunnel GRE Tunnel VLAN VRRP Bonding LTE									
	Name	Type	L2	MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	
R	br_vlanDOCE...	Bridge		1584	3.2 kbps	0 bps	3	0	
R	br_vlanIMPRES...	Bridge		1584	3.3 kbps	0 bps	3	0	
R	ether1-Link	Ethernet		1588	3.8 Mbps	92.4 Mbps	7 312	7 912	
	ether2-master4...	Ethernet		1588	0 bps	0 bps	0	0	
RS	ether3-slave-lo...	Ethernet		1588	51.7 kbps	14.2 kbps	40	17	
R	ether4-slave-lo...	Ethernet		1588	92.3 Mbps	3.9 Mbps	7 900	7 425	
S	ether5-slave-lo...	Ethernet		1588	0 bps	0 bps	0	0	

Figura 97. Lista de interfaces en Winbox

De la misma manera para visualizar el monitor de Icecast2 y constatar cuantos usuarios se conectan, nos dirigimos hacia el navegador de nuestra preferencia y digitamos la siguiente dirección URL <http://radio.unl.edu.ec:8080/status.xsl>, donde nos aparece la información del servidor como: nombre del stream, descripción, contenido, genero, etc., pero lo más importante de esta información son los usuarios concurrentes que se encuentran conectados en un momento específico.

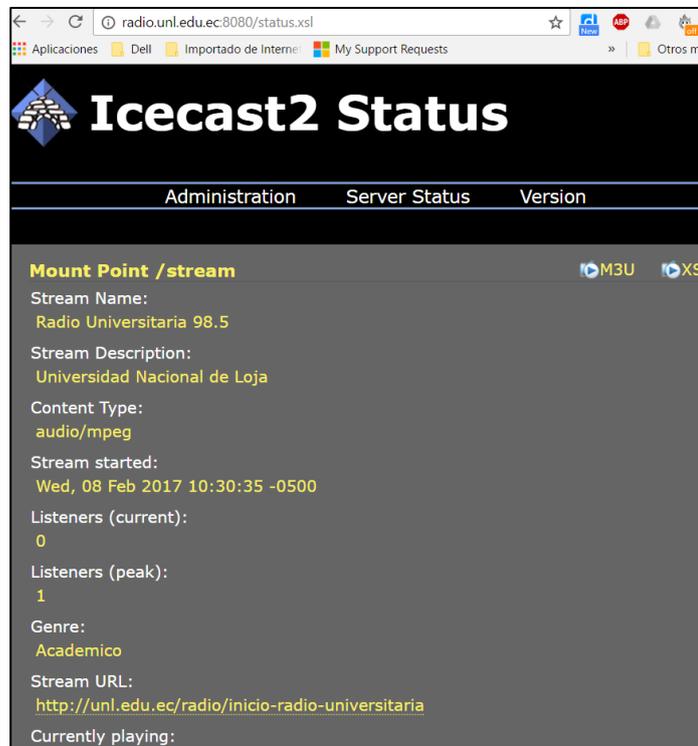


Figura 98. Monitor de status del servidor Icecast2

ANEXO 6

Integración del streaming en TuneIn

Paso 1: Entrar a página principal

Primeramente, abrimos un navegador a continuación, ingresamos la siguiente dirección www.tunein.com

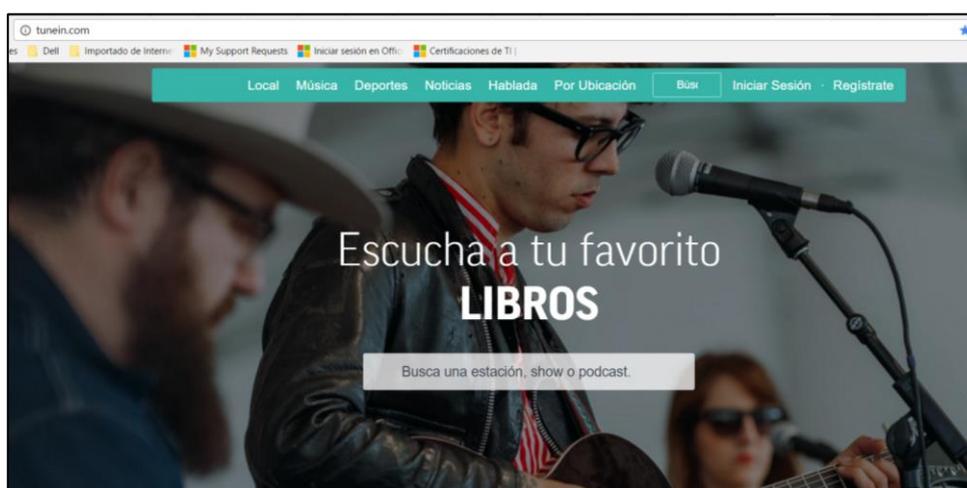


Figura 99. Página Principal de TuneIn

Paso 2: Ingresar a Radiodifusoras

Seguidamente procedemos a ingresar a la opción Radiodifusoras la misma que se encuentra al final de la página.

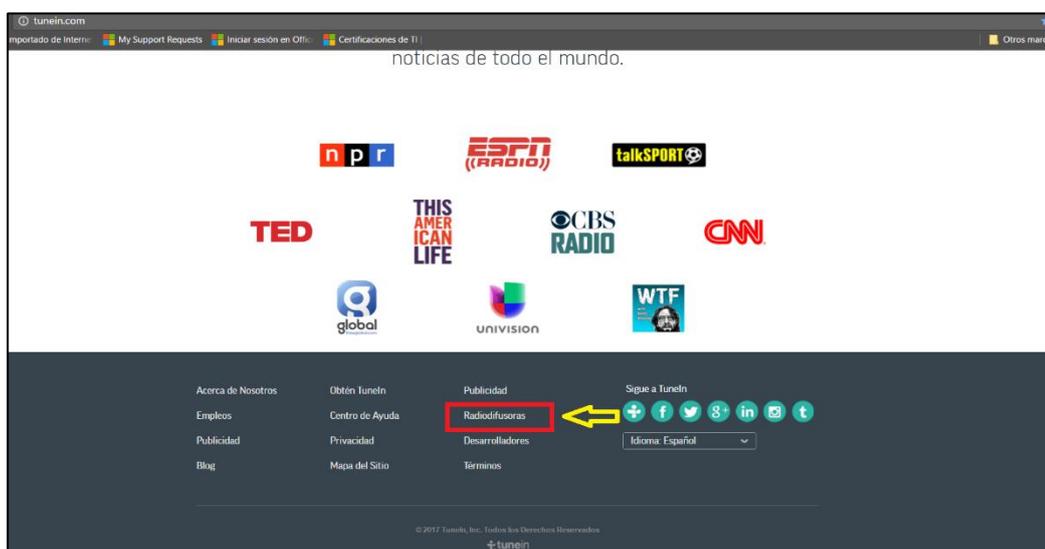


Figura 100. Página de TuneIn opción (Radiodifusoras)

Paso 3: Añadir una estación (streaming de la radio)

En este paso se realiza el registro de nuestra radio (streaming), para lo cual ingresamos en la opción **ADD A STATION** a continuación, se abrirá un formulario donde crearemos una nueva estación para ello necesitaremos varios datos obligatorios para poder crear nuestra radio, una vez llenado todo el formulario procedemos aceptar las condiciones y enviamos nuestro formulario.

Una vez enviado el formulario TuneIn nos muestra un mensaje que la propuesta de nuestra radio se ha enviado con éxito, la cual será analizada en un periodo de 3 a 5 días y una vez analizada TuneIn enviará un correo electrónico en el cual se acepta nuestra estación y nos informa que ahora es parte de la red de TuneIn, por lo que está disponible en aplicaciones TuneIn para iPhone, Blackberry y los teléfonos Android y en muchas otras aplicaciones accionados por TuneIn.

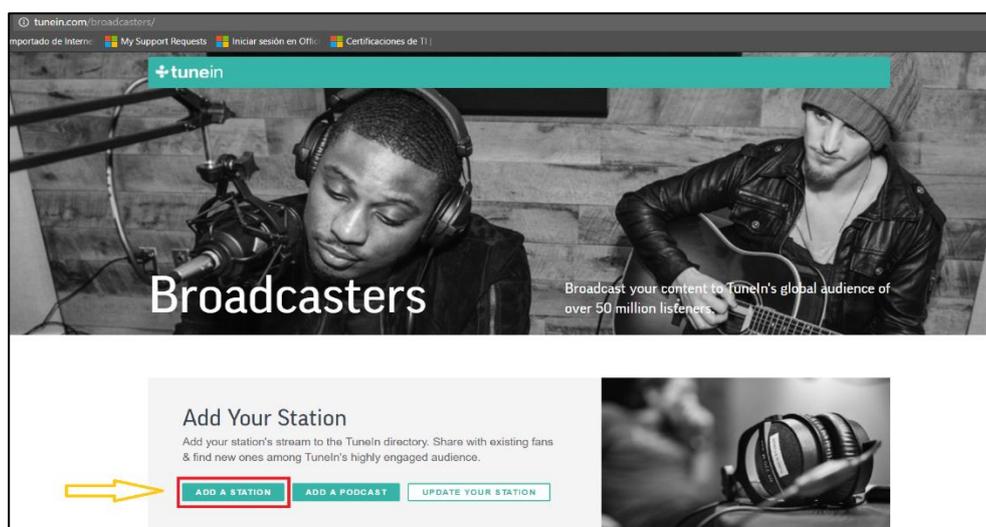


Figura 101. Página de TuneIn Radiodifusoras opción (Add a station)

TABLA LXV. DATOS PARA AGREGAR RADIO UNIVERSITARIA A TUNEIN

Campo	Dato ingresado	Opción del campo
Nombre de emisora	Radio Universitaria	Requerido
Correo electrónico registro	<u>apps.moviles@unl.edu.ec</u>	Obligatorio
Página Web	<u>http://unl.edu.ec/radio/inicio-radio-universitaria</u>	Opcional

URL de la transmisión (streaming)	http://radio.unl.edu.ec:8080/stream	Requerido
Tipo de Trasmisión	Solo Internet	Requerido
País	Ecuador	Requerido
Idioma	Español	Requerido
Formato (audio)	MP3	Requerido
Logo	En esta opción se cargó el logo de la Radio Universitaria	Requerido
Estación de contactos de correo electrónico	universitaria98.5@unl.edu.ec	Requerido
Gorjeo	-----	Opcional
Teléfono	072545055	Requerido
Eslogan	Radio Universitaria la académica de Loja	Requerido
Descripción	Radio Universitaria 98.5 es un medio de comunicación público de la Universidad Nacional de Loja, para comunicarse con la comunidad a través de la coordinación, cooperación, consulta, intercambio y promoción del arte, la ciencia, la cultura y el desarrollo de Loja y la Región Sur.	Requerido

tunein.com/syndication/new/?isBroadcaster=true

Dell Importado de Internet My Support Requests Iniciar sesión en Office Certificaciones de TI

tunein

Si desea instrucciones sobre cómo añadir su podcast de TuneIn, por favor visite [aquí](#).

Lo Esencial

Su nombre de la emisora: *(requerido)*
Radio Universitaria 98.5

Su correo electrónico: *(obligatorio)*
apps.moviles@unl.edu.ec

Su página web:
http://unl.edu.ec/radio/inicio-radio-universitaria

URL de la transmisión: *(requerido)*
http://radio.unl.edu.ec:8080/stream

Haga clic [aquí](#) para obtener más información.
+ Añadir otra corriente

Figura 102. Formulario para agregar una nueva estación

tunein.com/syndication/new/?isBroadcaster=true

es Dell Importado de Internet My Support Requests Iniciar sesión en Office Certificaciones de TI

tunein

Tipo De Transmisión:
 Sólo Internet AM / FM / TV

país:
Ecuador

idioma:
español

formatos:
MP3

Tu logo:
Seleccionar archivo Logo Radio Universitaria vertical.png

Estación de contactos de correo electrónico:
universitaria98.5@unl.edu.ec

Gorjeo:

Teléfono:
072545055

Eslogan:
Radio Universitaria la académica de Loja

Breve descripción, 64 caracteres como máximo

Figura 103. Formulario para agregar una nueva estación

tunein.com/syndication/new/?IsBroadcaster=true

MP3

Tu logo:
 Logo Radio Universitaria vertical.png

Estación de contactos de correo electrónico:
 universitaria98.5@unl.edu.ec

Gorjeo:
 https://twitter.com/RadioUNL

Teléfono:
 072545055

Eslogan:
 Radio Universitaria la académica de Loja

Descripción:
 Radio Universitaria 98.5 es un medio de comunicación público de la Universidad Nacional de Loja, para comunicarse con la comunidad a través de la coordinación, cooperación, consulta, intercambio y promoción del arte, la ciencia, la cultura y el desarrollo de Loja y la Región Sur.

700 caracteres como máximo: 699 restante

Figura 104. Formulario para agregar una nueva estación

tunein.com/syndication/new/?IsBroadcaster=true

comentarios:

[Aceptar Los Terminos](#)

Por favor marque esta casilla para indicar que quiere que tengamos su estación en la base de datos y entendemos nuestras condiciones.

Acepto y acepto los [Términos y condiciones](#) .

Figura 105. Envío del Formulario

tunein.com/syndication/Complete/

¡Gracias!

Su Propuesta Ha Sido Recibida

Esto es lo que sucede a continuación:
 Nuestro equipo está revisando su presentación y le enviará un correo electrónico de confirmación dentro de 3-5 días laborales.
 Una vez que los cambios son aprobados, por favor de hasta 24 horas antes de que aparezca su estación en Tunein.
 Por favor, póngase en contacto con broadcaster-support@tunein.com si no ha recibido un correo electrónico de confirmación después de 3-5 días hábiles.

Consejos Útiles Una Vez Que Su Estación Está En La Lista

Mantenga su estación actualiza.
 Enviar información permanente para la estación, espectáculo, o la canción actual con nuestra API Transmisor.
 Si usted tiene mayor estación cambia enviar un correo electrónico a Transmisor de apoyo .

