



**UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN**

Implementación de un sistema web apoyado en pruebas unitarias para la administración de guiones y publicidad en la radio ciudad.

Trabajo de investigación previo a la obtención del título de Ingeniero en Sistemas y Computación.

**Autor:**

Campos Curizaca Julio David

**Tutor:**

PhD. Ximena Quintana López

**Riobamba – Ecuador**

**2022**

## DERECHOS DE AUTORÍA

Yo, Julio David Campos Curizaca, con cédula de ciudadanía 060422721-5, autor del trabajo de investigación titulado: “Implementación de un sistema web apoyado en pruebas unitarias para la administración de guiones y publicidad en la radio ciudad”, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 18 de mayo del 2022.



---

Campos Curizaca Julio David  
C.I: 060422721-5

## DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Implementación de un sistema web apoyado en pruebas unitarias para la administración de guiones y publicidad en la radio ciudad, presentado por Campos Curizaca Julio David, con cédula de identidad número 060422721-5, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 18 de mayo del 2022.

PhD. Paola Vinueza Naranjo PRESIDENTE DEL TRIBUNAL DE GRADO	
PhD. Fernando Molina Granja MIEMBRO DEL TRIBUNAL DE GRADO	
Mgs. Wayner Bustamante MIEMBRO DEL TRIBUNAL DE GRADO	
PhD. Ximena Quintana López TUTORA	



Campos Curizaca Julio David  
C.I: 060422721-5

## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Implementación de un sistema web apoyado en pruebas unitarias para la administración de guiones y publicidad en la radio ciudad, presentado por Campos Curizaca Julio David, con cédula de identidad número 060422721-5, bajo la tutoría de PhD. Ximena Quintana López; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 18 de mayo del 2022.

PhD. Paola Vinueza Naranjo PRESIDENTE DEL TRIBUNAL DE GRADO	
PhD. Fernando Molina Granja MIEMBRO DEL TRIBUNAL DE GRADO	
Mgs. Wayner Bustamante MIEMBRO DEL TRIBUNAL DE GRADO	



# CERTIFICACIÓN

Que, **CAMPOS CURIZACA JULIO DAVID** con CC: **060422721-5**, estudiante de la Carrera **INGENIERÍA EN SISTEMAS Y COMPUTACIÓN, NO VIGENTE**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**IMPLEMENTACIÓN DE UN SISTEMA WEB APOYADO EN PRUEBAS UNITARIAS PARA LA ADMINISTRACIÓN DE GUIONES Y PUBLICIDAD EN LA RADIO CIUDAD**", cumple con el **8%**, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 25 de abril de 2022

---

PhD. Ximena Quintana  
**TUTORA**

## **DEDICATORIA**

El presente trabajo de investigación va dedicado con amor y cariño a toda mi familia en especial a mis padres Dolores Curizaca y Julio Campos, por ser el eje fundamental de mi vida y el motor de aliento de mi ser, brindándome su amor de padres, apoyo incondicional, por todas sus enseñanzas y valores impartidos sobre mí, quién con sus consejos, sacrificio, y apoyo me ha permitido seguir adelante en mi carrera universitaria, contribuyendo en el logro de este paso de mi vida.

*David Campos*

## **AGRADECIMIENTO**

Primeramente, Agradezco a Dios por darme salud y mantenerme con vida además por ser mi guía a lo largo de mi vida.

A mis padres Dolores Curizaca y Julio Campos por darme la vida, por estar siempre conmigo, por ser mis primeros maestros de vida, por cada uno de los valores que me enseñaron, por todo ese apoyo sincero e incondicional que me regalaron en mi día a día y en si por ser unos buenos padres, a mis hermanos y a toda mi familia por su apoyo total que me han brindado.

Agradezco a la Universidad Nacional de Chimborazo que durante esta etapa universitaria se convirtió en un segundo hogar para mí, a mis docentes por compartir sus conocimientos y experiencias enfocadas en mi formación profesional.

De igual manera un agradecimiento a la Ing. Ximena Quintana, por su rectitud en su profesión como docente, por su conocimiento y motivación para guiarme durante todo el desarrollo de la tesis.

*David Campos*

## ÍNDICE GENERAL

DERECHOS DE AUTORÍA .....	II
DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL .....	III
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL .....	IV
CERTIFICADO ANTIPLAGIO .....	V
DEDICATORIA .....	VI
AGRADECIMIENTO .....	VII
RESUMEN .....	XIV
ABSTRACT .....	XV
INTRODUCCIÓN.....	16
CAPÍTULO I.....	18
1. PLANTEAMIENTO DEL PROBLEMA .....	18
1.1 Problema y Justificación .....	18
1.2 Objetivos .....	19
1.2.1 Objetivo General.....	19
1.2.2 Objetivos Específicos .....	19
CAPÍTULO II.....	20
2 MARCO TEÓRICO .....	20
2.1 PostgreSQL.....	20
2.1.1 Definición .....	20
2.1.2 Arquitectura .....	20
2.1.3 Características.....	21
2.1.4 Ventajas.....	22
2.2 Lenguaje de programación Java.....	22
2.2.1 Origen .....	22
2.2.2 Definición .....	23
2.2.3 Características.....	23
2.3 Java Server Face.....	24
2.3.1 Características Java Server Face.....	24
2.3.2 Funcionamiento .....	24
2.3.3 Elementos de JSF .....	24
2.3.3.1 La vista .....	25
2.3.3.2 El modelo.....	25
2.3.3.3 El controlador .....	25
2.3.4 Ventajas de usar de JSF.....	25
2.4 Bootstrap.....	26



2.4.1	Estructura de un archivo Bootstrap .....	26
2.4.2	Ventajas de usar Bootstrap.....	27
2.5	Metodología ágil SCRUM .....	27
2.5.1	Definición .....	27
2.5.2	El equipo SCRUM.....	27
2.5.3	Artefactos de SCRUM .....	28
2.5.3.1	Lista de Producto (Product Backlog).....	28
2.5.3.2	Lista de Pendientes del Sprint (Sprint Backlog).....	28
2.5.3.3	Incremento .....	28
2.5.4	Eventos SCRUM .....	29
2.6	Modelo Verificación y validación .....	29
2.6.1	Verificación.....	30
2.6.2	Validación .....	30
2.6.3	Modelo V&V .....	30
2.7	Pruebas de Software.....	32
2.7.1	Enfoques de pruebas de software .....	33
2.7.2	Pruebas unitarias.....	33
2.7.2.1	Definición .....	33
2.7.2.2	Características que tiene las pruebas unitarias .....	33
2.7.2.3	Beneficios de las pruebas unitarias.....	34
2.8	Herramientas para la automatización de las pruebas unitarias .....	35
2.8.1	Junit. ....	35
2.9	Diagrama de casos de uso .....	36
CAPÍTULO III .....		37
3	METODOLOGÍA.....	37
3.1	Tipo y diseño de la investigación.....	37
3.2	Métodos y técnicas.....	37
3.2.1	Métodos aplicados en el proyecto .....	37
3.2.2	Técnicas .....	38
3.3	Fase de planificación del sistema web .....	38
3.3.1	Reunión de planificación .....	38
3.3.2	Equipo SCRUM del Proyecto .....	40
3.3.3	Tipo de usuario y roles del proyecto.....	40
3.3.4	Lista del producto (Product Backlog).....	41
3.4	Fase de desarrollo del sistema web. ....	44
3.4.1	Arquitectura del sistema .....	44
3.4.2	Diseño de la interfaz de usuario .....	45

3.4.3	Diseño de la base de datos .....	47
3.4.4	Diagrama de casos de uso.....	48
3.4.4.1	Procesos del Productor del sistema .....	48
3.4.4.2	Procesos del locutor de la radio.....	48
3.4.4.3	Procesos del coordinador de la radio .....	49
3.4.5	Historias de usuario .....	49
3.4.6	Historias Técnicas.....	50
3.4.7	Sprint del Proyecto .....	51
3.4.7.1	Sprint 1 .....	51
3.4.7.2	Sprint 2 .....	52
3.4.7.3	Sprint 3 .....	53
3.4.7.4	Sprint 4 .....	54
3.5	Fase de cierre e implementación .....	55
CAPÍTULO IV .....		56
4	RESULTADOS Y DISCUSIÓN .....	56
4.1	Evaluar el beneficio del software con respecto al proceso manual.....	56
4.1.1	Proceso de administración de guiones.....	56
4.1.2	Muestra .....	56
4.1.3	Análisis de Datos .....	56
4.1.4	Procesos con el rol de Productor .....	56
4.1.4.1	Proceso de gestión de usuarios .....	57
4.1.4.2	Proceso de gestión de programas radiales .....	59
4.1.4.3	Proceso de gestión de publicidad.....	61
4.1.5	Procesos con el rol de Locutor .....	63
4.1.5.1	Proceso de gestión de guiones .....	63
4.1.5.2	Proceso de gestión de diálogos .....	64
4.1.6	Procesos con el rol de Coordinador.....	66
4.1.6.1	Proceso de Revisión de guiones .....	66
4.1.7	Contraste de normalidad.....	68
4.1.8	Pruebas en el proceso de Gestión de Usuarios .....	68
4.1.8.1	Prueba de normalidad del proceso insertar y editar.....	69
4.1.8.2	Prueba de muestras emparejadas del proceso insertar.....	69
4.1.8.3	Prueba de wilcoxon en el proceso Editar.....	70
4.1.9	Pruebas en el proceso de Gestión de Programas de Radio .....	70
4.1.9.1	Prueba de normalidad del proceso insertar y editar.....	70
4.1.9.2	Prueba de muestras emparejadas del proceso insertar y editar.....	70
4.1.10	Pruebas en el proceso de Gestión de Publicidad .....	71

4.1.10.1	Prueba de normalidad del proceso insertar y editar.....	71
4.1.10.2	Prueba de muestras emparejadas del proceso insertar y editar.....	71
4.1.11	Pruebas en el proceso de Gestión de Guiones .....	72
4.1.11.1	Prueba de normalidad del proceso insertar y editar.....	72
4.1.11.2	Prueba de muestras emparejadas del proceso insertar y editar.....	72
4.1.12	Pruebas en el proceso de Gestión de Diálogos .....	73
4.1.12.1	Prueba de normalidad del proceso insertar y editar.....	73
4.1.12.2	Prueba de muestras emparejadas del proceso insertar y editar.....	73
4.1.13	Pruebas en el proceso de Revisión de Guiones .....	73
4.1.13.1	Prueba de normalidad del proceso insertar y editar.....	73
4.1.13.2	Prueba de muestras emparejadas del proceso insertar y editar.....	74
4.2	Porcentajes de las pruebas unitarias.....	74
4.3	Discusión.....	76
CAPÍTULO V.....		77
5 CONCLUSIONES Y RECOMENDACIONES .....		77
5.1	Conclusiones .....	77
5.2	Recomendaciones.....	78
Bibliografía.....		79
ANEXOS.....		81
Anexo A Historia de Usuario del Proyecto .....		81
Anexo B Historias Técnicas del Proyecto.....		87
Anexo C Casos de Pruebas Unitarias.....		89
Anexo D Manual Técnico.....		120

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Características de Java. -----	23
<b>Tabla 2:</b> Miembros el equipo SCRUM. -----	27
<b>Tabla 3:</b> Eventos de SCRUM -----	29
<b>Tabla 4:</b> Simbología UML. -----	36
<b>Tabla 5:</b> Reuniones de Planificación. -----	38
<b>Tabla 6:</b> Equipo SCRUM del proyecto. -----	40
<b>Tabla 7:</b> Tipos de usuario y sus funciones. -----	41
<b>Tabla 8:</b> Talla de la camiseta del proyecto. -----	42
<b>Tabla 9:</b> Lista del producto del proyecto. -----	42
<b>Tabla 10:</b> Historia de usuario del proyecto.-----	50
<b>Tabla 11:</b> Historias técnicas. -----	50
<b>Tabla 12:</b> Sprint Backlog del proyecto. -----	51
<b>Tabla 13:</b> Sprint 1 del proyecto. -----	51
<b>Tabla 14:</b> Sprint 2 del proyecto. -----	52
<b>Tabla 15:</b> Sprint 3 del proyecto. -----	53
<b>Tabla 16:</b> Sprint 4 del proyecto. -----	54
<b>Tabla 17:</b> Registro de tiempo en la gestión de usuarios.-----	57
<b>Tabla 18:</b> Registro de tiempo en la gestión de programas de radio -----	59
<b>Tabla 19:</b> Registro de tiempo en la gestión de publicidad. -----	61
<b>Tabla 20:</b> Registro de tiempo en la gestión de guiones. -----	63
<b>Tabla 21:</b> Registro de tiempo en la gestión de diálogos.-----	65
<b>Tabla 22:</b> Registro de tiempo en el proceso de revisión de guiones -----	67
<b>Tabla 23:</b> Prueba de normalidad de la variable insertar y editar. -----	69
<b>Tabla 24:</b> Prueba de muestras emparejadas del proceso insertar.-----	69
<b>Tabla 25:</b> Prueba de wilcoxon del proceso editar. -----	70
<b>Tabla 26:</b> Prueba de normalidad de la variable insertar y editar. -----	70
<b>Tabla 27:</b> Prueba de muestras emparejadas del proceso insertar y editar.-----	70
<b>Tabla 28:</b> Prueba de normalidad de la variable insertar y editar. -----	71
<b>Tabla 29:</b> Prueba de muestras emparejadas del proceso insertar y editar.-----	71
<b>Tabla 30:</b> Prueba de normalidad de la variable insertar y editar. -----	72
<b>Tabla 31:</b> Prueba de muestras emparejadas del proceso insertar y editar.-----	72
<b>Tabla 32:</b> Prueba de normalidad de la variable insertar y editar. -----	73
<b>Tabla 33:</b> Prueba de muestras emparejadas del proceso insertar y editar.-----	73
<b>Tabla 34:</b> Prueba de normalidad de la variable insertar y editar. -----	74
<b>Tabla 35:</b> Prueba de muestras emparejadas del proceso insertar y editar.-----	74
<b>Tabla 36:</b> Porcentaje de las pruebas unitarias.-----	75

## ÍNDICE DE FIGURAS

<b>Gráfico 1:</b> Arquitectura de PostgreSQL .....	20
<b>Gráfico 2:</b> Vista en JSF .....	25
<b>Gráfico 3:</b> Estructura de Bootstrap .....	26
<b>Gráfico 4:</b> Modelo V&V .....	30
<b>Gráfico 5:</b> Arquitectura del proyecto .....	44
<b>Gráfico 6:</b> Pantalla de inicio de sesión del Sistema.....	45
<b>Gráfico 7:</b> Pantalla de gestión del Productor. ....	46
<b>Gráfico 8:</b> Pantalla de gestión del locutor. ....	46
<b>Gráfico 9:</b> Pantalla de gestión del Coordinador .....	47
<b>Gráfico 10:</b> Base de datos de sistema de la Radio.....	47
<b>Gráfico 11:</b> Diagrama del productor del sistema.....	48
<b>Gráfico 12:</b> Diagrama del locutor del sistema.....	48
<b>Gráfico 13:</b> Diagrama del coordinador del sistema.....	49
<b>Gráfico 14:</b> Proceso insertar en la gestión de usuarios.....	58
<b>Gráfico 15:</b> Proceso editar en la gestión de usuarios.....	58
<b>Gráfico 16:</b> Proceso insertar en la gestión de programas. ....	60
<b>Gráfico 17:</b> Proceso editar en la gestión de programas. ....	60
<b>Gráfico 18:</b> Proceso insertar en la gestión de publicidad. ....	62
<b>Gráfico 19:</b> Proceso editar en la gestión de publicidad. ....	62
<b>Gráfico 20:</b> Proceso insertar en la gestión de guiones.....	64
<b>Gráfico 21:</b> Proceso editar en la gestión de guiones.....	64
<b>Gráfico 22:</b> Proceso insertar en la gestión de diálogos.....	65
<b>Gráfico 23:</b> Proceso editar en la gestión de diálogos. ....	66
<b>Gráfico 24:</b> Proceso insertar en la revisión de guiones. ....	67
<b>Gráfico 25:</b> Proceso editar en la revisión de guiones. ....	68

## RESUMEN

En el presente trabajo de titulación se tuvo como objetivo desarrollar un sistema web apoyándose en las pruebas unitarias para ofrecer una herramienta informática que permita automatizar el proceso de administración de guiones, programas de radio y publicidad y así establecer un control de los procesos los mismos que son transmitidos en la Radio Ciudad 106.5 F.M.

Se utilizó el lenguaje de programación Java como entorno de desarrollo integrado se trabajó en Netbeans IDE 8.2, además se utilizó la tecnología para sistemas web Java Server Face aplicados para el sistema Web de la radio, se usó el framework de diseño de interfaces Bootstrap para los diferentes estilos que el sistema web de la radio va a tener y para la elaboración de la base de datos para el proyecto fue implementada con la tecnología PostgreSQL.

Teniendo en cuenta el ciclo de vida del software en V, acompañado con la metodología de desarrollo SCRUM, el mismo que se planificó la ejecución de los requerimientos obtenidos del productor de la radio, de esta forma, con la ayuda del productor de la radio y todas las personas comprometidas con la radio junto con el desarrollador se procede a la construcción del sistema teniendo como resultado un sistema web para la radio de acuerdo con las necesidades que tiene la entidad especificadas anteriormente. Para la implementación de las pruebas unitarias se realizó con JUnit y así verificar y validar el diseño de los diferentes módulos del sistema garantizando un porcentaje aceptable del correcto funcionamiento de los diferentes módulos que están conformados en el sistema web de la radio.

Además, con la utilización de las pruebas unitarias se procedió a ejecutar a cada historia de usuario del proyecto donde como resultado se obtuvo un rango del 80% - 100% de ejecución. Después de haber realizado un estudio estadístico se demostró que el tiempo empleado en cada proceso utilizando el sistema se obtuvo una diferencia significativa con respecto a los procesos realizados manualmente comprobando de esta manera la hipótesis planteada.

**Palabras Clave:** Administración de guiones y publicidad, sistema web, pruebas unitarias, java server faces (framework), Scrum (metodología de desarrollo ágil), Bootstrap, PostgreSQL.

## ABSTRACT

This research objective was to develop a web system based on unit tests to offer a computer tool that allows automating the process of managing scripts, radio programs, and advertising, thus establishing control of the same techniques broadcast on Radio Ciudad 106.5 F.M.

The Java programming language is using as an integrated development environment. Netbeans IDE 8.2 was worked on, Java Server Face technology was also used for web systems applied to the radio's Web system, and the Bootstrap interface design framework was used for the different styles for the web system of the radio will have. For the elaboration of the database for the project, it is implementing with PostgreSQL technology.

Taking into account the life cycle of the software in V, accompanied by the SCRUM development methodology, the same that the execution of the requirements obtained from the radio producer was planned, in this way, with the help of the radio producer and All the people committed to the radio, together with the developer, proceed to build the system, resulting in a web system for the radio by the needs of the entity specified above. The implementation of the unit tests was carried out with JUnit and thus verified and validated the design of the different modules of the system, guaranteeing an acceptable percentage of the correct functioning of the other modules that are made up of the web system of the radio.

In addition, with the use of unit tests, each user story of the project was executed, and as a result, a range of 80% - 100% execution was obtained. After having carried out a statistical study, it was shown that the time spent in each process using the system showed a significant difference concerning the procedures carried out manually, thus verifying the proposed hypothesis.

**Keywords:** Scripting and advertising management, web system, unit testing, java server faces (framework), Scrum (agile development methodology), Bootstrap, PostgreSQL.



Reviewed by:  
Ms.C. Ana Maldonado León  
ENGLISH PROFESSOR  
C.I.060197598

## INTRODUCCIÓN

En el desarrollo de software la fase de pruebas es imprescindible, ya que a través de ellas garantizamos que la aplicación abarca las diferentes funcionalidades para las cuales fue creada. Durante la etapa de pruebas puede demorarse más de lo esperado ya que al momento de identificar un error se necesita identificar el lugar exacto en el código para mostrar el error y así no devuelve el resultado deseado. Es por esta razón que una vez desarrollado cada componente del producto del software se incorporara una fase de pruebas de ese componente específico esto va a permitir reducir el costo del desarrollo, desarrollar la confianza en los clientes pues obviará los molestos errores de retrocesión, pero sobre todo nos brindará seguridad incremental en el desarrollo del producto final de software. (Niel & Stones, 2005, p. 14).

Al tratarse de una radio con poco tiempo al aire la radio ciudad realiza mayoritariamente procesos manualmente específicamente el proceso de presentación, revisión, análisis y aprobación de los guiones en la radio y la administración de la publicidad.

Contamos con una vasta bibliografía que nos permite conocer que hacer durante la fase de pruebas en un producto de software sin embargo la inversión por parte de las entidades en esta fase aún son escasos ya que la misma involucra recursos como tiempo, recursos humanos e infraestructura.

Una mala práctica que se realiza normalmente es el iniciar la fase de pruebas una vez finalizado el desarrollo del producto software, sin embargo, esta fase sería imponderable si las pruebas de software se ejecuten durante el desarrollo del sistema, finalizado cada módulo o componente que forman parte del producto final, evitaremos con resultado un software desprovisto de calidad, incremento en el costo al efectuar los cambios al realizar los cambios y sobre todo la insatisfacción del cliente.

En este caso, tenemos a Radio ciudad que pertenece al departamento de comunicación del Municipio de Riobamba que ha visto la necesidad de automatizar algunos de sus procesos como son la administración de guiones y el acceso a la información de la publicidad dando paso esta demanda a la implementación de un sistema web para la administración de guiones y publicidad en Radio ciudad con la idea de constituir una herramienta para el Ahorro de tiempo, recursos humanos.

Se requiere una automatización en los protocolos de revisión y aprobación de los guiones además del control de la publicidad transmitida a fin de acelerar el proceso de administración con el propósito de alcanzar eficacia y garantizar exactitud de respuesta del sistema.



El proyecto de investigación tendrá como objetivo el implementar un sistema web basándose en pruebas unitarias para ofrecer una herramienta informática que permita automatizar el proceso de administración de guiones de programas de radio y establecer un control de la publicidad. Entre las tecnologías a utilizarse están la tecnología para sistemas web Java Server Face, el frameworks de diseño de interfaces responsivas Bootstrap, Java como lenguaje de programación, PostgreSQL para la implementación de Base de Datos y la metodología de desarrollo ágil SCRUM, para que cumpla los requerimientos especificados.

Las pruebas se planificarán en el desarrollo del software para validar el diseño de los módulos del software garantizando un alto porcentaje en el correcto funcionamiento de las unidades codificadas.

El desarrollo del software donde está basado en las pruebas unitarias permitirá comprobar la funcionalidad del software de todos los módulos que posee. Este tipo de pruebas en esencia son fragmentos de código diseñados para validar que una parte específica del sistema desarrollado devuelva los valores exactos para los cuales fue creado.

El trabajo de investigación se basará en el análisis de la aplicación de pruebas unitarias y herramientas que se usarán en este proyecto. En la creación, diseño y desarrollo del sistema web de la administración de guiones y publicidad en la radio ciudad con pruebas unitarias y finalmente se comprobará y se analizará los resultados obtenidos dentro de un entorno de desarrollo basado en casos de pruebas unitarias.

En este documento se presentará la siguiente estructura: Capítulo I Introducción, Capítulo II Problema, Capítulo III objetivos del proyecto, Capítulo IV Estado del arte, Capítulo V Metodología, Capítulo VI Cronograma, Capítulo VII Fuentes bibliográficas, Capítulo VIII Anexos.

# CAPÍTULO I

## 1. PLANTEAMIENTO DEL PROBLEMA

### 1.1 Problema y Justificación

¿Cómo apoyará la aplicación de pruebas unitarias en el desarrollo del sistema web para la gestión de guiones y publicidad en Radio Ciudad?, ¿Cómo se realiza actualmente el proceso de gestión de guiones y publicidad? ¿Cómo el sistema web favorecerá en la gestión de los guiones y publicidad en radio ciudad?

El desarrollo de software apoyado en pruebas unitarias es la vía adecuada para comprobar cada uno de los componentes el software. Este tipo de pruebas consisten en tomar partes pequeñas de código comprobable en la aplicación, aislarla del resto del código y determinar si se comporta exactamente como esperamos y si cumple los objetivos con el cual fue creado. Cada unidad se prueba por separado antes de integrarlas en los componentes para probar las interfaces entre las unidades.

Para el desarrollo de los guiones de la radio se requiere que los mismos pasen por una revisión y aprobación, se determina que los procesos son llevados manualmente y no existe ningún proceso que maneje la sección de publicidad. por lo que el proceso de gestión demanda de tiempo y retraso, además de carencia de información oportuna y veraz. En este proceso de gestión participan varios actores por lo tanto para facilitar su desempeño es necesario desarrollar una aplicación web que cuente con eficiencia en cuanto al manejo de información y que asegure la exactitud de respuesta del sistema. Radio Ciudad, está consciente que el uso de las Tecnologías de la Información y Comunicación son una gran ayuda para realizar parte de su actividad en la gestión de guiones y publicidad, permitiendo llevar a cabo las tareas necesarias optimizando el problema de tiempo y recursos. Aplicando el desarrollo de software con pruebas unitarias, como solución al problema planteado se propone desarrollar un sistema web de gestión de guiones para la presentación, revisión, análisis y aprobación de los mismos. Dentro del proceso de desarrollo se implementarán pruebas unitarias automatizadas, las mismas que serán ejecutadas luego de crear una unidad de código del software, en este caso, la unidad de código serán las clases creadas para el software. El sistema constará con los siguientes módulos:

- Módulo de autenticación de usuarios.
- Módulo de gestión de usuarios.
- Módulo de gestión de programas de radio.

- Módulo de gestión de guiones.
- Módulo de revisión de guiones.
- Módulo de gestión de publicidad.
- Módulo de reportes (usuarios, programas, publicidad y guiones).

El presente proyecto se basa en una investigación exploratoria, descriptiva y aplicada. El tema de investigación principal cuenta con el suficiente acceso de información tanto en internet, revistas, artículos, libros, etc. EL presente trabajo de investigación se lo realizará en un plazo de 400 horas para la ejecución de todos los procesos de investigación tales como: el planteamiento del problema, marco teórico, marco metodológico, resultados y conclusiones. El proyecto de investigación no requiere de un financiamiento mayor o ser auspiciado por alguna entidad.

Este tipo de investigación ayudará a formar una base de conocimiento y experiencia, el mismo que con la ejecución de pruebas unitarias al software nos ayudará durante el desarrollo a la detección temprana de errores y en la reducción de los costos de producción y mantenimiento del software.

Se espera que esto contribuya a que el componente de Radio Ciudad perteneciente al departamento de comunicación del municipio de Riobamba en la administración de guiones y publicidad obtenga una mayor integridad de datos, menor consumo de tiempos en los procesos y en la generación de reportes A demás de una mejor accesibilidad a la información.

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

Implementar un sistema web para la administración de guiones y publicidad en la radio Ciudad 106.5 utilizando un desarrollo de software apoyado en pruebas unitarias para la validación y verificación de la funcionalidad y eficacia del sistema.

### **1.2.2 Objetivos Específicos**

- Analizar la aplicación de pruebas unitarias a través de la recopilación de información necesaria para el desarrollo e implementación del sistema web.
- Diseñar e implementar el sistema web con las tecnologías PostgreSQL, Java Server Face y Bootstrap para la administración de guiones y publicidad en la radio “Ciudad 106.5 FM”.
- Evaluar el sistema web mediante pruebas unitarias para certificar la funcionalidad del software implementado.

## CAPÍTULO II

### 2 MARCO TEÓRICO

A continuación, se describe todo lo que se va a usar para poder completar con nuestro tema de investigación, así como la tecnología de diseño e implementación de la base de datos, el lenguaje de programación, la tecnología de sistemas web junto con el frameworks de diseño web responsiva.

#### 2.1 PostgreSQL.

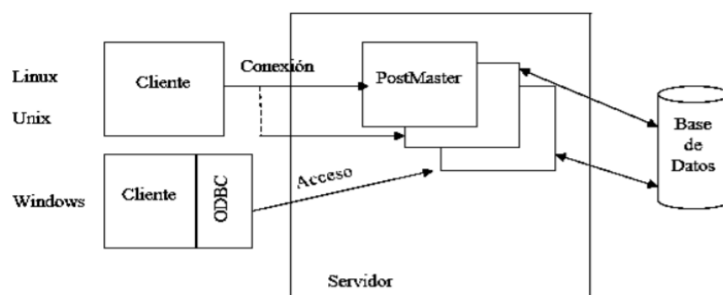
##### 2.1.1 Definición

Es un sistema de gestión de base de datos (DBMS) libre y de código abierto que incorpora un modelo relacional para sus bases de datos y soporta el lenguaje de consultas SQL (Niel & Stones, 2005, p. 11).

Este DBMS está diseñado para ser ejecutado sobre UNIX como plataforma, sin embargo, está también diseñado para ser portable y podría ejecutarse en varias plataformas, como, por ejemplo, MAC OS X, Solaris y Windows (POSTGRESQL, 2018).

##### 2.1.2 Arquitectura

PostgreSQL puede ser usado en un entorno cliente/servidor, ejecutándose en un solo servidor. A continuación, se expone la arquitectura de PostgreSQL, de acuerdo con (Niel & Stones, 2005, p. 14).



**Gráfico 1:** Arquitectura de PostgreSQL  
**Fuente:** (Niel & Stones, 2005, p. 14)

En el Gráfico 1 nos explica que diferentes clientes se conectan a un servidor haciendo uso de una red TCP/IP, en PostgreSQL, mediante una red LAN o posiblemente haciendo uso del internet. Cada cliente en su momento se conecta al proceso principal del servidor de la base de datos que es conocido como PostMaster como nos indica en la Figura 1, este crea un nuevo proceso específico para las peticiones de servicio donde da el acceso para los diferentes cliente.

De esta forma se hace el manejo de datos en un servidor, intenta controlar a diferentes

clientes que acceden a los mismos datos que se encuentran almacenados en un directorio compartido o a la vez en un servidor. Esto permite que el administrador de bases de datos PostgreSQL, mantenga la integridad de datos aun con muchos usuarios conectados.

Los clientes hacen uso de un mensaje específico de protocolo para PostgreSQL, se puede también instalar un software en el cliente para que facilite una interfaz estándar y así la aplicación trabaje, como el estándar Open Database Connectivity (ODBC) o el estándar Java Database Connectivity (JDBC) los mismos que son usado por los programas Java. La disponibilidad de un driver ODBC permite que las aplicaciones existentes usen PostgreSQL como base de datos. (Niel & Stones, 2005, p. 14).

La arquitectura cliente / servidor para el administrador de bases de datos PostgreSQL permite una división de labor. Una máquina de servidor adaptada para el almacenamiento y acceso a grandes cantidades de informaciones puede ser utilizado como un repositorio de datos seguro. Las diferentes aplicaciones gráficas sofisticadas pueden ser desarrolladas por los clientes, así como también un web los mismos que son basado en el front – end, estos pueden ser creado para acceder a los datos y devolver resultados a páginas web como (Niel & Stones, 2005, p. 14)

### **2.1.3 Características**

Este administrador de bases de datos PostgreSQL tiene diferentes características que son únicas marcando la diferencia en relación con otros DBMS, por lo que se presenta las características generales, propias de programación y características de secuencias SQL.

Entre las características generales más importantes de PostgreSQL tenemos:

- Unicode
- Integridad
- Posee varios métodos de autenticación al sistema
- Tiene el acceso encriptado vía SSL
- Licencia BSD

Además, presenta características de programación muy notables las cuales se puede mencionar tres rasgos de mayor importancia con respecto al desarrollo, que este sistema de gestión de base de datos PostgreSQL tiene (Martinez, 2009).

- Se puede asociar funciones y procedimientos almacenados en diferentes tipos de lenguajes de programación.
- Además, soporta el almacenamiento de objetos binarios grandes como son los gráficos, los videos y sonido.

- Se puede hacer uso de diferentes interfaces de programación de aplicaciones o conocidos como API, por ejemplo entre estos tenemos: C, C++, Java, Perl, Net, Python, Ruby, entre otros.

#### **2.1.4 Ventajas**

Según (POSTGRESQL, 2018), se pueden dar a conocer algunas de las ventajas que tiene este gestor de bases de datos a diferencia de otros:

- PostgreSQL tiene implementada un control de concurrencias multiversión (MVCC), la misma que es conocida esta característica como snapshot isolation.
- Permite agregar diferentes funciones las mismas que pueden ser personalizadas y desarrolladas usando diferentes tipos de lenguajes de programación como C++, Java, etc.
- Este gestor de bases de datos es designado para ser extensible, donde nos permite definir tipos de datos propios, tipos de índice, lenguajes funcionales entre otros.
- En el caso de la consulta para resolver diferentes tipos de problemas o dudas con este gestor de bases de datos, la comunidad involucrada con PostgreSQL es muy extensa los mismos que ayuda resolviendo muchas dudas e inquietudes, además existen empresas que tienen como objetivo brindar soporte comercial en todo momento.