Promúevase
 Si usted está interesado en la promoción de su estación a través de Tunein para aumentar su distribución, por favor, póngase en contacto con Transmisor de apoyo .

¿Preguntas adicionales?
 Por favor, visite nuestro [CENTRO DE AYUDA](#)

Figura 106. Mensaje de finalización de registro de nueva estación

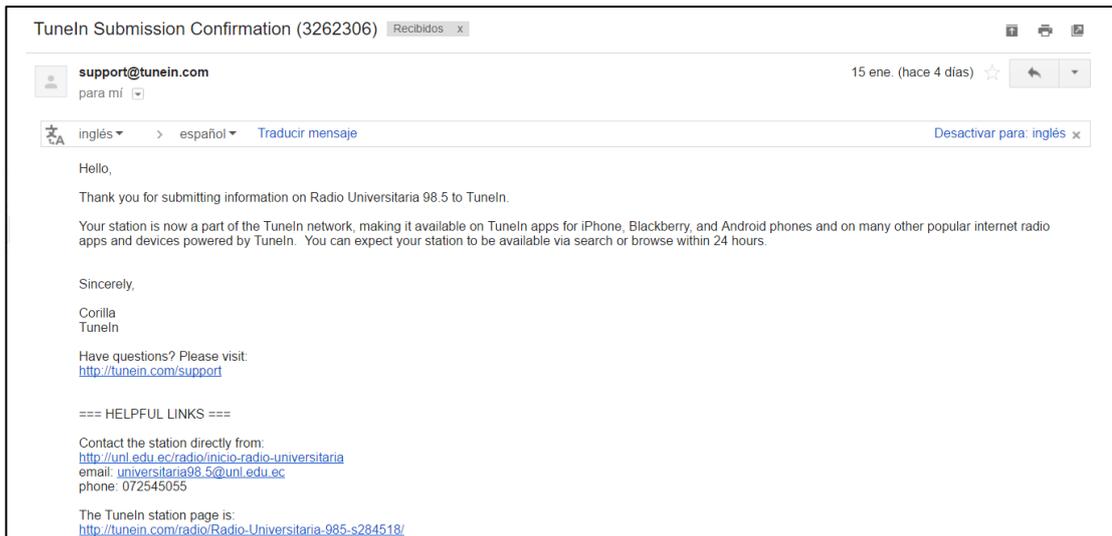


Figura 107. Correo electrónico de aceptación de la estación de radio en TuneIn

Paso 5: Ingreso al streaming mediante la plataforma TuneIn

Este se lo realiza para constatar e ingresar al streaming en vivo de la Radio Universitaria para ello ingresamos a la página principal de TuneIn, una vez dentro de la página principal procedemos a buscar la radio para ello la buscamos por el nombre **Radio Universitaria 98.5** la cual nos enviara a la página para reproducir el streaming de la radio o directamente podemos ingresar mediante la siguiente dirección <http://tunein.com/radio/Radio-Universitaria-985-s284518/> y reproducir el streaming.

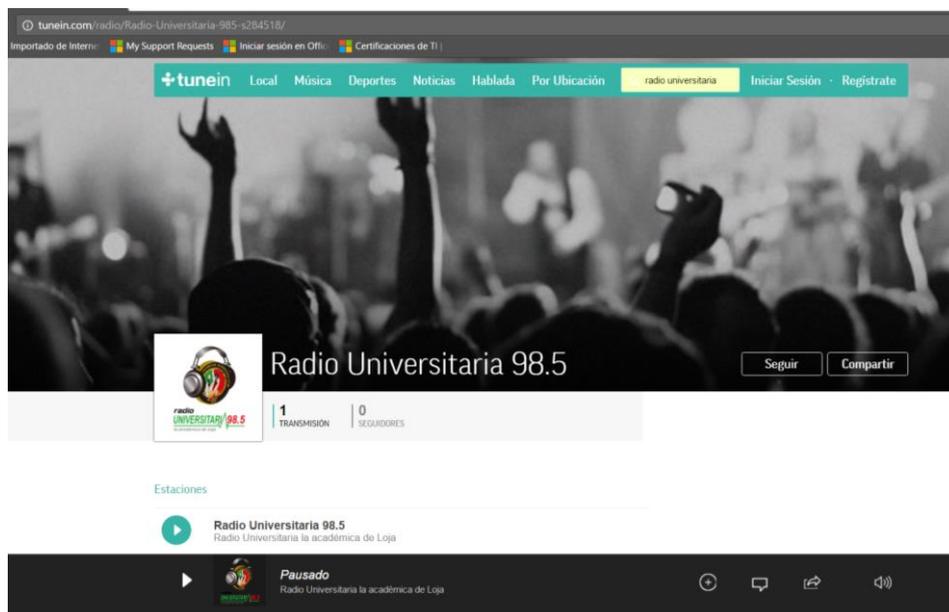


Figura 108. Radio Universitaria en plataforma TuneIn

ANEXO 7

Emulación de Aplicación en Sistema Operativo iOS

Para poder realizar las pruebas de la aplicación móvil en el sistema operativo iOS se utiliza una máquina virtual con sistema operativo Mac, en la cual se emula un dispositivo iPhone, para verificar el correcto funcionamiento de la aplicación.

Paso 1: Iniciar máquina virtual

Se inicia la máquina virtual con sistema operativo Mac OS X 10.11, donde tenemos instalado todas las dependencias que se necesita como son: Ionic y Xcode, primordiales para poder realizar la emulación.

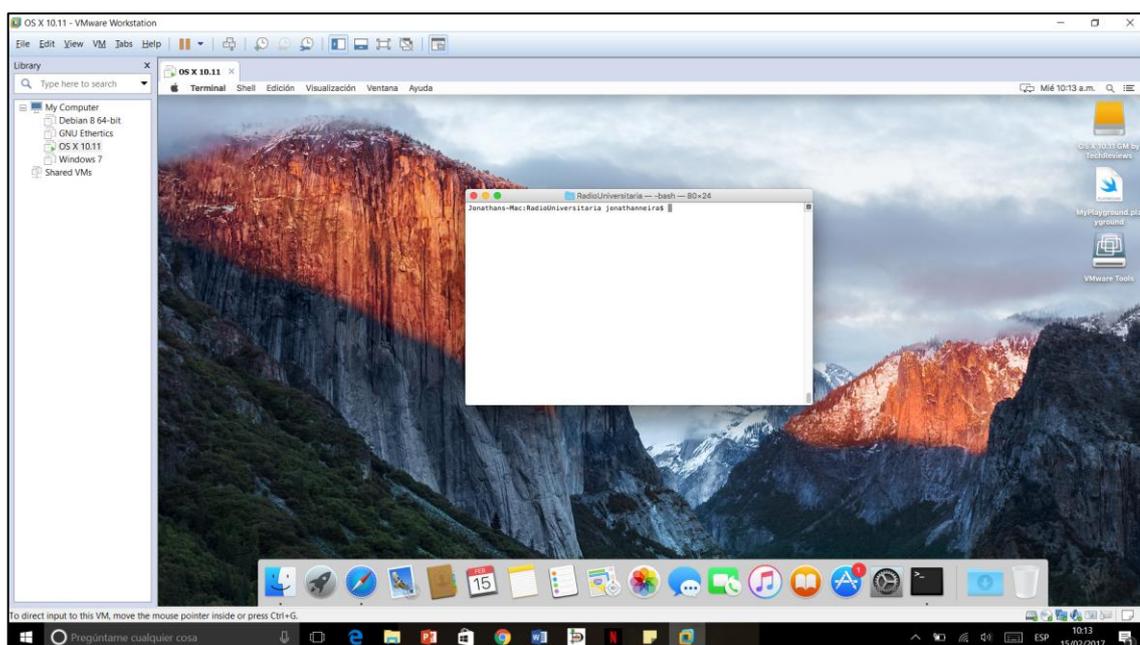


Figura 109. Máquina virtual Mac OS X10.11

Paso 2: Construir y emular la aplicación en iOS

Para construir y emular la aplicación en iOS, se procede a abrir una terminal y nos dirigimos hacia la ruta de nuestro proyecto a continuación, construimos el proyecto para el sistema operativo iOS, para el cual digitamos el comando **ionic build ios**, una vez construido el proyecto, se procede a emular, para ello digitamos el comando **ionic emulate ios**

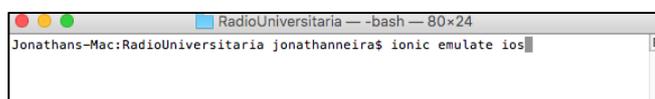


Figura 110. Comando para emular en iOS

Una vez iniciada la emulación este despliega la pantalla de un iPhone SE 10.1, donde automáticamente se inicia la aplicación de la Radio Universitaria.

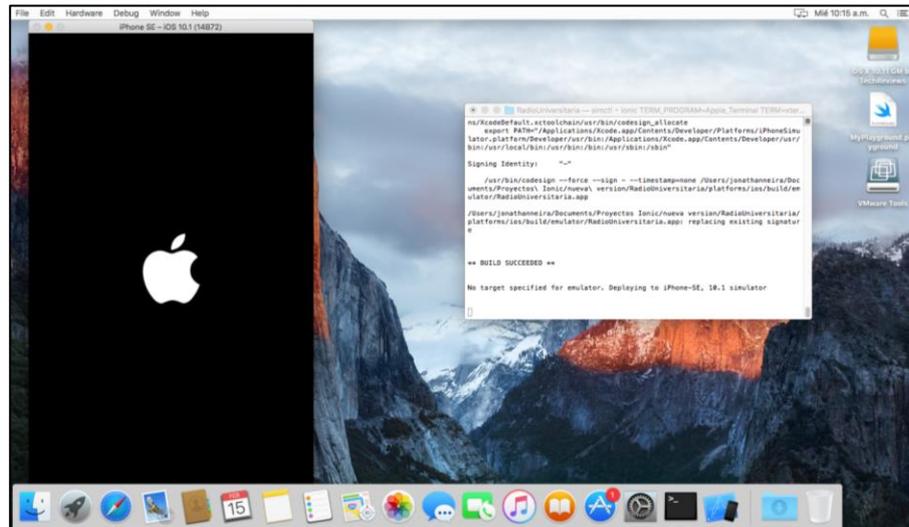


Figura 111. Inicio de emulación en iOS



Figura 112. Pantalla principal Online

Paso 3: Probar funciones de la aplicación

Una vez ya emulada la aplicación se procede a probar todas las funciones de la aplicación, en donde en primera instancia se comprueba el streaming, seguido de las demás funcionalidades como son: visualización de información quienes son la radio, parrilla de programación, visualización de las redes sociales y pagina institucional.



Figura 113. Ejecución de streaming

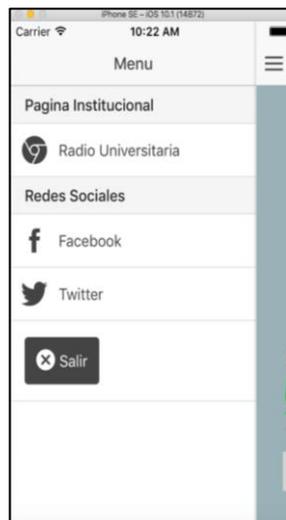


Figura 114. Submenú de aplicación



Figura 115. Visualización de red social Facebook



Figura 116. Visualización de información Quienes Somos



Figura 117. Parrilla de programación



Figura 118. Programación diaria



Figura 119 Noticias

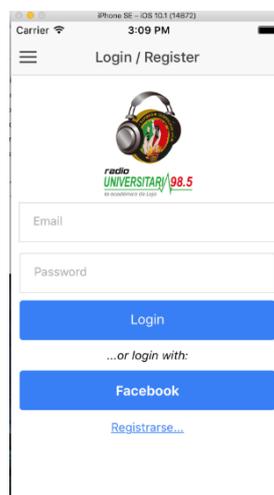


Figura 120 Login/Register

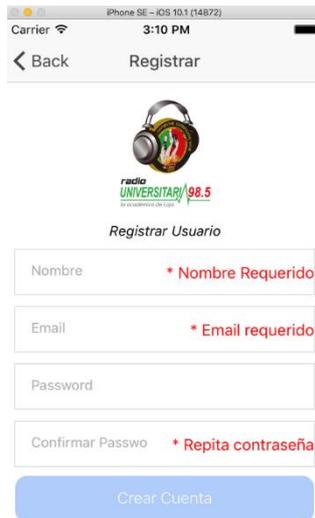


Figura 121 Registrar nuevo usuario



Figura 122 Logueo con red social Facebook

En esta emulación se comprueba que la aplicación cumple con todas sus funciones para la cual fue creada.

ANEXO 8

Publicar la aplicación en Google Play

Para publicar la aplicación en Google Play se deben de seguir los siguientes pasos:

Paso 1: Exportar y firmar la aplicación

Para exportar y firmar la aplicación se ha seguido el proceso de publicación que se encuentra en la documentación de Ionic para ello se debe seguir los pasos que se encuentra en la siguiente dirección [<https://ionicframework.com/docs/guide/publishing.html>].

Paso 2: Acceder a la consola de Google Play para desarrolladores.

Para acceder a la consola de Google Play se requiere tener cuenta en Gmail y acceder a la siguiente dirección [<https://play.google.com/apps/publish>]. Al ingresar se debe iniciar sesión con la cuenta de Gmail y se mostrara una pantalla como se indica a continuación.



Figura 123. Registrar cuenta de desarrollador en Google Play

En la pantalla mostrada en la figura 106 se debe aceptar las condiciones y proceder a llenar los datos de la tarjeta de crédito para realizar el pago correspondiente, luego se debe llenar algunos datos de la cuenta para completar el registro. Finalmente se mostrará una pantalla como se indica en la figura 107, en donde ya se puede proceder a publicar la aplicación.

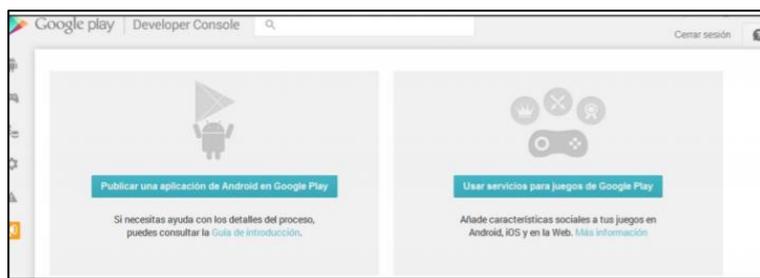


Figura 124. Finalización del registro en la consola de Google Play

Paso 3: Publicar la aplicación

Para publicar la aplicación presionamos el botón **Publicar una aplicación de Android en Google Play**, ubicado en la pantalla mostrada en la figura 124. A continuación, se presenta una pantalla para ubicar el nombre de la aplicación

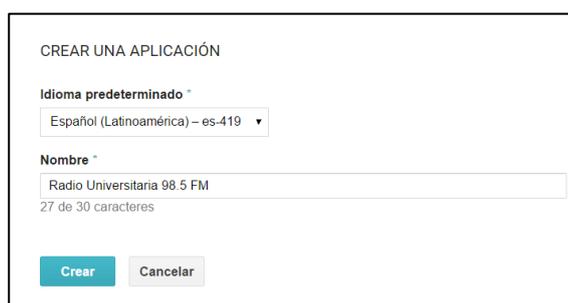


Figura 125. Añadir nueva aplicación Android

Luego presionamos el botón **Subir APK**, mostrándose la pantalla indicada en la figura 126.



Figura 126. APK para producción

Seguidamente se presiona el botón **Subir tu primer archivo APK en fase de producción**, y se procede a subir el archivo APK previamente firmado.



Figura 127. Subir APK de producción

Luego de haber subido el APK, se procede a llenar la ficha de la aplicación.

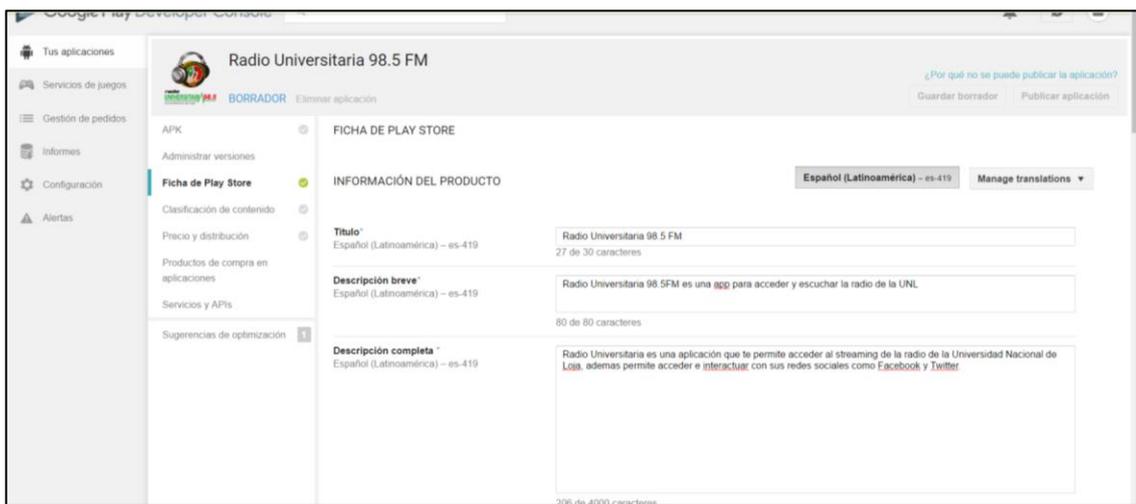


Figura 128. Ficha de Play Store

Finalmente se procede a publicar la aplicación y en el transcurso de una hora la aplicación se muestra en Google Play.

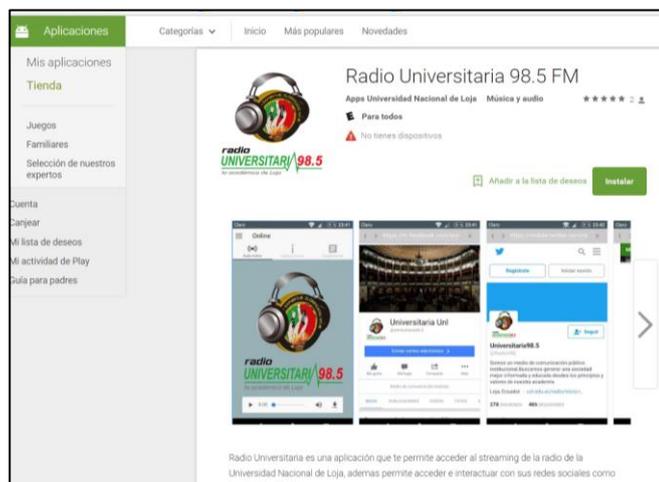


Figura 129. Aplicación publicada en Google Play

ANEXO 9

Proyecciones en HW y SW para la inclusión de nuevos usuarios (escalabilidad)

La escalabilidad se define como la facilidad con que una solución puede crecer a medida que surja la necesidad.

Se entiende por escalabilidad a la capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan de forma significativa el número de usuarios del mismo. Aunque parezca un concepto claro, la escalabilidad de un sistema es un aspecto complejo e importante del diseño.[37]

La escalabilidad está íntimamente ligada al diseño del sistema. Influye en el rendimiento de forma significativa. Si una aplicación está bien diseñada, la escalabilidad no constituye un problema.

La escalabilidad supone un factor crítico en el crecimiento de un sistema. Si un sistema tiene como objetivo crecer en el número de usuarios manteniendo su rendimiento actual, tiene que evaluar dos posibles opciones:

- Con un hardware de mayor potencia o
- Con una mejor combinación de hardware y software.

Se pueden distinguir dos tipos de escalabilidad, vertical y horizontal:

- El escalar verticalmente o escalar hacia arriba, significa el añadir más recursos a un solo nodo en particular dentro de un sistema, tal como el añadir memoria o un disco duro más rápido a una computadora.



Figura 130 Escalamiento Horizontal

Fuente:<https://www.oscarblancarteblog.com/2017/03/07/escalabilidad-horizontal-y-vertical/>

- La escalabilidad horizontal, significa agregar más nodos a un sistema, tal como añadir una computadora nueva a un programa de aplicación para espejo.

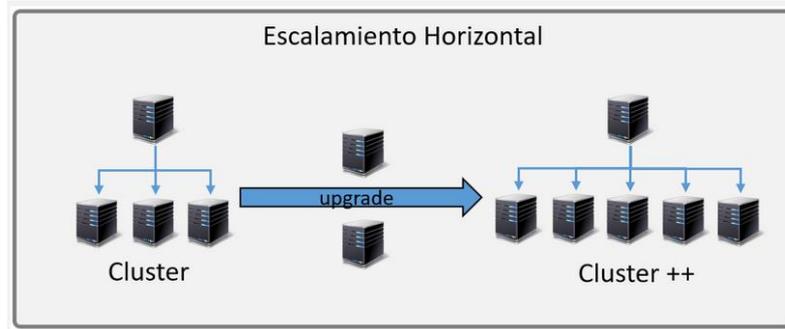


Figura 131 Escalamiento Horizontal

Fuente: <https://www.oscarblancarteblog.com/2017/03/07/escalabilidad-horizontal-y-vertical/>

Ventajas y desventajas de escalabilidad vertical

Ventajas

- No implica un gran problema para las aplicaciones, pues todo el cambio es sobre el hardware
- Es mucho más fácil de implementar que el escalamiento horizontal.
- Puede ser una solución rápida y económica (compara con modificar el software)

Desventajas

- El crecimiento está limitado por el hardware.
- Una falla en el servidor implica que la aplicación se detenga.
- No soporta la Alta disponibilidad.
- Hacer un upgrade del hardware al máximo puede llegar a ser muy caro, ya que las partes más nuevas suelen ser caras con respecto al rendimiento de un modelo anterior.

Ventajas y desventajas de escalabilidad vertical

Ventajas

- El crecimiento es prácticamente infinito, podríamos agregar cuantos servidores sean necesarios
- Es posible combinarse con el escalamiento vertical.
- Soporta la alta disponibilidad
- Si un nodo falla, los demás sigue trabajando.

- Soporta el balanceo de cargas.

Desventajas

- Requiere de mucho mantenimiento
- Es difícil de configurar
- Requiere de grandes cambios en las aplicaciones (si no fueron diseñadas para trabajar en cluster)
- Requiere de una infraestructura más grande

Red Hat en su página oficial menciona que hace mejoras a Red Hat Enterprise Linux tanto para escalabilidad horizontal como para escalabilidad vertical; sin embargo, la escalabilidad horizontal suele ser el caso de uso más aplicable. La idea detrás de la escalabilidad horizontal es usar múltiples computadores estándar para distribuir cargas de trabajo pesadas con el fin de mejorar el rendimiento y la confiabilidad. [38]

Proyecciones software

Nombre	Estado actual del software	Proyección	Valor	Dependencia
Butt Broadcast	En ejecución, con capacidad de enviar gran capacidad de flujo de audio con un bitrate de 128 kbps.	Mantener el bitrate de audio en 128 kbps para seguir obteniendo un audio de salida de calidad.	Gratis	-----
Servidor Icecast2	En ejecución con capacidad de soportar 943 usuarios concurrentes en el streaming.	Aumentar la capacidad de usuarios concurrentes al streaming.	Licencia Gratuita del software Icecast2	Depende del ancho de banda que se disponga para el streaming y características hardware del servidor streaming.

Proyecciones Hardware

Equipo	Estado actual del hardware	Características Hardware actual	Proyección	Características de proyección	Valor
Servidor Radio Universitaria	Funcionando eficazmente con la capacidad de soportar un numero de 943 usuarios concurrentes, consumiendo un 66.50% de CPU, 0.90% de memoria RAM.	Procesador 1 GHz Memoria RAM 1GB Disco Duro 10 GB Sistema Operativo GNU Linux Debian	Se proyecta tener un hardware con una capacidad de soportar el doble de usuarios que actualmente soporta, además de almacenar audios entrevistas y programas grabados.	Procesador 2 GHz Memoria RAM 2 GB Disco Duro 40 GB Sistema Operativo GNU Linux Debian	\$2000 "Cotización por encargado de la Subdirección del departamento de la Unidad de Telecomunicaciones e Información (UTI)"
Cliente Procesador de señal digital (DSP)	Funcionando eficazmente para la codificación y envío de audio hacia el servidor.	Procesador 3.40 GHz Memoria RAM 2 GB Disco Duro 80 GB Sistema Operativo GNU Ethertics	En este aspecto no se pretende realizar una proyección del hardware ya que el equipo actual está dedicado únicamente al streaming es decir a la	-----	-----

			codificación y envió de audio al servidor por lo que no consume muchos recursos hardware.		
--	--	--	---	--	--

Conclusión

Se determina que la proyección de software Icecast2 depende del valor del hardware que lo contenga a dicho software no obstante cabe recalcar que el software Icecast2 tiene una licencia GPL por lo que tiene un costo gratuito en el mercado. No obstante, sucede lo contrario con la proyección de hardware donde si existe un valor del equipo del servidor, donde se toma en cuenta que debe adaptarse a la plataforma Blade y con las siguientes características memoria RAM 2GB, Procesador de 2GHz y Disco Duro de 40GB el precio aproximado es de \$2000 cotizado por personal del departamento de la Unidad de Telecomunicaciones e Información (UTI).

Análisis de costo para transmisión de Streaming de Radio utilizando servicios de Amazon

Precios de transferencia de datos regionales al origen (por GB)

Transferencias de datos regionales al origen (por GB)								
	Estados Unidos	Canadá	Europa	Hong Kong, Filipinas, Corea del Sur, Singapur y Taiwán	Japón	América del Sur	Australia	India
Transferencia de todos los datos	0,020 USD	0,020 USD	0,020 USD	0,060 USD	0,060 USD	0,125 USD	0,100 USD	0,160 USD

Figura 132 Precios de transferencia de datos regionales al origen (por GB) en Amazon

Fuente: <https://aws.amazon.com/es/cloudfront/pricing/>

Análisis de transferencia actual en Mbps									
	Conexiones	Ancho de Banda Mbps	CPU %	Memoria %	Transferencia de datos en 1 min Mbps	Transferencia de datos en 1 hora Mbps	Transferencia en 1 día Mbps	Transferencia en 1 mes Mbps	Transferencia en 1 año Mbps
ACTUAL	943-1418	92,3	66.5 -100	0,9	5538	332280	7974720	247216320	2966595840

Análisis de transferencia actual en Gbps y Costo					
Conexiones	Transferencia de datos en 1 min Gbps	Transferencia de datos en 1 hora Gbps	Transferencia en 1 día Gbps	Transferencia en 1 mes Gbps	Transferencia en 1 año Gbps
943-1418	5538	332280	7974720	247216320	2966595840
Costo	0,6760253906	40,56152344	973,4765625	30177,77344	362133,2813

Análisis de transferencia en Mbps para Escalabilidad Duplicando conexiones									
	Conexiones	Ancho de Banda Mbps	CPU %	Memoria %	Transferencia de datos en 1 min Mbps	Transferencia de datos en 1 hora Mbps	Transferencia en 1 día Mbps	Transferencia en 1 mes Mbps	Transferencia en 1 año Mbps
Escalabilidad	1866-2836	184,6	66.5 -100	0,9	11076	664560	664560	664560	664560

Análisis de transferencia en Gbps y Costo para la escalabilidad Duplicando Conexiones					
Conexiones	Transferencia de datos en 1 min Gbps	Transferencia de datos en 1 hora Gbps	Transferencia en 1 día Gbps	Transferencia en 1 mes Gbps	Transferencia en 1 año Gbps
1866-2836	10,81640625	10,81640625	10,81640625	10,81640625	10,81640625
Costo	1,352050781	81,12304688	1946,953125	60355,54688	724266,5625

Conclusión del análisis de costos:

Tomando en cuenta la cantidad de transferencia de datos que se produce al realizar streaming y el costo que genera realizar dicha transferencia de datos se cree conveniente comprar equipos físicos para la entrega del servicio ya que, realizando una comparación de costos con un servicio alquilado como Amazon el costo anual para la Universidad sería de \$ 724.266,5625, resultando ser demasiado costoso en relación con equipos físicos que costarían \$ 2.000 los cuales pueden cumplir o abastecer la misma capacidad de conexiones simultaneas para el streaming de la Radio Universitaria.