## **2.2 Lenguaje de programación Java**

### **2.2.1 Origen**

Este lenguaje de programación fue creado como respuesta a las diferentes necesidades que se presentaban en ese tiempo considerando que sea un lenguaje capaz de ser compatible con las diferentes arquitecturas físicas como son los procesadores, tamaño de memoria y los sistemas operativos. Así como también el nivel de adaptación de la interfaz gráfica de las aplicaciones y la interconexión con otros dispositivos (Groussard, 2014).

En su tiempo Bill Joy era ingeniero de Sun Microsystems, y junto con su equipo de investigadores trabajaban en el “Proyecto Green” el mismo que consistía en desarrollar aplicaciones a una amplia variedad de periféricos y sistemas embebidos. Este equipo de investigadores desarrollaron el lenguaje multiplataforma que en principio se conocía como C++-, y finalmente lo llamaron Java y así nace en el año de 1991 (Groussard, 2014).

### 2.2.2 Definición

Java es el lenguaje de programación orientado a objetos el mismo que es creado en 1991 y publicado en 1995 por la empresa Sun Microsystems la misma que fue adquirida por Oracle en el año 2010, este lenguaje de programación permite escribir el código solo una vez y puede ser ejecutado en cualquier dispositivo, gracias a la máquina virtual de Java (JVM) que brinda la portabilidad del lenguaje. (Guevara, 2018).

### 2.2.3 Características

El lenguaje de programación conocido como Java tiene características únicas el mismo que lo hacen diferente al resto de lenguajes de programación, lo que ha permitido que su popularidad se mantenga en estos tiempos tecnológicos. A continuación en la Tabla 1 se presenta sus características más importantes.

**Tabla 1:** Características de Java.

<b>Características</b>	<b>Descripción</b>
Orientado a Objetos	En Java se puede decir que todo es un objeto, así garantiza un mejor dominio de la complejidad, reutilización más sencilla y mayor facilidad de corrección y evolución.
Distribuido	Java implementa los protocolos de red estándar, de esa forma permitiendo el desarrollo de aplicaciones cliente / servidor en arquitecturas distribuidas.
Interpretado	Cuando se realiza un programa utilizando lenguaje Java, este no se ejecuta sino más bien se lo interpreta por la máquina virtual o JVM. Por lo general, en este lenguaje solo hay dos etapas, la etapa de compilación y la etapa de la ejecución. El JVM se encarga de la operación de edición de enlaces en tiempo de ejecución del programa.
Robusto	Java es un lenguaje altamente tipado y estricto. La declaración de las variables es obligatoriamente. Con el lenguaje de programación java en el momento de la etapa de compilación y de la ejecución se verifica el código para evitar los errores y problemas de compatibilidad.
Seguro	tiene un mecanismo que alerta la seguridad de las aplicaciones de los sistemas, este es el motor de ejecución de Java o conocido como el JRE.
Independiente de las arquitecturas	Con Java no se genera un código para un tipo de arquitectura específico. El compilador genera un bytecode que es independiente y es de fácil interpretación o transformación dinámica en código nativo para aumentar el rendimiento. La JVM se encarga de traducir e interpretar el bytecode.
Portable	Java es portable gracias a que se trata de un lenguaje interpretado facilitando al cliente.
Eficaz	Java es interpretado, esto quiere decir que es un poco lento a referencia con un programa nativo, pero Java tiene un proceso de optimización de interpretación de código, llamado JIT (Just in Time) o HotSpot. Dicho proceso compila el bytecode Java en código nativo en tiempo de ejecución, lo que permite alcanzar el mismo rendimiento que un programa escrito en lenguaje C o C++.
Multitarea	Java permite desarrollar aplicaciones que poseen una ejecución simultánea de varios hilos o procesos mas ligeros. (Groussard, 2014, pp. 16-19)

**Realizado por:** Campos David

El lenguaje de programación Java se lleva de una buena manera y este es compatible con toda la tecnología de desarrollo de sistemas web robustos, esto es el framework JSF o conocido como Java Server Face, el mismo que permite realizar un buen diseño front end.

## **2.3 Java Server Face**

Java Server Face o JSF es un framework modelo, vista controlador (MVC) basado en el API de Servlets que facilita diferentes componentes en forma de etiquetas definidas en páginas XHTML mediante el framework Facelets. Este framework Facelets se define como un elemento indispensable que proporciona tipos de plantillas y de la creación de componente. (Universidad de Alicante, 2014).

### **2.3.1 Características Java Server Face**

JSF facilita rasgos destacables para el desarrollo de aplicaciones web, a continuación, se presenta sus características más relevantes que son: (Universidad de Alicante, 2014).

- Define las interfaces de usuario mediante vistas que juntan componentes gráficos.
- Mediante los beans admite la conexión de los diferentes componentes gráficos con los datos que tiene la aplicación.
- La conversión de los datos y la validación automática de la entrada del usuario.
- Facilita la navegación entre diferentes vistas.
- A partir de la especificación 2.0 un modelo estándar de comunicación Ajax entre la vista y el servidor.

### **2.3.2 Funcionamiento**

El funcionamiento del framework Java Server Faces tiene el siguiente proceso que se explica a continuación: (Universidad de Alicante, 2014)

1. Se ejecuta la página de arriba hacia abajo y se crea un árbol de los componentes JSF en forma de objetos instanciados de clases del framework Java Server Faces.
2. Se adquieren los valores ingresados por el usuario y se actualizan todos los beans que han sido gestionados con ellos.
3. Se actualizan todos los componentes con los valores originarios de las propiedades de los beans gestionados.
4. Se solicita a los componentes que se rendericen, el mismo que se genera el código HTML que se envía de vuelta al navegador como resultado de dicha petición.
5. El árbol de componentes JSF se guarda en la memoria para que más adelante las peticiones a la misma página JSF no tengan que crearlo, sino que haga uso de las peticiones existente.

### **2.3.3 Elementos de JSF**

Cuando se tiene una aplicación que es realizada con el framework de JSF, hay que tener en cuenta tres elementos indispensables que son: la vista, el modelo y el controlador.



### 2.3.3.1 La vista

A la vista se define mediante las páginas con diferentes componentes de Java Server Faces que utilizan beans gestionados para almacenar los diferentes tipos de datos. Los beans se expresan en el fichero de configuración faces-config.xml.

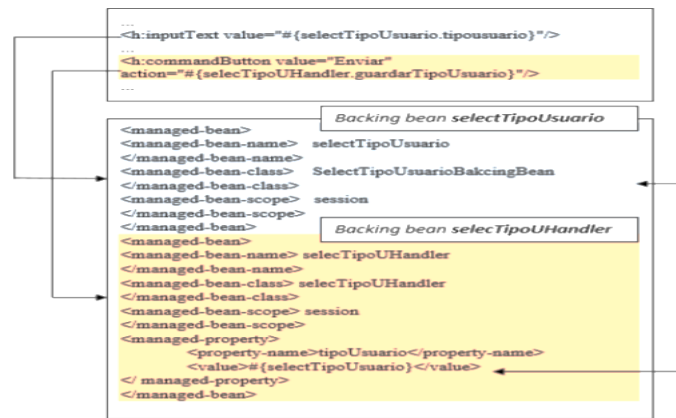


Gráfico 2: Vista en JSF

Fuente: (Universidad de Alicante, 2014)

En el Gráfico 2 se muestra las relaciones que existen entre la vista y la configuración, en donde el cuadro superior representa a la vista (select-tipoUsuario.jsp) y el cuadro inferior representa a la configuración (faces-config.xml). Este tipo de lenguaje de expresiones de Java Server Faces se puede utilizar en el fichero faces-config.xml para inicializar los valores de las propiedades de los beans, permitiendo acceder de un bean hacia otro (Universidad de Alicante, 2014).

### 2.3.3.2 El modelo

Constituyen a las propiedades y comportamientos que se encuentran representados en un beans de Java normal, se debe tomar en cuenta que, al referirse a propiedades y comportamientos, son atributos y métodos conocidos como los getters y setters. (Universidad de Alicante, 2014).

### 2.3.3.3 El controlador

Este está definido mediante los métodos de los beans atados a acciones de la vista. La acción que se va a ejecutar se puede definir en el código del método y la vista resultante depende de la cadena devuelta y del fichero de la configuración (Universidad de Alicante, 2014).

### 2.3.4 Ventajas de usar de JSF.

Entre las ventajas mas relevantes al momento de usar Java Server Face según (Munoz Simo, 2012) son:

- El código JSF es casi similar al estándar HTML, el mismo que hace su fácil utilización para los desarrolladores y diseñadores web.
- Java Server Faces permite ingresar javascript en la página lo que permite acelerar la respuesta de la interfaz en el cliente.
- Java Server Faces permite crear nuevos componentes a medida, permitiendo que sea extensible, mediante las APIs se puede alterar el comportamiento y controlar su funcionamiento.
- Las nuevas versiones del framework JSF almacenan la funcionalidad de versiones, las mismas que garantizan un mayor porcentaje de compatibilidad, permitiendo que el mantenimiento de las aplicaciones sea más sencillo.

## 2.4 Bootstrap

Se puede decir que Bootstrap es un producto de código abierto el mismo que fue creado por Mark Otto y Jacob Thornton, esta herramienta nace con la necesidad de estandarizar el conjunto de herramientas para el trabajo que se realiza en el front end.

El objetivo principal de Bootstrap es facilitar al desarrollador el diseño web permitiendo la creación de páginas webs adaptables, esto quiere decir que se debe ajustar a cualquier dispositivo y también a cualquier tamaño de pantalla (Spurlock, 2013, p. 1).

### 2.4.1 Estructura de un archivo Bootstrap

Un archivo Bootstrap tiene una estructura bien definida que hace posible el buen funcionamiento en el diseño de las páginas web. A continuación, se presenta la estructura que tiene un archivo Bootstrap.

```
bootstrap/
  |-- css/
  |   |-- bootstrap.css
  |   |-- bootstrap.min.css
  |-- js/
  |   |-- bootstrap.js
  |   |-- bootstrap.min.js
  |-- img/
  |   |-- glyphs-halflings.png
  |   |-- glyphs-halflings-white.png
  |-- README.md
```

**Gráfico 3:** Estructura de Bootstrap

**Fuente:** (Spurlock, 2013, p. 2)

El Gráfico 3 se puede apreciar la estructura que tiene el archivo de Bootstrap, el mismo que incluye las carpetas de css, js, e img. En la carpeta denominada css se contiene las diferentes hojas de estilo haciendo posible el diseño de la web; en la carpeta llamada js se encuentra el java script y por último en la carpeta img se localiza todos los iconos que

se pueden contener en un sitio web. Es de mucha importante mencionar que durante la etapa de desarrollo se pueden utilizar los archivos no comprimidos, pero para la etapa de producción de la web desarrollada se puede incluir los archivos comprimidos (bootstrap.min.css, bootstrap.min.js) (Spurlock, 2013, p. 2).

## 2.4.2 Ventajas de usar Bootstrap

Dentro de las ventajas principales en el momento de usar Bootstrap tenemos las siguientes: (Alba, 2014).

- Este framework tiene un mantenimiento y servicio de actualización.
- Además tiene un paquete de elementos web que pueden ser personalizables.
- Incluye Grid system, el mismo que permite realizar la maquetación de las columnas.
- Las diferentes plantillas son de fácil adaptación responsiva.
- También utiliza el lenguaje Less, teniendo en cuenta que este lenguaje permite mejorar los estilos CSS de la web.
- Tiene una gran variedad de documentación, tutoriales y plugins los mismos que enseñan a utilizar con facilidad.

## 2.5 Metodología ágil SCRUM

### 2.5.1 Definición

Se considera como una metodología ágil y flexible de desarrollo de software, la misma que permite la construcción de un producto incrementalmente en ambientes donde los requisitos cambian constantes. El objetivo principal de la metodología SCRUM es generar un proceso beneficioso en el momento de la elaboración de los proyectos informáticos, los mismos que se trabajan en equipos pequeños con requisitos poco estables y con iteraciones cortas (Laínez Fuentes, 2015, p. 127).

### 2.5.2 El equipo SCRUM

Este equipo SCRUM está conformado por tres roles muy importantes los mismos que hacen lo posible para trabajar y así alcanzar el objetivo planificado. A continuación, en la Tabla 2 estos roles son explicados de una mejor manera.

**Tabla 2:** Miembros el equipo SCRUM.

Nombre	Descripción
Dueño del producto (Product Owner)	Este es el encargado de gestionar un listado de trabajo (Product Backlog), el mismo que va expresando de una manera clara cada uno de sus elementos para después ordenarlos de una mejor manera que sea conveniente para así alcanzar las metas establecidas. También se debe tener en cuenta la coordinación del proyecto, el análisis de la visión del producto y la optimización del equipo de trabajo son las tareas en las cuales debe trabajar

	el dueño del producto.
Scrum Master	Este rol tiene como objetivo fundamental asesorar y formar a los miembros del equipo de desarrollo para trabajar de una manera organizada y con responsabilidad. También debe ser capaz de resolver los obstáculos que representen un peligro en la ejecución de un sprint (iteración o entregable).
Equipo de desarrollo (Development Team)	Este equipo está conformado por diferentes profesionales que hacen posible la entrega de una parte del producto, que sin ningún problema puede ser ubicado en la etapa de producción al finalizar cada iteración o entregable (Schwaber & Sutherland, 2017, p. 6)

**Realizado por:** Campos David

### 2.5.3 Artefactos de SCRUM

Los conocidos artefactos Scrum son los elementos físicos que se generan para garantizar la transparencia de información y permitir la oportunidad de la inspección y adaptación. Dentro de la metodología Scrum se consideran tres artefactos de suma importancia que son los siguientes:

#### 2.5.3.1 Lista de Producto (Product Backlog)

Esto hace referencia a una lista de ítems los mismos que representa a los requisitos que generalmente el cliente ha solicitado y marca todo el periodo completo de trabajo. (Requena Mesa, 2018)

Cada uno de estos requisitos deben ser descritos los mismos que incluyen, la descripción, el orden, la estimación y el valor, permitiendo agrupar y priorizar los elementos de la lista de producto. El equipo desarrollador es el encargado de la estimación de los elementos de la lista de producto, ya que son ellos los que van a llevar a cabo el trabajo y son los que se comprometen a cumplir el objetivo de cada Sprint (Ramos Vega, 2017).

#### 2.5.3.2 Lista de Pendientes del Sprint (Sprint Backlog)

Este tipo de artefacto consiste en un subconjunto de elementos elegidos, junto con un plan para entregarlos como incremento y conseguir el objetivo principal del Sprint.

Esta lista de pendientes del Sprint está establecida dentro del plan total de lista de producto o Product Backlog, ya que representa una estimación de las funcionalidades que formarán parte de un sprint. Mediante el Sprint Backlog se hace notorio el trabajo del equipo desarrollador incluyendo las distintas mejoras a los procesos de mayor prioridad (Schwaber & Sutherland, 2017).

#### 2.5.3.3 Incremento

Este tipo de incremento es una parte de software acabado al final de cada sprint. Esto se puede apreciar que el incremento es la suma de las tareas, casos de uso, historias de usuario y cualquier elemento que se ha elaborado durante el Sprint y que será puesto a disposición del usuario final. (Roche, 2016).

## 2.5.4 Eventos SCRUM

Estos tipos de eventos que tiene Scrum son considerados como fases o bloques de tiempo muy bien definidos teniendo como objetivo crear regularidad y minimizar la necesidad de las reuniones que no están definidas. Es considerado el Sprint como el evento principal que contiene a los demás, así constituyendo una oportunidad formal para la revisión y adaptación de cualquier faltante en el desarrollo del producto de software. (Schwaber & Sutherland, 2017, p. 9). En la Tabla 3 se presenta los eventos importantes que componen la metodología y son:

**Tabla 3:** Eventos de SCRUM

<b>Evento</b>	<b>Descripción</b>
Sprint	Este Comprende un periodo máximo de tiempo el mismo que es recomendado 1 mes, en la duración de cada Sprint se crea un aumento de producto de software, el mismo que debe ser funcional y utilizable y listo para ser puesto durante la etapa de producción.
Reunión de planificación de Sprint	Son reuniones que se ejecuta durante el inicio de cada Sprint del proyecto, para elaborar las planificaciones de 30 días de trabajo, además en esta reunión debe tener una duración máxima de 8 horas. Es de suma importancia tener en cuenta que si el sprint es más corto lógicamente el periodo de tiempo debe ser más corto. (Palacios, 2016)
Objetivo del Sprint	Este representa a la meta que se desea alcanzar con todo el equipo de trabajo mediante la implementación de ítems que indican a un entregable funcional del producto software (Revueltas, 2018).
Scrum Diario	Esto es apreciado como una guía entre los integrantes del equipo para alcanzar el objetivo deseado del Sprint. Se dice que esta reunión diaria es corta y el tiempo máximo de duración es de 15 minutos los mismos que sirve como un lineamiento para que el equipo mejore el entendimiento compartido de los aspectos más relevantes que van a ser realizadas. (Francia, 2017)
Revisión de Sprint	En este tipo de revisión del sprint tiene como finalidad presentar el entregable correspondiente finalizado sin olvidar la lista de requerimientos para las próximas presentaciones de los avances del proyecto que esta en ejecución. (Ramos Vega, 2017)
Retrospectiva de Sprint	Esto Permite a todo el equipo Scrum examinarse a sí mismo para poder crear un plan de mejoras que será puesto en marcha de inmediato en el siguiente Sprint del proyecto.

**Realizado por:** Campos David

## 2.6 Modelo Verificación y validación

Este aseguramiento de la calidad incluye la realización de casos de prueba a un producto software, permitiendo consolidar un porcentaje alto de funcionalidad y exactitud. Las pruebas unitarias que comprenden el enfoque de este trabajo de titulación son utilizadas en una etapa específica en el ciclo de vida del desarrollo del software, esto es, el diseño de los módulos de un sistema, para la comprensión adecuada de este nivel de pruebas se expone brevemente el modelo de verificación y validación (V&V). (Agarwal, et al., 2010)

Es importante conceptualizar brevemente los procesos de verificación y validación por separado, como consecuencia también se expone una definición acertada de este modelo, esto incluye las respectivas etapas con el nivel de prueba correspondiente que las validan.

En los siguientes puntos, se expone lo anteriormente mencionado con el detalle necesario para establecer un lineamiento claro sobre los casos de test unitarios.

### 2.6.1 Verificación.

La verificación es un proceso de valoración de un sistema o componente teniendo como objetivo determinar si un producto en cada etapa de desarrollo compensa las condiciones establecidas al inicio de la misma. Se considera una técnica de análisis estático que se realiza sin ejecutar ningún tipo de código.

### 2.6.2 Validación

Esta se considera como una técnica de análisis dinámico, donde puede ser funcional y no funcional los mismos que involucra ejecutar pruebas mediante la ejecución de código. Este es considerado como un proceso para evaluar el software que se ejecuta después de la finalización de una etapa en el ciclo de vida de desarrollo, al final esto permite verificar si se está cumpliendo con las perspectivas y los requisitos del cliente (Mera-Paz, 2016).

### 2.6.3 Modelo V&V

Este tipo de modelo es considerado como de verificación y validación (V&V), el mismo que representa un tipo de ciclo de vida de desarrollo de software. Su objetivo es permitir la creación de una manera secuencial de estados o fases por los cuales la construcción de un sistema informático avanza.

Esta basado en la agrupación de las fases de pruebas para cada etapa de desarrollo correspondiente. La fase siguiente comienza después de completar la fase anterior, para cada una de las actividades de desarrollo existe una actividad de pruebas específica. (Pauta Ayabaca & Moscoso Bernal, 2017).



**Gráfico 4:** Modelo V&V  
**Fuente:** (Mera-Paz, 2016)

En el Gráfico 4, se presenta el modelo de verificación y validación V&V para el desarrollo de software, el cual se encuentra en la izquierda en sentido descendente se puede ver cada una de las fases que se lleva a cabo en la construcción de un sistema, cada etapa de la elaboración es comprobada con la siguiente etapa inmediata y además es validada con los diferentes tipos de pruebas que se encuentran en el lado derecho de la figura.

A continuación, se da una breve explicación de cada una de estas etapas (Kumar, 2017)

- **Análisis de requerimientos:** en esta etapa es definido todos los requerimientos y las expectativas que tiene el usuario con relación al software. Para eso es recomendable una buena comunicación y el uso de estándares.
- **Diseño del sistema:** en esta etapa se tiene el diseño del sistema, y además también la configuración completa de comunicación.
- **Diseño de la arquitectura:** durante esta etapa el diseño del sistema es apartado en módulos que tienen diferentes funciones, además se tiene en cuenta la transferencia de los datos y la comunicación de los módulos con el mundo exterior los mismos que están claramente definidos.
- **Diseño de los módulos del sistema:** en esta etapa se permite definir o separar el diseño del sistema en algunos módulos pequeños, a esto se le conoce cómo diseño detallado o también es considerado como un diseño de bajo nivel (Low-Level Design).
- **Codificación:** en esta etapa comprende el desarrollo del sistema de según el diseño establecido, haciendo uso de un lenguaje de programación y herramientas informáticas los mismos que conjuntamente hagan posible la consecución de un software según las especificaciones establecidas anteriormente.

La fase de pruebas de dicho modelo está en la parte derecha de la Figura 4, el mismo que debe ser comprendido cada etapa, los mismos que valida cada una de las fases de desarrollo.

a continuación, se tiene presenta una breve explicación de cada una de las etapas de pruebas del software, entre ellas tenemos (Kumar, 2017).

- **Pruebas unitarias:** este tipo de pruebas son ejecutadas durante la fase del diseño de los diferentes módulos del sistema. Este tipo de pruebas unitarias son ejecutados para eliminar algunos tipos de errores a nivel de unidades de código.
- **Pruebas de integración:** este tipo de pruebas denominadas de integración son ejecutadas en la fase de diseño de la arquitectura del sistema, después de ejecutar

las pruebas unitarias. Permitiendo verificar la comunicación que existe entre los diferentes módulos del sistema los mismos que están integrado.

- **Pruebas del sistema:** este tipo de pruebas sirven para validar la aplicación completa con toda su funcionalidad, interdependencia y comunicación entre los módulos del sistema. Además aquí se comprueba los requerimientos funcionales y no funcionales del sistema que fue desarrollada.
- **Pruebas de aceptación:** Este tipo de pruebas son desarrolladas con el usuario final similar a un ambiente de producción, teniendo como objetivo principal validar si el sistema entregado cumple con todos los requisitos del usuario y que el sistema está apto y listo para su uso en el mundo real.

Las pruebas unitarias son un nivel de pruebas que de acuerdo a lo expuesto en el modelo en V tienen su aplicación en los módulos del sistema. Este proyecto técnico es planificado con la metodología ágil de desarrollo Scrum, cada historia de usuario identificada es considerada como una unidad, por lo tanto, las pruebas unitarias son incluidas como historias técnicas. Por otra parte, formular los enfoques de las pruebas de software junto con los niveles de aplicación también permiten tener aún mayor comprensión de realizar pruebas, especialmente las que comprenden el nivel de unitarias.

## 2.7 Pruebas de Software

Una prueba de software es considerada como un proceso que permite asegurar la entrega del producto al cliente libre de defectos. Es decir, es una actividad que con circunstancias previamente especificadas todo el sistema o uno de sus componentes son ejecutados. Para realizar una evaluación de un criterio de calidad específico, los resultados de las pruebas son registrados y documentados. (Pauta Ayabaca & Moscoso Bernal, 2017, p. 27)

Las pruebas de software deben cumplir con los siguientes objetivos (Pauta Ayabaca & Moscoso Bernal, 2017, p. 27)

- Encontrar defectos mediante la información obtenida producto del seguimiento y documentación de una prueba.
- Brindar confianza al cliente y al equipo de desarrollo de un producto software debido a que se asegura un nivel aceptable de calidad previniendo y corrigiendo errores.
- Como se ha expuesto los tipos de pruebas que pueden ser aplicados en el modelo de verificación y validación, las pruebas sirven para metas específicas que varían dependiendo de cada etapa de un proceso involucrado.



Siguiendo el modelo de desarrollo de verificación y validación se abordará brevemente dos enfoques de pruebas de software importantes para la consecución de los objetivos planteados.

### **2.7.1 Enfoques de pruebas de software**

En las pruebas de software, que consiste en la técnica de ejecutar un conjunto de condiciones previamente establecidas junto con el propósito de validación de una etapa específica en el desarrollo de un sistema informático, es importante asegurar la funcionalidad como atributo de calidad, para ello se considera dos enfoques (Ghahrai,2018)

- Pruebas de caja negra: consisten en el hecho de ingresar datos de entrada y verificar los resultados para comprobar si la funcionalidad probada funciona correctamente.
- Pruebas de caja blanca: puede ser diseñado y ejecutado para evaluar una unidad que compone el sistema o un componente del sistema.

En los siguientes puntos se describe de manera acertada los enfoques de software antes mencionados.

### **2.7.2 Pruebas unitarias.**

La idea principal de las pruebas unitarias se empieza a tener en cuenta desde muchos años atrás específicamente desde el año de 1976, cuando David J. Panzal expuso su trabajo llamado “Procedures: A New Approach to Software Verification” en la que se llevo a cabo la segunda edición de la conferencia internacional de la ingeniería de software, este trabajo explica el procedimiento que se debe seguir en las pruebas como una manera de llamar casos en un módulo respectivo para generar un reporte detallado indicando cuando el caso de prueba haya fallado y su respectivo porcentaje de error que tiene. (Gulati & Sharma, 2017, p. 5)

#### **2.7.2.1 Definición**

Este tipo de pruebas unitarias son técnicas que se utilizan con el objetivo de comprobar la funcionalidad de unidades de código que se encuentra en el sistema. Estas pruebas en el nivel de unidades de código son ejecutadas en las diferentes clases y los métodos los mismos que captura los resultados de los mismos. El propósito principal de las pruebas unitarias es revisar cada sección del código y posteriormente a verificar si devuelve los resultados correctos que se encuentra en el código. (Dietrich, 2014)

#### **2.7.2.2 Características que tiene las pruebas unitarias**

Este tipo de pruebas unitarias cuentan con las siguientes características que son de mucha

relevancia y son las siguientes (Dietrich, 2014).

- **Automatizable:** los resultados son definidos para cada uno de los casos de pruebas, además este tipo de pruebas unitarias se pueden automatizar para alcanzar los resultados deseados de forma individual o grupal.
- **Legible:** el objetivo principal de una prueba unitaria es indicar al tester sobre lo que hace la unidad de código, si la prueba unitaria no es legible, el tester encargado no sabrá cuando dicha prueba unitaria a fallado.
- **Rápido:** los códigos para realizar las pruebas unitarias no deben tomar más de 5 minutos para ser creados y ejecutados.
- **Independiente y aislado:** estas pruebas unitarias trabajan independientemente de otros diferentes casos de prueba, teniendo un ambiente único para cada caso.
- **Correcto:** una prueba unitaria hace exactamente lo que tiene que hacer, un diferente caso de prueba corresponde a un caso único.
- **Repetible:** cuando exista el caso de repetir las pruebas unitarias de manera individual o grupal, el resultado final siempre debe ser el mismo dando igual el orden en que se realicen los test, las pruebas unitarias se guardan para poder realizar estas repeticiones y así usarlas en otras ocasiones.

### 2.7.2.3 Beneficios de las pruebas unitarias

Los beneficios que tiene este tipo de pruebas unitarias al momento de realizarlas a nuestro software son: (Apiumhub, 2017)

- **Comprueba las especificaciones:** antes de empezar con su respectiva codificación de los componentes es necesario tratar de determinar que va a hacer cada uno de los componentes. Además se debe tratar de construir un caso de prueba unitaria con todas las posibles entradas y salidas que tiene el software para de esa manera clarificar el comportamiento esperado de cada uno de los componentes que hay.
- **Proporciona una anticipada detección de errores:** las pruebas unitarias al momento de ser ejecutadas pueden identificar fallas en el primer momento, además, permiten saber si un componente está completo o a la vez incompleto.
- **Da mantenimiento de soporte:** teniendo un conjunto de pruebas unitarias que estén bien escritas esto sirve como un aumento a la productividad para el equipo de desarrollo, debido al aumento de los requerimientos en todo el tiempo.

- Pueden mejorar el diseño: estas pruebas unitarias permiten perfeccionar las especificaciones del sistema mediante la exhibición de las fallas en el diseño de la interfaz del software.
- Documentación del producto: este tipo de pruebas unitarias enseña cómo trabaja una línea de código, de una misma manera sincronizada se guarda en memoria el último estado de la especificación junto con los cambios del código.

Las pruebas unitarias al ser ejecutadas permiten conocer la cobertura de una unidad de código que pertenece a una funcionalidad específica del sistema, por lo tanto, basado en este estudio se puede concluir la exactitud que tiene un caso de prueba específico.

## **2.8 Herramientas para la automatización de las pruebas unitarias**

### **2.8.1 Junit.**

Esta herramienta JUnit es un framework de código abierto el mismo que se utilizó para ejecutar las pruebas unitarias en el lenguaje Java. Esta herramienta es demasiado útil para todos los desarrolladores al momento de usarlas en el lenguaje Java cuando se trata de escribir y ejecutar pruebas repetibles dentro de un software. (Pandya, 2017)

Esta herramienta fue desarrollada por Kent Beck y Erich Gamma, teniendo como referencia a SUnit, el cual es un framework de pruebas unitarias escrito en Smalltalk.

Esta versión de la herramienta de JUnit fue desarrollada en el año de 1997 y con el pasar de tiempo se ha transformado en un estándar para diferentes lenguajes y muchas herramientas. (Gulati & Sharma, 2017)

Este tipo de herramienta se puede agregar a diferentes tipos de entornos integrados de desarrollo de software, entre los cuales los más utilizados son: Eclipse y Netbeans.

El framework de JUnit incluyen características notorias que les hace únicas a comparación de las demás entre estas tenemos las siguientes: (JUnit, 2018)

- Tiene afirmaciones para comprobar todos los resultados esperados del sistema.
- Proporciona elementos de prueba para compartir datos comunes de las pruebas.
- Son ejecutores de los casos de prueba automatizados.

Al momento de hacer uso de este framework de Junit en nuestro sistema se tiene una gran variedad de ventajas entre las principales tenemos: (Pandya, 2017)




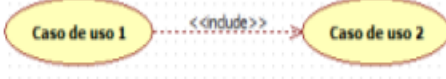

- Los diferentes tipos de errores son detectados en una etapa temprana la misma que permite que el código sea más confiable.
- Tiene una gran utilidad este framework en diferentes entornos de trabajo los mismos que estos se basen en pruebas.

- Las pruebas unitarias exigen a grupo de desarrolladores a leer el código más que a su escritura.
- Al momento de usar este framework admite escribir un código más legible confiable y lo más importante libre de errores lo que implica mayor confianza durante todo el transcurso del desarrollo del sistema.

## 2.9 Diagrama de casos de uso

En el presente proyecto se describe los procesos de los tres roles principales del sistema, donde se hace uso del lenguaje de modelado unificado o conocido como UML. Donde en este apartado se indican todos los diagramas de caso de uso de los usuarios que participan en el sistema, los mismos que son: productor, coordinador y locutor. Este tipo de diagramas de casos de uso explica las acciones que el sistema debe cumplir. A continuación, en la Tabla 4 se da una breve explicación de la simbología utilizada.

**Tabla 4:** Simbología UML.

Símbolo	Nombre	Descripción
	Actor.	Este permite representar a los roles de los usuarios los mismos que hacen uso del sistema.
	Caso de uso	Esta simbología representa una acción específica que el sistema va a realiza
	Relación de asociación	Esto representa las distintas relaciones los mismos que están asociados entre el actor y un caso de uso.
	Relación de inclusión	Esta simbología señala que en un caso de uso incluye dos o más.
	Relación de dependencia	Este tipo de refacion indica que un caso de usos puede depender de una o varias acción que lo precede.

**Realizado por:** Campos David

## CAPÍTULO III

### 3 METODOLOGÍA

#### 3.1 Tipo y diseño de la investigación

El proyecto de investigación este basado en un enfoque cualitativo, para la validación del sistema web nos apoyaremos en pruebas unitarias. Este tipo de pruebas en esencia son fragmentos de código diseñados para validar que una parte específica del sistema desarrollado devuelva los valores exactos para los cuales fue creado.

- **Investigación exploratoria** se usó este tipo de investigación porque responde a las siguientes preguntas: para qué, cuál es el problema y qué se podría investigar, sobre un problema en específico planteado en este caso el sistema web para la radio.
- **Investigación descriptiva** se manejó esta técnica para la recolección de requerimientos de la radio, y además saber los usuarios que están implicados en el presente proyecto de la radio, de la misma manera saber también los diferentes roles que tienen los usuarios, las diferentes entidades y funcionalidades propias que se tiene en este proyecto.
- **Investigación aplicativa** se utilizó este tipo de investigación el cual permite usar el conocimiento obtenido para poder alcanzar un objetivo en general. Después de haber realizado la formulación del marco teórico relacionado al desarrollo de dicho software, se continuo con la construcción del sistema para la gestión de guiones y publicidad de la radio. En el presente proyecto se emplea, fundamentos teóricos y además con la ayuda de diferente tipo de tecnologías y de herramientas que se utilizaron para el desarrollo del producto siendo un sistema web.