ANEXO 10

Entrevista Radio Universitaria

Nombre: Ing. Paulina Jara

Institución Laboral: Radio Universitaria

Cargo: Secretaria con funciones de Manejo Administrativo de la Radio

Objetivo de la entrevista: Recabar información de la situación actual de equipos presentes en la radio e información adicional de la misma.

Esta entrevista se dio lugar el día 02 de mayo de 2016 en las instalaciones de la Radio Universitaria conjuntamente con la secretaria de la radio la Ing. Paulina Jara, el objetivo de esta entrevista fue recabar información de la situación actual de equipos presentes en la radio e información adicional de la misma, a continuación, se redacta la entrevista antes mencionada.

La radio Universitaria se crea de un proyecto titulado “Implementación de medios para la Universidad Nacional de Loja”, la radio nace dentro del centro Radio Cine y Televisión que antes funcionaba, el cual tenía dos brazos ejecutores tanto la Radio Universitaria como la imagen misma, la escritura de la radio fue entregada luego de los papeleos correspondientes el 27 de Julio del 2000 sin embargo no se inaugura de manera inmediata por cuestiones de requerimientos posteriores que exigía en ese entonces CONATEL, dicha entidad en la actualidad ya no existe, de esta manera la radio se inaugura formalmente el 4 de Diciembre del 2001, en el edificio del CINFA en su último piso, con la presencia Lic. Milton Andrade Tapia director en ese momento, estas instalaciones en la actualidad pertenecen al Área Agropecuaria de la universidad antes mencionada, en la actualidad la radio está funcionando en las instalaciones del ex Cater de la Universidad Nacional de Loja.

El representante legal de la radio es el Rector de la Universidad Nacional de Loja, pero se ha realizado un comunicado que la radio está contemplada en el departamento de Comunicación e Imagen Institucional, actualmente en la Radio Universitaria laboran dos personas el Sr. Stewart Naranjo y la Ing. Paulina Jara las mismas que administran la radio.

La cobertura de la radio es en la ciudad de Loja y sus alrededores, pero existió la posibilidad de expandir su cobertura con cuatro repetidoras en el año 2008 y así de esta manera abarcar cuatro provincias, sin embargo, por motivos internos y por autoridades administrativas anteriores que dirigían la radio no se pudo culminar con la expansión de la cobertura, pero cabe recalcar que existieron los estudios necesarios para su realización.

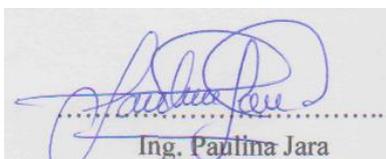
La radio cuenta con el servicio de Internet, pero existen problemas por estar un poco distante del campus universitario de la misma manera, la energía eléctrica no es estable por lo que existen bajas de voltaje afectando a los equipos y a la misma vez generando intermitencia en Internet, pero en la actualidad se está mejorando la estabilidad del Internet.

El medio de difusión digital para dar a conocer su imagen y el servicio que presta la radio es la página web institucional de la Universidad Nacional de Loja en la sección de servicios, adicional a esto se hace uso de las cuentas oficiales de la Radio Universitaria en Facebook y Twitter.

El streaming de la radio se perdió desde el momento que se cambiaron las instalaciones de la radio desde sus anteriores oficinas hacia las actuales ocurriendo esto desde la fecha del 4 de febrero del 2011, desde ese instante se realizó la petición correspondiente a los directivos de la Unidad de Telecomunicaciones sin obtener una respuesta satisfactoria a la petición de este servicio que es muy importante debido a la limitante en capacidad de cobertura de dicha radio.

Para la producción de la radio existen dos computadoras Mac de Apple 1419 Core i7 última generación, dos consolas de 30 canales, dos computadoras con tecnología un poco desfasada que sirven para producción y para la práctica de estudiantes, una consola digital ip con la posibilidad de administración remota mediante internet cuenta con 8 canales con 22 entradas marca Axia compuesta por un panel de control modelo iQ 8 MEN, discos duros externos, ups americano para la variación de voltaje que se presenta, audífonos, brazos, micrófonos profesionales marca electro voice modelo RE-20, antena Yagi para transmisiones en vivo, procesador de audio marca Omnia modelo 4.5 estéreo, equipos de enlace, etc., estas son las últimas adquisiciones que cuenta la radio y con las cuales se está saliendo en producción al aire para todos los radioescuchas.

Programas adicionales para producción se utiliza Pro Tools para ediciones de audio, creaciones de cuñas, etc. Así mismo como alternativa en presencia de fallos se utiliza programa de Zara Radio para salir con la transmisión al aire.



Ing. Paulina Jara

ANEXO 11

Cuestionario Realizado a Subdirector UTI

Nombre: Ing. Jhon Calderón

Institución Laboral: Unidad de Telecomunicaciones e Información

Cargo: Subdirector

Objetivo de la entrevista: Conocer la arquitectura de red, ancho de banda y otra información afine a la misma con la que cuenta la Radio Universitaria.

1. ¿Con que arquitectura de red cuenta la Universidad Nacional de Loja y específicamente las instalaciones de la Radio Universitaria?

Es un modelo jerárquico de 3 capas: Capa de núcleo, capa de distribución y capa de acceso, en el caso de la Radio Universitaria se encuentra equipos de red que operan en la capa de acceso.

2. ¿Con que ancho de banda cuenta la radio?

La Universidad cuenta con 300Mbps de Internet comercial y 1Gbps de Internet avanzado, este ancho de banda se encuentra distribuido en todo el campus Universitario. Cabe indicar que no existe segmentación de AB para la Radio Universitaria.

3. ¿Se han presentado problemas en la red de la radio?

Sí No

Que problemas ha tenido:

Los inconvenientes presentados se han debido a que los equipos de red no cuentan con un respaldo de energía para los equipos de red.

4. Si se contestó SI en la pregunta 3 Contestar: ¿En qué tiempo se han solucionado dichos problemas?

Entre 1 a 60 minutos (X)

Entre 1 a 2 horas ()

Más de 2 horas ()

Entre 1 a 2 días ()

Más de 2 días ()

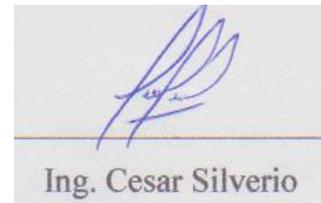
5. En qué sistema operativo corren los servidores que cuenta la unidad de Telecomunicaciones e Información.

Debían (x)

Centos (x)

Windows ()

Otros: _____



ANEXO 12

Certificación Unidad de Telecomunicaciones e Información



UNL
UNIVERSIDAD
NACIONAL
DE LOJA

*Unidad de
Telecomunicaciones e
Información*

Milton Leonardo Labanda Jaramillo Mg.
DIRECTOR DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN.

CERTIFICA.

Que el Señor estudiante **JONATHAN EZEQUIEL NEIRA BALCAZAR** con cédula de Identidad. 1104835945, cuyo proyecto de titulación versa sobre el tema **"IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES "**, está implementado y en funcionamiento dentro de nuestra Infraestructura cumpliendo con los lineamientos dispuestos por nuestra Unidad Administrativa

LO CERTIFICO, para los fines pertinentes.

Loja, 5 de enero del 2017

Ing. Milton Labanda
DIRECTOR T.I.



ANEXO 13

Certificación de la Radio Universitaria 98.5



UNIVERSIDAD NACIONAL DE LOJA
ESTACIÓN DE RADIO Y TELEVISIÓN UNIVERSITARIA 98.5

La Radio Académica de Loja

Ingeniera.
Paulina Lourdes Jara Delgado
SECRETARIA DE LA ESTACIÓN
DE RADIO Y TELEVISIÓN UNIVERSITARIA 98.5

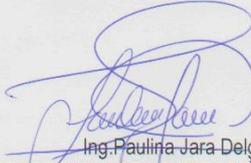
CERTIFICO:

Que el Señor estudiante **JONATHAN EZEQUIEL NEIRA BALCÁZAR** portador de la cédula de Identidad N° **1104835945**, cuyo sistema titulado "IMPLEMENTACIÓN DEL SERVICIO DE RADIO DE LA UNIVERSIDAD NACIONAL DE LOJA PARA LOS MEDIOS DE DIFUSIÓN DIGITAL Y DISPOSITIVOS MÓVILES", esta implementado y en funcionamiento dentro de las instalaciones de la Radio Universitaria 98.5FM cumpliendo con las siguientes características:

- Streaming de la radio online
- Administración de contenido de la aplicación móvil
 - Gestión de Noticias
 - Gestión de Programas (Parrilla de Programación)
 - Gestión de Programas grabados
 - Información de la radio (Contactos, correos, etc) y
 - Gestión de perfil de usuario

Lo certifico en honor a la verdad, facultando al interesado hacer uso del presente en lo que estime pertinente.

Loja noviembre 9, 2017


Ing. Paulina Jara Delgado
SECRETARIA DE LA ESTACIÓN DE RADIO Y
TELEVISIÓN UNIVERSITARIA 98.5



ANEXO 14

Especificación de requisitos de software

Proyecto:

Implementación del servicio de radio de la Universidad Nacional de Loja para los medios de difusión digital y dispositivos móviles.

1. Introducción

Una radio en Internet dentro de una universidad permite amplias posibilidades de intercambio de información, no solo dando a conocer acontecimientos pertenecientes a la institución, sino generar un ambiente de colaboración donde se puedan compartir contenidos para fomentar la parte cultural y educativa, cumpliendo un rol fundamental en la difusión del conocimiento y de esta manera tener una relación continua con la colectividad en general.

La Radio de la Universidad Nacional de Loja en los últimos años ha contribuido mucho en la parte de difusión de acontecimientos institucionales, así como también tener una relación continua con la colectividad es por ello que se plantea el desarrollo de un SISTEMA DE ADMINISTRACIÓN WEB PARA LA APLICACIÓN MOVIL DE LA RADIO UNIVERSITARIA (SWAR).

Esta herramienta basada en web facilitará a personal de la Radio gestionar sus programas (parrilla de programación) y audios de programas o entrevistas grabadas.

1.1. Propósito

El presente documento tiene como propósito definir de forma clara y precisa todos los requisitos y restricciones que posee el Sistema web para la administración de la Radio Universitaria.

1.2. Alcance

Esta especificación de requisitos está dirigida para personal de la radio de la Universidad Nacional de Loja. El sistema permitirá a los usuarios gestionar los programas y horarios (parrilla de programación) y audios grabados de programas, entrevistas, noticias, etc.

1.3. Personal Involucrado

Nombre	Jonathan Ezequiel Neira Balcázar
Rol	Desarrollador de Software
Categoría Profesional	Ingeniero en Sistemas
Información de Contacto	jeneirab@unl.edu.ec

1.4. Definiciones, Acrónimos y Abreviaturas

Usuario: Persona que usará el sistema para gestionar la radiO

AUP: Proceso Unificado Ágil, Metodología de desarrollo de software

ERS: Especificación de Requerimientos de Software

API: Interfaz de aplicaciones de Programación

REST: Representational State Transfer

PHP: Hypertext Preprocessor (Lenguaje de Programación)

Codeigniter: Framework de aplicaciones web basado en PHP

Javascript: Lenguajes de Programación

Angular: Framework web basado en Javascript

Responsive Web Design: Enfoque para la construcción de aplicaciones web que respondan a las diferentes resoluciones de dispositivos.

1.5. Referencias

Título de Documento	Referencia
Standard IEEE 830-1998	IEEE

1.6. Resumen

El documento presente consta de tres secciones. Esta sección es la introducción y proporciona una visión general del ERS. En la segunda sección se provee una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados, y los factores, restricciones de la administración de la aplicación móvil para la radio.

La sección tres presenta detalladamente los requisitos que deben satisfacer las necesidades

2. Descripción General

A continuación, vamos a ver los factores que afectan al producto y requerimientos.

2.1. Perspectiva del Producto

Lo que se necesita es poder administrar aspectos como programas (parrilla de programación) y audios grabados de entrevistas, programas, etc., como solución a ello el Módulo de Administración estará desarrollado en un entorno web.

Este sistema Interactuará con datos extraídos de la página oficial de la Universidad Nacional de Loja la cual cuenta con información necesaria para poder ser consumida por la aplicación móvil de la radio.

2.1.1. Interfaces del sistema

El módulo de administración estará desarrollado en un entorno web construido con el lenguaje de programación PHP usando Codeigniter como framework de aplicación esta tecnología estará del lado del servidor, mientras que del lado del cliente se usará javascript usando el framework Angular, además de las tecnologías HTML5, CSS3.

2.2. Funcionalidad del Producto

El sistema para la administración de la aplicación móvil de la radio universitaria deberá proporcionar las siguientes tareas:

Gestión de Usuarios

Para las cuentas y datos de tanto del administrador y clientes de la aplicación móvil.

Gestionar los programas y sus horarios (parrilla de programación)

El sistema proporcionará a los usuarios administración de los programas y horarios (parrilla de programación). Para poder administrarlo se necesita la información del programa como título, días que se emite el programa y la hora en cada día que se emite el programa.

Gestionar los audios de programas, entrevistas, noticias, etc., grabados para que estén disponible en la aplicación móvil.

El sistema proporcionará a los usuarios administración de los audios de programas y entrevistas, noticias, etc. Para poder administrarlo se necesita la información del audio como su título y el archivo de audio en formato mp3.

2.3. Características de los Usuarios

Tipo de Usuario	Administrador del Sistema
Formación	Persona con nivel técnico encargado de manejar el sistema
Habilidades	Utilización del equipo informático (ordenador)
Actividades	- Maneja la información del sistema

2.4. Restricciones

Para el desarrollo del sistema web se usará la metodología AUP ya que provee un desarrollo ágil en la construcción del sistema y que es óptima debido al tiempo y al cambio de requisitos que podría surgir.

Para la programación del lado del servidor (Backend) se usará el framework Codeigniter y javascript, mientras que del lado del cliente la aplicación móvil (Frontend) se usará el framework Ionic el cual consta de Html5, Css3, y Angular.js

Para la instalación del presente sistema en un servidor de aplicaciones se necesita los requerimientos siguientes

Requisitos del Hardware	
Nombre	Requisitos Mínimos
Procesador	Procesador Intel
Velocidad	1.6 GHz o superior
Caché	512 MB
Sistema Operativo	Debian, Centos

Requisitos del Software	
Nombre	Requisitos Mínimos
Base de Datos	Postgresql o MySQL
Servidor Web	Apache

Interprete de Ordenes	SSH
Sistema Operativo	Debian

2.5. Suposiciones y Dependencias

Los requisitos del sistema se basan en ciertos lineamientos sugeridos por la Universidad Nacional de Loja los cuales para el correcto funcionamiento habrán sido evaluados.

2.6. Evolución previsible del Sistema

El sistema brindará diferentes facilidades para la integración de nuevas funcionalidades tanto en la parte del Backend como del Frontend en el futuro el mismo que escalará y evolucionará con facilidad.

3. Requisitos Específicos

3.1 Requisitos Comunes de las interfaces

3.1.2. Interfaces de usuario

Al realizarse un Backend para la administración de la aplicación móvil de la radio universitaria se establece un estándar de diseño y usabilidad que permita una gran experiencia con el usuario con el fin que sea fácil de usar y sea atractivo al usuario. Para el desarrollo de la interfaz se usará HTML5, CSS3 y Bootstrap con Material Design.

El usuario interactuará con las interfaces del sistema por medio del mouse, teclado para la entrada de datos y pantalla para la salida de datos.

Las interfaces de usuario contarán con los siguientes componentes:

- Texto legible
- Áreas de texto para entrada de datos
- Cuadros de diálogo
- Paneles
- Formularios
- Botones interactivos
- Menús desplegables
- Imágenes

3.1.3. Interfaces de Hardware

Monitor	El sistema deberá mostrar información de la administración del contenido de la aplicación móvil en cuanto a programación y audios de programas, entrevistas, noticias, etc.
Teclado	Ingresar datos en el sistema por parte de los usuarios
Mouse	Ayudará al usuario a interactuar con diferentes componentes del sistema como botones, links, menús, etc.

3.1.4. Interfaces de Comunicación

El sistema se comunicará con la base de datos MySQL mediante el Framework Codeigniter. Además, para la construcción de la aplicación se hará uso de servicios REST del propio Framework Codeigniter el cual permitirá la comunicación con la aplicación móvil de la radio.

3.2. Requisitos Funcionales

El sistema permitirá:

TABLA LXVI. REQUISITO FUNCIONAL 001

Número del requisito	RF001
Nombre del Requisito	Gestionar perfil de usuario
Descripción del requisito	Al usuario tendrá un perfil el cual contendrá sus datos personales además de su usuario y contraseña.
Prioridad del Requisito	Alta

TABLA LXVII. REQUISITO FUNCIONAL 002

Número del requisito	RF002
Nombre del requisito	Autenticar Usuario
Descripción del requisito	Un usuario deberá poder ingresar a al sistema utilizando un user y un password con el fin de preservar los datos dentro

	del sistema.
Prioridad del Requisito	Alta

2. GESTION DE PROGRAMAS Y HORARIOS (PARRILLA DE PROGRAMACIÓN)

TABLA LXVIII. REQUISITO FUNCIONAL 003

Número del requisito	RF003
Nombre del requisito	Administrar programas y horarios
Descripción	Al usuario administrar cada programa de acuerdo a su día y horario de salida al aire para ello se necesitará los datos: <ul style="list-style-type: none"> - Título de programa. - Días en los que está al aire. - Hora del programa.
Prioridad del Requisito	Alta

3. GESTION DE AUDIOS GRABADOS DE PROGRAMAS, ENTREVISTAS, NOTICIAS, ETC.

TABLA LXIX. REQUISITO FUNCIONAL 004

Número del requisito	RF004
Nombre del requisito	Administrar audios grabados de programas, entrevistas, noticias, etc.
Descripción del Requisito	Al usuario administrar los audios de programas, entrevistas, noticias, etc. Para ello se necesitará: <ul style="list-style-type: none"> - Título del audio - Archivo de audio en formato mp3.

Prioridad del Requisito	Alta
--------------------------------	-------------

Requerimientos no Funcionales.

Requisitos de Rendimiento

- El servidor en producción que se utilizará para el despliegue de la aplicación será Radio Universitaria.
- El sistema trabajará con una alta concurrencia capaz de soportar peticiones de más de 500 usuarios.
- El gestor de base de datos elegido es MySQL, en el cual se aloja la base de datos.

Seguridad

- Cada usuario tendrá sus propios privilegios los cuales no puedan afectar directamente al sistema.
- El usuario deberá loguearse usando un username y un password, en este proceso se realizará diferentes controles para que el ingreso sea seguro.

Disponibilidad

- El sistema ya que este desplegado en producción deberá estar disponible los 365 días del año, en el momento que necesite los usuarios.

Portabilidad

- El sistema estará construido completamente en un Web por lo que estará disponible en cualquier plataforma.

ANEXO 14

Documento de Arquitectura de Software

SWAR – Administración web para la
aplicación móvil de la Radio
Universitaria

Documento de Arquitectura de Software

Introducción

La arquitectura de software cumple un papel fundamental en el desarrollo de un sistema software ya que permite a un alto nivel describir, ver la manera cómo interactúan y se comportan los componentes del sistema. Además, permite tener diferentes perspectivas a medida que se pueda lograr entender el objetivo q debe cumplir el mismo lo que permite dar valor de calidad al producto.

Propósito.

Este documento al usar el modelo de vistas 4+1 proporciona una descripción de la arquitectura del sistema desde diferentes vistas arquitectónicas que permiten obtener diferentes perspectivas que cumplen un mismo objetivo.

Alcance

Este documento define la arquitectura de software que es utilizado para el sistema web SWAR el mismo que cuenta con las diferentes vistas del sistema (Casos de Uso, vista lógica, vista física, vista de despliegue, vista de procesos).

Definiciones, Siglas y Abreviatura.

Todas las definiciones, siglas y abreviaturas se mostrarán en el documento de glosario de términos del sistema SWAR.

Referencias

TABLA LXX. REFERENCIAS

Título de Documento	Referencia
Especificación de requerimientos SWAR	-
Glosario de Términos SWAR	-
Arquitectura de Software 4+1	[1] P. Kruchten, "Architectural Blueprints—The; 4+1; View Model of Software Architecture," <i>IEEE Softw.</i> , vol. 12, no. 6, pp. 42–50, 1995.

Vista global.

El presente documento está organizado para ayudar al lector a entender correctamente cada característica importante de la arquitectura del software, ya que se detalla cada una de las vistas a desarrollar. De acuerdo al modelo 4+1 se mostrará la descripción de la vista de casos de uso que contiene todos los casos de uso del sistema, la vista lógica donde se describe la funcionalidad del sistema, la vista física donde se muestra los componentes físicos que están interactuando en el sistema, la vista de despliegue muestra como esta interactuando los componentes software del sistema, y la vista de procesos que muestra el flujo que hay entre cada una de las operaciones de negocio del sistema. Todas estas vistas muestran sus modelos UML que representan a las mismas

Representación de la Arquitectura

La arquitectura a utilizada para el desarrollo de SWAR es la arquitectura cliente-servidor la cual para demostrar interacción se utiliza las vistas 4+1 de Kruchten: vista de casos de uso, vista lógica, vista de procesos, vista de despliegue, vista física que están basadas en el lenguaje unificado de Modelado UML.

TABLA LXXI. REPRESENTACION DE LA ARQUITECTURA

Vista	Elemento Modelado	Descripción
Vista de casos de Uso.	Casos de uso	Muestra la interacción que tienen los diferentes actores en el sistema.
Vista lógica.	Diagrama de clases,	Representa las funcionalidades y servicios que ofrece el sistema al usuario.
Vista física.	Diagrama de despliegue.	Representa la interacción de los componentes físicos que se usan en el sistema.
Vista de despliegue o Desarrollo.	Diagrama de Interacción.	Describe los componentes software para que el programador pueda entender las interacciones que existen.
Vista de procesos.	Diagrama de actividad.	Define y describe la comunicación de todos los procesos de negocio del sistema.

Objetivos y Restricciones de la Arquitectura

Para cumplir con un desarrollo de un buen software el sistema SWAR debe cumplir las siguientes características:

- **Rendimiento:** El sistema trabajará con una alta concurrencia capaz de soportar peticiones de más de 500 usuarios.
- **Disponibilidad:** El sistema ya que este desplegado en producción deberá estar disponible los 365 días del año, en el momento que necesite los usuarios.
- **Portabilidad:** El sistema estará construido completamente en un Web por lo que estará disponible en cualquier plataforma.
- **Calidad:** El sistema debe cumplir con todos los requisitos planteados en la especificación de requerimientos utilizando estándares que permitan asegurar la calidad del software.

Restricciones:

- **Restricción de Seguridad:** Cada usuario tendrá sus propios privilegios los cuales no puedan afectar directamente al sistema.
- **Restricciones de tecnología:** El sistema debe ser compatible con la mayoría de los navegadores web modernos.
- **Restricciones de tecnología:** La metodología a usar es la AUP (Proceso Unificado Ágil)
- **Restricciones de tecnología:** Las tecnologías de desarrollo estarán basadas en lenguajes de programación de PHP y javascript (Angular).
- **Restricciones de diseño:** Diseñar una arquitectura flexible que permita escalar el sistema para futuras adiciones de módulos.

Vista de Casos de Uso

En esta sección se describen los Casos de Uso del sistema SWAR, en donde se muestra todas las posibles acciones entre los actores y el sistema.

Modelo de casos de Uso

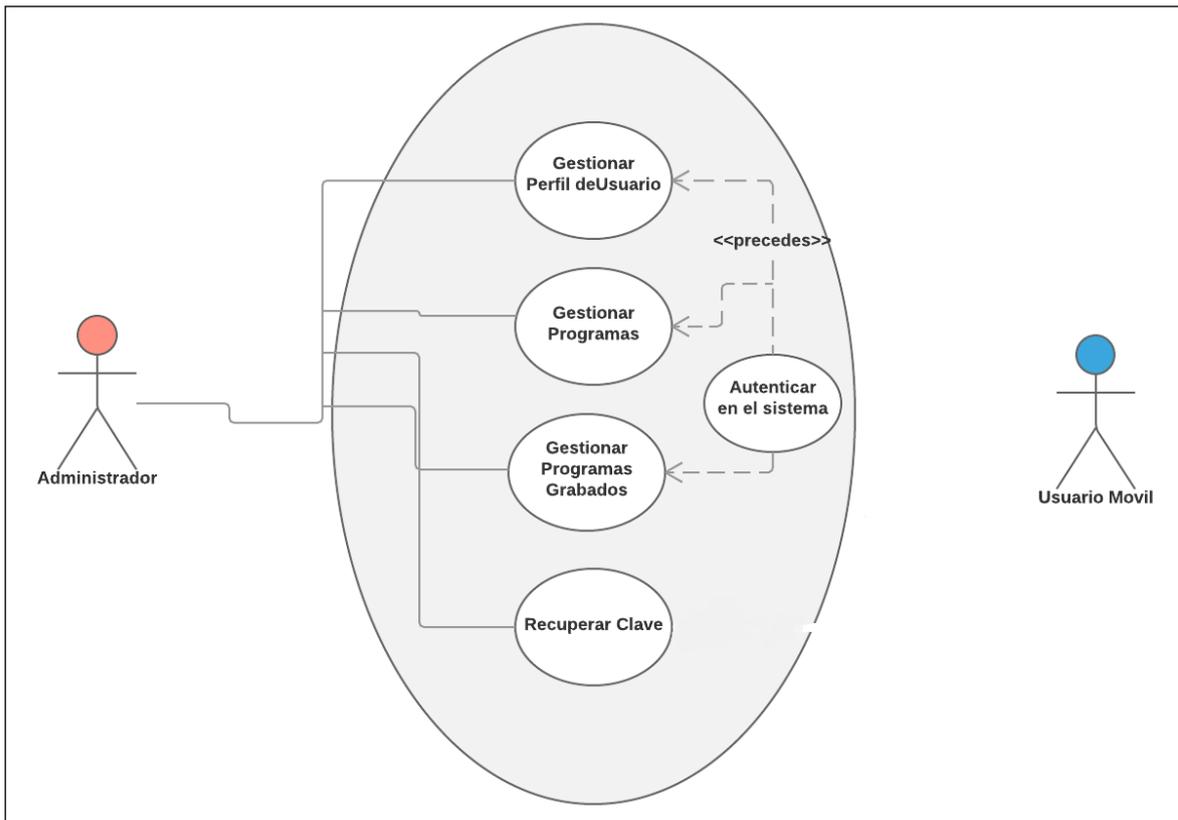


Figura 133 Modelo de casos de uso

Especificación de casos de uso.