#### 3.2 Métodos y técnicas

##### 3.2.1 Métodos aplicados en el proyecto

- **Método analítico** Este tipo de método permite detectar las líneas de código de cada funcionalidad del proyecto, para según eso permitir diseñar y ejecutar las pruebas.
- **Método sintético** En el presente proyecto este tipo de método nos ha permitido recolectar características para así poder definir las respectivas funcionalidades que va a poseer el sistema. Los entregables o Sprint del presente proyecto han sido desarrollados siguiendo la orientación del método sintético.

- **Método deductivo** en el desarrollo del proyecto, el planteamiento para los diseños de pruebas unitarias dentro de cada uno de los requisitos funcionales los mismo que fueron identificados para el proyecto.
- **Método inductivo** este tipo de método se utilizó para la elaboración de las diferentes conclusiones que arrojaron los resultados obtenidos en el momento de ejecutar las pruebas en el presenta proyecto.
- **Metodología SCRUM** esta metodología propone la construcción de un producto de manera incremental, los mismos que son planificados cada cierto periodo de tiempo. Al momento que se finaliza el último sprint que estuvo planificado se puede garantizar la entrega de un producto totalmente funcional e integrado.

### 3.2.2 Técnicas

- **Observación** este tipo de técnica tiene como objetivo permitir divisar los diferentes resultados que arrojan las pruebas unitarias durante el desarrollo del sistema de la radio.
- **Entrevista** esta es dirigida al personal encargado de la Radio Ciudad 106.5 FM, para poder obtener la información necesaria sobre los procesos de la gestión de guiones, publicidad y de los programas de radio que se maneja en la entidad.
- **Análisis Documental** con la utilización de dicha técnica se pudo tener diferente información de varias fuentes bibliográficas como por ejemplo de libros, de documentos científicos foros y tesis las mismas que tienen algún tipo de relación con nuestro proyecto, dicha información servirá para la fundamentación teórica y así poder establecer la guía metodológica del proyecto realizado.

## 3.3 Fase de planificación del sistema web

### 3.3.1 Reunión de planificación

Para poder establecer los requerimientos funcionales y no funcionales para el sistema, se realiza una serie de reuniones donde se trabaja conjuntamente con las personas que se encuentran involucradas en el proyecto logrando una respectiva planificación. En la Tabla 5 se presenta el desarrollo de las respectivas actividades del proyecto.

**Tabla 5:** Reuniones de Planificación.

<b>Personal</b>	<b>Actividades</b>	<b>Resultados</b>
- <b>Tutor del Proyecto:</b> PhD. Ximena Quintana López - <b>Encargado de la Radio:</b> Patricio Vargas	Recopilación de información del proceso que se maneja al momento de la gestión de guiones,	Determinación de los diferentes módulos que va a tener el sistema.

- <b>Desarrollador del sistema:</b> David Campos	publicidad y programas de la radio.	
- <b>Tutor del Proyecto:</b> PhD. Ximena Quintana López - <b>Productor de la Radio:</b> Patricio Vargas - <b>Desarrollador del sistema:</b> David Campos	Determinación de las diferentes necesidades que tiene la radio y además el alcance del sistema de la radio.	Descripción de los requerimientos del sistema.
- <b>Tutor del Proyecto:</b> PhD. Ximena Quintana López - <b>Productor de la Radio:</b> Patricio Vargas - <b>Desarrollador del sistema:</b> David Campos	Exposición de los requerimientos y el alcance del sistema.	Aprobación de los diferentes requerimientos y además del alcance que va a tener el sistema de la radio.
- <b>Tutor del Proyecto:</b> PhD. Ximena Quintana López -- <b>Productor de la Radio:</b> Patricio Vargas - <b>Desarrollador del sistema:</b> David Campos	Presentación del avance del sistema al encargado de la radio,	Corrección de errores en las vistas del sistema.

**Realizado por:** Campos David

El alcance del sistema web de gestión de guiones y publicidad de la radio se pudo establecer como alcance los siguientes módulos que se explica a continuación:

- **Módulo de inicio de sesión:** este módulo permite iniciar sesión a los usuarios del sistema, teniendo en cuenta el rol que tiene cada usuario (productor, coordinador, locutor).
- **Módulo de gestión de usuarios:** este módulo tiene como objetivo permitir el ingreso, la modificación y la eliminación de los usuarios de la radio dentro del sistema, según el rol que tenga el usuario de la radio.
- **Módulo de gestión de programas de radio:** la función de este módulo es permitir el ingreso, la actualización y la eliminación de los programas de radio los mismos que son emitidos en la radio.
- **Módulo de gestión de publicidad:** este módulo el encargado de permitir el ingreso, la actualización y la eliminación de las diferentes publicidades que son emitidos en la dicha radio.
- **Módulo de gestión de guiones:** atreves de este módulo se permite el ingreso, la actualización y la eliminación de los guiones de una programación radial, los mismos que son enviados al coordinador para su respectiva revisión.

- **Módulo de gestión de diálogos:** este módulo permite el registro, actualización y eliminación los diálogos los mismos que son asignados al guion de una programación radial, el cual el locutor puede subir al sistema cualquier tipo de archivo y guardarlo en el sistema.
- **Módulo de revisión de guiones:** este módulo permite la revisión de los guiones de una programación radial que el locutor envía para su revisión y aprobación al coordinador de la radio.
- **Módulo de reportes:** este módulo permite realizar los diferentes reportes de los usuarios, programas de radios, guiones y publicidad que se encuentran registrados en el sistema de la radio.

### 3.3.2 Equipo SCRUM del Proyecto

De acuerdo con la metodología de desarrollo de software que se está siguiendo, donde se define los roles teniendo como resultados el Dueño del producto, el Scrum Master y el Desarrollador del sistema para la radio.

A continuación, en la Tabla 6 se puede distinguir el nombre de las personas y sus respectivos roles los mismos que están involucrados en el desarrollo del sistema.

**Tabla 6:** Equipo SCRUM del proyecto.

<b>Personal/Cargo</b>	<b>Rol</b>	<b>Responsabilidad</b>
Productor Patricio Vargas	Encargado de recibir el sistema para la radio	Definir los ítems necesarios los mismos que representan las especificaciones que va a tener el sistema.
PhD. Ximena Quintana	Scrum Master	Es el encargado de brindar la asesoría en todo el transcurso y además hacer el seguimiento del proyecto de la radio.
David Campos	Programador del sistema web para la radio.	Codificar, diseñar y ejecutar el sistema web de la radio.

**Realizado por:** Campos David

### 3.3.3 Tipo de usuario y roles del proyecto

Para poder continuar con el desarrollo del sistema web para la radio se pudo definir tres tipos de roles de usuarios que existe en la radio, hay que tener en cuenta que cada tipo de rol de usuario tienen funciones específicas en la radio.

A Continuación, en la Tabla 7 se explica los diferentes tipos de usuarios con su respectivo rol que se tiene para este proyecto de la radio.



**Tabla 7:** Tipos de usuario y sus funciones.

<b>Tipo de usuario</b>	<b>Funciones</b>
Productor	<ul style="list-style-type: none"><li>- Gestionar los usuarios en el sistema.</li><li>- Gestionar los programas de radio.</li><li>- Gestionar las publicidades de la radio.</li></ul>
Locutor	<ul style="list-style-type: none"><li>- Gestionar los guiones del programa radial.</li><li>- Visualizar la sugerencia del guion por parte del coordinador</li><li>- Modificar el guion.</li><li>- Gestionar los diálogos que se va a asignar al guion.</li></ul>
Coordinador	<ul style="list-style-type: none"><li>- Visualizar el listado de guiones.</li><li>- Revisar un guion de la radio para su posterior aprobación.</li><li>- Aprobar o cancelar un guion de un programa radial.</li><li>- Reportes generales de los usuarios que se encuentran registrados en el sistema</li><li>- Reportes generales de los programas de radio que se encuentran registrados en el sistema.</li><li>- Reportes generales de los guiones que se encuentran registrados en el sistema.</li><li>- Reportes generales de las publicidades que se encuentran registrados en el sistema</li></ul>

**Realizado por:** Campos David

### **3.3.4 Lista del producto (Product Backlog).**

Este tipo de lista contiene todas las historias de usuario, donde de una manera ordenada se describe todos los requerimientos que han sido establecidos para el desarrollo del sistema web de la radio. Para establecer de manera definitiva todos los requerimientos del sistema se organizó varias reuniones con el productor de la radio analizando las necesidades de la institución radial. Para conocer las historias de usuario, ver **ANEXO A**. Para definir este tipo de lista de producto donde esta detallado los requerimientos funcionales y técnicos para lograr esto se utilizó puntos de estimación de acuerdo a la técnica llamada “Talla de la camiseta”. Teniendo en cuenta esta técnica de estimación de los puntos anteriormente mencionada se puede definir las categorías con las siguientes letras: XL, L, M, S, donde nos indica que cada talla significa la duración que debe tener el Sprint o fracción del mismo, cada punto estimado está representado por una hora de trabajo.

En la Tabla 8 se presenta las diferentes tallas de camisetas con las que se hace referencia en el proyecto.

**Tabla 8:** Talla de la camiseta del proyecto.

<b>Talla</b>	<b>Puntos Estimados</b>	<b>Prioridad</b>
XL	60	Muy Alta
L	30	Alta
M	15	Media
S	10	Baja
XS	5	Muy Baja

**Realizado por:** Campos David

A continuación, en la tabla 9 se presenta la lista del producto las mismas que están involucradas en el desarrollo del proyecto. Para un mejor entendimiento se asigna las nomenclaturas HT para hacer referencia a las historias técnicas y la nomenclatura HU para hacer referencia a las historias de usuario, cada ítem tiene su estimación y prioridad.

**Tabla 9:** Lista del producto del proyecto.

<b>Id</b>	<b>Lista de Producto</b>	<b>Puntos Estimados</b>	<b>Prioridad</b>
HT-01	Como desarrollador del proyecto necesito realizar la recopilación de requerimientos del sistema.	L	Alta
HT-02	Como desarrollador necesito realizar la planificación de los entregables del sistema.	M	Media
HT-03	Como desarrollador necesito definir las historias de usuario del sistema de la radio.	M	Alta
HT-04	Como desarrollador necesito realizar la planificación de las pruebas unitarias del sistema.	L	Alta
HT-05	Como desarrollador necesito diseñar la arquitectura del sistema.	XS	Muy Baja
HT-06	Como desarrollador necesito diseñar las interfaces de usuario.	XS	Muy Baja
HT-07	Como desarrollador necesito diseñar la base de datos para el proyecto del sistema.	S	Baja
HT-08	Como desarrollador necesito realizar la implementación de la base de datos.	XS	Muy Baja
HT-09	Como desarrollador necesita realizar la conexión a la base de datos.	XS	Muy Baja
HU-01	Como usuario necesito autenticar las credenciales correctas para poder ingresar al sistema de la radio.	XS	Muy Baja

HU-02	Como Productor del sistema necesito ingresar nuevos usuarios al sistema de la radio.	M	Media
HU-03	Como Productor necesito listar los usuarios que están registrados en el sistema de la radio.	XS	Muy Baja
HU-04	Como Productor del sistema necesito editar la información de los usuarios del sistema.	S	Baja
HU-05	Como Productor necesito eliminar los usuarios del sistema.	M	Media
HU-06	Como Productor necesito ingresar al sistema un nuevo programa de radio.	M	Media
HU-07	Como Productor necesito listar los programas de radio que están registrados en el sistema.	XS	Muy Baja
HU-08	Como Productor del sistema necesito editar la información de los programas de radio que tiene el sistema.	S	Baja
HU-09	Como Productor necesito eliminar los programas de radio del sistema.	M	Media
HU-10	Como Productor necesito ingresar al sistema una nueva publicidad de la radio.	M	Media
HU-11	Como Productor necesito listar las publicidades de la radio que están registrados en el sistema.	XS	Muy Baja
HU-12	Como Productor del sistema necesito editar la información de las publicidades de radio que tiene el sistema.	S	Baja
HU-13	Como Productor necesito eliminar las publicidades que tiene la radio en el sistema.	M	Media
HU-14	Como Locutor necesito ingresar al sistema un nuevo guion.	M	Media
HU-15	Como locutor necesito listar los guiones de la radio que están registrados en el sistema.	XS	Muy Baja
HU-16	Como locutor necesito editar la información de los guiones que está en el sistema.	S	Baja
HU-17	Como locutor necesito eliminar los guiones que tiene la radio en el sistema.	M	Media
HU-18	Como Locutor necesito ingresar al sistema un nuevo dialogo.	L	Alta

HU-19	Como Locutor necesito listar los diálogos que están registrados en el sistema.	XS	Muy Baja
HU-20	Como locutor necesito editar la información de los diálogos que está en el sistema.	S	Baja
HU-21	Como Productor necesito eliminar los diálogos elegidos del sistema.	M	Media
HU-22	Como Coordinador necesito revisar los guiones que el Locutor ingresa al sistema y así poder aprobar o rechazar el guion.	XL	Muy Alta
HU-23	Como Coordinador necesito hacer un reporte general de los usuarios que hay en el sistema de la radio.	M	Media
HU-24	Como Coordinador necesito hacer un reporte general de los programas radiales que hay en el sistema de la radio.	M	Media
HU-25	Como Coordinador necesito hacer un reporte general de los guiones que hay en el sistema de la radio.	M	Media
HU-26	Como Coordinador necesito hacer un reporte general de la publicidad que hay en el sistema de la radio.	M	Media

Realizado por: Campos David

### 3.4 Fase de desarrollo del sistema web.

#### 3.4.1 Arquitectura del sistema

El sistema de la radio está basado en la arquitectura denominada cliente – servidor, teniendo en cuenta que es una aplicación web, la misma que se sigue el patrón que tiene el modelo, vista controlador o MVC.

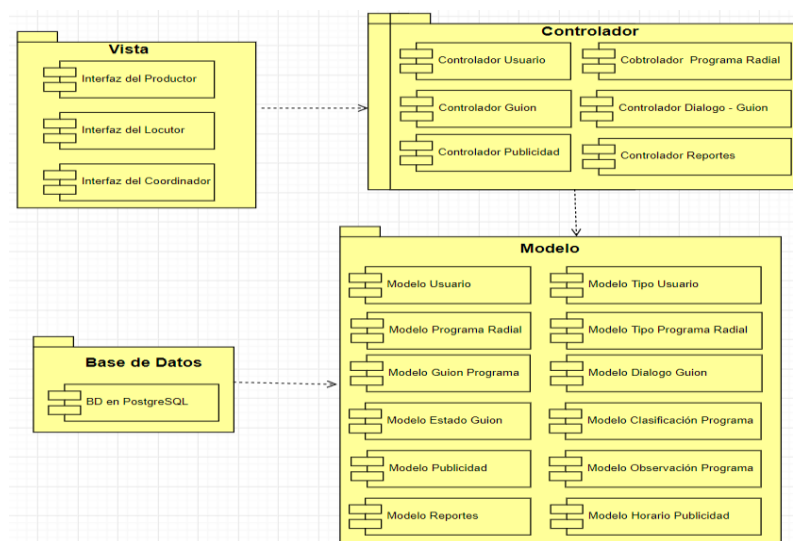


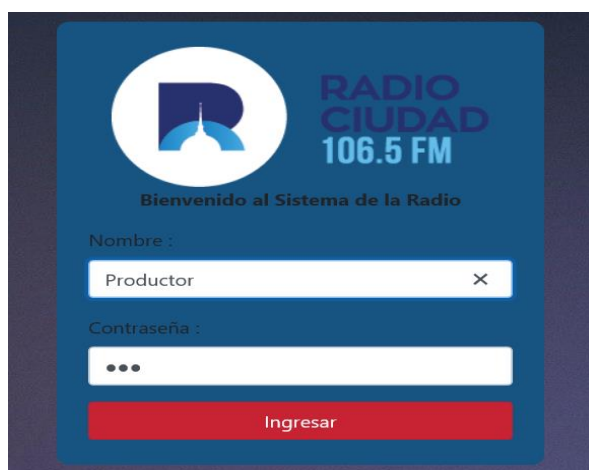
Gráfico 5: Arquitectura del proyecto  
Realizado por: Campos David

En el Gráfico 5 se puede apreciar el esquema general que va a tener el sistema web para gestión de guiones, programas de radio y la publicidad de la radio Ciudad 106.5 F.M, este sistema web está conformado de una base de datos donde será alojada en un servidor de base de datos con conexión a internet, de esa manera permitirá gestionar la información necesaria con respecto a los procesos que se implementará en el sistema web.

En el servidor de aplicaciones se alojará la aplicación desarrollada la misma que se utilizó el lenguaje Java, esto hace referencia al controlador y modelo del sistema, utilizando la tecnología Java Server Face junto con el framework Bootstrap.

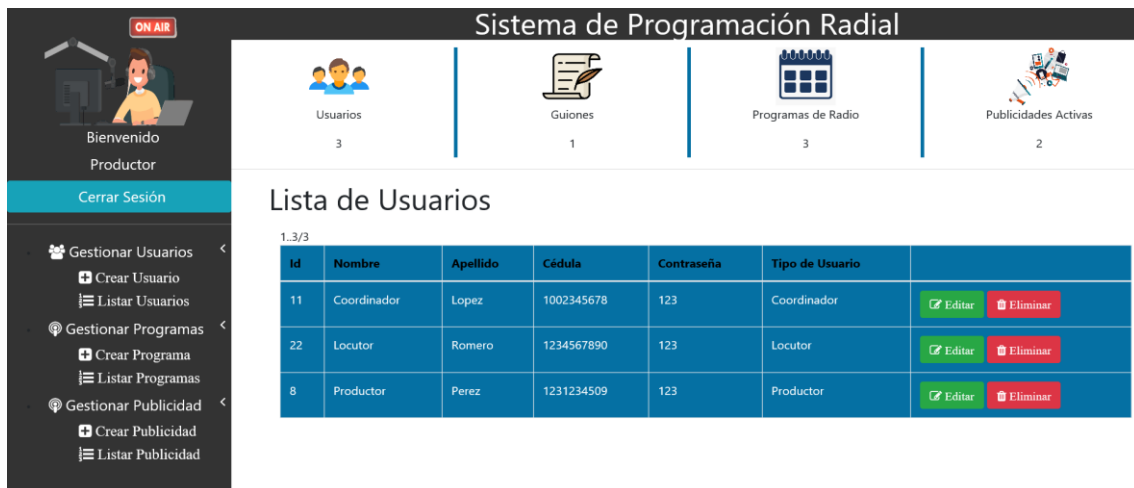
### 3.4.2 Diseño de la interfaz de usuario

Se conoce como interfaz de usuario al conjunto de elementos gráficos como puede ser colores, iconos, botones, etc. Los mismos que hace posible la comunicación del usuario con el sistema web, donde permite realizar tareas y así cumplir con los objetivos para los cuales el sistema ha sido construido sin ningún problema.



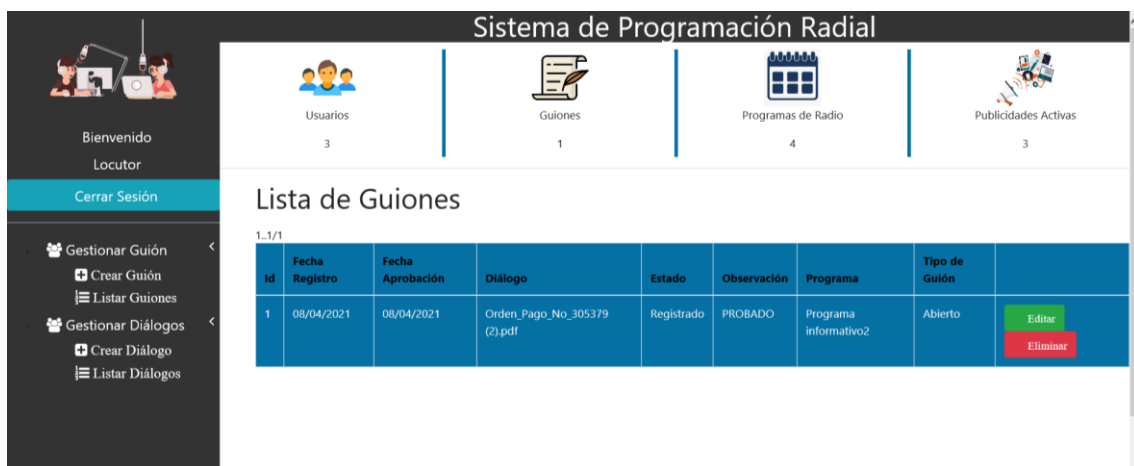
**Gráfico 6:** Pantalla de inicio de sesión del Sistema.  
**Realizado por:** Campos David

En el Gráfico 6 se muestra la pantalla de inicio de sesión al sistema de la radio, esta es la pantalla de inicio la que por defecto aparecerá al inicio. Donde que para entrar al sistema debe digitar las credenciales de cada tipo de usuario donde será verificadas y si es correcto las credenciales, se tendrá el acceso a diferentes módulos que tiene el sistema de acuerdo con el rol de usuario que se desee ingresar.



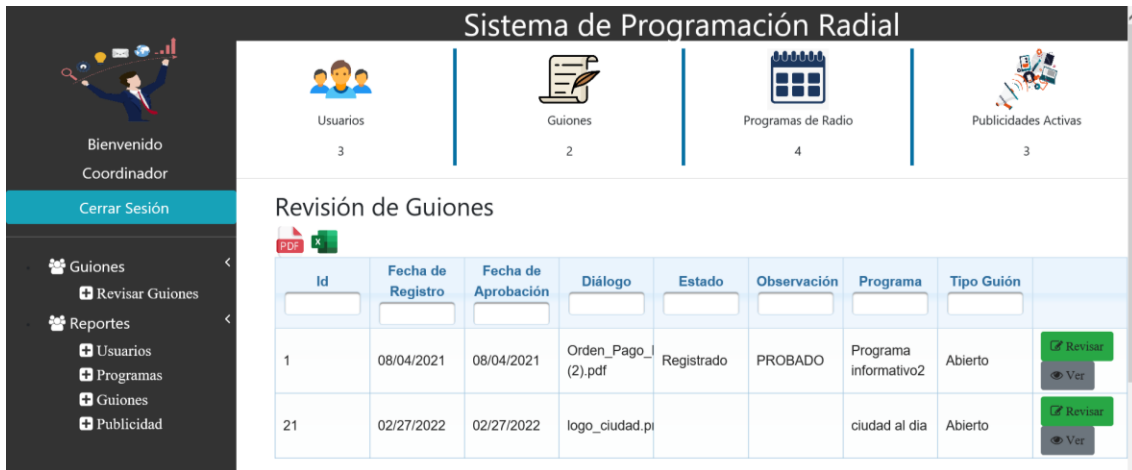
**Gráfico 7:** Pantalla de gestión del Productor.  
**Realizado por:** Campos David

En el Gráfico 7 indica la pantalla con los diferentes tipos de módulos los mismos que son propios del productor del sistema, se puede ver las diferentes opciones como son gestionar usuario, gestionar programas y gestionar la publicidad de la radio. El mismo que se tiene las opciones de crear, modificar listar y eliminar los registros.



**Gráfico 8:** Pantalla de gestión del locutor.  
**Realizado por:** Campos David

En el Gráfico 8 muestra la pantalla del locutor de la radio con los módulos de trabajo respectivos. El panel del locutor permite la gestión de guiones de radio y la gestión de los diálogos los mismos que son asignados a los guiones. Estos módulos tienen las opciones de crear, modificar listar y eliminar los registros del sistema.

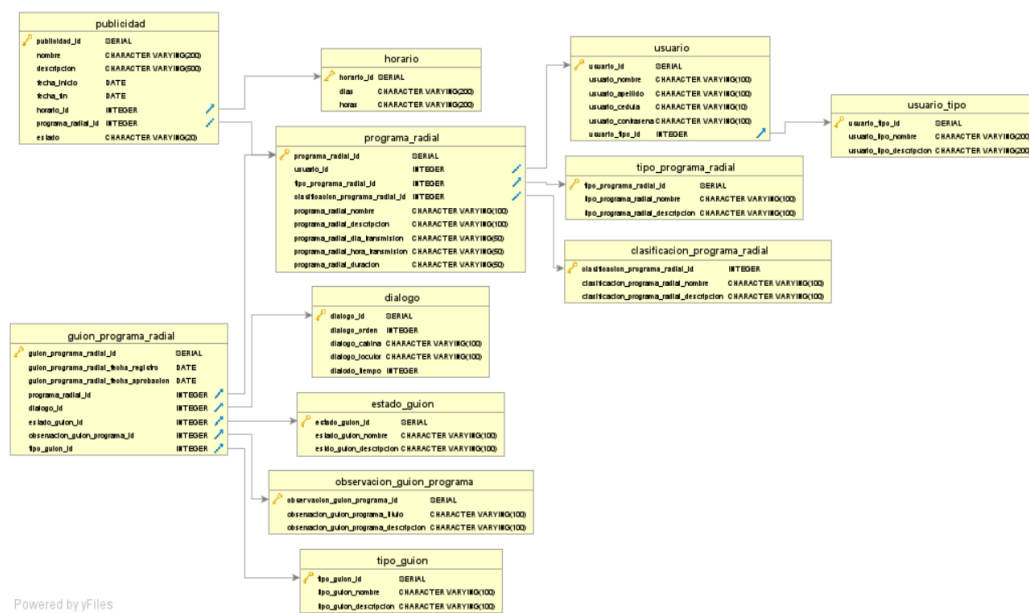


**Gráfico 9:** Pantalla de gestión del Coordinador  
**Realizado por:** Campos David

En el Gráfico 9 muestra la pantalla del Coordinador de la radio con los módulos de trabajo respectivos. El panel del Coordinador permite la revisión de guiones que el locutor ingresa al sistema de la radio, donde puede poner las observaciones del guion. Además, la gestión de reportes generales de: Usuarios, Programas, Guiones y Publicidad.

### 3.4.3 Diseño de la base de datos

La base de datos permite almacenar grandes cantidades de información, teniendo en cuenta que las bases de datos que más se utiliza son las denominadas relacionales. La base de datos para el sistema de la Radio Ciudad ha sido posible implementar en PostgreSQL, teniendo como un resultado un total de 12 tablas principales, las mismas que son relacionadas entre sí como se muestra a continuación en el gráfico 10.

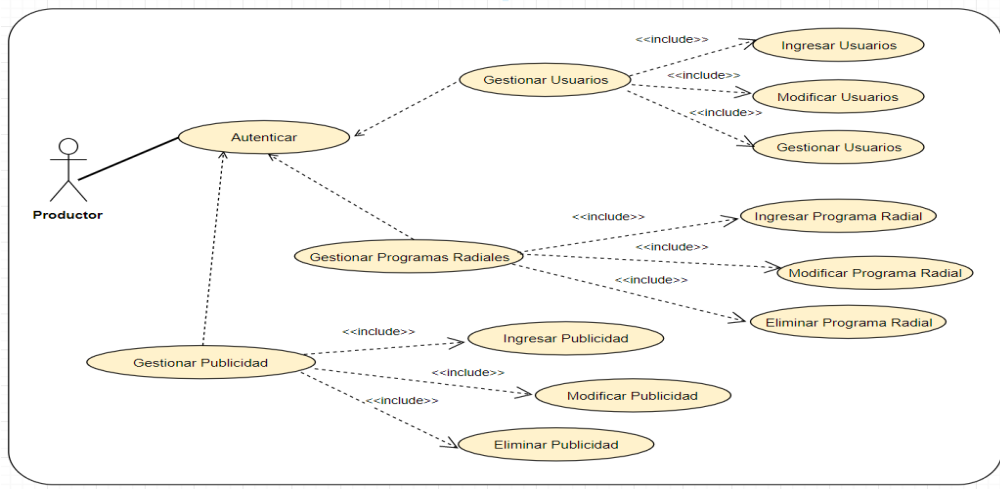


**Gráfico 10:** Base de datos de sistema de la Radio  
**Realizado por:** Campos David

### 3.4.4 Diagrama de casos de uso

#### 3.4.4.1 Procesos del Productor del sistema

Este tipo de diagramas de casos de uso tiene como objetivo indicar el comportamiento del sistema con relación del rol de usuario en este caso cuando el productor de la radio acceda en el sistema.

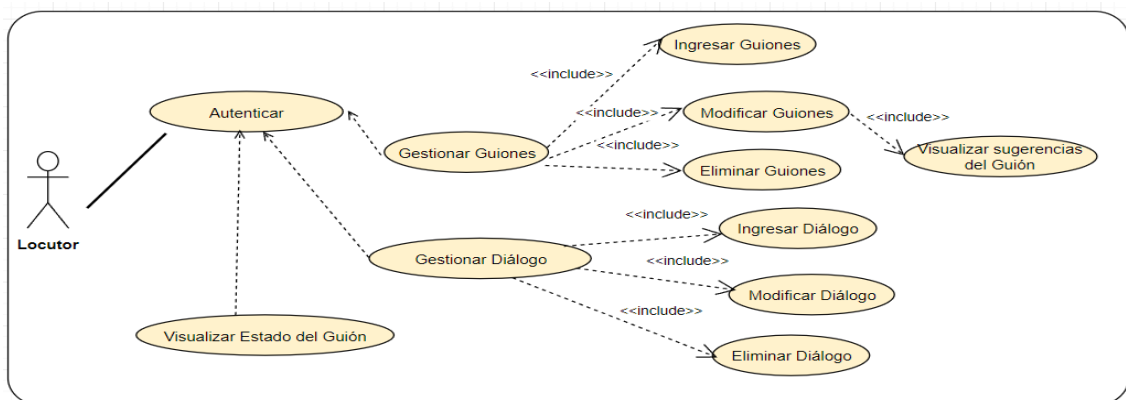


**Gráfico 11:** Diagrama del productor del sistema.  
**Realizado por:** Campos David

En el Gráfico 11 explica el diagrama de caso de uso del rol de productor de la radio. Se indica las diferentes acciones que tiene este rol como son: la gestión de los usuarios, la gestión de programas de radio y la gestión de publicidades. Estas acciones dependen del inicio de sesión según el rol del usuario. Cualquier categoría de este segmento incluye diferentes acciones como son: ingresar, listar, modificar y eliminar la información.

#### 3.4.4.2 Procesos del locutor de la radio

En este tipo de diagrama de casos de uso nos indica las diferentes acciones que el sistema tiene según el rol de usuario denominado locutor de la radio.



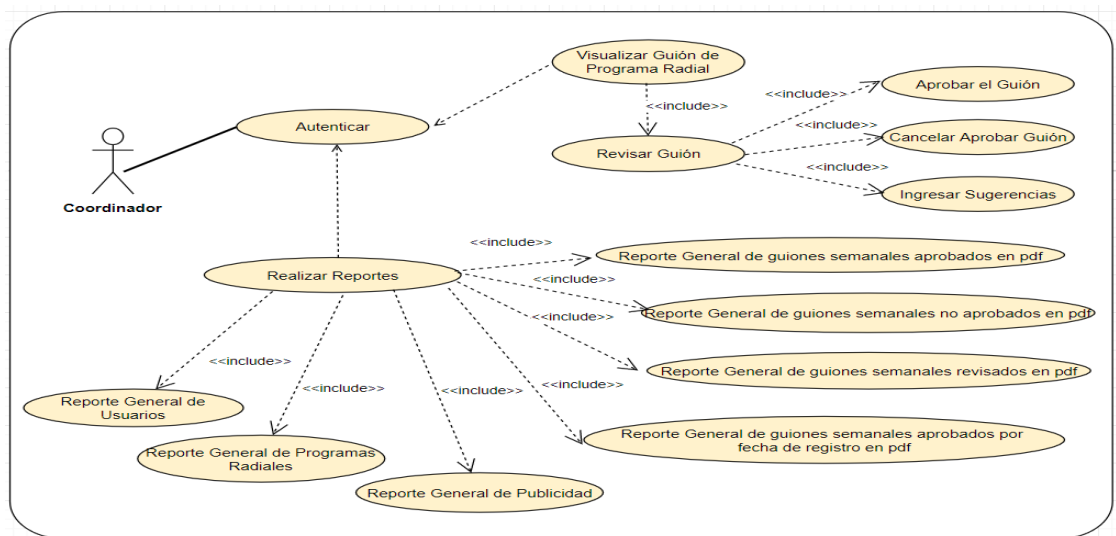
**Gráfico 12:** Diagrama del locutor del sistema  
**Realizado por:** Campos David



En el Gráfico 12 nos explican las acciones que va a tener el sistema teniendo en cuenta el rol de usuario en este caso del locutor de la radio, este tipo de acciones son: la gestión de guiones y la gestión de diálogos los mismos que son asignados a los guiones.

### 3.4.4.3 Procesos del coordinador de la radio

Este tipo de diagrama de casos de uso nos indica las diferentes acciones que cumple el módulo el mismo que pertenece al rol de usuario denominado coordinador. Entre estas acciones tenemos: la revisión de guiones y los reportes generales de los usuarios, programas, guiones y publicidad que están en el sistema.



**Gráfico 13:** Diagrama del coordinador del sistema.  
Realizado por: Campos David

El Gráfico 13 se visualiza el proceso de revisión de guion, así como también e proceso de los reportes generales del sistema.

### 3.4.5 Historias de usuario

Estas historias de usuario nos indican los requisitos que han sido definidos por el dueño del producto, se está utilizando las diferentes recomendaciones que nos brinda la metodología scrum. Donde cada historia de usuario nos facilita ver en forma detallada las funcionalidades durante el desarrollo del proyecto.

En la Tabla 10 se presentan las historias de usuario, el mismo que consta de: tipo de historia de usuario el mismo que puede ser de funcionalidad o técnica, Consta el ID de identificación, la descripción de la historia de usuario, también la asignación del usuario, el sprint en cual están siendo ejecutadas y puntos de estimación, además consta de una descripción y observaciones respectivas para cada historia de usuario.

**Tabla 10:** Historia de usuario del proyecto.

<b>Historia de Usuario</b>		
<b>ID:</b> HU-01	<b>Descripción:</b> Como usuario necesito autenticar mis credenciales para poder ingresar al sistema de la radio.	
<b>Usuario:</b> Locutor, productor, coordinador	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como usuario necesito autenticar las credenciales correctas para poder ingresar al sistema de la radio.		
<b>Observación:</b> El sistema debe permitir el acceso a las diferentes tipos de ventanas los mismos que corresponden según los roles asignados en este caso que son: productor, locutor y coordinador.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Al momento de ingresar las credenciales correctas, el sistema debe permitir el ingreso correcto.</li> <li>○ Al ingresar las credenciales incorrectas, el sistema debe emitir el mensaje respectivo.</li> </ul>		

**Realizado por:** Campos David

Para el desarrollo del este proyecto de investigación consta un total de 26 historias de usuario las mismas que representan a las diferentes funcionalidades que debe tener dicho sistema. Para más de estas historias de usuario ver en el **ANEXO A**.

### 3.4.6 Historias Técnicas

Este tipo de historias técnicas fueron definidas con el propósito de poder realizar la respectiva planificación de la creación del presente proyecto. A continuación, en la Tabla 11 nos permite apreciar la documentación de una historia técnica la misma que se utiliza para el desarrollo de dicho proyecto para la radio.

**Tabla 11:** Historias técnicas.

<b>Historia Técnica</b>		
<b>ID:</b> HT-01	<b>Descripción:</b> Como desarrollador es importante realizar la recolección de los diferentes requerimientos del sistema.	
<b>Usuario:</b> Desarrollador.	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 30
<b>Descripción:</b> Como desarrollador necesito realizar la recopilación de los diferentes requerimientos que el sistema debe tener.		
<b>Observación:</b> Se debe tener una serie de reuniones con todas las personas involucradas para el desarrollo del producto.		

**Realizado por:** Campos David

El desarrollo del presente proyecto involucra 9 historias técnicas que representan las funcionalidades del sistema. Para más historias técnicas ver el **ANEXO B**.

### 3.4.7 Sprint del Proyecto

Una vez realizado los respectivos análisis, se ha definido la respectiva lista de producto para el desarrollo del proyecto, además se establece un plan de entregas para el respectivo proyecto. En el dicho plan de entregas se detalla el alcance de las historias de usuarios y las historias técnicas donde consta la fecha de inicio y la fecha de culminación de cada sprint.

**Tabla 12:** Sprint Backlog del proyecto.

<b>Sprint</b>	<b>Id</b>	<b>Fecha Inicio</b>	<b>Fecha fin</b>	<b>Horas</b>
Sprint 1	HT-01, HT-02, HT-03, HT-04, HT-05 HT-06, HT-07, HT-08, HT-09	04/10/2021	22/10/2021	125
Sprint 2	HU-01, HU-02, HU-03, HU-04, HU-05, HU-06, HU-07, HU-08, HU-09, HU-10, HU-11, HU-12, HU-13	25/10/2021	24/11/2021	150
Sprint 3	HU-01, HU-14, HU-15, HU-16, HU-17, HU-18, HU-19, HU-20, HU-21	29/11/2021	18/12/2021	125
Sprint 4	HU-01, HU-22, HU23, HU-24, HU-25, HU-26	03/01/2022	17/01/2022	125

**Realizado por:** Campos David

En la Tabla 12 se expone el Sprint Backlog del proyecto. Se puede ver los diferentes campos como son: Sprint el mismo que hace referencia a los nombres de cada sprint que se a planificado, el Id que indica el código de cada historia de usuario o historias técnicas que contiene cada sprint, además tenemos los campos para el registro de la fecha de inicio, el campo para el registro de la fecha de finalización y por ultimo las horas estimadas que se va a demorar en cada esprint.

#### 3.4.7.1 Sprint 1

Este sprint contiene las diferentes historias de usuario técnicas, las mismas que fueron planificadas con el único objetivo para establecer una bases sobre como va a ser desarrollado nuestro proyecto para el beneficio de la radio.

Entre las especificaciones importantes que se determinan en la Tabla 10 se encuentran, la identificación y especificación de los requerimientos necesarios, el diseño de la base de datos, entre otro. En la tabla 13 se indica toda la historia técnica, las mismas que pertenece a este sprint 1 del proyecto.

**Tabla 13:** Sprint 1 del proyecto.

<b>Sprint 1</b>			
<b>Fecha Inicio:</b> 04/10/2021	<b>Fecha Fin:</b> 22/10/2021	<b>Esfuerzo Estimado (H):</b> 125	<b>Esfuerzo Real (H):</b> 120

<b>Pila del sprint</b>				
<b>ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HT-01	Como desarrollador necesito realizar la recopilación de requerimientos del sistema.	30	Análisis	David Campos
HT-02	Como desarrollador necesito realizar la planificación de los entregables del sistema	15	Análisis	David Campos
HT-03	Como desarrollador necesito definir las historias de usuario del sistema de la radio	15	Análisis	David Campos
HT-04	Como desarrollador necesito realizar la planificación de las pruebas unitarias automatizadas del sistema.	30	Análisis	David Campos
HT-05	Como desarrollador necesito diseñar la arquitectura del sistema	5	Diseño	David Campos
HT-06	Como desarrollador necesito diseñar las interfaces de usuario	5	Diseño	David Campos
HT-07	Como desarrollador necesito diseñar la base de datos para el proyecto del sistema	10	Diseño	David Campos
HT-08	Como desarrollador necesito realizar la implementación de la base de datos.	5	Desarrollo	David Campos
HT-09	Como desarrollador necesita realizar la conexión a la base de datos.	5	Desarrollo	David Campos

**Realizado por:** Campos David

### 3.4.7.2 Sprint 2

Este sprint contiene la planificación de las historias de usuario relacionadas con todas las actividades que están asignadas al rol del productor. Para el desarrollo de este sprint hay un total de 14 historias de usuario las mismas que indican las diferentes funcionalidades que va tener el sistema con relación al rol del productor. A continuación, en la Tabla 14 se detalla el alcance del segundo sprint del proyecto.

**Tabla 14:** Sprint 2 del proyecto.

<b>Sprint 2</b>				
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>	<b>Esfuerzo Estimado (H):</b>	<b>Esfuerzo Real (H):</b>	
25/10/2021	24/11/2021	150	140	
<b>Pila del sprint</b>				
<b>ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HU-01	Como usuario necesito autenticar mis credenciales de productor para poder ingresar al sistema de la radio	5	Desarrollo	David Campos
HU-02	Como Productor del sistema necesito ingresar nuevos usuarios al sistema de la radio.	15	Desarrollo	David Campos
HU-03	Como Productor necesito listar los usuarios que están registrados en el sistema de la radio.	5	Desarrollo	David Campos

HU -04	Como Productor del sistema necesito editar la información de los usuarios del sistema.	10	Desarrollo	David Campos
HU -05	Como Productor necesito eliminar los usuarios del sistema.	15	Desarrollo	David Campos
HU -06	Como Productor necesito ingresar al sistema un nuevo programa de radio.	15	Desarrollo	David Campos
HU -07	Como Productor necesito listar los programas de radio que están registrados en el sistema.	5	Desarrollo	David Campos
HU -08	Como Productor del sistema necesito editar la información de los programas de radio que tiene el sistema.	10	Desarrollo	David Campos
HU -09	Como Productor necesito eliminar los programas de radio del sistema.	15	Desarrollo	David Campos
HU -10	Como Productor necesito ingresar al sistema una nueva publicidad de la radio.	15	Desarrollo	David Campos
HU -11	Como Productor necesito listar las publicidades de la radio que están registrados en el sistema.	5	Desarrollo	David Campos
HU -12	Como Productor del sistema necesito editar la información de las publicidades de radio que tiene el sistema.	10	Desarrollo	David Campos
HU -13	Como Productor necesito eliminar las publicidades que tiene la radio en el sistema.	15	Desarrollo	David Campos

**Realizado por:** Campos David

### 3.4.7.3 Sprint 3

Este sprint contiene la planificación de las historias de usuario relacionadas con todas las actividades que están asignadas al rol del locutor. Para el desarrollo de este sprint hay un total de 9 historias de usuario las mismas que indican las diferentes funcionalidades que va a tener el sistema con relación al rol del locutor. A continuación, en la Tabla 15 se detalla el alcance del tercer sprint del proyecto.

**Tabla 15:** Sprint 3 del proyecto.

<b>Sprint 3</b>				
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>	<b>Esfuerzo Estimado (H):</b>	<b>Esfuerzo Real (H):</b>	
29/11/2021	18/12/2021	125	110	
<b>Pila del sprint</b>				
<b>ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HU-01	Como usuario necesito autenticar mis credenciales de locutor para poder ingresar al sistema de la radio	5	Desarrollo	David Campos

HU-14	Como Locutor necesito ingresar al sistema un nuevo guion.	15	Desarrollo	David Campos
HU -15	Como Locutor necesito listar los guiones de la radio que están registrados en el sistema.	5	Desarrollo	David Campos
HU -16	Como locutor necesito editar la información de los guiones que está en el sistema.	10	Desarrollo	David Campos
HU -17	Como locutor necesito eliminar los guiones que tiene la radio en el sistema.	15	Desarrollo	David Campos
HU -18	Como Locutor necesito ingresar al sistema un nuevo dialogo.	30	Desarrollo	David Campos
HU -19	Como Locutor necesito listar los diálogos que están registrados en el sistema.	5	Desarrollo	David Campos
HU -20	Como locutor necesito editar la información de los diálogos que está en el sistema.	10	Desarrollo	David Campos
HU -21	Como Productor necesito eliminar los diálogos elegidos del sistema.	15	Desarrollo	David Campos

**Realizado por:** Campos David

#### 3.4.7.4 Sprint 4

Este sprint contiene la planificación de las historias de usuario relacionadas con todas las actividades que están asignadas al rol del coordinador. Para el desarrollo de este sprint hay un total de 6 historias de usuario las mismas que indican las diferentes funcionalidades que va a tener el sistema con relación al rol del coordinador. A continuación, en la Tabla 16 se detalla el alcance del cuarto sprint del proyecto.

**Tabla 16:** Sprint 4 del proyecto.

<b>Sprint 4</b>				
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>	<b>Esfuerzo Estimado (H):</b>	<b>Esfuerzo Real (H):</b>	
03/01/2022	17/01/2022	140	125	
<b>Pila del sprint</b>				
<b>ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HU-01	Como usuario necesito autenticar mis credenciales de locutor para poder ingresar al sistema de la radio.	5	Desarrollo	David Campos
HU-22	Como Coordinador necesito revisar los guiones que el Locutor ingresa al sistema y así poder aprobar o rechazar el guion.	60	Desarrollo	David Campos
HU-23	Como Coordinador necesito hacer un reporte general de los usuarios que hay en el sistema de la radio.	15	Desarrollo	David Campos

HU -24	Como Coordinador necesito hacer un reporte general de los programas radiales que hay en el sistema de la radio	15	Desarrollo	David Campos
HU -25	Como Coordinador necesito hacer un reporte general de los guiones que hay en el sistema de la radio.	15	Desarrollo	David Campos
HU -26	Como Coordinador necesito hacer un reporte general de la publicidad que hay en el sistema de la radio.	15	Desarrollo	David Campos

**Realizado por:** Campos David

### **3.5 Fase de cierre e implementación**

Para la implementación de dicho sistema el mismo que servirá para la gestión de guiones y publicidad de la radio Ciudad 106.5 F.M. En esta fase final se prueban las diferentes funcionalidades que tiene el producto en un periodo de una semana, con usuarios corresponde al rol de productor, Locutor y coordinador de la radio.

## CAPÍTULO IV

### 4 RESULTADOS Y DISCUSIÓN

#### 4.1 Evaluar el beneficio del software con respecto al proceso manual

Para evaluar el beneficio que va a tener el software con respecto al proceso manual que se efectúa en la administración de guiones y de la publicidad de la radio, se hace un análisis estadístico para aceptar o rechazar la hipótesis planteada.

##### 4.1.1 Proceso de administración de guiones

Para evaluar el beneficio con respecto al tiempo en el proceso de administración de guiones en Radio Ciudad se establece:

- **Elemento de Experimentación:** Sistema de Administración de Guiones para Radio Ciudad
- **Individuos de Experimentación:** Usuarios

##### 4.1.2 Muestra

En el proceso de análisis estadístico se dispone de una muestra de 10 usuarios, los mismos que simbolizan a la población que tiene relación con la entidad, los registros de las tablas representan el tiempo en segundos el proceso de administración de guiones y publicidad.

##### 4.1.3 Análisis de Datos

Planteamos la siguiente pregunta ¿Existe diferencia significativa entre el tiempo promedio del proceso manual y el proceso con el sistema web implementado en la administración de guiones y publicidad en la radio ciudad? y se plantea las siguientes hipótesis:

- **H<sub>0</sub>:** No hay diferencia significativa entre el tiempo promedio del proceso manual y el proceso con el sistema web implementado en la administración de guiones y publicidad en la radio ciudad.
- **H<sub>1</sub>:** Hay diferencia significativa el tiempo promedio del proceso manual y el proceso con el sistema web implementado en la administración de guiones y publicidad en la radio ciudad.

##### 4.1.4 Procesos con el rol de Productor

Estando en el sistema con el rol del productor se realizó el registro de los tiempos haciendo uso del sistema web y el proceso manual, tanto en el proceso insertar como en el proceso editar de los siguientes módulos: gestión de usuario, gestión de programas de radio y gestión de publicidad.



#### 4.1.4.1 Proceso de gestión de usuarios

En este proceso de análisis estadístico se aplica una muestra de 10 usuarios, los mismos que representan a todo el personal involucrado dentro de la población de la radio, los datos representan los tiempos en segundos del proceso de gestión de usuarios y de esa manera establecer si se obtiene un beneficio del sistema web con respecto al proceso manual, obteniendo los siguientes resultados.

Tabla 17: Registro de tiempo en la gestión de usuarios.

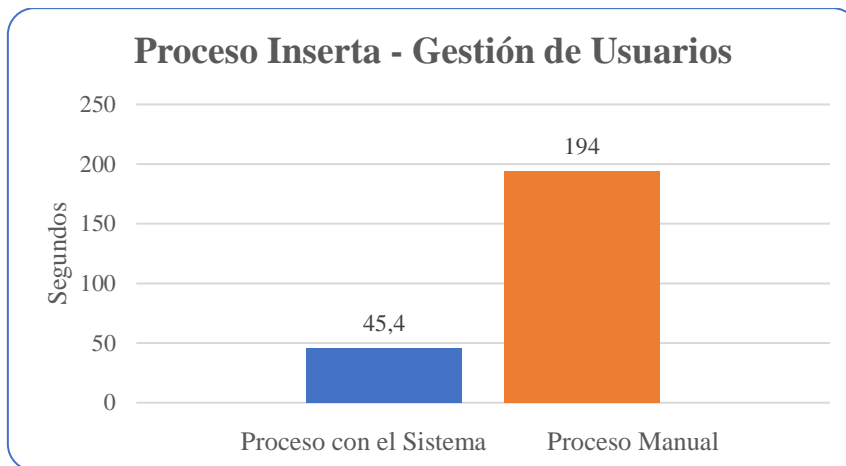
<b>Proceso de Gestión de Usuarios</b>				
<b>Productor de la Radio</b>				
<b>Usuarios</b>	<b>Proceso con el Sistema (Segundos)</b>		<b>Proceso Manual (Segundos)</b>	
<b>Nombre</b>	<b>Insertar</b>	<b>Editar</b>	<b>Insertar</b>	<b>Editar</b>
Patricio Vargas	49	65	190	240
David Campos	41	58	195	239
Edison Soria	45	58	193	235
Carlos Pérez	45	63	191	240
Paola Moreno	43	60	193	236
Emily Romero	47	65	195	236
Nelly Pilco	48	65	200	238
Lorena Narváez	43	63	198	240
Erick Cando	43	60	195	235
Cesar Méndez	50	65	190	239
<b>Promedio:</b>	45,4	62,2	194	237,8

Realizado por: Campos David.

En la Tabla 17 se puede ver los tiempos en segundos que emplearon 10 usuarios realizando el proceso de gestión de usuarios utilizando el sistema web y de forma manual. Se realizó el registro de los segundos en los dos procesos de gestión de usuarios: insertar y editar.

- **Proceso Insertar**

Dentro del módulo de gestión de usuarios en el proceso insertar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 17.

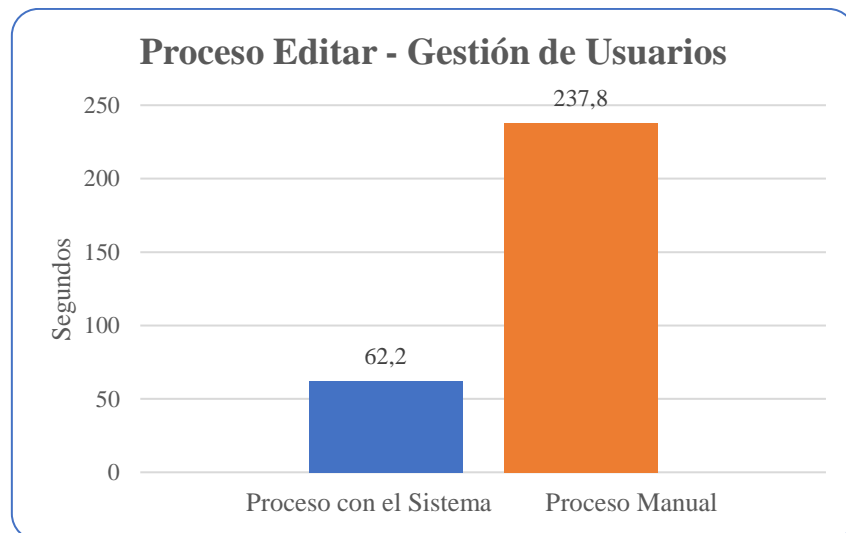


**Gráfico 14:** Proceso insertar en la gestión de usuarios.  
**Realizado por:** Campos David.

En el Gráfico 14 se representa el resultado del tiempo empleado en segundos del proceso insertar en la gestión de usuarios que el productor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 45,4 segundos y el proceso manual con un tiempo total de 194 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

- **Proceso Editar**

Dentro del módulo de gestión de usuarios en el proceso editar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 17.



**Gráfico 15:** Proceso editar en la gestión de usuarios.  
**Realizado por:** Campos David.

En el Gráfico 15 se representa el resultado del tiempo empleado en segundos del proceso editar en la gestión de usuarios que el productor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 62,2 segundos y el proceso manual con un tiempo

total de 237,8 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

#### 4.1.4.2 Proceso de gestión de programas radiales

En este proceso de análisis estadístico se aplica una muestra de 10 usuarios, los mismos que representan a todo el personal involucrado dentro de la población de la radio, los datos representan los tiempos en segundos del proceso de gestión de programas radiales y de esa manera establecer si se obtiene un beneficio del sistema web con respecto al proceso manual, obteniendo los siguientes resultados.

**Tabla 18:** Registro de tiempo en la gestión de programas de radio

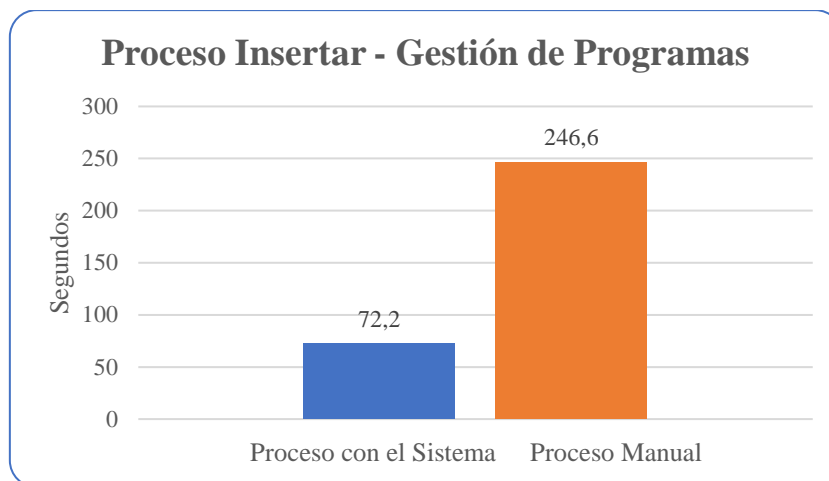
<b>Proceso de Gestión de Programas de Radio</b>				
<b>Productor de la Radio</b>				
<b>Usuarios</b>	<b>Proceso con el Sistema (Segundos)</b>		<b>Proceso Manual (Segundos)</b>	
<b>Nombre</b>	<b>Insertar</b>	<b>Editar</b>	<b>Insertar</b>	<b>Editar</b>
Patricio Vargas	78	120	245	275
David Campos	65	105	250	273
Edison Soria	75	110	240	278
Carlos Pérez	69	110	245	274
Paola Moreno	65	105	250	275
Emily Romero	70	108	255	279
Nelly Pilco	74	115	245	274
Lorena Narváez	71	110	248	271
Erick Cando	75	118	243	276
Cesar Méndez	80	115	245	275
<b>Promedio:</b>	72,2	111,6	246,6	275

**Realizado por:** Campos David.

En la Tabla 18 se puede ver los tiempos en segundos que emplearon 10 usuarios realizando el proceso de gestión de programas de radio utilizando el sistema web y de forma manual. Se realizó el registro de los segundos en los dos procesos de gestión de programas de radio: insertar y editar.

- **Proceso Insertar**

Dentro del módulo de gestión de programas de radio en el proceso insertar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 18.

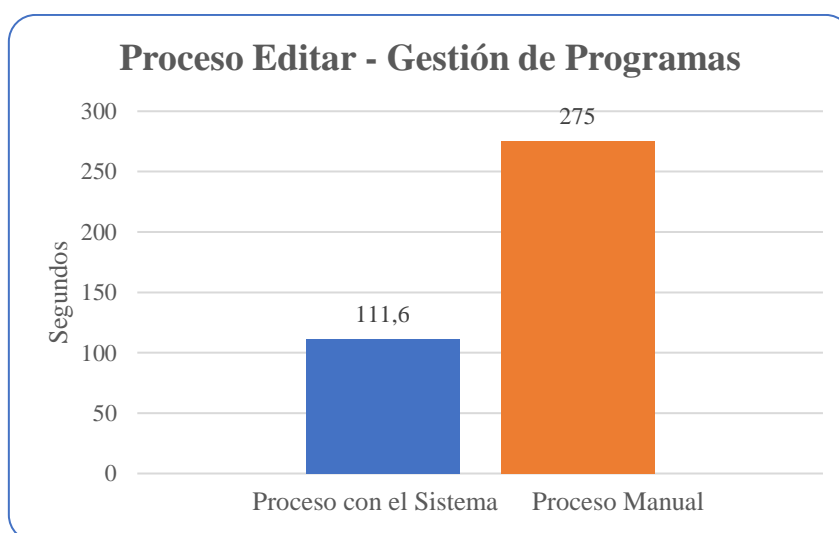


**Gráfico 16:** Proceso insertar en la gestión de programas.  
**Realizado por:** Campos David.

En el Gráfico 16 se representa el resultado del tiempo empleado en segundos del proceso insertar en la gestión de programas de radio que el productor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 72,2 segundos y el proceso manual con un tiempo total de 246,6 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

- **Proceso Editar**

Dentro del módulo de gestión de programas de radio en el proceso editar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 18.



**Gráfico 17:** Proceso editar en la gestión de programas.  
**Realizado por:** Campos David.

En el Gráfico 17 se representa el resultado del tiempo empleado en segundos del proceso editar en la gestión de programas de radio que el productor realiza. Se puede ver el

proceso realizado en el sistema con un tiempo total de 111,6 segundos y el proceso manual con un tiempo total de 275 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

#### 4.1.4.3 Proceso de gestión de publicidad.

En este proceso de análisis estadístico se aplica una muestra de 10 usuarios, los mismos que representan a todo el personal involucrado dentro de la población de la radio, los datos representan los tiempos en segundos del proceso de gestión de publicidad y de esa manera establecer si se obtiene un beneficio del sistema web con respecto al proceso manual, obteniendo los siguientes resultados.

**Tabla 19:** Registro de tiempo en la gestión de publicidad.

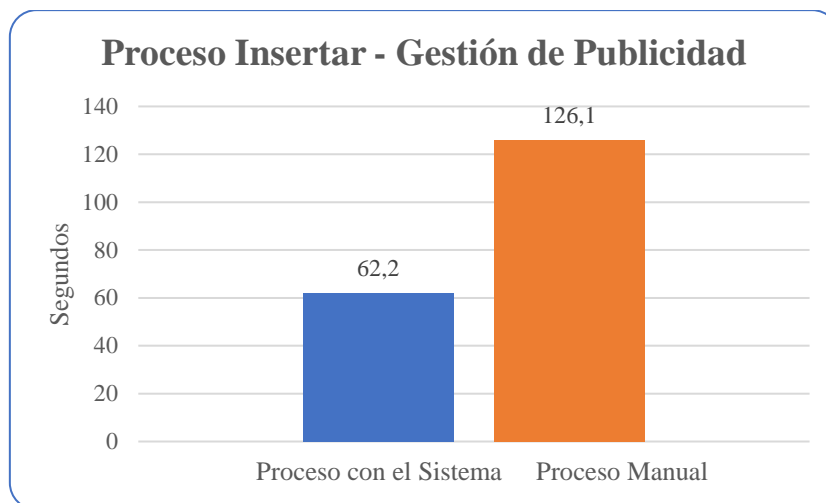
<b>Proceso de Gestión de Publicidad</b>				
<b>Productor de la Radio</b>				
<b>Usuarios</b>	<b>Proceso con el Sistema (Segundos)</b>		<b>Proceso Manual (Segundos)</b>	
<b>Nombre</b>	<b>Insertar</b>	<b>Editar</b>	<b>Insertar</b>	<b>Editar</b>
Patricio Vargas	55	95	125	180
David Campos	65	100	120	175
Edison Soria	61	90	124	170
Carlos Pérez	59	95	124	176
Paola Moreno	58	98	129	179
Emily Romero	65	93	125	175
Nelly Pilco	60	97	130	170
Lorena Narváez	65	92	127	173
Erick Cando	65	100	129	180
Cesar Méndez	69	94	128	174
<b>Promedio:</b>	62,2	95,4	126,1	175,2

**Realizado por:** Campos David.

En la Tabla 19 se puede ver los tiempos en segundos que emplearon 10 usuarios realizando el proceso de gestión de publicidad utilizando el sistema web y de forma manual. Se realizó el registro de los segundos en los dos procesos de gestión de publicidad: insertar y editar.

- **Proceso Insertar**

Dentro del módulo de gestión de publicidad en el proceso insertar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 19.

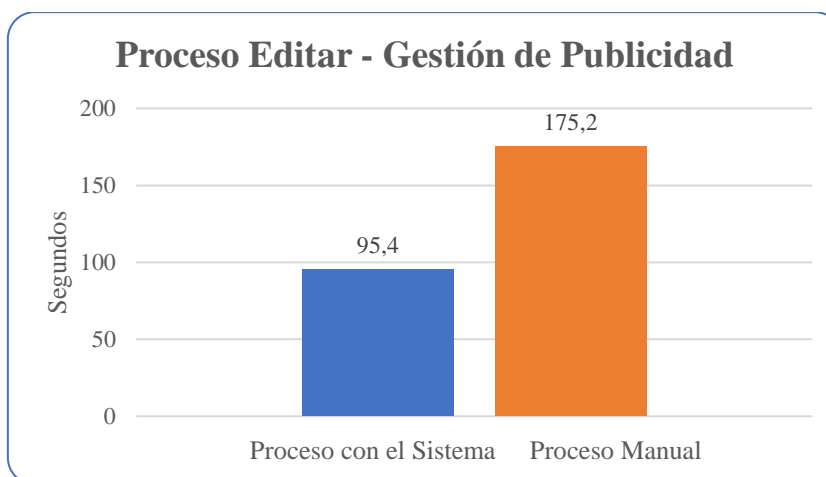


**Gráfico 18:** Proceso insertar en la gestión de publicidad.  
**Realizado por:** Campos David.

En el Gráfico 18 se representa el resultado del tiempo empleado en segundos del proceso insertar en la gestión de publicidad que el productor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 62,2 segundos y el proceso manual con un tiempo total de 126,1 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

- **Proceso Editar**

Dentro del módulo de gestión de publicidad en el proceso editar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 19.



**Gráfico 19:** Proceso editar en la gestión de publicidad.  
**Realizado por:** Campos David.

En el Gráfico 19 se representa el resultado del tiempo empleado en segundos del proceso editar en la gestión de publicidad que el productor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 95,4 segundos y el proceso manual con un

tiempo total de 175.2 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

#### 4.1.5 Procesos con el rol de Locutor

Teniendo en cuenta el rol de usuario se realizó el registro de los tiempos haciendo uso del sistema web y el proceso manual, tanto en el proceso insertar como en el editar de los siguientes módulos: gestión de guiones y gestión de diálogos.

##### 4.1.5.1 Proceso de gestión de guiones

En este proceso de análisis estadístico se aplica una muestra de 10 usuarios, los mismos que representan a todo el personal involucrado dentro de la población de la radio, los datos representan los tiempos en segundos del proceso de gestión de guiones y de esa manera establecer si se obtiene un beneficio del sistema web con respecto al proceso manual, obteniendo los siguientes resultados.

**Tabla 20:** Registro de tiempo en la gestión de guiones.

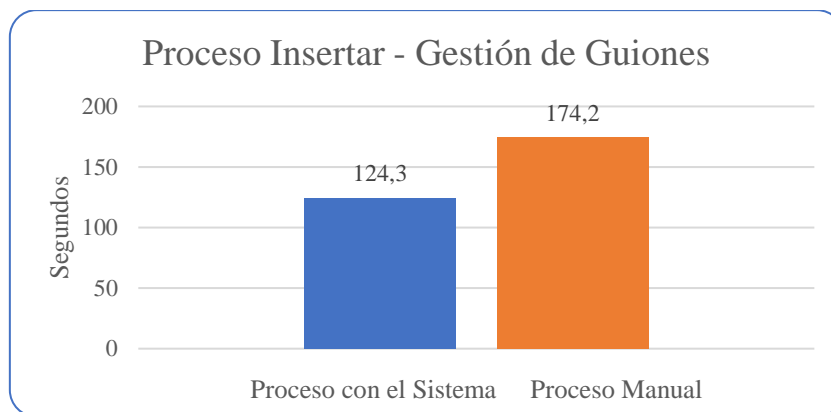
<b>Proceso de Gestión de Guiones</b>				
<b>Locutor de la Radio</b>				
<b>Usuarios</b>	<b>Proceso con el Sistema (Segundos)</b>		<b>Proceso Manual (Segundos)</b>	
<b>Nombre</b>	<b>Insertar</b>	<b>Editar</b>	<b>Insertar</b>	<b>Editar</b>
Patricio Vargas	125	156	180	220
David Campos	120	150	170	215
Edison Soria	128	158	176	215
Carlos Pérez	125	155	173	219
Paola Moreno	120	150	179	217
Emily Romero	127	157	170	220
Nelly Pilco	124	160	175	220
Lorena Narváez	126	169	173	218
Erick Cando	128	158	175	216
Cesar Méndez	120	160	171	218
<b>Promedio:</b>	124,3	157,3	174,2	217,8

**Realizado por:** Campos David.

En la Tabla 20 se puede ver los tiempos en segundos que emplearon 10 usuarios realizando el proceso de gestión de guiones utilizando el sistema web y de forma manual. Se realizó el registro de los segundos en los dos procesos de gestión de guiones: insertar y editar.

- **Proceso Insertar**

Dentro del módulo de gestión de guiones en el proceso insertar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso, detallado en la tabla 20.