TABLA LXXII. CASO DE USO AUTENTICACIÓN CON CORREO

Nombre	Autenticación con Correo.
Actores	Usuarios.
Objetivos	Autenticarse en el Sistema.
Precondiciones	Ninguna.
Postcondición	Usuario Autenticado.
Flujo Básico	<ol style="list-style-type: none"> 1. El usuario ingresa el usuario y el password. 2. El usuario presiona el botón ingresar. 3. El usuario digita la pregunta de verificación del sistema. 4. El usuario presiona el botón iniciar sesión.

	5. El sistema carga la página principal.
Flujo Alternativo	2.1. El sistema verifica si los datos del usuario son correctos. 2.2. Si los datos no son correctos muestra un mensaje de datos incorrectos.

Tabla LXXIII. CASO DE USO REESTABLECER CONTRASEÑA

Nombre	Restablecer Contraseña
Actores	Usuarios
Objetivos	Restablecer Contraseña olvidada
Precondiciones	Ninguna
Postcondición	El usuario puede usar la contraseña para loguearse
Flujo Básico	1.El Usuario presiona el enlace olvidaste tu contraseña 2. Se abre una ventana donde se ingresa el email de respaldo del Usuario 3. El Usuario digita la respuesta de pregunta de seguridad. 4. El Usuario presiona el botón iniciar sesión. 5. El Sistema carga la página principal.
Flujo Alternativo	3.1. Si el usuario no está registrado se mostrará un mensaje de error.

TABLA LXXIV. CASO DE USO GESTIONAR PERFIL DE USUARIO

Nombre	Gestionar perfil de Usuario
Actores	Administrador
Objetivos	Modificar la información personal del usuario.
Precondiciones	Debe estar autenticado en el sistema.
Postcondición	Gestión de perfil de Usuario completado.
Flujo Básico	<ol style="list-style-type: none"> 1. Debe autenticarse. 2. Se le mostrará página principal del sistema. 3. Selecciona donde se encuentra el nombre del usuario. 4. Se le muestra dos opciones Datos personales o Salir del sistema. 5. Selecciona Datos personales. 6. Se le mostrara un formulario con todos sus datos personales. 7. Procederá a modificar sus datos o cambiar su contraseña. 8. Presionara guardar datos. 9. El sistema guarda la información y muestra un mensaje satisfactorio.
Flujo Alternativo	<p>Flujo.</p> <p>Si los datos no se encuentran correctos, el sistema muestra un mensaje de error.</p>

TABLA LXXV. CASO DE USO GESTIONAR PROGRAMAS

Nombre	Gestionar programas
Actores	Administrador
Objetivos	Crear, Modificar, Eliminar, Programas.
Precondiciones	El administrador debe estar logueado.
Postcondición	Gestión satisfactoria de programas.
Flujo Básico	<ol style="list-style-type: none"> 1. El administrador ingresa al sistema. 2. El sistema presenta un panel de administración. 3. El administrador selecciona del menú de administración la opción parrilla de programación. 4. El administrador pulsa el botón de añadir programa. 5. El sistema presenta un formulario para ingresar los datos del nuevo programa. 6. El administrador rellena los datos del programa. 7. El sistema valida los datos. 8. El administrador pulsa el botón guardar programa. 9. El sistema guarda el nuevo programa. 10. El sistema presenta un mensaje que realiza la acción exitosamente.
Flujo Alternativo	<p>Flujo Modificar Programa</p> <ol style="list-style-type: none"> 4A.1. El administrador busca el programa modificar. 4A.2. El administrador pulsa en el título del programa que lleva al detalle de la mismo. 4A.3. El sistema le presenta el detalle del programa. 4A.4. El administrador pulsa en el botón de modificar. 4A.5. El sistema presenta el formulario con datos del programa.

	<p>4A.6. El administrador modifica los datos.</p> <p>4A.7. El sistema valida los campos.</p> <p>4A.8. El administrador pulsa en el botón guardar</p> <p>4A.9. El sistema muestra que ha sido guardada satisfactoriamente.</p> <p>Flujo 4B Eliminar Programa.</p> <p>4B.1. El administrador busca el programa a eliminar.</p> <p>4B.2. El administrador pulsa el botón eliminar</p> <p>4B.3. El sistema muestra un mensaje que se ha eliminado el programa.</p>
--	---

TABLA LXXVI. CASO DE USO GESTIONAR PROGRAMAS GRABADOS

Nombre	Gestionar programas grabados
Actores	Administrador
Objetivos	Crear, Modificar, Eliminar, Programas grabados.
Precondiciones	El administrador debe estar logueado.
Postcondición	Gestión satisfactoria de programas grabados.
Flujo Básico	<ol style="list-style-type: none"> 1. El administrador ingresa al sistema. 2. El sistema presenta un panel de administración. 3. El administrador selecciona del menú de administración la opción programas grabados. 4. El administrador pulsa el botón de añadir programa grabado. 5. El sistema presenta un formulario para ingresar los datos del nuevo programa grabado. 6. El administrador rellena los datos del programa grabado.

	<p>7. El sistema valida los datos.</p> <p>8. El administrador pulsa el botón guardar programa grabado.</p> <p>9. El sistema guarda el nuevo programa.</p> <p>10. El sistema presenta un mensaje que realiza la acción exitosamente.</p>
<p>Flujo Alternativo</p>	<p>Flujo Modificar Programa grabado</p> <p>4A.1. El administrador busca el programa grabado a modificar.</p> <p>4A.2. El administrador pulsa en el título del programa grabado que lleva al detalle de la mismo.</p> <p>4A.3. El sistema le presenta el detalle del programa grabado.</p> <p>4A.4. El administrador pulsa en el botón de modificar.</p> <p>4A.5. El sistema presenta el formulario con datos del programa grabado.</p> <p>4A.6. El administrador modifica los datos.</p> <p>4A.7. El sistema valida los campos.</p> <p>4A.8. El administrador pulsa en el botón guardar</p> <p>4A.9. El sistema muestra que ha sido guardada satisfactoriamente.</p> <p>Flujo 4B Eliminar Programa grabado.</p> <p>4B.1. El administrador busca el programa grabado a eliminar.</p> <p>4B.2. El administrador pulsa el botón eliminar</p> <p>4B.3. El sistema muestra un mensaje que se ha eliminado el programa grabado.</p>

Vista Lógica

La vista lógica de SWAR representa los requerimientos funcionales del sistema. Para esta vista se ha empleado el modelo de clases y de secuencia.