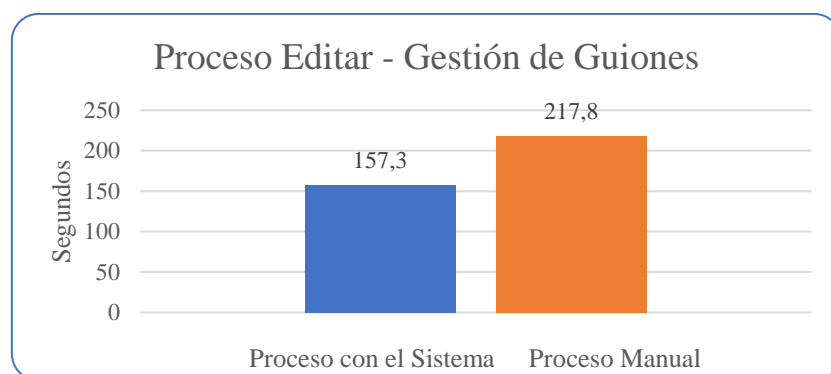


**Gráfico 20:** Proceso insertar en la gestión de guiones.  
**Realizado por:** Campos David.

En el Gráfico 20 se representa el resultado del tiempo empleado en segundos del proceso insertar en la gestión de guiones que el locutor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 124,3 segundos y el proceso manual con un tiempo total de 174,2 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

- **Proceso Editar**

Dentro del módulo de gestión de guiones en el proceso editar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 20.



**Gráfico 21:** Proceso editar en la gestión de guiones.  
**Realizado por:** Campos David.

En el Gráfico 21 se representa el resultado del tiempo empleado en segundos del proceso editar en la gestión de guiones que el locutor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 157,3 segundos y el proceso manual con un tiempo total de 217,8 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

#### 4.1.5.2 Proceso de gestión de diálogos

En este proceso de análisis estadístico se aplica una muestra de 10 usuarios, los mismos que representan a todo el personal involucrado dentro de la población de la radio, los



datos representan los tiempos en segundos del proceso de gestión de diálogos y de esa manera establecer si se obtiene un beneficio del sistema web con respecto al proceso manual, obteniendo los siguientes resultados.

**Tabla 21:** Registro de tiempo en la gestión de diálogos.

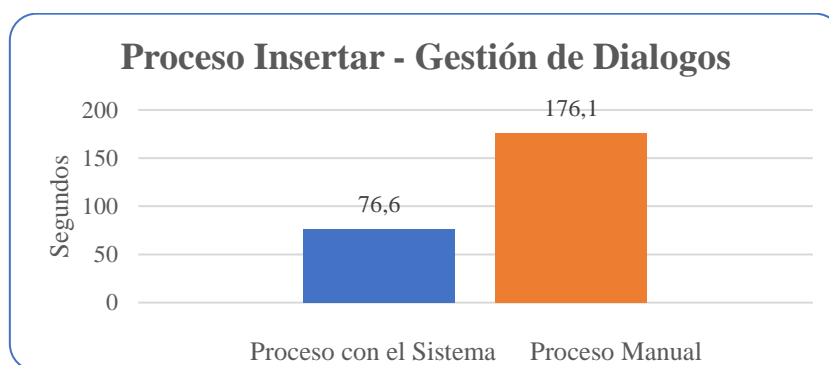
<b>Proceso de Gestión de Diálogos</b>				
<b>Locutor de la Radio</b>				
<b>Usuarios</b>	<b>Proceso con el Sistema (Segundos)</b>		<b>Proceso Manual (Segundos)</b>	
<b>Nombre</b>	<b>Insertar</b>	<b>Editar</b>	<b>Insertar</b>	<b>Editar</b>
Patricio Vargas	90	120	180	200
David Campos	60	115	185	220
Edison Soria	75	118	175	210
Carlos Pérez	70	115	170	215
Paola Moreno	80	120	175	215
Emily Romero	80	119	173	215
Nelly Pilco	68	119	179	210
Lorena Narváez	79	117	174	220
Erick Cando	75	116	170	200
Cesar Méndez	89	120	180	216
<b>Promedio:</b>	<b>76,6</b>	<b>117,9</b>	<b>176,1</b>	<b>212,1</b>

**Realizado por:** Campos David.

En la Tabla 21 se puede ver los tiempos en segundos que emplearon 10 usuarios realizando el proceso de gestión de diálogos utilizando el sistema web y de forma manual. Se realizó el registro de los segundos en los dos procesos de gestión de diálogos: insertar y editar.

- **Proceso Insertar**

Dentro del módulo de gestión de diálogos en el proceso insertar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 21.



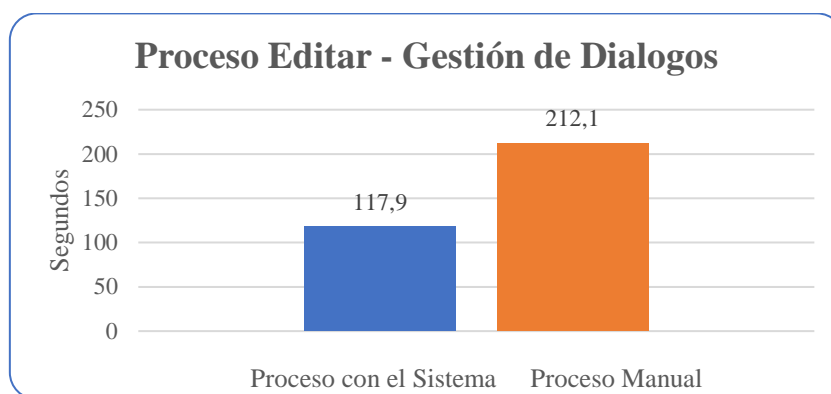
**Gráfico 22:** Proceso insertar en la gestión de diálogos.

**Realizado por:** Campos David.

En el Gráfico 22 se representa el resultado del tiempo empleado en segundos del proceso insertar en la gestión de diálogos que el locutor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 76,6 segundos y el proceso manual con un tiempo total de 176,1 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

- **Proceso Editar**

Dentro del módulo de gestión de diálogos en el proceso editar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 21.



**Gráfico 23:** Proceso editar en la gestión de diálogos.  
**Realizado por:** Campos David.

En el Gráfico 23 se representa el resultado del tiempo empleado en segundos del proceso editar en la gestión de diálogos que el locutor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 117,9 segundos y el proceso manual con un tiempo total de 212,1 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

#### **4.1.6 Procesos con el rol de Coordinador**

Teniendo en cuenta el rol de usuario se realizó el registro de los tiempos haciendo uso del sistema web y el proceso manual, tanto en el proceso insertar como en el editar del siguiente módulo: revisión de guiones.

##### **4.1.6.1 Proceso de Revisión de guiones**

En este proceso de análisis estadístico se aplica una muestra de 10 usuarios, los mismos que representan a todo el personal involucrado dentro de la población de la radio, los datos representan los tiempos en segundos del proceso de revisión de guiones y de esa manera establecer si se obtiene un beneficio del sistema web con respecto al proceso manual, obteniendo los siguientes resultados.

**Tabla 22:** Registro de tiempo en el proceso de revisión de guiones

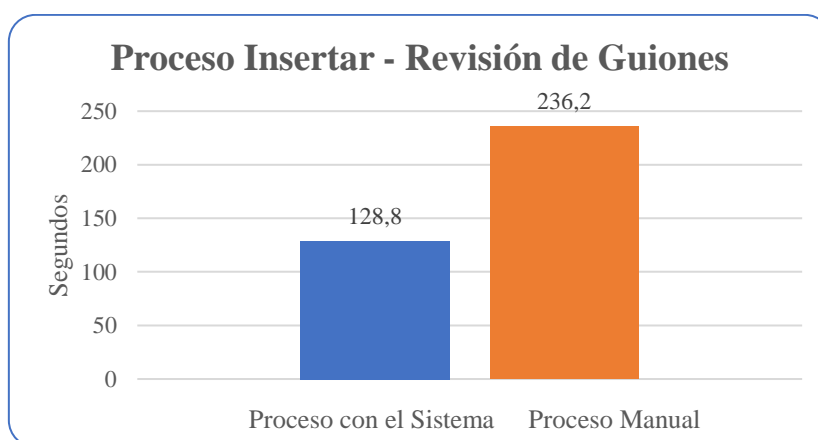
Proceso de Gestión de Revisión de Guiones				
Productor de la Radio				
Usuarios	Proceso con el Sistema (Segundos)		Proceso Manual (Segundos)	
	Insertar	Editar	Insertar	Editar
Patricio Vargas	130	180	240	260
David Campos	125	175	235	268
Edison Soria	139	170	239	267
Carlos Pérez	140	179	237	268
Paola Moreno	125	172	230	263
Emily Romero	128	178	239	260
Nelly Pilco	121	173	234	264
Lorena Narváez	124	176	230	268
Erick Cando	127	174	238	261
Cesar Méndez	129	180	240	265
<b>Promedio:</b>	128,8	175,7	236,2	264,4

Realizado por: Campos David.

En la Tabla 22 se puede ver los tiempos en segundos que emplearon 10 usuarios realizando el proceso de revisión de guiones utilizando el sistema web y de forma manual. Se realizó el registro de los segundos en los dos procesos de gestión de usuarios: insertar y editar.

- **Proceso Insertar**

Dentro del módulo de revisión de guiones en el proceso insertar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 22.

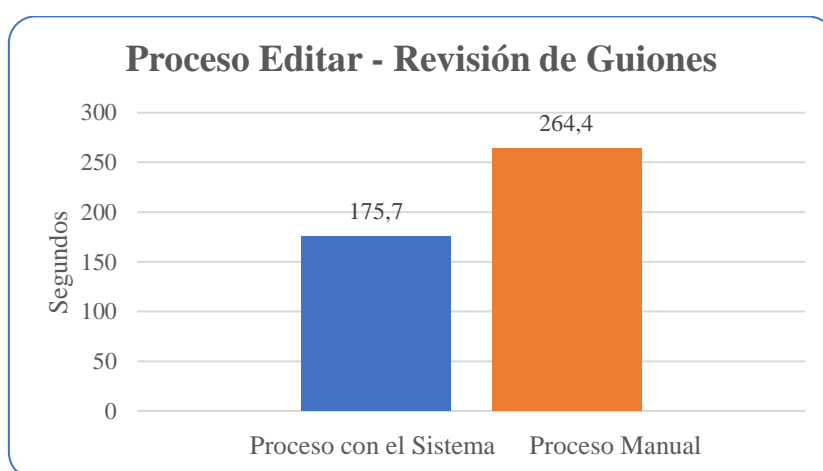


**Gráfico 24:** Proceso insertar en la revisión de guiones.  
Realizado por: Campos David.

En el Gráfico 24 se representa el resultado del tiempo empleado en segundos del proceso insertar en la revisión de guiones que el coordinador realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 128,8 segundos y el proceso manual con un tiempo total de 236,2 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

- **Proceso Editar**

Dentro del módulo de revisión de guiones en el proceso editar se calculó el promedio del tiempo en segundos de las dos formas de realizar el proceso: utilizando el sistema web y de forma manual detallado en la tabla 22.



**Gráfico 25:** Proceso editar en la revisión de guiones.  
**Realizado por:** Campos David.

En el Gráfico 25 se representa el resultado del tiempo empleado en segundos del proceso editar en la revisión de guiones que el productor realiza. Se puede ver el proceso realizado en el sistema con un tiempo total de 175,7 segundos y el proceso manual con un tiempo total de 264,4 segundos, se puede establecer una gran diferencia entre el tiempo empleado en estos dos procesos.

#### 4.1.7 Contraste de normalidad

En el procedimiento para el análisis estadístico y poder deducir si existe o no un beneficio en la gestión de guiones, primero necesitamos conocer si los datos corresponden a una población que sigue una distribución normal. El test de *Shapiro-Wilks* se aplica cuando el tamaño de la población es menor a 50 y plantea las siguientes hipótesis:

- **H<sub>0</sub>** La muestra proviene de una distribución normal
- **H<sub>1</sub>** La muestra no proviene de una distribución normal.

#### 4.1.8 Pruebas en el proceso de Gestión de Usuarios

#### 4.1.8.1 Prueba de normalidad del proceso insertar y editar

Tabla 23: Prueba de normalidad de la variable insertar y editar.

Pruebas de normalidad			
Procesos	Shapiro-Wilk		
	Estadístico	gl	Sig.
Insertar_Manual	.931	10	.458
Editar_Manual	.843	10	.049
Insertar_Sistema	.941	10	.569
Editar_Sistema	.826	10	.030

Realizado por: Campos David.

Las variables Insertar dado que el valor p calculado es mayor a 0.05 por lo que aceptamos nuestra hipótesis nula en el contraste de normalidad que dice la muestra proviene de una distribución normal.

Las variables Editar tienen una distribución poblacional No Normal ya que el valor p calculado es menor a 0.05 por lo que aceptamos nuestra hipótesis alternativa que dice la muestra no proviene de una distribución normal.

#### 4.1.8.2 Prueba de muestras emparejadas del proceso insertar

Tabla 24: Prueba de muestras emparejadas del proceso insertar.

Prueba de muestras emparejadas									
		Diferencias emparejadas					t	gl	Sig. (bilateral)
		Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
					Inferior	Superior			
Par 1	Insertar_Manual Insertar_Sistema	148.6000	5.1034	1.6138	144.9493	152.2507	92.079	9	.000

Realizado por: Campos David.

En este caso la población posee una distribución normal, para realizar el análisis estadístico se utiliza la distribución de T de Student.

La media del tiempo que toma realizar el proceso Insertar en la gestión de usuarios es estadísticamente menor después que se realizó la implementación del sistema, ya que el valor p calculado es menor que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

### 4.1.8.3 Prueba de wilcoxon en el proceso Editar

**Tabla 25:** Prueba de wilcoxon del proceso editar.

Estadísticos de prueba	
	Editar_Sistema Editar_Manual
Z	-2.812 <sup>b</sup>
Sig. asintótica (bilateral)	.005
a. Prueba de rangos con signo de Wilcoxon	
b. Se basa en rangos positivos.	

**Realizado por:** Campos David

En este caso la población posee una distribución no normal, para realizar el análisis estadístico se utilizó el Test del signo de rangos de Wilcoxon.

La media del tiempo que toma realizar el proceso Editar en la gestión de usuarios es menor después que se realizó la implementación del sistema, ya que el valor p calculado es igual que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

### 4.1.9 Pruebas en el proceso de Gestión de Programas de Radio

#### 4.1.9.1 Prueba de normalidad del proceso insertar y editar

**Tabla 26:** Prueba de normalidad de la variable insertar y editar.

Pruebas de normalidad			
Procesos	Shapiro-Wilk		
	Estadístico	gl	Sig.
Insertar_Manual	.939	10	.542
Editar_Manual	.959	10	.779
Insertar_Sistema	.951	10	.677
Editar_Sistema	.928	10	.427

**Realizado por:** Campos David

Las variables Insertar y Editar dado que el valor p calculado es mayor a 0.05 por lo que aceptamos nuestra hipótesis nula en el contraste de normalidad que dice la muestra proviene de una distribución normal.

#### 4.1.9.2 Prueba de muestras emparejadas del proceso insertar y editar

**Tabla 27:** Prueba de muestras emparejadas del proceso insertar y editar.

Prueba de muestras emparejadas						
	Diferencias emparejadas			t	gl	Sig. (bilateral)
	Media	Desviación estándar	Media de error estándar			

					Inferior	Superior			
Par 1	Insertar_Manual Insertar_Sistema	174.4000	8.3693	2.6466	168.4130	180.3870	65.896	9	.000
Par 2	Editar_Manual - Editar_Sistema	163.4000	5.5817	1.7651	159.4071	167.3929	92.573	9	.000

**Realizado por:** Campos David

En este caso la población posee una distribución normal, para realizar el análisis estadístico se utiliza la distribución de T de Student.

La media del tiempo que toma realizar el proceso Insertar y Editar en la gestión de programas de radio es estadísticamente menor después que se realizó la implementación del sistema, ya que el valor p calculado es menor que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

#### 4.1.10 Pruebas en el proceso de Gestión de Publicidad

##### 4.1.10.1 Prueba de normalidad del proceso insertar y editar

**Tabla 28:** Prueba de normalidad de la variable insertar y editar.

Pruebas de normalidad			
Procesos	Shapiro-Wilk		
	Estadístico	gl	Sig.
Insertar_Manual	.933	10	.483
Editar_Manual	.917	10	.331
Insertar_Sistema	.939	10	.543
Editar_Sistema	.958	10	.767

**Realizado por:** Campos David

Las variables Insertar y Editar dado que el valor p calculado es mayor a 0.05 por lo que aceptamos nuestra hipótesis nula en el contraste de normalidad que dice la muestra proviene de una distribución normal.

##### 4.1.10.2 Prueba de muestras emparejadas del proceso insertar y editar

**Tabla 29:** Prueba de muestras emparejadas del proceso insertar y editar.

Prueba de muestras emparejadas									
		Diferencias emparejadas					t	gl	Sig. (bilateral)
		Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
					Inferior	Superior			
Par 1	Insertar_Manual Insertar_Sistema	63.9000	5.2589	1.6630	60.1380	67.6620	38.425	9	.000
Par 2	Editar_Manual - Editar_Sistema	79.8000	3.4254	1.0832	77.3496	82.2504	73.670	9	.000

**Realizado por:** Campos David

En este caso la población posee una distribución normal, para realizar el análisis estadístico se utiliza la distribución de T de Student.

La media del tiempo que toma realizar el proceso Insertar y Editar en la gestión de publicidad es estadísticamente menor después que se realizó la implementación del sistema, ya que el valor p calculado es menor que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

#### 4.1.11 Pruebas en el proceso de Gestión de Guiones

##### 4.1.11.1 Prueba de normalidad del proceso insertar y editar

**Tabla 30:** Prueba de normalidad de la variable insertar y editar.

Pruebas de normalidad			
Procesos	Shapiro-Wilk		
	Estadístico	gl	Sig.
Insertar_Manual	.930	10	.447
Editar_Manual	.886	10	.153
Insertar_Sistema	.855	10	.066
Editar_Sistema	.908	10	.270

Realizado por: Campos David

Las variables Insertar y Editar dado que el valor p calculado es mayor a 0.05 por lo que aceptamos nuestra hipótesis nula en el contraste de normalidad que dice la muestra proviene de una distribución normal.

##### 4.1.11.2 Prueba de muestras emparejadas del proceso insertar y editar

**Tabla 31:** Prueba de muestras emparejadas del proceso insertar y editar.

Prueba de muestras emparejadas									
		Diferencias emparejadas					t	gl	Sig. (bilateral)
		Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
					Inferior	Superior			
Par 1	Insertar_Manual Insertar_Sistema	49.9000	4.5080	1.4256	46.6752	53.1248	35.004	9	.000
Par 2	Editar_Manual - Editar Sistema	60.5000	5.2757	1.6683	56.7260	64.2740	36.264	9	.000

Realizado por: Campos David

En este caso la población posee una distribución normal, para realizar el análisis estadístico se utiliza la distribución de T de Student.

La media del tiempo que toma realizar el proceso Insertar y Editar en la gestión de guiones es estadísticamente menor después que se realizó la implementación del sistema, ya que



el valor p calculado es menor que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

#### 4.1.12 Pruebas en el proceso de Gestión de Diálogos

##### 4.1.12.1 Prueba de normalidad del proceso insertar y editar

Tabla 32: Prueba de normalidad de la variable insertar y editar.

Pruebas de Normalidad			
Procesos	Shapiro-Wilk		
	Estadístico	gl	Sig.
Insertar_Manual	.940	10	.548
Editar_Manual	.853	10	.063
Insertar_Sistema	.961	10	.801
Editar_Sistema	.485	10	.000

Realizado por: Campos David

Las variables Insertar y Editar dado que el valor p calculado es mayor a 0.05 por lo que aceptamos nuestra hipótesis nula en el contraste de normalidad que dice la muestra proviene de una distribución normal.

##### 4.1.12.2 Prueba de muestras emparejadas del proceso insertar y editar

Tabla 33: Prueba de muestras emparejadas del proceso insertar y editar.

Prueba de muestras emparejadas									
		Diferencias emparejadas					t	gl	Sig. (bilateral)
		Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
					Inferior	Superior			
Par 1	Insertar_Manual - Insertar_Sistema	99.5000	10.7935	3.4132	91.7788	107.2212	29.151	9	.000
Par 2	Editar_Manual - Editar_Sistema	89.3000	20.3199	6.4257	74.7640	103.8360	13.897	9	.000

Realizado por: Campos David

En este caso la población posee una distribución normal, para realizar el análisis estadístico se utiliza la distribución de T de Student.

La media del tiempo que toma realizar el proceso *Insertar y Editar* en la gestión de diálogos es estadísticamente menor después que se realizó la implementación del sistema, ya que el valor p calculado es menor que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

#### 4.1.13 Pruebas en el proceso de Revisión de Guiones

##### 4.1.13.1 Prueba de normalidad del proceso insertar y editar

**Tabla 34:** Prueba de normalidad de la variable insertar y editar.

Pruebas de normalidad			
Procesos	Shapiro-Wilk		
	Estadístico	gl	Sig.
Insertar_Manual	.859	10	.073
Insertar_Sistema	.875	10	.114
Editar_Manual	.870	10	.100
Editar_Sistema	.942	10	.580

**Realizado por:** Campos David

Las variables Insertar y Editar dado que el valor p calculado es mayor a 0.05 por lo que aceptamos nuestra hipótesis nula en el contraste de normalidad que dice la muestra proviene de una distribución normal.

#### 4.1.13.2 Prueba de muestras emparejadas del proceso insertar y editar

**Tabla 35:** Prueba de muestras emparejadas del proceso insertar y editar.

Prueba de muestras emparejadas									
		Diferencias emparejadas					t	gl	Sig. (bilateral)
		Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
					Inferior	Superior			
Par 1	Insertar_Manual Insertar_Sistema	107.4000	5.3166	1.6813	103.5967	111.2033	63.880	9	.000
Par 2	Editar_Manual - Editar Sistema	88.7000	5.2292	1.6536	84.9593	92.4407	53.640	9	.000

**Realizado por:** Campos David

En este caso la población posee una distribución normal, para realizar el análisis estadístico se utiliza la distribución de T de Student.

La media del tiempo que toma realizar el proceso *Insertar y Editar* en la gestión de revisión de guiones es estadísticamente menor después que se realizó la implementación del sistema, ya que el valor p calculado es menor que el nivel de significancia  $\alpha=0.05$ , rechazamos la hipótesis nula y aceptamos la hipótesis alternativa.

#### 4.2 Porcentajes de las pruebas unitarias.

Como se presentó anteriormente, el presente proyecto está formado por un total de 26 historias de usuario con sus respectivos casos de pruebas unitarias que fueron desarrollados durante todo el transcurso.

A continuación, se indica la tabla con el resumen de todas las historias de usuarios y sus respectivos porcentajes.

**Tabla 36:** Porcentaje de las pruebas unitarias.

<b>Id</b>	<b>Descripción</b>	<b>Porcentaje</b>
HU-01	Como usuario necesito autenticar las credenciales correctas para poder ingresar al sistema de la radio.	99.83 %
HU-02	Como Productor del sistema necesito ingresar nuevos usuarios al sistema de la radio.	98.90 %
HU-03	Como Productor necesito listar los usuarios que están registrados en el sistema de la radio.	100.00 %
HU-04	Como Productor del sistema necesito editar la información de los usuarios del sistema.	90.97 %
HU-05	Como Productor necesito eliminar los usuarios del sistema.	100.00 %
HU-06	Como Productor necesito ingresar al sistema un nuevo programa de radio.	90.94 %
HU-07	Como Productor necesito listar los programas de radio que están registrados en el sistema.	100.00 %
HU-08	Como Productor del sistema necesito editar la información de los programas de radio que tiene el sistema.	98.65 %
HU-09	Como Productor necesito eliminar los programas de radio del sistema.	99.82 %
HU-10	Como Productor necesito ingresar al sistema una nueva publicidad de la radio.	80.79 %
HU-11	Como Productor necesito listar las publicidades de la radio que están registrados en el sistema.	100.00 %
HU-12	Como Productor del sistema necesito editar la información de las publicidades de radio que tiene el sistema.	100.00 %
HU-13	Como Productor necesito eliminar las publicidades que tiene la radio en el sistema.	100.00 %
HU-14	Como Locutor necesito ingresar al sistema un nuevo guion.	80.64 %
HU-15	Como locutor necesito listar los guiones de la radio que están registrados en el sistema.	99.96 %
HU-16	Como locutor necesito editar la información de los guiones que está en el sistema.	80.68 %
HU-17	Como locutor necesito eliminar los guiones que tiene la radio en el sistema.	99.97 %
HU-18	Como Locutor necesito ingresar al sistema un nuevo dialogo.	80.54 %
HU-19	Como Locutor necesito listar los diálogos que están registrados en el sistema.	99.06 %
HU-20	Como locutor necesito editar la información de los diálogos que está en el sistema.	97.86 %
HU-21	Como Productor necesito eliminar los diálogos elegidos del sistema.	89.70 %
HU-22	Como Coordinador necesito revisar los guiones que el	90.79 %

	Locutor ingresa al sistema y así poder aprobar o rechazar el guion.	
HU-23	Como Coordinador necesito hacer un reporte general de los usuarios que hay en el sistema de la radio.	100.00 %
HU-24	Como Coordinador necesito hacer un reporte general de los programas radiales que hay en el sistema de la radio.	100.00 %
HU-25	Como Coordinador necesito hacer un reporte general de los guiones que hay en el sistema de la radio.	100.00 %
HU-26	Como Coordinador necesito hacer un reporte general de la publicidad que hay en el sistema de la radio.	100.00 %

**Realizado por:** Campos David

En la Tabla 36 se detalla el Id de las historias de usuario la descripción y el porcentaje de cada historia de usuario donde se puede apreciar que el porcentaje está en un rango del 80% al 100 %. Además, se demostró que el tiempo empleado en cada proceso utilizando el sistema tuvo una diferencia significativa con respecto a los procesos realizados manualmente.

### 4.3 Discusión

La investigación de (Chela, 2019) concluye que el tiempo empleado en cada proceso de la radio utilizando el sistema tiene una diferencia significativa con respecto al proceso manual, concluyendo en resultados similares al del presente trabajo de investigación con un porcentaje del 95,35% de fiabilidad que nos ofrece el sistema web al momento de usar

El proyecto de investigación de (Paucar, 2017) nos indica la facilidad con lo que permite la comprobación del código a través de las pruebas unitarias en los procesos del sistema web, obteniendo un resultado del 99.87% que se puede mejorar en el tiempo de la generación de las pruebas unitarias para los diferentes tipos de métodos que existen. Así comprobando que existe una gran disminución considerable entre los tiempos invertidos en la realización de estos tipos de procesos, de esta manera se puede apreciar que existe una mejora de eficiencia planteada en los objetivos.

Según (Llalo, 2015) menciona que con la ayuda de las pruebas unitarias al momento de ejecutarlas todos ganan. Afirmando que para el desarrollador será mucho más fácil, ya que la calidad de su código mejorará, se reducirán los tiempos de depuración y la corrección de las incidencias que pueden existir. Este tipo de prueba es un test que comprueba el correcto funcionamiento de una funcionalidad incluida en la aplicación.

## CAPÍTULO V

### 5 CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

- Se realizó el proceso del análisis y recopilación de la información necesaria con respecto al uso de las pruebas unitarias y de esa manera se determinó que el modelo del ciclo de vida del desarrollo de software en V teniendo en cuenta para el diseño de los módulos del sistema web de la radio.
- Una vez realizado el proceso de la recolección de la información sobre el proceso de gestión de guiones y publicidad en la Radio Ciudad 106.5 FM, se logró realizar el diseño del sistema web planificando un total de 4 Sprints que abarcan 26 historias de usuario y 9 historias técnicas y además se seleccionó 10 usuarios para registrar el tiempo que se demora en cada proceso, se llevó a cabo junto con la metodología SCRUM.
- En la parte de la implementación del sistema, para la elaboración de la base de datos se ha procedido implementar en PostgreSQL, además, se ha desarrollado el sistema teniendo en cuenta el patrón de arquitectura modelo vista controlador; donde para el modelo y el controlador del sistema fueron codificados en el lenguaje de programación Java Netbeans IDE 8.2, así como también las diferentes vistas que tiene el proyecto ha sido desarrollada con Java Server Faces(JSF) y Bootstrap permitiendo tener una mejor visualización y estilos de la interfaz web para el sistema.
- La implementación del sistema web de la radio ha permitido optimizar el tiempo en los procesos de gestión de guiones, programas de radio y publicidad; después de haber realizado el análisis se ha podido comprobar que el tiempo empleado con el sistema web es reducido notablemente a diferencia con el tiempo del proceso manual

## 5.2 Recomendaciones

- Para tener una mejor comprensión de las pruebas unitarias se recomienda hacer un mejor estudio sobre el tema para entender adecuadamente sus procesos y pasos a seguir para tener un mejor resultado en el sistema web de la radio.
- Debería implementar otros tipos de pruebas, como por ejemplo las pruebas de mutación ya que este tipo de prueba tiene como objetivo principal asegurar la calidad de las pruebas al momento de crear diferentes casos en los cuales un test unitario debería fallar.
- Para la implementación de un sistema web se recomienda utilizar nuevas tecnologías que día a día aparecen, ya que las que se utilizó para este proyecto son un poco antiguas y ya no hay muchas actualizaciones dando como resultado la pérdida de tiempo al programador del proyecto.
- Además, se recomienda que cada cierto periodo de tiempo se vaya sacando respaldos de la información del proyecto ya que a veces por falla técnica o falla humana se suele perder toda la información y retrasando al programador.

## Bibliografía

APIUMHUB. Beneficios de las pruebas unitarias. [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://apiumhub.com/es/tech-blog-barcelona/beneficios-de-las-pruebas-unitarias/>.

AGARWAL, B. B.; TAYAL, S. P.; & GUPTA, MAHESH. Software Engineering & Testing. 1ª ed. Sudbury, Massachusetts-USA: Jones and Bartlett Publishers, 2010 pp.166-167.

ALBA, Teresa. 10 Razones para desarrollar una web con Bootstrap. [en línea]. 2014. [Consulta: 01 de 08 de 2020.] Disponible en: <https://xn--diseocreativo-lkb.com/developar-web-bootstrap/>.

DIETRICH, Erik. Starting to Unit Test: Not as Hard as You Think. 1ª ed. United States of America: Blog Into Book, 2014 pp. 7-12.

FRANCIA, Joel. El Daily Scrum No es una reunión de estado [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://www.scrum.org/resources/blog/el-daily-scrum-no-es-una-reunion-de-estado>.

GROUSSARD, Thierry. Java 8 Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos). Barcelona-España: Ediciones ENI, 2014 pp. 11-40.

GULATI, SHEKHAR; & SHARMA, RAUL. Java Unit Testing with JUnit 5 . 1ª ed. Delhi, India : Apress, 2017 pp. 25-44.

JUNIT. JUnit 4. [en línea]. 2018. [Consulta: 01 de 08 de 2020.] Disponible: [https://junit.org/junit4/faq.html#overview\\_1](https://junit.org/junit4/faq.html#overview_1).

KUMAR, Dharmendra. Software Engineering | SDLC V-Model. [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/>.

LAÍÑEZ FUENTES, José Rubén. Desarrollo de Software Ágil. Extremme Programing y Scrum. 2ª ed. s.l. : IT Campus Academy, 2015 pp. 127-129.

MARTINEZ, Rafael. PostgreSQL-es. [en línea]. 2009. [Consulta: 01 de 08 de 2020.] Disponible en: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).

MERA-PAZ, J.A. "Análisis del proceso de pruebas de calidad de software". Ingeniería Solidaria, vol. 12, n° 20 ( 2016 ), (Colombia) pp. 163-176.

MUNOZ SIMO, José Manuel. JSF, características principales, ventajas y puntos a destacar. [en línea]. 2012. [Consulta: 10 de 08 de 2020.] Disponible en: <https://josemmsimo.wordpress.com/2012/07/30/jsf-caracteristicas-principales-ventajas-y-puntos-a-destacar/>.

NIEL, MATTHEW; & STONES, RICHARD. Beginning Databases with PostgreSQL from

- Novice to Professional. 2ª ed. New York-United States of America : Apress, 2005. p. 11.
- PALACIOS, Jerónimo. Guía fundamental de Scrum. [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://jeronimopalacios.com/scrum/>.
- PANDYA, Rahul. Quora, What is JUnit? [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://www.quora.com/What-is-JUnit>.
- PAUTA AYABACA, LEOPOLDO; & MOSCOSO BERNAL, SANTIAGO. "Verificación y Validación de Software". Revista Killkana Técnica, vol.1, n° 3 (2017), (Ecuador) pp. 25-32.
- POSTGRESQL. PostgreSQL Tutorial. [en línea]. 2018. [Consulta: 01 de 08 de 2020.] Disponible en: <http://www.postgresqtutorial.com/what-is-postgresql/>.
- RAMOS VEGA, Cristina. Los eventos en Scrum. [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://cristinaramosvega.com/los-eventos-scrum/>.
- REQUENA MESA, Abraham. Qué es un sprint en scrum? [en línea]. 2018. [Consulta: 01 de 08 de 2020.] Disponible en: <https://openwebinars.net/blog/que-es-un-sprint-scrum/>.
- REVUELTAS, Pedro. El poder del sprint goal. [en línea]. 2018. [Consulta: 01 de 08 de 2020.] Disponible en: <https://www.paradigmadigital.com/techbiz/el-poder-del-sprint-goal/>.
- ROCHE, Julio. Artefactos de Scrum. [en línea]. 2016. [Consulta: 31 de Enero de 2019.] Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>.
- SCHWABER, KEN; & SUTHERLAND, JEFF. Guía de Scrum. [en línea]. 2017. [Consulta: 01 de 08 de 2020.] Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>.
- SPURLOCK, Jake. Responsive Web Development, Bootstrap. 1ª ed. California-USA : O'Reilly, 2013, pp. 1-10.
- UNIVERSIDAD DE ALICANTE. El MVC en JavaServer Faces. [en línea]. 2014. [Consulta: 01 de 08 de 2018.] Disponible en: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion02-apuntes.html#Resumen+de+los+elementos+de+JSF>.
- Llalao Cushpa, A. L., & Luisataxi Cevallos, J. R. (2013). Comparación de herramientas Benchmarking para pruebas unitarias web de aplicaciones JSF. Caso práctico: Sistema de gestión de viáticos de la ESPOCH (Bachelor's tesis)
- Paucar Ati, E. F. (2017). Desarrollo de pruebas unitarias automáticas para el sistema web escolar del Instituto Tecnológico Superior Stanford (Bachelor's thesis, Escuela Superior Politécnica de Chimborazo).
- Llumiguano, C., & Paúl, J. (2019). Desarrollo basado en pruebas unitarias de un sistema web para la gestión de guiones en la radio Crisólito Celestial 93.1 FM de la ciudad de Guaranda (Bachelor's thesis, Escuela Superior Politécnica de Chimborazo).



## ANEXOS

### Anexo A Historia de Usuario del Proyecto

Historia de Usuario		
<b>ID:</b> HU-01	<b>Descripción:</b> Como usuario necesito autenticar mis credenciales para poder ingresar al sistema de la radio.	
<b>Usuario:</b> Locutor, productor, coordinador	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como usuario necesito autenticar las credenciales correctas para poder ingresar al sistema de la radio.		
<b>Observación:</b> El sistema debe permitir ingresar a las ventanas correspondientes según los roles asignados que son: productor, locutor y coordinador.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"><li>○ Al momento de ingresar las credenciales correctas, el sistema debe permitir el ingreso correcto.</li><li>○ Al ingresar las credenciales incorrectas, el sistema debe emitir el mensaje respectivo.</li></ul>		

Historia de Usuario		
<b>ID:</b> HU-02	<b>Descripción:</b> Gestionar Usuarios.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Productor del sistema necesito ingresar nuevos usuarios al sistema de la radio.		
<b>Observación:</b> El sistema debe permitir ingresar el registro de la información de nuevos usuarios.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"><li>○ Ingresar datos de un usuario y el sistema debe guardar la información.</li><li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente del registro.</li></ul>		

Historia de Usuario		
<b>ID:</b> HU-03	<b>Descripción:</b> Gestionar Usuarios	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como Productor necesito listar los usuarios que están registrados en el sistema de la radio.		
<b>Observación:</b> El sistema debe permitir visualizar la información de los usuarios que tiene en el sistema		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"><li>○ Listar la información de los usuarios del sistema.</li></ul>		

Historia de Usuario		
<b>ID:</b> HU-04	<b>Descripción:</b> Gestionar Usuarios	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 10
<b>Descripción:</b> Como Productor del sistema necesito editar la información de los usuarios del sistema.		
<b>Observación:</b> El sistema debe permitir modificar la información de usuarios que existen en la radio.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"><li>○ Modificar y guardar la información de los usuarios del sistema</li><li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente de la modificación.</li></ul>		

Historia de Usuario		
<b>ID:</b> HU-05	<b>Descripción:</b> Gestionar Usuarios	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Productor necesito eliminar los usuarios del sistema.		
<b>Observación:</b> El sistema debe permitir eliminar del sistema la información de los usuarios.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Elimina la información de los usuarios del sistema</li> <li>○ Si se desea eliminar un usuario el sistema emitirá un mensaje de alerta donde el productor puede confirmar o cancelar la eliminación.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-06	<b>Descripción:</b> Gestionar Programas.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Productor necesito ingresar al sistema un nuevo programa de radio.		
<b>Observación:</b> El sistema debe permitir ingresar un nuevo programa de radio al sistema de la radio.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Ingresar los datos de un programa de radio y se debe guardar en el sistema la información.</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente del registro.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-07	<b>Descripción:</b> Gestionar Programas.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como Productor necesito listar los programas de radio que están registrados en el sistema.		
<b>Observación:</b> El sistema debe permitir visualizar la información de los programas de la radio que esta registrado en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Listar la información de los usuarios del sistema.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-08	<b>Descripción:</b> Gestionar Programas.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 10
<b>Descripción:</b> Como Productor del sistema necesito editar la información de los programas de radio que tiene el sistema.		
<b>Observación:</b> El sistema debe permitir modificar la información de los programas de radio que existen en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Modificar y guardar la información de los programas de radio</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente de la modificación.</li> </ul>		

Historia de Usuario
---------------------

<b>ID:</b> HU-09	<b>Descripción:</b> Gestionar Programas.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Productor necesito eliminar los programas de radio del sistema.		
<b>Observación:</b> El sistema debe permitir eliminar los programas de radio que se deseen del sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Elimina la información de los programas de radio del sistema</li> <li>○ Si se desea eliminar un programa de radio del sistema, el mismo emitirá un mensaje de alerta donde el productor puede confirmar o cancelar la eliminación.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-10	<b>Descripción:</b> Gestionar Publicidad.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Productor necesito ingresar al sistema una nueva publicidad de la radio.		
<b>Observación:</b> El sistema debe permitir ingresar una nueva publicidad de la radio al sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Ingresar la información correspondiente a la publicidad y se debe guardar en el sistema.</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente del registro.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-11	<b>Descripción:</b> Gestionar Publicidad.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como Productor necesito listar las publicidades de la radio que están registrados en el sistema.		
<b>Observación:</b> El sistema debe permitir visualizar la información de las publicidades de la radio que está registrado en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Listar la información de las publicidades de la radio que se encuentra en el sistema.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-12	<b>Descripción:</b> Gestionar Publicidad.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 10
<b>Descripción:</b> Como Productor del sistema necesito editar la información de las publicidades de radio que tiene el sistema.		
<b>Observación:</b> El sistema debe permitir modificar la información de la publicidad de la radio que existen en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Modificar y guardar la información de los programas de radio</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente de la modificación.</li> </ul>		

<b>Historia de Usuario</b>	
<b>ID:</b>	<b>Descripción:</b>

HU-13	Gestionar Publicidad.	
<b>Usuario:</b> Productor	<b>Sprint Asignado:</b> 2	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Productor necesito eliminar las publicidades que tiene la radio en el sistema.		
<b>Observación:</b> El sistema debe permitir eliminar las publicidades que se deseen del sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Elimina la información de las publicidades del sistema.</li> <li>○ Si se desea eliminar una publicidad del sistema, el mismo emitirá un mensaje de alerta donde el productor puede confirmar o cancelar la eliminación.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-14	<b>Descripción:</b> Gestionar Guiones.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Locutor necesito ingresar al sistema un nuevo guion.		
<b>Observación:</b> El sistema debe permitir ingresar un nuevo guion con su respectiva información al sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Ingresar la información correspondiente del guion y se debe guardar en el sistema.</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente del registro.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-15	<b>Descripción:</b> Gestionar Guiones.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como locutor necesito listar los guiones de la radio que están registrados en el sistema.		
<b>Observación:</b> El sistema debe permitir visualizar la información de los guiones que está registrado en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Listar la información de los guiones de la radio que se encuentra en el sistema.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-16	<b>Descripción:</b> Gestionar Guiones.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 10
<b>Descripción:</b> Como locutor necesito editar la información de los guiones que está en el sistema.		
<b>Observación:</b> El sistema debe permitir modificar la información del guion que se elija en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Modificar y guardar la información del guion seleccionado en el sistema.</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente de la modificación.</li> </ul>		

<b>Historia de Usuario</b>		
<b>ID:</b> HU-17	<b>Descripción:</b> Gestionar Guiones.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 15

<b>Descripción:</b> Como locutor necesito eliminar los guiones que tiene la radio en el sistema.
<b>Observación:</b> El sistema debe permitir eliminar los guiones que se desee del sistema.
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Elimina la información del guion del sistema.</li> <li>○ Si se desea eliminar un guion del sistema, el mismo emitirá un mensaje de alerta donde el locutor puede confirmar o cancelar la eliminación.</li> </ul>

Historia de Usuario		
<b>ID:</b> HU-18	<b>Descripción:</b> Gestionar Diálogos.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 30
<b>Descripción:</b> Como Locutor necesito ingresar al sistema un nuevo dialogo.		
<b>Observación:</b> El sistema debe permitir ingresar un nuevo dialogo con su respectiva información, además el locutor va poder subir el dialogo en cualquier formato al sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Ingresar la información correspondiente al dialogo y se debe guardar en el sistema.</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente del registro.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-19	<b>Descripción:</b> Gestionar Diálogos.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 5
<b>Descripción:</b> Como Locutor necesito listar los diálogos que están registrados en el sistema.		
<b>Observación:</b> El sistema debe permitir visualizar la información de los diálogos que está registrado en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Listar la información de los diálogos de la radio que se encuentra en el sistema.</li> <li>○ Se puede ver el archivo del dialogo que se subió al sistema.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-20	<b>Descripción:</b> Gestionar Diálogos.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 10
<b>Descripción:</b> Como locutor necesito editar la información de los diálogos que está en el sistema.		
<b>Observación:</b> El sistema debe permitir modificar la información del dialogo que se elija en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Modificar y guardar la información del guion seleccionado en el sistema.</li> <li>○ Si la información es correcta el sistema debe emitir el mensaje correspondiente de la modificación.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-21	<b>Descripción:</b> Gestionar Diálogos.	
<b>Usuario:</b> Locutor	<b>Sprint Asignado:</b> 3	<b>Puntos estimados:</b> 15
<b>Descripción:</b>		

Como Productor necesito eliminar los diálogos elegidos del sistema.
<b>Observación:</b> El sistema debe permitir eliminar los diálogos que se desee del sistema.
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Elimina la información del dialogo del sistema.</li> <li>○ Si se desea eliminar un dialogo del sistema, el mismo emitirá un mensaje de alerta donde el locutor puede confirmar o cancelar la eliminación.</li> </ul>

Historia de Usuario		
<b>ID:</b> HU-22	<b>Descripción:</b> Revisar Guiones.	
<b>Usuario:</b> Coordinador	<b>Sprint Asignado:</b> 4	<b>Puntos estimados:</b> 60
<b>Descripción:</b> Como Coordinador necesito revisar los guiones que el Locutor ingresa al sistema y así poder aprobar o rechazar el guion.		
<b>Observación:</b> El sistema debe permitir visualizar la lista de guiones para poder revisar cada guion que el locutor a creado, además se podrá ver el archivo con el dialogo que esta subido en el sistema.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Visualizar el archivo con el dialogo que va a tener el guion</li> <li>○ Seleccionar el guion que se desea revisar.</li> <li>○ Llenar las sugerencias del guion y si la información es correcta el sistema debe emitir el mensaje correspondiente de la revisión.</li> <li>○ El Locutor podrá ver la observación del coordinador del guion que fue revisado.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-23	<b>Descripción:</b> Reportes de Usuarios.	
<b>Usuario:</b> Coordinador	<b>Sprint Asignado:</b> 4	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Coordinador necesito hacer un reporte general de los usuarios que hay en el sistema de la radio.		
<b>Observación:</b> El sistema debe permitir hacer un reporte general de los usuarios que están registrados en el sistema en pdf y una hoja de Excel.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ En caso de que el Coordinador desee un reporte especifico puede hacer uso de los filtros de búsqueda y generar el reporte.</li> <li>○ Se podrá descargar y visualizar el reporte en formato pdf y en una hoja de cálculo.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-24	<b>Descripción:</b> Reportes de Programas de Radio.	
<b>Usuario:</b> Coordinador	<b>Sprint Asignado:</b> 4	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Coordinador necesito hacer un reporte general de los programas radiales que hay en el sistema de la radio.		
<b>Observación:</b> El sistema debe permitir hacer un reporte general de los programas radiales que están registrados en el sistema en formato pdf y una hoja de Excel.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ Cuando el Coordinador desee un reporte especifico puede hacer uso de los filtros de búsqueda.</li> <li>○ Se podrá descargar y visualizar el reporte en formato pdf y en una hoja de cálculo.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-25	<b>Descripción:</b> Reportes de Guiones.	
<b>Usuario:</b> Coordinador	<b>Sprint Asignado:</b> 4	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Coordinador necesito hacer un reporte general de los guiones que hay en el sistema de la radio.		
<b>Observación:</b> El sistema debe permitir hacer un reporte general de los guiones que están registrados en el sistema en formato pdf y una hoja de Excel.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ En caso de que el Coordinador desee un reporte específico puede hacer uso de los filtros de búsqueda y generar el reporte.</li> <li>○ Se podrá descargar y visualizar el reporte en formato pdf y en una hoja de cálculo.</li> </ul>		

Historia de Usuario		
<b>ID:</b> HU-26	<b>Descripción:</b> Reportes de Publicidad.	
<b>Usuario:</b> Coordinador	<b>Sprint Asignado:</b> 4	<b>Puntos estimados:</b> 15
<b>Descripción:</b> Como Coordinador necesito hacer un reporte general de la publicidad que hay en el sistema de la radio.		
<b>Observación:</b> El sistema debe permitir hacer un reporte general de la publicidad que están registrados en el sistema en formato pdf y una hoja de Excel.		
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>○ En caso de que el Coordinador desee un reporte específico puede hacer uso de los filtros de búsqueda y generar el reporte.</li> <li>○ Se podrá descargar y visualizar el reporte en formato pdf y en una hoja de cálculo.</li> </ul>		

## Anexo B Historias Técnicas del Proyecto

Historia Técnica		
<b>ID:</b> HT-01	<b>Descripción:</b> Como desarrollador necesito realizar la recopilación de requerimientos del sistema.	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 30
<b>Descripción:</b> Como desarrollador necesito realizar la recopilación de requerimientos del sistema.		
<b>Observación:</b> Se debe seguir el plan de reuniones con el product owner y con los stakeholders involucrados en el desarrollo del producto.		

Historia Técnica		
<b>ID:</b> HT-02	<b>Descripción:</b> Como desarrollador necesito realizar la planificación de los entregables del sistema	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 15
<b>Descripción:</b> Como desarrollador necesito realizar la planificación de los entregables del sistema.		
<b>Observación:</b> Se debe seguir los lineamientos planteados en la metodología SCRUM para la planificación de los		

incrementos, además se debe realizar la priorización de los requerimientos con la técnica más adecuada

Historia Técnica		
<b>ID:</b> HT-03	<b>Descripción:</b> Como desarrollador necesito definir las historias de usuario del sistema de la radio.	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 15
<b>Descripción:</b> Como desarrollador necesito definir las historias de usuario del sistema de la radio.		
<b>Observación:</b> Las historias de usuario deben priorizarse según las necesidades que tiene la radio.		

Historia Técnica		
<b>ID:</b> HT-04	<b>Descripción:</b> Como desarrollador necesito realizar la planificación de las pruebas unitarias automatizadas en el sistema.	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 30
<b>Descripción:</b> Como desarrollador necesito realizar la planificación de las pruebas unitarias automatizadas del sistema.		
<b>Observación:</b> Las pruebas unitarias deben ser consideradas como historias técnicas en la planificación de cada Sprint.		

Historia Técnica		
<b>ID:</b> HT-05	<b>Descripción:</b> Como desarrollador necesito diseñar la arquitectura del sistema	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 5
<b>Descripción:</b> Como desarrollador necesito diseñar la arquitectura del sistema		
<b>Observación:</b> La arquitectura del sistema debe ser diseñada según lo que la radio pida al desarrollador		

Historia Técnica		
<b>ID:</b> HT-06	<b>Descripción:</b> Como desarrollador necesito diseñar las interfaces de usuario	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 5
<b>Descripción:</b> Como desarrollador necesito diseñar las interfaces de usuario		
<b>Observación:</b> La interfaz completa del sistema debe ser amigable con el usuario y además debe ser clara y precisa en el momento de utilizar el sistema.		

Historia Técnica		
<b>ID:</b> HT-07	<b>Descripción:</b> Como desarrollador necesito diseñar la base de datos	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 10



<b>Descripción:</b> Como desarrollador necesito diseñar la base de datos para el proyecto del sistema
<b>Observación:</b> La base de datos debe ser implementada en PostgreSQL.

Historia Técnica		
<b>ID:</b> HT-08	<b>Descripción:</b> Como desarrollador necesito realizar la implementación de la base de datos.	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 5
<b>Descripción:</b> Como desarrollador necesito realizar la implementación de la base de datos.		
<b>Observación:</b> La base de datos debe ser implementada en PostgreSQL.		

Historia Técnica		
<b>ID:</b> HT-09	<b>Descripción:</b> Como desarrollador necesita realizar la conexión a la base de datos.	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1	<b>Puntos Estimados:</b> 5
<b>Descripción:</b> Como desarrollador necesita realizar la conexión a la base de datos.		
<b>Observación:</b> La base de datos debe ser implementada en PostgreSQL y debe conectarse Netbeans.		

### Anexo C Casos de Pruebas Unitarias

Caso de Prueba	
<b>Id</b>	CPU-01
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-01
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Iniciar Sesión</li> <li>• Cerrar Sesión</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad del Inicio de Sesión y Cerrar Sesión, el usuario puede ingresar al sistema y salir del sistema según el rol asignado.
<b>Acción de Entrada</b>	Credenciales de usuario, nombre de usuario y contraseña.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Iniciar Sesión: se redirige al panel de trabajo dependiendo del rol de usuario asignado.</li> <li>• Cerrar Sesión: se redirige a la página inicial, dando por finalizada la sesión en el sistema.</li> </ul>
<b>Código</b>	<pre>@Test public void testValidar() {     System.out.println("validar");     loginController instance = new loginController();     String expResult = "";     String result = instance.validar();     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testRetornar() {</pre>

	<pre> System.out.println("retornar"); loginController instance = new loginController(); String expectedResult = ""; String result = instance.retornar(); assertEquals(expResult, result); fail("The test case is a prototype."); } @Test public void testGetUser() {     System.out.println("getUser");     loginController instance = new loginController();     Usuario expectedResult = null;     Usuario result = instance.getUser();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testSetUser() {     System.out.println("setUser");     Usuario user = null;     loginController instance = new loginController();     instance.setUser(user);     fail("The test case is a prototype."); } } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al panel de inicio de sesión del sistema.</li> <li>2. Ingresar las credenciales del usuario según el rol asignado.</li> <li>3. Dar clic en el botón “Ingresar”.</li> <li>4. Para finalizar la sesión, dar clic en el botón “Cerrar Sesión”.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-02
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-02
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Ingresar nuevos usuarios al sistema.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de ingresar nuevos usuarios al sistema, el software permite registrar la información correspondiente al nuevo usuario.
<b>Datos /Acción de Entrada</b>	Nombre, Apellido, Cédula, Contraseña, Tipo de Usuario.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Registro exitoso del nuevo usuario en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testSetUsuarioNombre() {     System.out.println("setUsuarioNombre");     String usuarioNombre = "David";     Usuario instance = new Usuario();     instance.setUsuarioNombre(usuarioNombre);     fail("The test case is a prototype."); } @Test public void testSetUsuarioApellido() {     System.out.println("setUsuarioApellido"); </pre>

	<pre>String usuarioApellido = "Campos"; Usuario instance = new Usuario(); instance.setUsuarioApellido(usuarioApellido); fail("The test case is a prototype."); } @Test public void testSetUsuarioCedula() {     System.out.println("setUsuarioCedula");     String usuarioCedula = "0604227215";     Usuario instance = new Usuario();     instance.setUsuarioCedula(usuarioCedula);     fail("The test case is a prototype."); } @Test public void testSetUsuarioContrasena() {     System.out.println("setUsuarioContrasena");     String usuarioContrasena = "123";     Usuario instance = new Usuario();     instance.setUsuarioContrasena(usuarioContrasena);     fail("The test case is a prototype."); } @Test public void testSetUsuarioTipoId() {     System.out.println("1");     UsuarioTipo usuarioTipoId = 1;     Usuario instance = new Usuario();     instance.setUsuarioTipoId(usuarioTipoId);     fail("The test case is a prototype."); } }</pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar usuarios.</li> <li>3. Clic en el Crear Usuario.</li> <li>4. Ingresar la información del usuario en los campos correspondientes.</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-03
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-03
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Listar los usuarios que se encuentran registrados en el sistema</li> </ul>
<b>Funcionalidad/ Característica</b>	Visualiza la información de los usuarios registrados en el sistema.
<b>Acción de Entrada</b>	Listar los usuarios registrados en el sistema
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Visualización exitosa de la información de los usuarios registrados en el sistema.</li> </ul>
<b>Código</b>	<pre>@Test public void testGetListafiltro() {     System.out.println("getListafiltro");     UsuarioController instance = new UsuarioController();     List&lt;Usuario&gt; expResult = null;     List&lt;Usuario&gt; result = instance.getListafiltro();     assertEquals(expResult, result); }</pre>

	<pre> fail("The test case is a prototype."); } @Test public void testSetListafiltro() {     System.out.println("setListafiltro");     List&lt;Usuario&gt; listafiltro = null;     UsuarioController instance = new UsuarioController();     instance.setListafiltro(listafiltro);     fail("The test case is a prototype."); } @Test public void testPrepareList() {     System.out.println("prepareList");     UsuarioController instance = new UsuarioController();     String expResult = "";     String result = instance.prepareList();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testPrepareView() {     System.out.println("prepareView");     UsuarioController instance = new UsuarioController();     String expResult = "";     String result = instance.prepareView();     assertEquals(expResult, result);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar usuarios.</li> <li>3. Clic en el listar Usuario.</li> <li>4. Y se visualiza la información de los usuarios del sistema.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-04
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-04
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Primero se debe listar los usuarios.</li> <li>• Seleccionar el usuario que se va a editar.</li> <li>• Modificar Usuario</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de modificar usuario se puede cambiar la información que se guardó anteriormente en el sistema.
<b>Datos /Acción de Entrada</b>	Los campos para modificar son: Nombre, Apellido, Cédula, Contraseña, Tipo de Usuario.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Modificación exitosa del usuario seleccionado en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testPrepareEdit() {     System.out.println("prepareEdit");     UsuarioController instance = new UsuarioController();     String expResult = "";     String result = instance.prepareEdit(); </pre>

```

    assertEquals(expResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testCancelUpdate() {
    System.out.println("cancelUpdate");
    UsuarioController instance = new UsuarioController();
    String expResult = "";
    String result = instance.cancelUpdate();
    assertEquals(expResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testUpdate() {
    System.out.println("update");
    UsuarioController instance = new UsuarioController();
    String expResult = "";
    String result = instance.update();
    assertEquals(expResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testSetUsuarioNombre() {
    System.out.println("setUsuarioNombre");
    String usuarioNombre = "David";
    Usuario instance = new Usuario();
    instance.setUsuarioNombre(usuarioNombre);
    fail("The test case is a prototype.");
}
@Test
public void testSetUsuarioApellido() {
    System.out.println("setUsuarioApellido");
    String usuarioApellido = "Campos";
    Usuario instance = new Usuario();
    instance.setUsuarioApellido(usuarioApellido);
    fail("The test case is a prototype.");
}
@Test
public void testSetUsuarioCedula() {
    System.out.println("setUsuarioCedula");
    String usuarioCedula = "0604227215";
    Usuario instance = new Usuario();
    instance.setUsuarioCedula(usuarioCedula);
    fail("The test case is a prototype.");
}
@Test
public void testSetUsuarioContrasena() {
    System.out.println("setUsuarioContrasena");
    String usuarioContrasena = "123";
    Usuario instance = new Usuario();
    instance.setUsuarioContrasena(usuarioContrasena);
    fail("The test case is a prototype.");
}
@Test
public void testSetUsuarioTipoId() {
    System.out.println("1");
    UsuarioTipo usuarioTipoId = 1;
    Usuario instance = new Usuario();
    instance.setUsuarioTipoId(usuarioTipoId);
    fail("The test case is a prototype."); }

```

<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar usuarios.</li> <li>3. Clic en el Listar Usuario.</li> <li>4. Seleccionar el usuario que se desee y presionamos en Editar.</li> <li>5. Ingresar la nueva información del usuario en los campos.</li> <li>6. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-05
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-05
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los usuarios.</li> <li>• Seleccionar el usuario que se va a eliminar.</li> <li>• Dar clic en eliminar usuario.</li> <li>• Confirmamos o cancelamos la eliminación.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de eliminar usuario se puede eliminar la información del usuario en el sistema.
<b>Datos /Acción de Entrada</b>	Permite eliminar toda la información de un usuario del sistema.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Eliminación exitosa del usuario seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testDestroy() {     System.out.println("destroy");     UsuarioController instance = new UsuarioController();     String expectedResult = "";     String result = instance.destroy();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); }  @Test public void testDestroyAndView() {     System.out.println("destroyAndView");     UsuarioController instance = new UsuarioController();     String expectedResult = "";     String result = instance.destroyAndView();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); } </pre>

<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar usuarios.</li> <li>3. Clic en el Listar Usuario.</li> <li>4. Seleccionar el usuario que se desee y presionamos en Eliminar.</li> <li>5. Confirmamos o cancelamos la eliminación del usuario.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>
-----------------------

<b>Id</b>	CPU-06
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-06
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Ingresar nuevos programas de radio al sistema.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de ingresar nuevos programas de radio al sistema, el software permite registrar la información correspondiente al nuevo programa de radio.
<b>Datos /Acción de Entrada</b>	Nombre, Descripción, Día, Hora, Duración, Clasificación, Contenido, Responsable
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Registro exitoso del nuevo programa de radio en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testSetProgramaRadialNombre() {     System.out.println("setProgramaRadialNombre");     String programaRadialNombre = "Ciudad al Dia";     ProgramaRadial instance = new ProgramaRadial();     instance.setProgramaRadialNombre(programaRadialNombre);     fail("The test case is a prototype."); }  @Test public void testSetProgramaRadialDescripcion() {     System.out.println("setProgramaRadialDescripcion");     String programaRadialDescripcion = "Todos los Dias";     ProgramaRadial instance = new ProgramaRadial();     instance.setProgramaRadialDescripcion(programaRadialDescripcion);     fail("The test case is a prototype."); }  @Test public void testSetProgramaRadialDiaTransmision() {     System.out.println("setProgramaRadialDiaTransmision");     String programaRadialDiaTransmision = "Lunes - Viernes";     ProgramaRadial instance = new ProgramaRadial();     instance.setProgramaRadialDiaTransmision(programaRadialDiaTransmision);     fail("The test case is a prototype."); }  @Test public void testSetProgramaRadialHoraTransmision() {     System.out.println("setProgramaRadialHoraTransmision");     String programaRadialHoraTransmision = "16:00";     ProgramaRadial instance = new ProgramaRadial();     instance.setProgramaRadialHoraTransmision(programaRadialHoraTransmision);     fail("The test case is a prototype."); }  @Test public void testSetProgramaRadialDuracion() {     System.out.println("setProgramaRadialDuracion");     String programaRadialDuracion = "00:05:00";     ProgramaRadial instance = new ProgramaRadial();     instance.setProgramaRadialDuracion(programaRadialDuracion);     fail("The test case is a prototype."); }  @Test public void testSetClasificacionProgramaRadialId() {     System.out.println("setClasificacionProgramaRadialId");     ClasificacionProgramaRadial clasificacionProgramaRadialId = 2;     ProgramaRadial instance = new ProgramaRadial();     instance.setClasificacionProgramaRadialId(clasificacionProgramaRadialId); </pre>

	<pre> fail("The test case is a prototype."); } @Test public void testSetTipoProgramaRadialId() {     System.out.println("setTipoProgramaRadialId");     TipoProgramaRadial tipoProgramaRadialId = 1;     ProgramaRadial instance = new ProgramaRadial();     instance.setTipoProgramaRadialId(tipoProgramaRadialId);     fail("The test case is a prototype."); } @Test public void testSetUsuarioId() {     System.out.println("setUsuarioId");     Usuario usuarioId = 2;     ProgramaRadial instance = new ProgramaRadial();     instance.setUsuarioId(usuarioId);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar programas.</li> <li>3. Clic en el Crear Programa.</li> <li>4. Ingresar la información del programa en los campos correspondientes.</li> <li>5. Dar clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-07
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-07
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Listar los programas de radio que se encuentran en el sistema</li> </ul>
<b>Funcionalidad/ Característica</b>	Visualiza la información de todos los programas de radio registrados en el sistema.
<b>Acción de Entrada</b>	Listar todos los programas de radio que están registrados en el sistema
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Visualización exitosa de la información de los programas de radio que están registrados en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testGetListafiltro() {     System.out.println("getListafiltro");     ProgramaRadialController instance = new ProgramaRadialController();     List&lt;ProgramaRadial&gt; expectedResult = null;     List&lt;ProgramaRadial&gt; result = instance.getListafiltro();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); } @Test public void testSetListafiltro() {     System.out.println("setListafiltro");     List&lt;ProgramaRadial&gt; listafiltro = null;     ProgramaRadialController instance = new ProgramaRadialController();     instance.setListafiltro(listafiltro);     fail("The test case is a prototype."); } </pre>



	<pre> @Test public void testPrepareList() {     System.out.println("prepareList");     ProgramaRadialController instance = new ProgramaRadialController();     String expectedResult = "";     String result = instance.prepareList();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testPrepareView() {     System.out.println("prepareView");     ProgramaRadialController instance = new ProgramaRadialController();     String expectedResult = "";     String result = instance.prepareView();     assertEquals(expResult, result);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar programas.</li> <li>3. Clic en el listar Programas.</li> <li>4. Y se visualiza la información de los programas de radio.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-08
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-08
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los programas de radio.</li> <li>• Seleccionar el programa de radio que se va a editar.</li> <li>• Modificar programa de radio</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de modificar programa se puede cambiar la información que se guardó anteriormente en el sistema.
<b>Datos /Acción de Entrada</b>	Los campos para modificar son: Nombre, Descripción, Dia, Hora, Duración, Clasificación, Contenido, Responsable
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Modificación exitosa del programa seleccionado en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testPrepareEdit() {     System.out.println("prepareEdit");     ProgramaRadialController instance = new ProgramaRadialController();     String expectedResult = "";     String result = instance.prepareEdit();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testCancelUpdate() {     System.out.println("cancelUpdate");     ProgramaRadialController instance = new ProgramaRadialController();     String expectedResult = ""; </pre>

```

String result = instance.cancelUpdate();
assertEquals(expResult, result);
fail("The test case is a prototype.");
}
@Test
public void testUpdate() {
    System.out.println("update");
    ProgramaRadialController instance = new ProgramaRadialController();
    String expResult = "";
    String result = instance.update();
    assertEquals(expResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testSetProgramaRadialNombre() {
    System.out.println("setProgramaRadialNombre");
    String programaRadialNombre = "Ciudad al Dia";
    ProgramaRadial instance = new ProgramaRadial();
    instance.setProgramaRadialNombre(programaRadialNombre);
    fail("The test case is a prototype.");
}
@Test
public void testSetProgramaRadialDescripcion() {
    System.out.println("setProgramaRadialDescripcion");
    String programaRadialDescripcion = "Todos los Dias";
    ProgramaRadial instance = new ProgramaRadial();
    instance.setProgramaRadialDescripcion(programaRadialDescripcion);
    fail("The test case is a prototype.");
}
@Test
public void testSetProgramaRadialDiaTransmision() {
    System.out.println("setProgramaRadialDiaTransmision");
    String programaRadialDiaTransmision = "Lunes - Viernes";
    ProgramaRadial instance = new ProgramaRadial();
    instance.setProgramaRadialDiaTransmision(programaRadialDiaTransmision);
    fail("The test case is a prototype.");
}
@Test
public void testSetProgramaRadialHoraTransmision() {
    System.out.println("setProgramaRadialHoraTransmision");
    String programaRadialHoraTransmision = "16:00";
    ProgramaRadial instance = new ProgramaRadial();
    instance.setProgramaRadialHoraTransmision(programaRadialHoraTransmision);
    fail("The test case is a prototype.");
}
@Test
public void testSetProgramaRadialDuracion() {
    System.out.println("setProgramaRadialDuracion");
    String programaRadialDuracion = "00:05:00";
    ProgramaRadial instance = new ProgramaRadial();
    instance.setProgramaRadialDuracion(programaRadialDuracion);
    fail("The test case is a prototype.");
}
@Test
public void testSetClasificacionProgramaRadialId() {
    System.out.println("setClasificacionProgramaRadialId");
    ClasificacionProgramaRadial clasificacionProgramaRadialId = 1;
    ProgramaRadial instance = new ProgramaRadial();
    instance.setClasificacionProgramaRadialId(clasificacionProgramaRadialId);
    fail("The test case is a prototype.");
}