Diagrama de clases

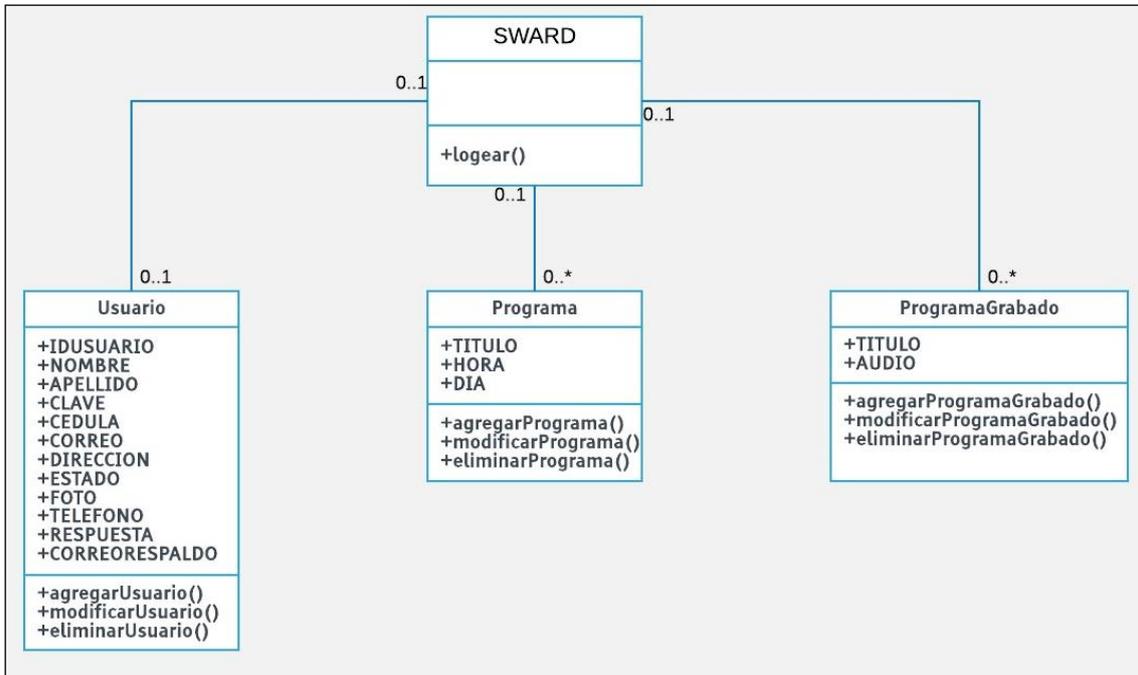


Figura 2 Diagrama de clases

Diagrama de secuencia

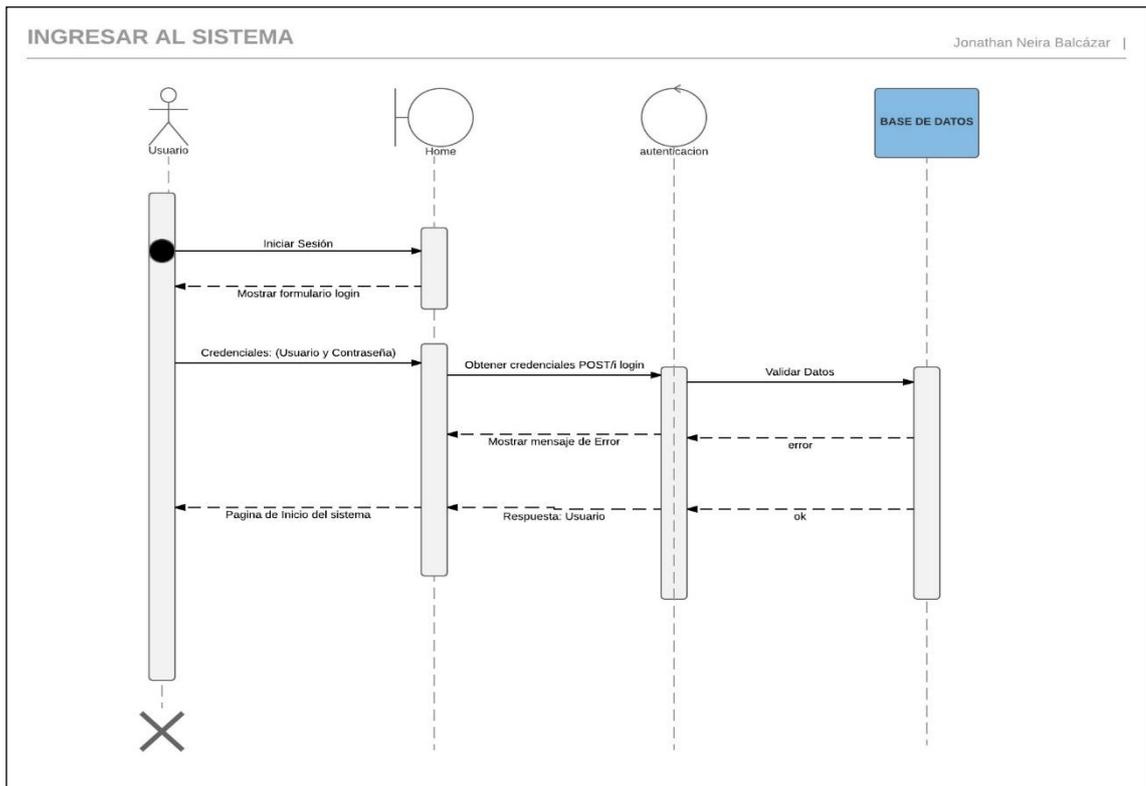


Figura 134 Diagrama de secuencia Ingresar al sistema

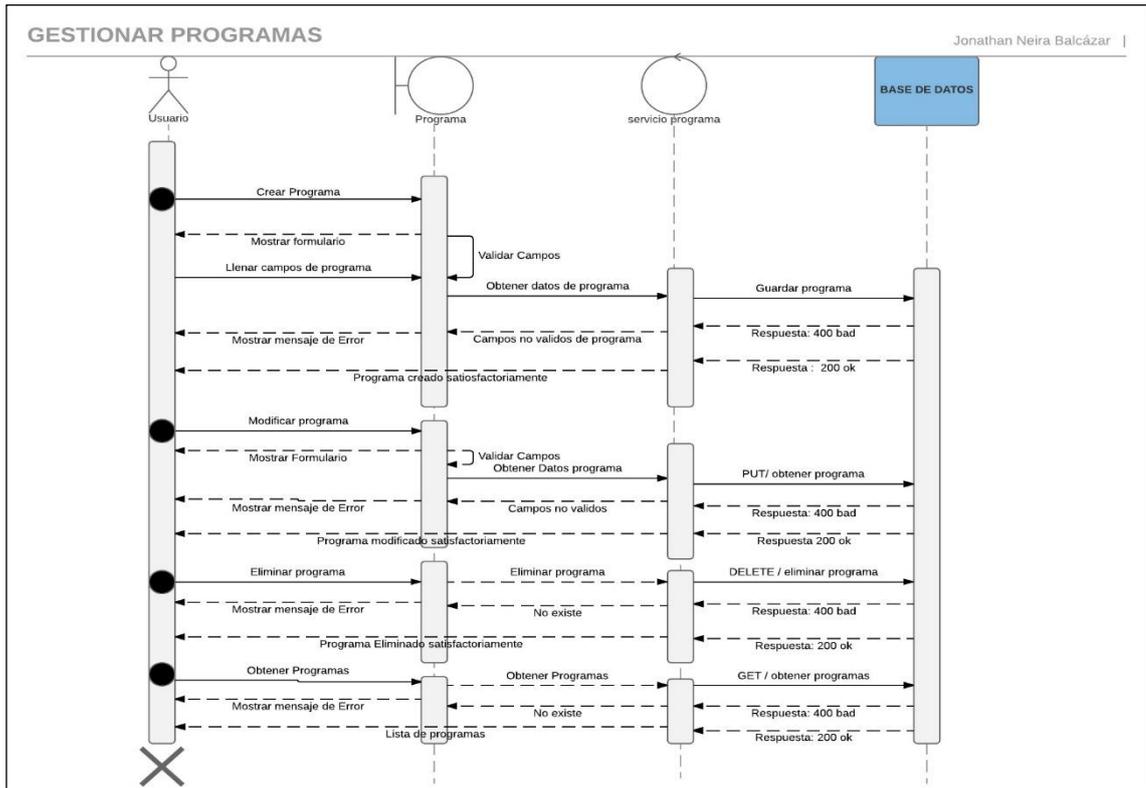


Figura 135 Diagrama de secuencia Gestionar programas

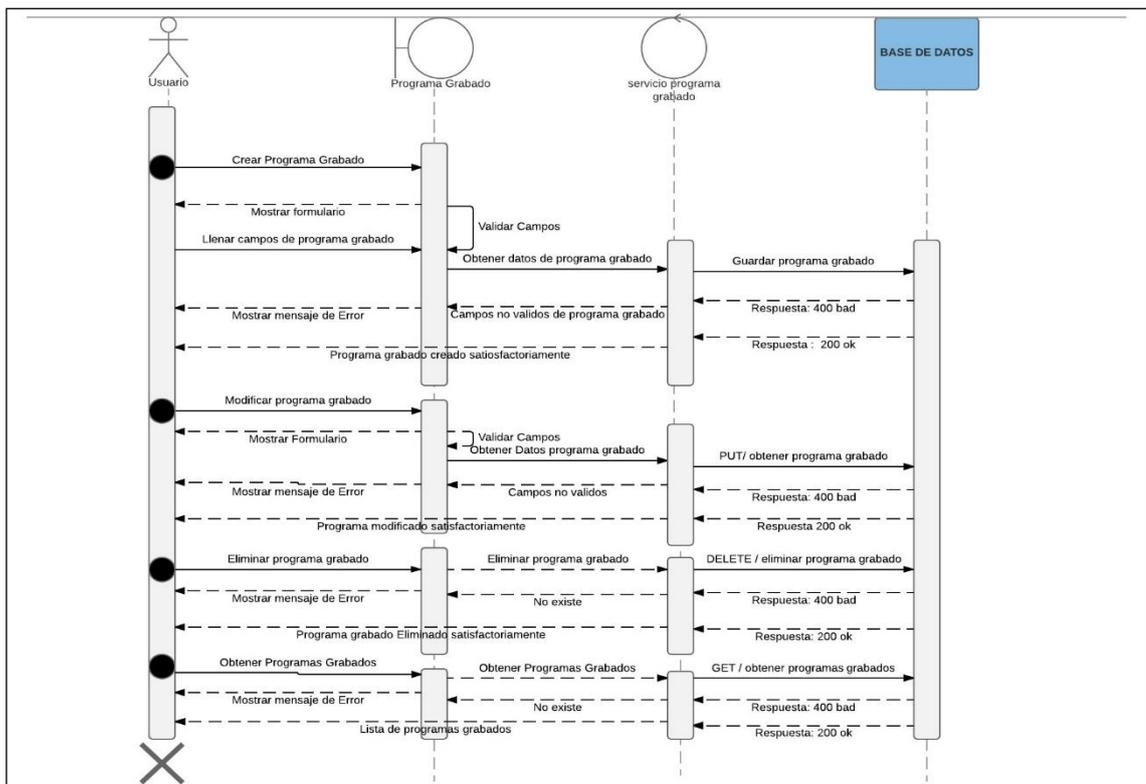


Figura 136 Diagrama de secuencia Gestionar programas grabados

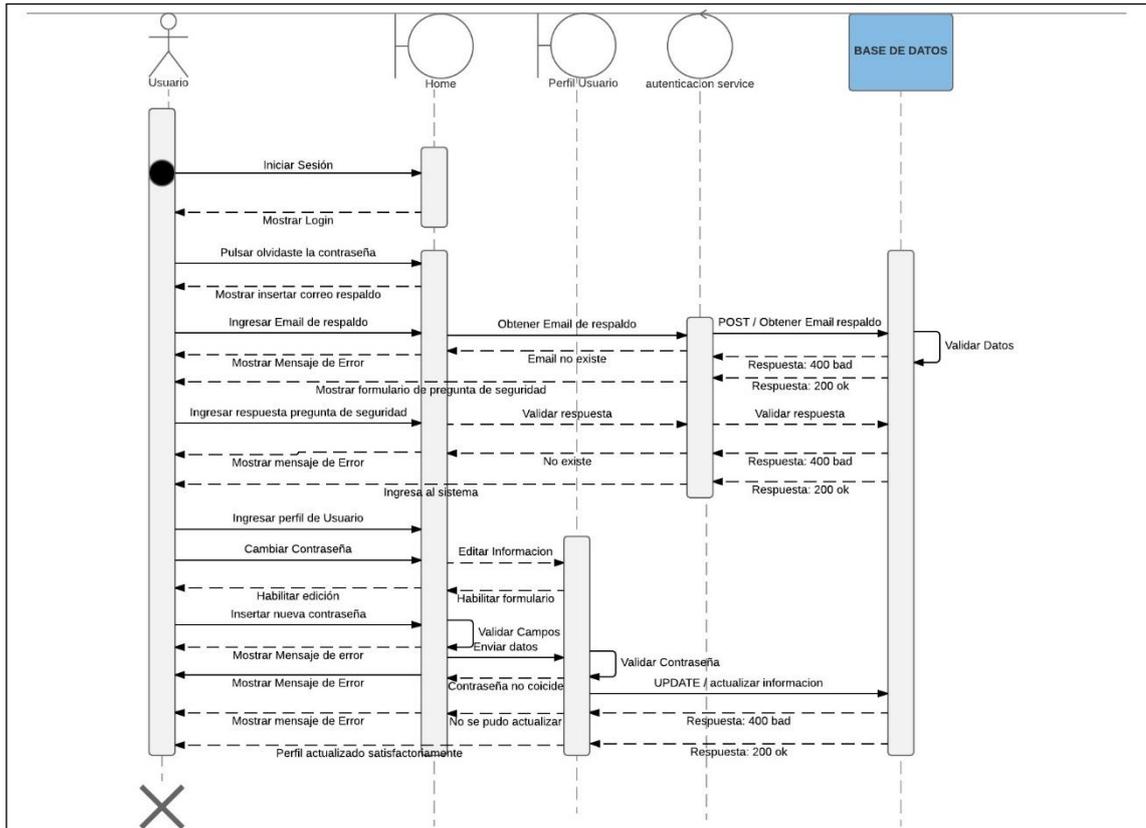


Figura 6 Diagrama de secuencia Cambiar contraseña

Vista de Procesos

Esta vista se ha modelado con el diagrama de procesos lo que ha permitido representar la descomposición de los procesos del sistema.

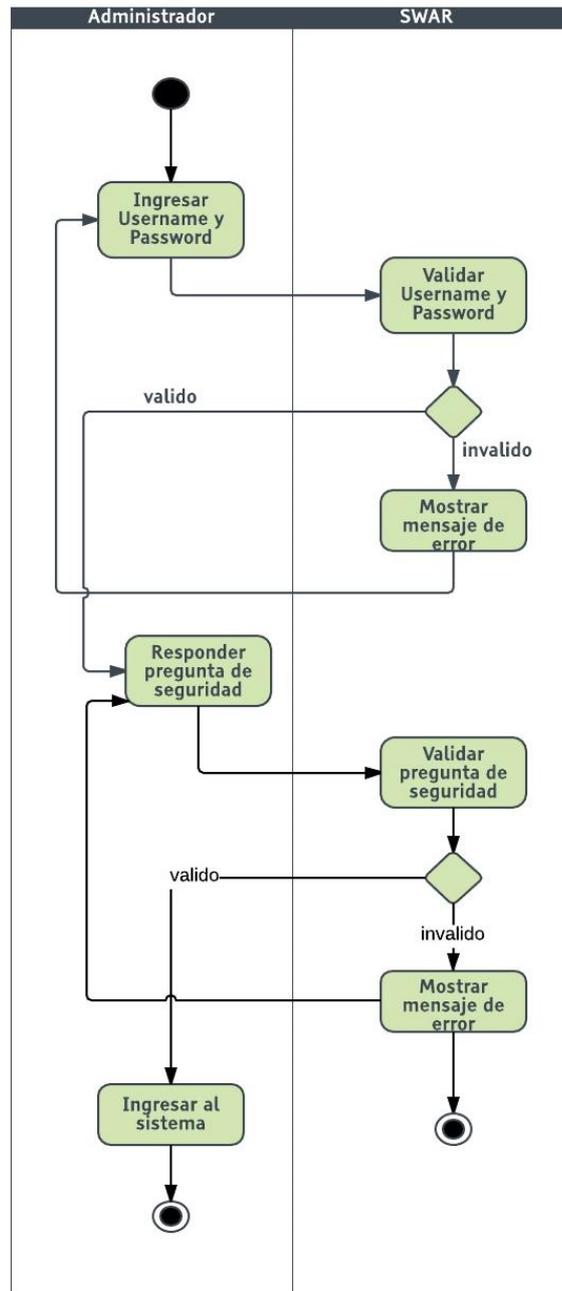


Figura 137 Proceso ingresar al sistema

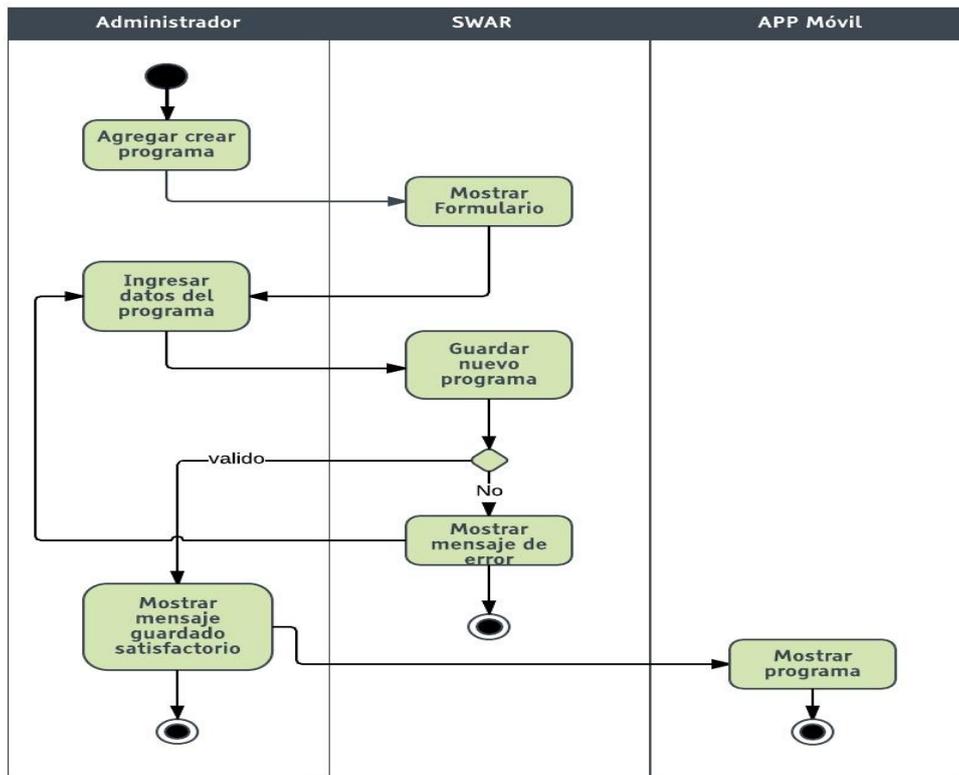


Figura 138 Proceso agregar programa

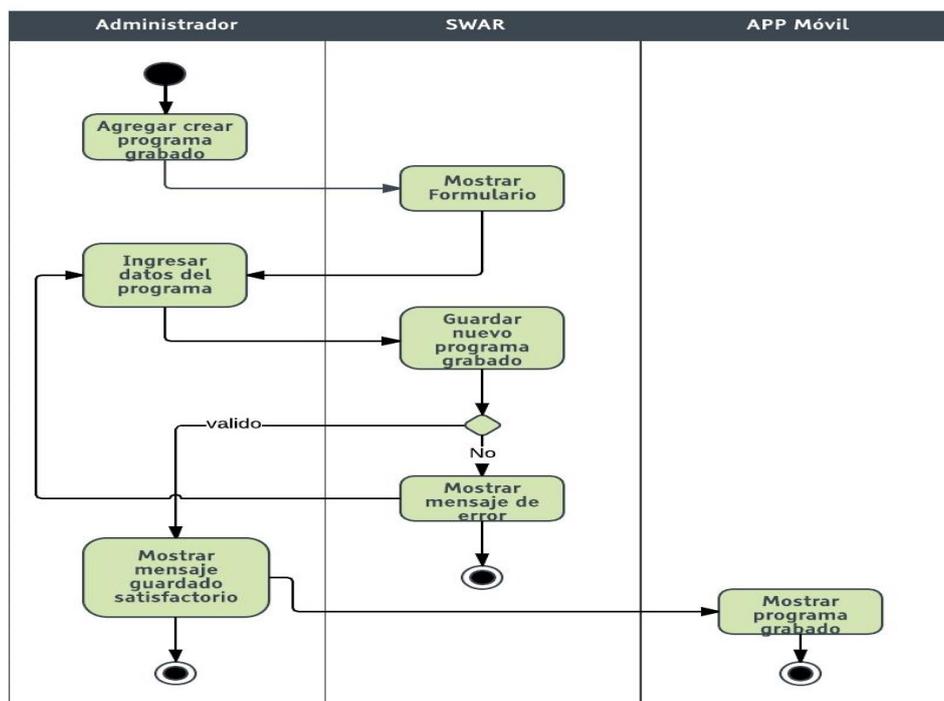


Figura 139 Proceso agregar programa grabado

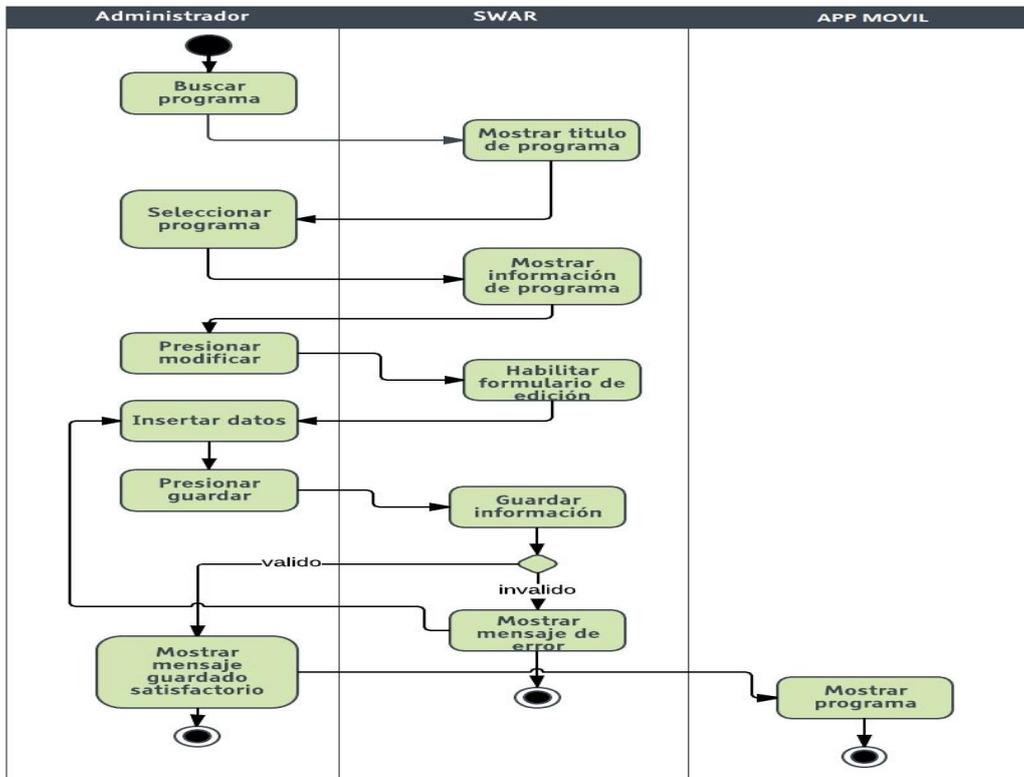


Figura 140 Proceso modificar programa

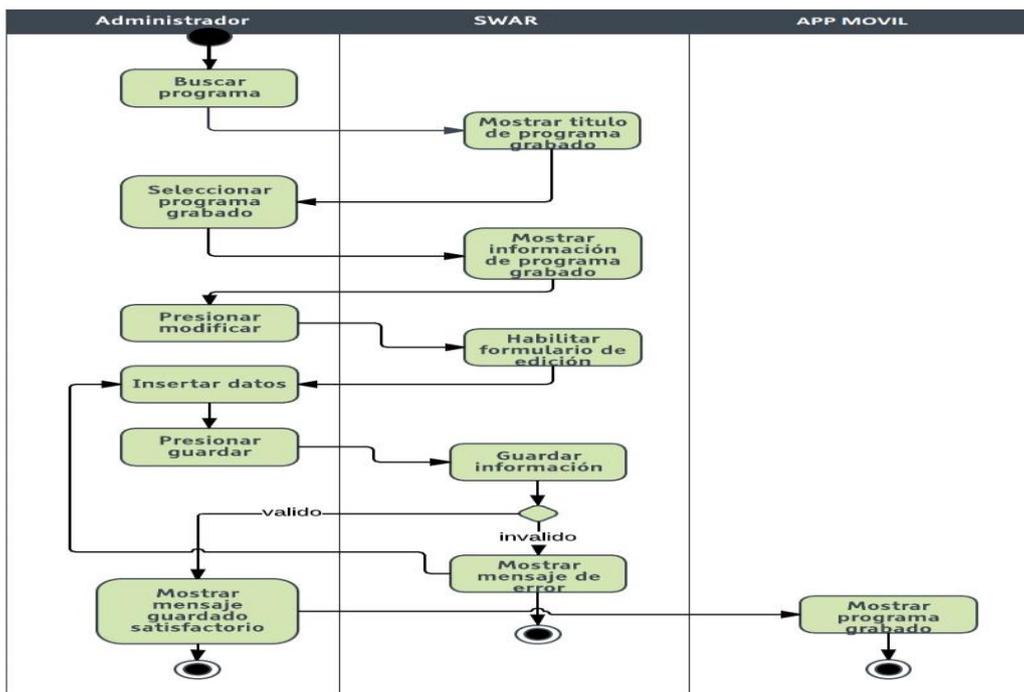


Figura 141 Proceso modificar programa grabado

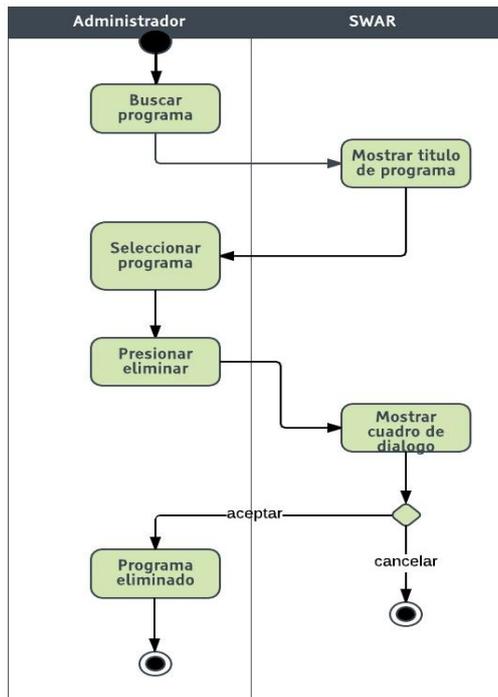


Figura 142 Proceso eliminar programa

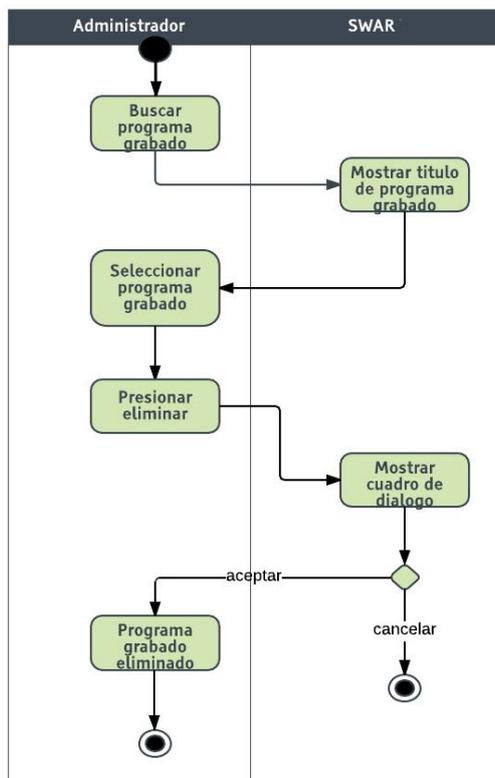


Figura 143 Proceso eliminar programa grabado

Vista de Despliegue
Diagrama de componentes

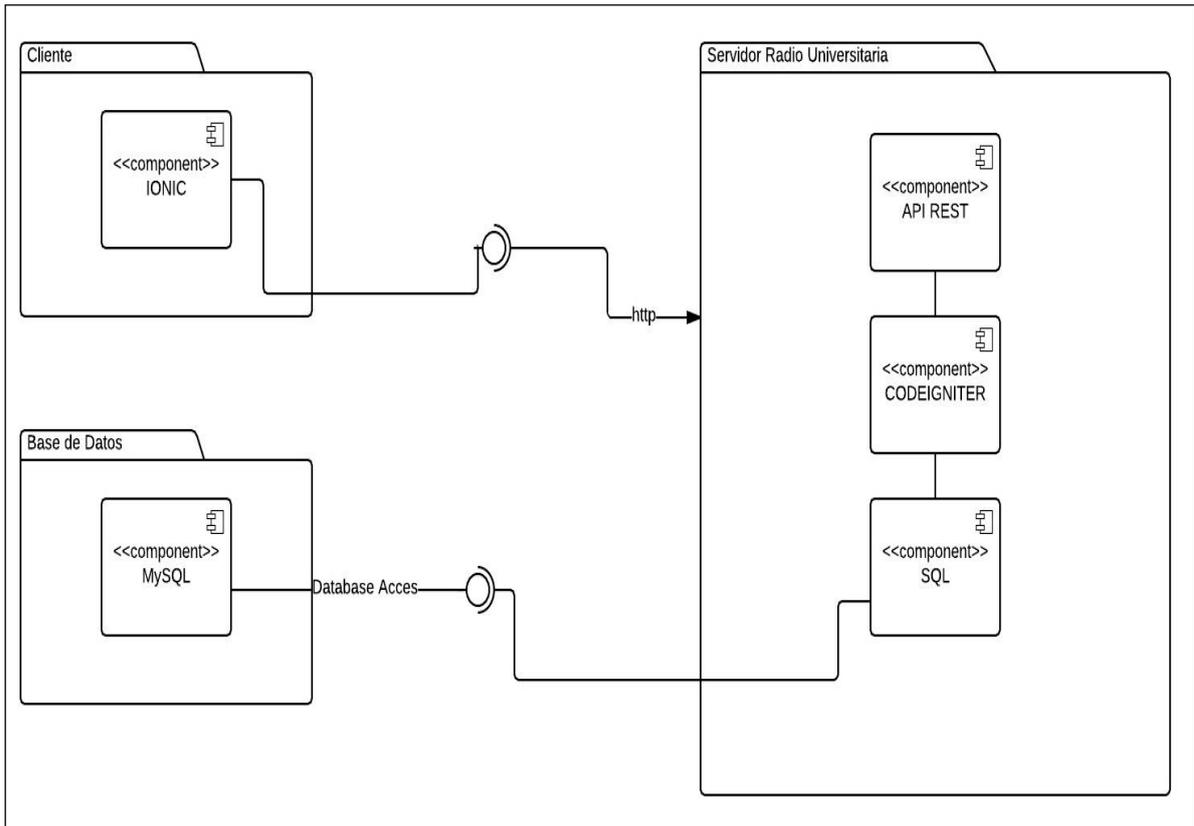


Figura 144 Diagrama de componentes y paquetes

Vista Física

Diagrama de Despliegue

En la siguiente imagen se muestra la vista de despliegue que muestra los nodos finales del Sistema de información SWAR.

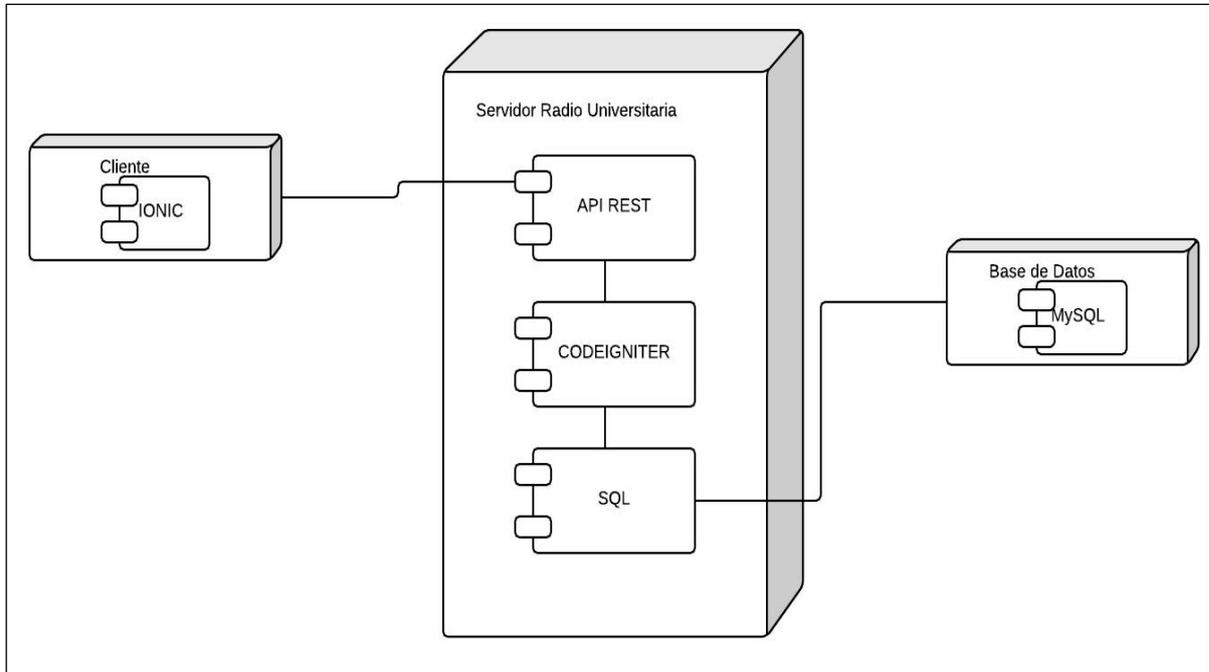


Figura 145 Diagrama de despliegue de SWAR

ANEXO 15

LICENCIA



IMPLEMENTACIÓN DEL SERVICIO
DE RADIO DE LA UNIVERSIDAD
NACIONAL DE LOJA PARA LOS
MEDIOS DE DIFUSIÓN DIGITAL Y
DISPOSITIVOS MÓVILES por
Jonathan Ezequiel Neira Balcázar se
distribuye bajo una [Licencia Creative
Commons Atribución-NoComercial-
CompartirIgual 4.0 Internacional](#).