```

	<pre> } @Test public void testSetTipoProgramaRadialId() {     System.out.println("setTipoProgramaRadialId");     TipoProgramaRadial tipoProgramaRadialId = 2;     ProgramaRadial instance = new ProgramaRadial();     instance.setTipoProgramaRadialId(tipoProgramaRadialId);     fail("The test case is a prototype."); } @Test public void testSetUsuarioId() {     System.out.println("setUsuarioId");     Usuario usuarioId = 1;     ProgramaRadial instance = new ProgramaRadial();     instance.setUsuarioId(usuarioId);     fail("The test case is a prototype."); } } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar programas.</li> <li>3. Clic en el Listar Programas.</li> <li>4. Seleccionar el programa que se desee y presionamos en Editar.</li> <li>5. Ingresar la nueva información del programa.</li> <li>6. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-09
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-09
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los programas.</li> <li>• Seleccionar el programa de radio que se va a eliminar.</li> <li>• Dar clic en eliminar programa.</li> <li>• Confirmamos o cancelamos la eliminación del programa.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de eliminar programa se puede eliminar la información del programa de radio que está en el sistema.
<b>Datos /Acción de Entrada</b>	Permite eliminar toda la información de un programa de radio del sistema.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Eliminación exitosa del programa seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testDestroy() {     System.out.println("destroy");     ProgramaRadialController instance = new ProgramaRadialController();     String expResult = "";     String result = instance.destroy();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testDestroyAndView() {     System.out.println("destroyAndView");     ProgramaRadialController instance = new ProgramaRadialController(); </pre>

	<pre>String expResult = ""; String result = instance.destroyAndView(); assertEquals(expResult, result); fail("The test case is a prototype."); }</pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar programas.</li> <li>3. Clic en el Listar Programas.</li> <li>4. Seleccionar el programa que se desee y presionamos en Eliminar.</li> <li>5. Confirmamos o cancelamos la eliminación del programa.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-010
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-10
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Ingresar nuevas publicidades de la radio al sistema.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de ingresar nuevas publicidades de la radio al sistema, el software permite registrar la información correspondiente a la publicidad.
<b>Datos /Acción de Entrada</b>	Nombre, Descripción, Fecha Inicio, Fecha Fin, Estado, Horario, Programa Radial.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Registro exitoso de la nueva publicidad de la radio en el sistema.</li> </ul>
<b>Código</b>	<pre>@Test public void testSetNombre() {     System.out.println("setNombre");     String nombre = "Anuncios";     Publicidad instance = new Publicidad();     instance.setNombre(nombre);     fail("The test case is a prototype."); }  @Test public void testSetDescripcion() {     System.out.println("setDescripcion");     String descripcion = "De ventas";     Publicidad instance = new Publicidad();     instance.setDescripcion(descripcion);     fail("The test case is a prototype."); }  @Test public void testSetFechaInicio() {     System.out.println("setFechaInicio");     Date fechaInicio = "03/10/2022";     Publicidad instance = new Publicidad();     instance.setFechaInicio(fechaInicio);     fail("The test case is a prototype."); }  @Test public void testSetFechaFin() {     System.out.println("setFechaFin");     Date fechaFin = "03/11/2022";     Publicidad instance = new Publicidad();     instance.setFechaFin(fechaFin); }</pre>

	<pre> fail("The test case is a prototype."); } @Test public void testSetEstado() {     System.out.println("setEstado");     String estado = "Activo";     Publicidad instance = new Publicidad();     instance.setEstado(estado);     fail("The test case is a prototype."); } @Test public void testSetHorarioId() {     System.out.println("setHorarioId");     Horario horarioId = 1;     Publicidad instance = new Publicidad();     instance.setHorarioId(horarioId);     fail("The test case is a prototype."); } @Test public void testSetProgramaRadialId() {     System.out.println("setProgramaRadialId");     ProgramaRadial programaRadialId = 1;     Publicidad instance = new Publicidad();     instance.setProgramaRadialId(programaRadialId);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar publicidad.</li> <li>3. Clic en el Crear Publicidad.</li> <li>4. Ingresar la información de la publicidad en los campos.</li> <li>5. Dar clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-11
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-11
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Listar las publicidades de radio que se encuentran en el sistema</li> </ul>
<b>Funcionalidad/ Característica</b>	Visualiza la información de todas las publicidades de la radio que se encuentran registrados en el sistema.
<b>Acción de Entrada</b>	Listar todas las publicidades de la radio que están registrados en el sistema
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Visualización exitosa de la información de todas las publicidades de la radio que están registrados en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testGetListafiltro() {     System.out.println("getListafiltro");     PublicidadController instance = new PublicidadController();     List&lt;Publicidad&gt; expectedResult = null;     List&lt;Publicidad&gt; result = instance.getListafiltro();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); } </pre>

	<pre> @Test public void testSetListafiltro() {     System.out.println("setListafiltro");     List&lt;Publicidad&gt; listafiltro = null;     PublicidadController instance = new PublicidadController();     instance.setListafiltro(listafiltro);     fail("The test case is a prototype."); }  @Test public void testPrepareList() {     System.out.println("prepareList");     PublicidadController instance = new PublicidadController();     String expectedResult = "";     String result = instance.prepareList();     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testPrepareView() {     System.out.println("prepareView");     PublicidadController instance = new PublicidadController();     String expectedResult = "";     String result = instance.prepareView();     assertEquals(expResult, result);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar publicidad.</li> <li>3. Clic en el listar publicidad.</li> <li>4. Y se visualiza la información de las publicidades de la radio.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-12
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-12
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar las publicidades de la radio.</li> <li>• Seleccionar la publicidad que se va a editar.</li> <li>• Modificar la información de la publicidad.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de modificar publicidad se puede cambiar la información que se guardó anteriormente en el sistema.
<b>Datos /Acción de Entrada</b>	Los campos para modificar son: Nombre, Descripción, Fecha Inicio, Fecha Fin, Estado, Horario, Programa Radial.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Modificación exitosa de la publicidad seleccionado en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testPrepareEdit() {     System.out.println("prepareEdit");     PublicidadController instance = new PublicidadController();     String expectedResult = "";     String result = instance.prepareEdit();     assertEquals(expResult, result);     fail("The test case is a prototype."); } </pre>

```

@Test
public void testCancelUpdate() {
    System.out.println("cancelUpdate");
    PublicidadController instance = new PublicidadController();
    String expectedResult = "";
    String result = instance.cancelUpdate();
    assertEquals(expectedResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testUpdate() {
    System.out.println("update");
    PublicidadController instance = new PublicidadController();
    String expectedResult = "";
    String result = instance.update();
    assertEquals(expectedResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testSetNombre() {
    System.out.println("setNombre");
    String nombre = "Anuncios";
    Publicidad instance = new Publicidad();
    instance.setNombre(nombre);
    fail("The test case is a prototype.");
}
@Test
public void testSetDescripcion() {
    System.out.println("setDescripcion");
    String descripcion = "De ventas";
    Publicidad instance = new Publicidad();
    instance.setDescripcion(descripcion);
    fail("The test case is a prototype.");
}
@Test
public void testSetFechaInicio() {
    System.out.println("setFechaInicio");
    Date fechaInicio = "03/10/2022";
    Publicidad instance = new Publicidad();
    instance.setFechaInicio(fechaInicio);
    fail("The test case is a prototype.");
}
@Test
public void testSetFechaFin() {
    System.out.println("setFechaFin");
    Date fechaFin = "03/11/2022";
    Publicidad instance = new Publicidad();
    instance.setFechaFin(fechaFin);
    fail("The test case is a prototype.");
}
@Test
public void testSetEstado() {
    System.out.println("setEstado");
    String estado = "Activo";
    Publicidad instance = new Publicidad();
    instance.setEstado(estado);
    fail("The test case is a prototype.");
}
@Test
public void testSetHorarioId() {

```

	<pre> System.out.println("setHorarioId"); Horario horarioId = 1; Publicidad instance = new Publicidad(); instance.setHorarioId(horarioId); fail("The test case is a prototype."); } @Test public void testSetProgramaRadialId() { System.out.println("setProgramaRadialId"); ProgramaRadial programaRadialId = 1; Publicidad instance = new Publicidad(); instance.setProgramaRadialId(programaRadialId); fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar publicidad.</li> <li>3. Clic en el Listar publicidad.</li> <li>4. Seleccionar la publicidad que se desee y presionamos en Editar.</li> <li>5. Ingresar la nueva información de la publicidad.</li> <li>6. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-13
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-13
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar las publicidades.</li> <li>• Seleccionar la publicidad que se va a eliminar.</li> <li>• Dar clic en eliminar publicidad.</li> <li>• Confirmamos o cancelamos la eliminación de la publicidad.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de eliminar publicidad se puede eliminar la información de la publicidad que está registrado en el sistema.
<b>Datos /Acción de Entrada</b>	Permite eliminar toda la información de una publicidad de la radio del sistema.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Eliminación exitosa de la publicidad seleccionada del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testDestroy() { System.out.println("destroy"); PublicidadController instance = new PublicidadController(); String expectedResult = ""; String result = instance.destroy(); assertEquals(expectedResult, result); fail("The test case is a prototype."); } @Test public void testDestroyAndView() { System.out.println("destroyAndView"); PublicidadController instance = new PublicidadController(); String expectedResult = ""; String result = instance.destroyAndView(); assertEquals(expectedResult, result); fail("The test case is a prototype."); } </pre>



<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de productor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar publicidad.</li> <li>3. Clic en el Listar Publicidad.</li> <li>4. Seleccionar la publicidad que se desee y damos en Eliminar.</li> <li>5. Confirmamos o cancelamos la eliminación de la publicidad.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-014
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-14
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Ingresar un nuevo guion de la radio al sistema.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de ingresar nuevos guiones de la radio al sistema, el software permite registrar la información correspondiente al guion.
<b>Datos /Acción de Entrada</b>	Fecha Registro, Fecha Aprobación, Dialogo, Estado, Observación, Programa, Tipo Guion.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Registro exitoso del nuevo guion de la radio en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testSetGuionProgramaRadialFechaRegistro() {     System.out.println("setGuionProgramaRadialFechaRegistro");     Date guionProgramaRadialFechaRegistro = "03/10/2022";     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setGuionProgramaRadialFechaRegistro (guionProgramaRadialFechaRegistro);     fail("The test case is a prototype."); } @Test public void testSetGuionProgramaRadialFechaAprobacion() {     System.out.println("setGuionProgramaRadialFechaAprobacion");     Date guionProgramaRadialFechaAprobacion = "03/11/2022";     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setGuionProgramaRadialFechaAprobacion (guionProgramaRadialFechaAprobacion);     fail("The test case is a prototype."); } @Test public void testSetDialogoId() {     System.out.println("setDialogoId");     Dialogo dialogoId = 1;     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setDialogoId(dialogoId);     fail("The test case is a prototype."); } @Test public void testSetEstadoGuionId() {     System.out.println("setEstadoGuionId");     EstadoGuion estadoGuionId = 1;     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setEstadoGuionId(estadoGuionId);     fail("The test case is a prototype."); } @Test public void testSetObservacionGuionProgramaId() { </pre>

	<pre> System.out.println("setObservacionGuionProgramaId"); ObservacionGuionPrograma observacionGuionProgramaId = 1; GuionProgramaRadial instance = new GuionProgramaRadial(); instance.setObservacionGuionProgramaId (observacionGuionProgramaId); fail("The test case is a prototype."); } @Test public void testSetProgramaRadialId() { System.out.println("setProgramaRadialId"); ProgramaRadial programaRadialId = 2; GuionProgramaRadial instance = new GuionProgramaRadial(); instance.setProgramaRadialId(programaRadialId); fail("The test case is a prototype."); } @Test public void testSetTipoGuionId() { System.out.println("setTipoGuionId"); TipoGuion tipoGuionId = 1; GuionProgramaRadial instance = new GuionProgramaRadial(); instance.setTipoGuionId(tipoGuionId); fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar guion.</li> <li>3. Clic en el Crear Guion.</li> <li>4. Ingresar la información del guion en los campos correspondientes.</li> <li>5. Dar clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-15
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-15
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Listar los guiones de radio que se encuentran registrados en el sistema</li> </ul>
<b>Funcionalidad/ Característica</b>	Visualiza la información de todos los guiones de la radio que se encuentran registrados en el sistema.
<b>Acción de Entrada</b>	Listar todos los guiones de la radio que están registrados en el sistema
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Visualización exitosa de la información de todos los guiones de la radio que están registrados en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testGetListafiltro() { System.out.println("getListafiltro"); GuionProgramaRadialController instance = new GuionProgramaRadialController(); List&lt;GuionProgramaRadial&gt; expectedResult = null; List&lt;GuionProgramaRadial&gt; result = instance.getListafiltro(); assertEquals(expectedResult, result); fail("The test case is a prototype."); } @Test public void testSetListafiltro() { System.out.println("setListafiltro"); List&lt;GuionProgramaRadial&gt; listafiltro = null; </pre>

	<pre>         GuionProgramaRadialController instance = new         GuionProgramaRadialController();         instance.setListafiltro(listafiltro);         fail("The test case is a prototype.");     }     @Test     public void testPrepareList() {         System.out.println("prepareList");         GuionProgramaRadialController instance = new         GuionProgramaRadialController();         String expectedResult = "";         String result = instance.prepareList();         assertEquals(expectedResult, result);         fail("The test case is a prototype.");     }     @Test     public void testPrepareView() {         System.out.println("prepareView");         GuionProgramaRadialController instance = new         GuionProgramaRadialController();         String expectedResult = "";         String result = instance.prepareView();         assertEquals(expectedResult, result);         fail("The test case is a prototype.");     }     } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de Locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar guiones.</li> <li>3. Clic en el listar guiones.</li> <li>4. Y se visualiza la información de los guiones de la radio.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-16
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-16
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Primero se debe listar los guiones de la radio.</li> <li>• Seleccionar el guion que se va a editar.</li> <li>• Modificar la información del guion.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de modificar guion se puede cambiar la información que se guardó anteriormente en el sistema.
<b>Datos /Acción de Entrada</b>	Los campos para modificar son: Fecha Registro, Fecha Aprobación, Dialogo, Estado, Observación, Programa, Tipo Guion.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Modificación exitosa del guion seleccionado en el sistema.</li> </ul>
<b>Código</b>	<pre>     @Test     public void testPrepareEdit() {         System.out.println("prepareEdit");         GuionProgramaRadialController instance = new         GuionProgramaRadialController();         String expectedResult = "";         String result = instance.prepareEdit();         assertEquals(expectedResult, result);         fail("The test case is a prototype.");     } </pre>

```

@Test
public void testCancelUpdate() {
    System.out.println("cancelUpdate");
    GuionProgramaRadialController instance = new
GuionProgramaRadialController();
    String expectedResult = "";
    String result = instance.cancelUpdate();
    assertEquals(expectedResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testUpdate() {
    System.out.println("update");
    GuionProgramaRadialController instance = new
GuionProgramaRadialController();
    String expectedResult = "";
    String result = instance.update();
    assertEquals(expectedResult, result);
    fail("The test case is a prototype.");
}
@Test
public void testSetGuionProgramaRadialFechaRegistro() {
    System.out.println("setGuionProgramaRadialFechaRegistro");
    Date guionProgramaRadialFechaRegistro = "03/10/2022";
    GuionProgramaRadial instance = new GuionProgramaRadial();
    instance.setGuionProgramaRadialFechaRegistro
(guionProgramaRadialFechaRegistro);
    fail("The test case is a prototype.");
}
@Test
public void testSetGuionProgramaRadialFechaAprobacion() {
    System.out.println("setGuionProgramaRadialFechaAprobacion");
    Date guionProgramaRadialFechaAprobacion = "03/811/2022";
    GuionProgramaRadial instance = new GuionProgramaRadial();
    instance.setGuionProgramaRadialFechaAprobacion
(guionProgramaRadialFechaAprobacion);
    fail("The test case is a prototype.");
}
@Test
public void testSetDialogoId() {
    System.out.println("setDialogoId");
    Dialogo dialogoId = 1;
    GuionProgramaRadial instance = new GuionProgramaRadial();
    instance.setDialogoId(dialogoId);
    fail("The test case is a prototype.");
}
@Test
public void testSetEstadoGuionId() {
    System.out.println("setEstadoGuionId");
    EstadoGuion estadoGuionId = 1;
    GuionProgramaRadial instance = new GuionProgramaRadial();
    instance.setEstadoGuionId(estadoGuionId);
    fail("The test case is a prototype.");
}
@Test
public void testSetObservacionGuionProgramaId() {
    System.out.println("setObservacionGuionProgramaId");
    ObservacionGuionPrograma observacionGuionProgramaId = 1;
    GuionProgramaRadial instance = new GuionProgramaRadial();
    instance.setObservacionGuionProgramaId

```

	<pre>(observacionGuionProgramaId);     fail("The test case is a prototype."); } @Test public void testSetProgramaRadialId() {     System.out.println("setProgramaRadialId");     ProgramaRadial programaRadialId = 2;     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setProgramaRadialId(programaRadialId);     fail("The test case is a prototype."); } @Test public void testSetTipoGuionId() {     System.out.println("setTipoGuionId");     TipoGuion tipoGuionId = 1;     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setTipoGuionId(tipoGuionId);     fail("The test case is a prototype."); } }</pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de Locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar de Guiones.</li> <li>3. Clic en el Listar Guiones.</li> <li>4. Seleccionar el guion que se desee y presionamos en Editar.</li> <li>5. Ingresar la nueva información del guion.</li> <li>6. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-17
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-17
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los guiones.</li> <li>• Seleccionar el guion que se va a eliminar.</li> <li>• Dar clic en eliminar guion.</li> <li>• Confirmamos o cancelamos la eliminación del guion.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de eliminar guion se puede eliminar la información del guion que está registrado en el sistema.
<b>Datos /Acción de Entrada</b>	Permite eliminar toda la información de un guion de la radio del sistema.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Eliminación exitosa del guion seleccionada del sistema.</li> </ul>
<b>Código</b>	<pre>@Test public void testDestroy() {     System.out.println("destroy");     GuionProgramaRadialController instance = new GuionProgramaRadialController();     String expectedResult = "";     String result = instance.destroy();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); } @Test public void testDestroyAndView() {     System.out.println("destroyAndView"); }</pre>

	<pre> GuionProgramaRadialController instance = new GuionProgramaRadialController(); String expectedResult = ""; String result = instance.destroyAndView(); assertEquals(expResult, result); fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar guiones.</li> <li>3. Clic en el Listar Guiones.</li> <li>4. Seleccionar el guion que se desee y presionamos en Eliminar.</li> <li>5. Confirmamos o cancelamos la eliminación del guion.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-018
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-18
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Ingresar un nuevo dialogo al sistema de la radio.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de ingresar nuevos diálogos al sistema, el software permite registrar la información correspondiente al dialogo además podrá cargar el documento con el dialogo.
<b>Datos /Acción de Entrada</b>	Numero de Cabina, Cargar Dialogo, Tiempo Min
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Registro exitoso del nuevo dialogo en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testSetDialogoCabina() {     System.out.println("setDialogoCabina");     String dialogoCabina = "1";     Dialogo instance = new Dialogo();     instance.setDialogoCabina(dialogoCabina);     fail("The test case is a prototype."); }  @Test public void testSetDialogoLocutor() {     System.out.println("setDialogoLocutor");     String dialogoLocutor = "Dialogo.pdf";     Dialogo instance = new Dialogo();     instance.setDialogoLocutor(dialogoLocutor);     fail("The test case is a prototype."); }  @Test public void testCargarArchivo() {     System.out.println("cargarArchivo");     String fuente = "d:/documentos/dialogo.pdf";     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.cargarArchivo(fuente);     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testSetDialogoTiempo() { </pre>

	<pre>System.out.println("setDialogoTiempo"); Integer dialogoTiempo = "00:05:00"; Dialogo instance = new Dialogo(); instance.setDialogoTiempo(dialogoTiempo); fail("The test case is a prototype."); }</pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar diálogos.</li> <li>3. Clic en el Crear dialogo.</li> <li>4. Ingresar la información necesaria del dialogo en los campos.</li> <li>5. Dar clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-19
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-19
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Listar los diálogos que se encuentran registrados en el sistema</li> <li>• Ver el archivo que fue subido al sistema.</li> </ul>
<b>Funcionalidad/ Característica</b>	Visualiza la información de todos los diálogos de la radio que se encuentran registrados en el sistema.
<b>Acción de Entrada</b>	Listar todos los diálogos de la radio que están registrados en el sistema
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Visualización exitosa de la información de todos los diálogos de la radio que están registrados en el sistema.</li> </ul>
<b>Código</b>	<pre>@Test public void testPrepareList() {     System.out.println("prepareList");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.prepareList();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); }  @Test public void testPrepareView() {     System.out.println("prepareView");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.prepareView();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); }</pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de Locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar diálogos.</li> <li>3. Clic en el listar diálogos.</li> <li>4. Y se visualiza la información de los guiones de la radio.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-20

<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-20
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Primero se debe listar los diálogos de la radio.</li> <li>• Seleccionar el dialogo que se va a editar.</li> <li>• Modificar la información del dialogo.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de modificar dialogo se puede cambiar la información que se guardó anteriormente en el sistema.
<b>Datos /Acción de Entrada</b>	Los campos para modificar son: Numero de Cabina, Cargar Dialogo, Tiempo Min
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Modificación exitosa del dialogo seleccionado en el sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testPrepareEdit() {     System.out.println("prepareEdit");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.prepareEdit();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); }  @Test public void testCancelUpdate() {     System.out.println("cancelUpdate");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.cancelUpdate();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); }  @Test public void testUpdate() {     System.out.println("update");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.update();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); }  @Test public void testSetDialogoCabina() {     System.out.println("setDialogoCabina");     String dialogoCabina = "2";     Dialogo instance = new Dialogo();     instance.setDialogoCabina(dialogoCabina);     fail("The test case is a prototype."); }  @Test public void testSetDialogoLocutor() {     System.out.println("setDialogoLocutor");     String dialogoLocutor = "Dialogo.pdf";     Dialogo instance = new Dialogo();     instance.setDialogoLocutor(dialogoLocutor);     fail("The test case is a prototype."); }  @Test public void testCargarArchivo() {     System.out.println("cargarArchivo");     String fuente = "d:/documentos/dialogo.pdf"; </pre>



	<pre> DialogoController instance = new DialogoController(); String expectedResult = ""; String result = instance.cargarArchivo(fuente); assertEquals(expResult, result); fail("The test case is a prototype."); } @Test public void testSetDialogoTiempo() {     System.out.println("setDialogoTiempo");     Integer dialogoTiempo = "00:10:00";     Dialogo instance = new Dialogo();     instance.setDialogoTiempo(dialogoTiempo);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de Locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar diálogos.</li> <li>3. Clic en el Listar diálogos.</li> <li>4. Seleccionar el dialogo que se desee y presionamos en Editar.</li> <li>5. Ingresar la nueva información del dialogo.</li> <li>6. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-21
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-21
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los diálogos.</li> <li>• Seleccionar el dialogo que se va a eliminar.</li> <li>• Dar clic en eliminar dialogo.</li> <li>• Confirmamos o cancelamos la eliminación del diálogo.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de eliminar dialogo se puede eliminar la información del dialogo que está registrado en el sistema.
<b>Datos /Acción de Entrada</b>	Permite eliminar toda la información de un dialogo de la radio del sistema.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Eliminación exitosa del dialogo seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testDestroy() {     System.out.println("destroy");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.destroy();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testDestroyAndView() {     System.out.println("destroyAndView");     DialogoController instance = new DialogoController();     String expectedResult = "";     String result = instance.destroyAndView();     assertEquals(expResult, result);     fail("The test case is a prototype."); } </pre>

	<pre>@Test public void testEliminarArchivo() {     System.out.println("eliminarArchivo");     String path = "";     DialogoController instance = new DialogoController();     instance.eliminarArchivo(path);     fail("The test case is a prototype."); }</pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol de locutor de la radio.</li> <li>2. Ubicarse en el Módulo de gestionar diálogos.</li> <li>3. Clic en el Listar diálogos.</li> <li>4. Seleccionar el dialogo que se desee y presionamos en Eliminar.</li> <li>5. Confirmamos o cancelamos la eliminación del dialogo.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-22
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-22
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los guiones.</li> <li>• Seleccionar el dialogo que se va a revisar.</li> <li>• Dar las observaciones del guion.</li> <li>• Guardar la observación del guion.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de revisar guion se puede dar sugerencias del guion o a la vez aprobar o rechazar el guion el mismo que este refleja al locutor.
<b>Datos /Acción de Entrada</b>	Permite revisar un guion de la radio los mismos que están registrados en el sistema.
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Revisión exitosa del guion seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre>@Test public void testGetSelected() {     System.out.println("getSelected");     EstadoGuionController instance = new EstadoGuionController();     EstadoGuion expResult = null;     EstadoGuion result = instance.getSelected();     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testPrepareEdit() {     System.out.println("prepareEdit");     GuionProgramaRadialController instance = new GuionProgramaRadialController();     String expResult = "";     String result = instance.prepareEdit();     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testUpdate() {     System.out.println("update");     GuionProgramaRadialController instance = new GuionProgramaRadialController();     String expResult = "";</pre>

	<pre> String result = instance.update(); assertEquals(expResult, result); fail("The test case is a prototype."); } @Test public void testSetGuionProgramaRadialFechaRegistro() {     System.out.println("setGuionProgramaRadialFechaRegistro");     Date guionProgramaRadialFechaRegistro = "03/10/2022";     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setGuionProgramaRadialFechaRegistro (guionProgramaRadialFechaRegistro);     fail("The test case is a prototype."); } @Test public void testSetGuionProgramaRadialFechaAprobacion() {     System.out.println("setGuionProgramaRadialFechaAprobacion");     Date guionProgramaRadialFechaAprobacion = "03/11/2022";     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setGuionProgramaRadialFechaAprobacion (guionProgramaRadialFechaAprobacion);     fail("The test case is a prototype."); } @Test public void testSetDialogoId() {     System.out.println("setDialogoId");     Dialogo dialogoId = 1;     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setDialogoId(dialogoId);     fail("The test case is a prototype."); } @Test public void testSetEstadoGuionId() {     System.out.println("setEstadoGuionId");     EstadoGuion estadoGuionId = 1;     GuionProgramaRadial instance = new GuionProgramaRadial();     instance.setEstadoGuionId(estadoGuionId);     fail("The test case is a prototype."); } } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol del Coordinado de la radio.</li> <li>2. Ubicarse en el Módulo de los Guiones.</li> <li>3. Clic en el Revisar Guiones.</li> <li>4. Para ver el dialogo que este asignado a ese guion dar clic en Ver.</li> <li>5. Seleccionar el guion que se desee revisar y presionamos en Revisar.</li> <li>6. Revisamos el guion y guardamos.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-23
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-23
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Primero se debe listar los usuarios que se encuentran registrados.</li> <li>• Usamos el filtro para la búsqueda según se desee o todo en general.</li> <li>• Damos clic en el archivo de pdf o Excel para que se descargue.</li> </ul>
<b>Funcionalidad/</b>	Con la funcionalidad de Reportes de Usuarios se puede descargar la

<b>Característica</b>	información en archivo pdf o Excel
<b>Datos /Acción de Entrada</b>	Permite descargar la información de los usuarios la misma que puede ser de todos o a la vez usando los filtros de búsqueda
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>Reporte exitoso de los usuarios seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testListarUsuarios() {     System.out.println("listarUsuarios");     CoordinadorController instance = new CoordinadorController();     instance.listarUsuarios();     fail("The test case is a prototype."); }  @Test public void testIsListarUsuarios() {     System.out.println("isListarUsuarios");     CoordinadorController instance = new CoordinadorController();     boolean expResult = false;     boolean result = instance.isListarUsuarios();     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testSetListarUsuarios() {     System.out.println("setListarUsuarios");     boolean listarUsuarios = false;     CoordinadorController instance = new CoordinadorController();     instance.setListarUsuarios(listarUsuarios);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>Ingresar al sistema con el rol del Coordinado de la radio.</li> <li>Ubicarse en el Módulo de los Reportes.</li> <li>Clic en el Reporte de Usuarios.</li> <li>Si se desea utilizar los filtros de búsqueda caso contrario el general</li> <li>Dar clic en el icono de pdf o Excel según lo que se desee.</li> <li>Se descarga el reporte de los usuarios que están en el sistema.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-24
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-24.
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>Primero se debe listar los programas radiales que se encuentran registrados en el sistema.</li> <li>Usamos el filtro para la búsqueda según se desee o ponemos todo en general.</li> <li>Damos clic en el archivo de pdf o Excel para que se descargue.</li> </ul>
<b>Fecha:</b>	
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de Reportes de programas radiales se puede descargar la información en archivo pdf o Excel.
<b>Datos /Acción de Entrada</b>	Permite descargar la información de los programas radiales la misma que puede ser de todos o a la vez usando los filtros de búsqueda si se desea una búsqueda personalizada
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>Reporte exitoso de los programas radiales seleccionado del sistema.</li> </ul>

<b>Código</b>	<pre> @Test public void testListarProg() {     System.out.println("listarProg");     CoordinadorController instance = new CoordinadorController();     instance.listarProg();     fail("The test case is a prototype."); }  @Test public void testListarProgs() {     System.out.println("listarProgs");     CoordinadorController instance = new CoordinadorController();     instance.listarProgs();     fail("The test case is a prototype."); }  @Test public void testSetListarProgramas() {     System.out.println("setListarProgramas");     boolean listarProgramas = false;     CoordinadorController instance = new CoordinadorController();     instance.setListarProgramas(listarProgramas);     fail("The test case is a prototype."); }  @Test public void testIsListarProgs() {     System.out.println("isListarProgs");     CoordinadorController instance = new CoordinadorController();     boolean expResult = false;     boolean result = instance.isListarProgs();     assertEquals(expResult, result);     fail("The test case is a prototype."); }  @Test public void testSetListarProgs() {     System.out.println("setListarProgs");     boolean listarProgs = false;     CoordinadorController instance = new CoordinadorController();     instance.setListarProgs(listarProgs);     fail("The test case is a prototype."); } </pre>
---------------	--

#### Información para el seguimiento

<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol del Coordinado de la radio.</li> <li>2. Ubicarse en el Módulo de los Reportes.</li> <li>3. Clic en el Reporte de Programas radiales.</li> <li>4. Si se desea utilizar los filtros de búsqueda.</li> <li>5. Dar clic en el icono de pdf o Excel según lo que se desee.</li> <li>6. Se descarga el reporte de los programas que está en el sistema.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

#### Caso de Prueba

<b>Id</b>	CPU-25
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-25.
<b>Descripción</b>	<p>Estos casos de prueba pertenecen a las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Primero se debe listar los guiones que se encuentran registrados en el sistema.</li> </ul>

	<ul style="list-style-type: none"> <li>• Usamos el filtro para la búsqueda según se desee o ponemos todo en general.</li> <li>• Damos clic en el archivo de pdf o Excel para que se descargue.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de Reportes de guiones se puede descargar la información en archivo pdf o Excel.
<b>Datos /Acción de Entrada</b>	Permite descargar la información de los guiones la misma que puede ser de todos los registros o a la vez usando los filtros de búsqueda si se desea una búsqueda personalizada
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>• Reporte exitoso de los guiones seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testListarGuiones() {     System.out.println("listarGuiones");     CoordinadorController instance = new CoordinadorController();     instance.listarGuiones();     fail("The test case is a prototype."); } @Test public void testIsListarGuiones() {     System.out.println("isListarGuiones");     CoordinadorController instance = new CoordinadorController();     boolean expectedResult = false;     boolean result = instance.isListarGuiones();     assertEquals(expectedResult, result);     fail("The test case is a prototype."); } @Test public void testSetListarGuiones() {     System.out.println("setListarGuiones");     boolean listarGuiones = false;     CoordinadorController instance = new CoordinadorController();     instance.setListarGuiones(listarGuiones);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema con el rol del Coordinado de la radio.</li> <li>2. Ubicarse en el Módulo de los Reportes.</li> <li>3. Clic en el Reporte de Guiones.</li> <li>4. Si se desea utilizar los filtros de búsqueda.</li> <li>5. Dar clic en el icono de pdf o Excel según lo que se desee.</li> <li>6. Se descarga el reporte de los programas que está en el sistema.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna

<b>Caso de Prueba</b>	
<b>Id</b>	CPU-26
<b>Caso de prueba:</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-26.
<b>Descripción</b>	Estos casos de prueba pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Primero se debe listar las publicidades que se encuentran registrados en el sistema.</li> <li>• Usamos el filtro para la búsqueda según se desee o ponemos todo en general.</li> <li>• Damos clic en el archivo de pdf o Excel para que se descargue.</li> </ul>
<b>Funcionalidad/ Característica</b>	Con la funcionalidad de Reportes de publicidad se puede descargar la información en archivo pdf o Excel.

<b>Datos /Acción de Entrada</b>	Permite descargar la información de las publicidades la misma que puede ser de todos los registros o a la vez usando los filtros de búsqueda si se desea una búsqueda personalizada
<b>Resultado Esperado</b>	<ul style="list-style-type: none"> <li>Reporte exitoso de las publicidades seleccionado del sistema.</li> </ul>
<b>Código</b>	<pre> @Test public void testListarPu() {     System.out.println("listarPu");     CoordinadorController instance = new CoordinadorController();     instance.listarPu();     fail("The test case is a prototype."); } @Test public void testIsListarPublicidad() {     System.out.println("isListarPublicidad");     CoordinadorController instance = new CoordinadorController();     boolean expResult = false;     boolean result = instance.isListarPublicidad();     assertEquals(expResult, result);     fail("The test case is a prototype."); } @Test public void testSetListarPublicidad() {     System.out.println("setListarPublicidad");     boolean listarPublicidad = false;     CoordinadorController instance = new CoordinadorController();     instance.setListarPublicidad(listarPublicidad);     fail("The test case is a prototype."); } </pre>
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>Ingresar al sistema con el rol del Coordinado de la radio.</li> <li>Ubicarse en el Módulo de los Reportes.</li> <li>Clic en el Reporte de Publicidad.</li> <li>Si se desea utilizar los filtros de búsqueda .</li> <li>Dar clic en el icono de pdf o Excel según lo que se desee.</li> <li>Se descarga el reporte de los programas de radio que están registrados en el sistema.</li> </ol>
<b>Estado</b>	Éxito
<b>Observación</b>	Ninguna



## **IMPLEMENTACIÓN DE UN SISTEMA WEB APOYADO EN PRUEBAS UNITARIAS PARA LA ADMINISTRACIÓN DE GUIONES Y PUBLICIDAD EN LA RADIO CIUDAD 106.5 FM.**

### **MANUAL TÉCNICO**

#### **Introducción**

El presente documento es un registro del funcionamiento del sistema web para la gestión y administración de guiones y publicidad realizado para la Radio Ciudad 106.5 FM.

Este proyecto ha sido desarrollado con tecnología OpenSource, se ha utilizado el lenguaje Java, para la construida de la base de datos se utilizó PostgreSQL, junto con la tecnología de JSF y para el diseño de una interfaz responsiva el framework Bootstrap. Para el uso de dicho sistema web se define tres tipos de roles de usuario que el sistema acepta las mismas que son: productor, locutor y coordinador donde cada usuario cumple un rol importante dentro de la radio y del sistema.

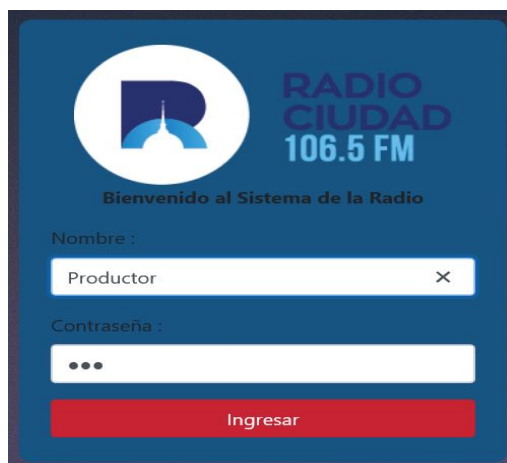
Con el fin de orientar al usuario en este manual se aborda, en el primer apartado de este documento contiene las funciones de acceso y autenticación de usuarios donde cada usuario debe ingresar sus credenciales para poder acceder al sistema, luego se aborda las diferentes funcionalidades propias de cada usuario dentro del sistema.

#### **Autenticación e Ingreso al Sistema**

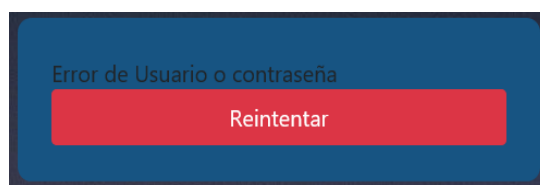
El proceso de autenticación hace referencia al ingreso de credenciales que permiten el ingreso al sistema, para ello, se debe ingresar al portal de inicio de sesión de la radio:

- La pantalla de ingreso al sistema consta de un formulario con el logo que identifica a la radio, además consta de dos cuadros de texto que permiten ingresar las credenciales, en este caso se deberá ingresar el nombre y contraseña del usuario, la credencial del usuario es creador por parte del productor de la radio.





- Si las credenciales son correctas, dependiendo del rol que el usuario ocupa en la entidad se da paso al perfil correspondiente. En caso de que las credenciales sean incorrectas el sistema emite mensajes de error.



### ➤ Rol del Productor de la Radio

El perfil del productor en el sistema permite realizar tareas como: la gestión de usuarios del sistema, la gestión de programas de radio y la gestión de publicidad que tiene la radio.

- Una vez ingresado las credenciales del productor, el sistema de paso al perfil del Productor del sistema. A continuación, se puede observar el área de trabajo del productor.

Id	Nombre	Apellido	Cédula	Contraseña	Tipo de Usuario	Editar	Eliminar
11	Coordinador	Lopez	1002345678	123	Coordinador	✓	✗
22	Locutor	Romero	1234567890	123	Locutor	✓	✗
8	Productor	Perez	1231234509	123	Productor	✓	✗

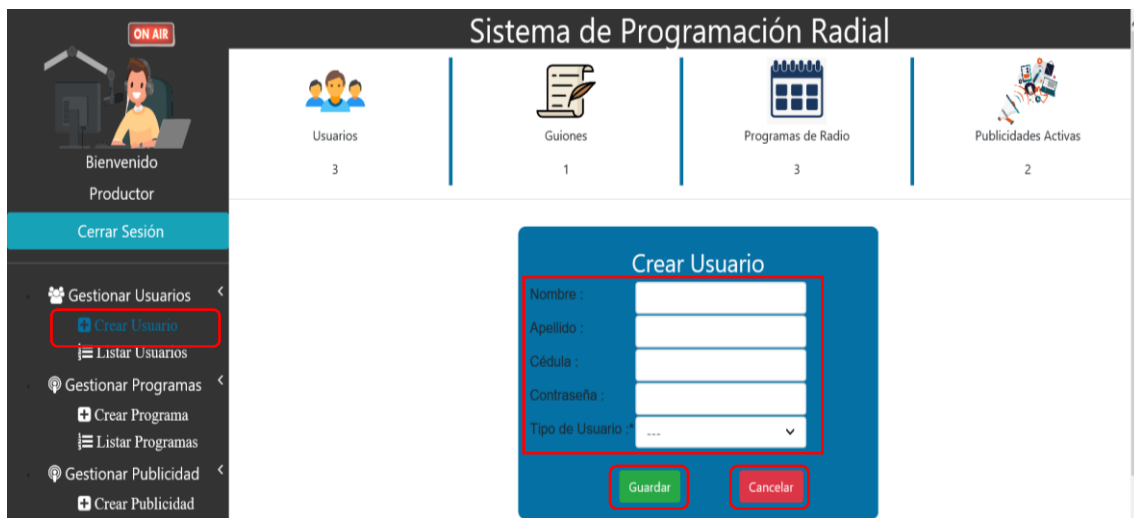
Este perfil consta en la parte superior izquierda del sistema se tiene el nombre, el tipo de usuario que está en el sistema y el botón de cerrar sesión. Debajo de esto se encuentran las tres categorías de trabajo propias del perfil del productor. En el centro se encuentra el área de trabajo donde se va a visualizar la información según la categoría que se seleccione, y además en la parte superior se encuentra la cantidad de usuarios, guiones, programas de radio y publicidad que tiene el sistema.

## 1. Gestión de Usuarios.

La categoría de gestionar usuarios permite crear un nuevo usuario, la modificación del registro de usuarios y la eliminación de los usuarios del sistema.

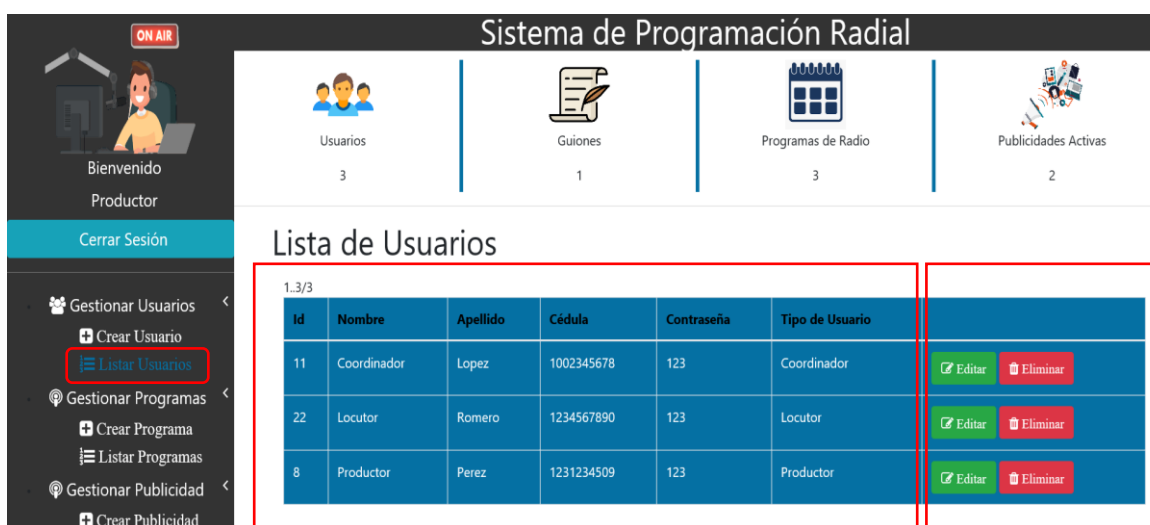
### Crear Usuario

- Para el ingreso de nuevos usuarios en el sistema, primero se debe dar clic en la opción Crear Usuarios y nos asomara un cuadro donde nos pide la información del nuevo usuario, una vez llenado la información damos en guardar, en caso de no querer guardar la información se debe dar clic en el botón Cancelar para salir del formulario. Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada.



### Listar Usuarios

- Permite visualizar la información de los usuarios que se encuentra registrados en el sistema, además de este apartado se puede modificar la información de los usuarios o a la vez eliminar los usuarios.



### Modificar Usuarios

- De la lista de usuarios seleccionamos el que queremos modificar y damos clic en el botón editar como se muestra a continuación.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 3 | Publicidades Activas: 2

**Lista de Usuarios**

1.3/3

Id	Nombre	Apellido	Cédula	Contraseña	Tipo de Usuario	
11	Coordinador	Lopez	1002345678	123	Coordinador	<input checked="" type="checkbox"/> Editar <input type="checkbox"/> Eliminar
22	Locutor	Romero	1234567890	123	Locutor	<input checked="" type="checkbox"/> Editar <input type="checkbox"/> Eliminar
8	Productor	Perez	1231234509	123	Productor	<input checked="" type="checkbox"/> Editar <input type="checkbox"/> Eliminar

- A continuación nos asomara el formulario con la información del usuario y podemos hacer la respectiva modificación. Una vez modificado la información damos en guardar, caso contrario damos clic en el botón Cancelar.
- Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada, y se puede verificar la modificación en el listar usuarios.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 3 | Publicidades Activas: 2

**Editar Usuario**

Nombre :

Apellido :

Cédula :

Contraseña :

Tipo de Usuario :

Guardar  Cancelar

### Eliminar Usuario

- Estando en el listado de usuarios elegimos el que se desee eliminar y damos clic en el botón Eliminar que se encuentra de color rojo.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 3 | Publicidades Activas: 2

**Lista de Usuarios**

1.3/3

Id	Nombre	Apellido	Cédula	Contraseña	Tipo de Usuario	
11	Coordinador	Lopez	1002345678	123	Coordinador	<input checked="" type="checkbox"/> Editar <input type="checkbox"/> Eliminar
22	Locutor	Romero	1234567890	123	Locutor	<input checked="" type="checkbox"/> Editar <input type="checkbox"/> Eliminar
8	Productor	Perez	1231234509	123	Productor	<input checked="" type="checkbox"/> Editar <input type="checkbox"/> Eliminar

- A continuación nos muestra una advertencia en una ventana emergente donde el productor puede confirmar o cancelar la eliminación del usuario.

The screenshot shows the 'Sistema de Programación Radial' dashboard. At the top, there are four main categories: Usuarios (3), Guiones (1), Programas de Radio (3), and Publicidades Activas (2). Below this is the 'Lista de Usuarios' table with the following data:

Id	Nombre	Apellido	Cedula	Tipo de Usuario	Acciones
11	Coordinador	Lopez	10023	Coordinador	[Editar] [Eliminar]
22	Locutor	Romero	12345	Locutor	[Editar] [Eliminar]
8	Productor	Perez	1231234509	Productor	[Editar] [Eliminar]

A confirmation dialog box is overlaid on the table, asking: '¿Está seguro que desea eliminar?' with 'Aceptar' and 'Cancelar' buttons.

## 2. Gestión de Programas de Radio.

La categoría de Gestionar Programas incluye, ingresar, modificar y eliminar la información de los programas que se emiten en la radio las mismas que se encuentran en el sistema.

### Crear Programa

- Para el ingreso de nuevos programas de radio en el sistema, primero se debe dar clic en la opción Crear Programa y nos asomara un cuadro donde nos pide la información del nuevo programa de radio.

The screenshot shows the 'Sistema de Programación Radial' dashboard with the 'Gestionar Programas' menu item highlighted. The 'Crear Programa Radial' form is displayed with the following fields:

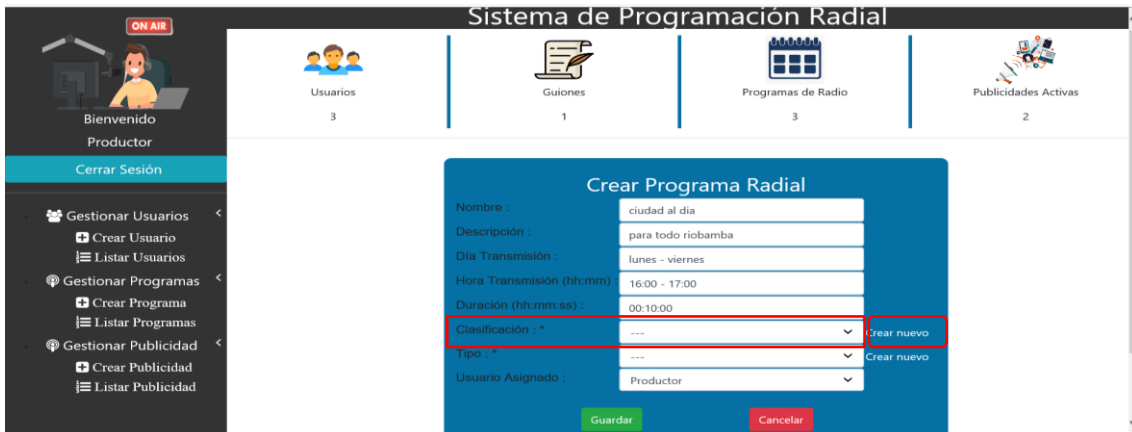
- Nombre: [Input field]
- Descripción: [Input field]
- Día Transmisión: [Input field]
- Hora Transmisión (hh:mm): [Input field]
- Duración (hh:mm:ss): [Input field]
- Clasificación: [Dropdown menu]
- Tipo: [Dropdown menu]
- Usuario Asignado: [Dropdown menu]

Buttons for 'Guardar' and 'Cancelar' are at the bottom of the form.

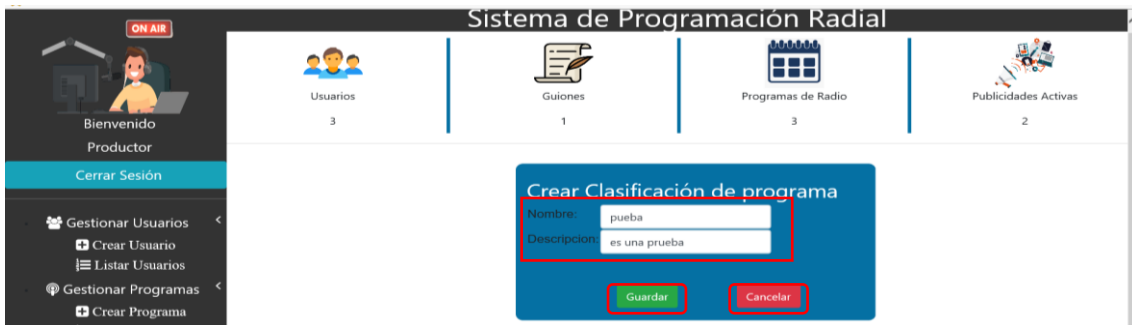
- Una vez llenado la información damos en guardar, en caso de no querer guardar la información se debe dar clic en el botón Cancelar para salir del formulario. Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada.

This screenshot is identical to the previous one, but with red boxes highlighting the 'Crear Programa' menu item in the sidebar and the 'Guardar' and 'Cancelar' buttons at the bottom of the form.

- En caso de que se desee crear una nueva clasificación para el programa de radio damos clic donde dice crear nuevo como nos indica a continuación.



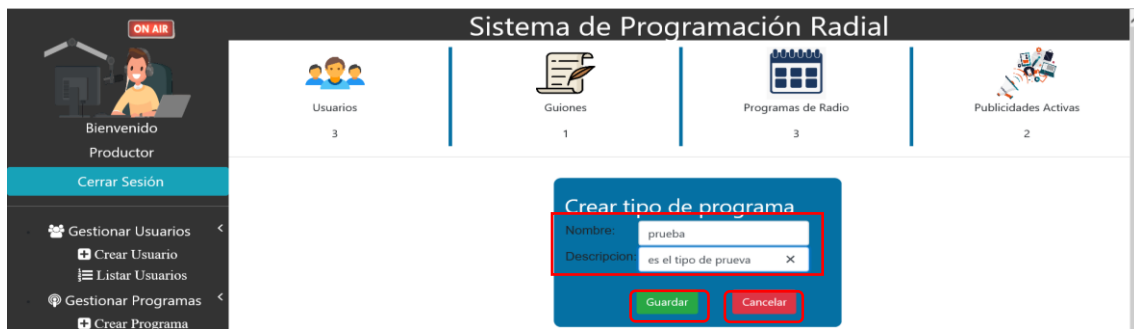
- A continuación, nos asoma una ventana donde ponemos el nombre y descripción de esa clasificación del programa. Una vez llenada la información damos en guardar, caso contrario damos clic en el botón Cancelar.



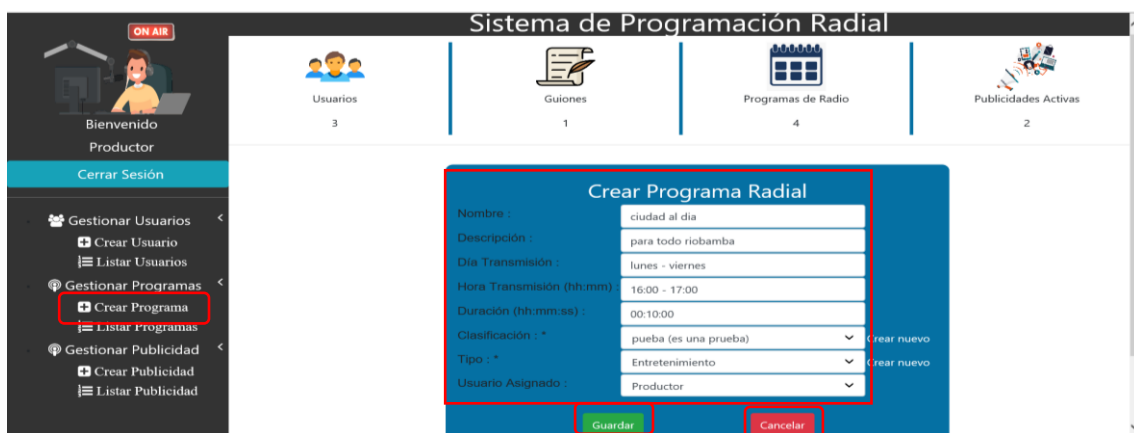
- De la misma manera si se desea crear un nuevo tipo de programa de radio damos clic donde dice Crear nuevo como nos indica a continuación.



- A continuación, nos asoma una ventana donde ponemos el nombre y descripción del nuevo tipo del programa. Una vez llenada la información damos en guardar, caso contrario damos clic en el botón Cancelar.

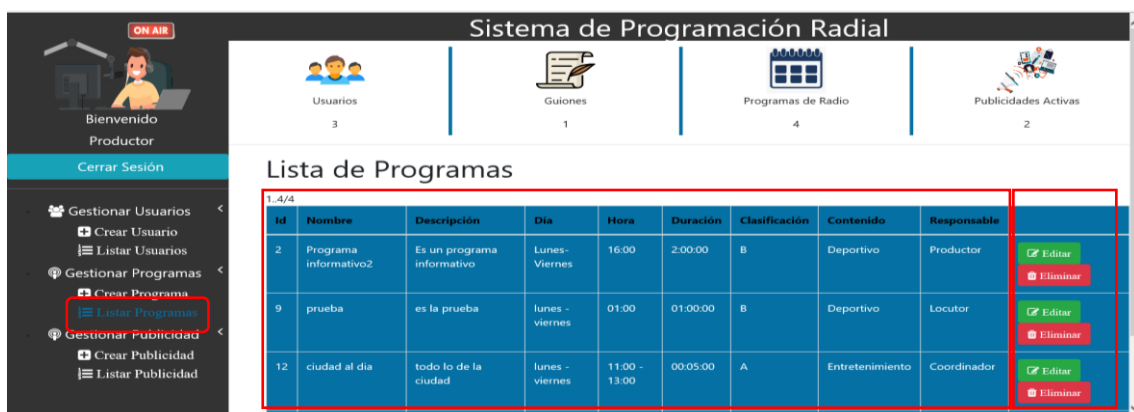


- Y para finalizar la creación del programa de radio llenamos toda la información del programa, una vez llenado la información damos en guardar, caso contrario damos en el botón Cancelar para salir del formulario. Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada.



### Listar Programas

- Permite visualizar la información de todos los programas de radio que se encuentra registrados en el sistema, además de este apartado se puede modificar la información de los usuarios o a la vez eliminar los usuarios.



### Modificar Programas

- De la lista de programas seleccionamos el que queremos modificar y damos clic en el botón verde que es el de editar como se muestra a continuación.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 4 | Publicidades Activas: 2

**Lista de Programas**

Id	Nombre	Descripción	Día	Hora	Duración	Clasificación	Contenido	Responsable	
2	Programa informativo2	Es un programa informativo	Lunes-Viernes	16:00	2:00:00	B	Deportivo	Productor	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>
9	prueba	es la prueba	lunes - viernes	01:00	01:00:00	B	Deportivo	Locutor	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>
12	ciudad al día	todo lo de la ciudad	lunes - viernes	11:00 - 13:00	00:05:00	A	Entretenimiento	Coordinador	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>

- A continuación, nos asomara el formulario con la información del usuario y podemos hacer la respectiva modificación. Una vez modificado la información damos en guardar, caso contrario damos clic en el botón Cancelar.
- Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada, y se puede verificar la modificación en el listar usuarios.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 4 | Publicidades Activas: 2

**Editar Programa Radial**

Nombre: ciudad al día  
 Descripción: todo lo de la ciudad  
 Día Transmisión: lunes - viernes  
 Hora transmisión (hh:mm): 11:00 - 13:00  
 Duración (hh:mm:ss): 00:05:00  
 Clasificación: A (Todo Público)   
 Tipo: Entretenimiento   
 Usuario Asignado: Coordinador

### Eliminar Programa

- Estando en el listado de los programas de radio elegimos el que se desee eliminar y damos clic en el botón Eliminar que se encuentra de color rojo.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 4 | Publicidades Activas: 2

**Lista de Programas**

Id	Nombre	Descripción	Día	Hora	Duración	Clasificación	Contenido	Responsable	
2	Programa informativo2	Es un programa informativo	Lunes-Viernes	16:00	2:00:00	B	Deportivo	Productor	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>
9	prueba	es la prueba	lunes - viernes	01:00	01:00:00	B	Deportivo	Locutor	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>
12	ciudad al día	todo lo de la ciudad	lunes - viernes	11:00 - 13:00	00:05:00	A	Entretenimiento	Coordinador	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>

- A continuación, nos muestra una advertencia en una ventana emergente donde el productor puede confirmar o cancelar la eliminación del usuario.

The screenshot shows the 'Sistema de Programación Radial' dashboard. At the top, there are four main sections: Usuarios (3), Guiones (1), Programas de Radio (4), and Publicidades Activas (2). Below this is a 'Lista de Programas' table with columns: Id, Nombre, Descripción, Fecha Inicio, Fecha Fin, Clasificación, Contenido, Responsable, and Acciones. A confirmation dialog box is overlaid on the table, asking '¿Está seguro que desea eliminar?' with 'Aceptar' and 'Cancelar' buttons. The 'Eliminar' button in the actions column of the first row is highlighted with a red box.

Id	Nombre	Descripción	Fecha Inicio	Fecha Fin	Clasificación	Contenido	Responsable	Acciones	
2	Programa informativo2	Es un programa informativo	01:00:00	01:00:00	B	Deportivo	Productor	[Editar] [Eliminar]	
9	prueba	es la prueba	lunes - viernes	01:00	01:00:00	B	Deportivo	Locutor	[Editar] [Eliminar]
12	ciudad al día	todo lo de la ciudad	lunes - viernes	11:00 - 13:00	00:05:00	A	Entretenimiento	Coordinador	[Editar] [Eliminar]

### 3. Gestión de Publicidad.

La categoría de Gestionar Publicidad incluye ingresar, modificar y eliminar la información de las publicidades que se emiten en la radio las mismas que se encuentran registradas en el sistema.

#### Crear Publicidad

- Para el ingreso de nuevas publicidades en el sistema, primero se debe dar clic en la opción Crear Programa y nos asomara un cuadro donde nos pide la información de la publicidad que se va a crear.

The screenshot shows the 'Sistema de Programación Radial' dashboard with the 'Crear Publicidad' form open. The form has the following fields: Nombre (input), Descripción (input), Fecha Inicio (mm/dd/yyyy), Fecha Fin (mm/dd/yyyy), Estado (dropdown menu with 'Activo' selected), Día (input), Horario (input), and Programa Radial (dropdown menu). There are 'Guardar' and 'Cancelar' buttons at the bottom of the form.

- Una vez llenado la información damos en guardar, caso contrario damos clic en el botón Cancelar para salir del formulario. Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada.

The screenshot shows the 'Sistema de Programación Radial' dashboard with the 'Crear Publicidad' form filled with data. The fields are: Nombre (ventas), Descripción (todas las ventas), Fecha Inicio (02/02/2022), Fecha Fin (02/05/2022), Estado (Activo), Día (lunes - viernes), Horario (06:00 - 07:00 - 10:00), and Programa Radial (Programa informativo2). The 'Guardar' and 'Cancelar' buttons are highlighted with red boxes.

#### Listar Publicidad

- Permite visualizar la información de todas las publicidades de la radio que se encuentra registrados en el sistema, además en este apartado se puede modificar la información de las publicidades o a la vez eliminar las publicidades.



**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Publicidad**

Id	Nombre	Descripción	Fecha de Inicio	Fecha de Fin	Estado	Horario	Programa Radial	Editar	Eliminar
26	Anuncios	todos los del municipio	02/12/2022	02/13/2022	Terminado	lunes - sábado 13:00 - 16:00	ciudad al día	Editar	Eliminar
28	sddd	sss	02/04/2022	02/04/2022	Activo	lunes - domingo 13:00	ciudad al día	Editar	Eliminar
29	ventas	todas las ventas	02/02/2022	02/05/2022	Activo	lunes - viernes 06:00 - 07:00 - 10:00	Programa informativo 2	Editar	Eliminar

### Modificar Publicidad

- De la lista de publicidades, seleccionamos el que queremos modificar y damos clic en el botón verde que es el de editar como se muestra a continuación.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Publicidad**

Id	Nombre	Descripción	Fecha de Inicio	Fecha de Fin	Estado	Horario	Programa Radial	Editar	Eliminar
26	Anuncios	todos los del municipio	02/12/2022	02/13/2022	Terminado	lunes - sábado 13:00 - 16:00	ciudad al día	Editar	Eliminar
28	sddd	sss	02/04/2022	02/04/2022	Activo	lunes - domingo 13:00	ciudad al día	Editar	Eliminar
29	ventas	todas las ventas	02/02/2022	02/05/2022	Activo	lunes - viernes 06:00 - 07:00 - 10:00	Programa informativo 2	Editar	Eliminar

- A continuación, nos asomara el formulario con la información de la publicidad y podemos hacer la respectiva modificación. Una vez modificado la información damos en guardar, caso contrario damos clic en el botón Cancelar.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 1 | Programas de Radio: 4 | Publicidades Activas: 3

**Editar Publicidad**

Nombre:

Descripción:

Fecha de Inicio (mm/dd/aaaa):

Fecha de Fin (mm/dd/aaaa):

Estado:

Día:

Hora:

Programa Radial:

- Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada, y se puede verificar la modificación en el listar publicidad.

### Eliminar Publicidad

- Estando en el listado de las publicidades de radio elegimos el que se desea eliminar y damos clic en el botón Eliminar que se encuentra de color rojo.

Id	Nombre	Descripción	Fecha de inicio	Fecha de Fin	Estado	Horario	Programa Radial		
26	Anuncios	todos los del municipio	02/12/2022	02/13/2022	Terminado	lunes - sabado 13:00 - 16:00	ciudad al dia	Editar	Eliminar
28	sddd	sss	02/04/2022	02/04/2022	Activo	lunes - domingo 13:00	ciudad al dia	Editar	Eliminar
29	ventas	todas las ventas	02/02/2022	02/05/2022	Activo	lunes - viernes 06:00 - 07:00 - 10:00	Programa informativo2	Editar	Eliminar

- A continuación, nos muestra una advertencia en una ventana emergente donde el productor puede confirmar o cancelar la eliminación de la publicidad.

### ➤ Rol del Locutor de la Radio.

Este perfil permite realizar tareas como la gestión de un guion y la gestión de diálogos los mismos que son asignados al guion. Además, se puede revisar el estado, la observación y las recomendaciones que el coordinador le da al guion tiene.

- Al ingresar en el formulario de inicio de sesión las credenciales correctas pertenecientes a un locutor, el sistema muestra la siguiente pantalla

Id	Fecha Registro	Fecha Aprobación	Diálogo	Estado	Observación	Programa	Tipo de Guión		
1	08/04/2021	08/04/2021	Orden_Pago_No_305379 (2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	Editar	Eliminar

Este perfil consta en la parte superior izquierda del sistema se tiene el nombre, el tipo de usuario que está en el sistema, el logo del locutor y el botón de cerrar sesión. Debajo de esto se encuentran las dos categorías de trabajo propias del perfil del locutor. En el centro se encuentra el área de trabajo donde se va a visualizar la información según la categoría que se seleccione, y además en la parte superior se encuentra la cantidad de usuarios, guiones, programas de radio y publicidad que tiene el sistema.

## 1. Gestión de Guiones.

La categoría de Gestionar Guion incluye ingresar, modificar y eliminar la información de los guiones que se emiten en la radio las mismas que se encuentran registradas en el sistema.

### Crear Guion

- Para el ingreso de nuevos guiones en el sistema, primero se debe dar clic en la opción Crear Guion y nos asomara un cuadro donde nos pide la información del guion que se va a crear.

The screenshot shows the 'Sistema de Programación Radial' dashboard. The top navigation bar includes 'Usuarios' (3), 'Guiones' (1), 'Programas de Radio' (4), and 'Publicidades Activas' (3). The left sidebar contains 'Gestionar Guión' with sub-options 'Crear Guión', 'Listar Guiones', 'Gestionar Diálogos', 'Crear Diálogo', and 'Listar Diálogos'. The 'Crear Guión' option is highlighted with a red box. The main content area displays the 'Crear Guión' form with the following fields: 'Fecha Registro(mm/dd/yyyy) : \*', 'Fecha Aprobación(mm/dd/yyyy)', 'Diálogo :', 'Estado Guión :', 'Observación :', 'Programa Radial :', and 'Tipo Guión : \*'. The 'Guardar' and 'Cancelar' buttons are at the bottom.

- Una vez llenado la información que nos pide del guion damos clic en guardar, caso contrario damos clic en el botón Cancelar para salir del formulario. Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada.

The screenshot shows the 'Sistema de Programación Radial' dashboard with the 'Crear Guión' form filled with data. The 'Fecha Registro(mm/dd/yyyy) : \*' field contains '02/28/2022', 'Fecha Aprobación(mm/dd/yyyy)' contains '02/28/2022', 'Diálogo :' contains 'logo\_ciudad.png', 'Estado Guión :' contains '---', 'Observación :', 'Programa Radial :' contains 'ciudad al día', and 'Tipo Guión : \*' contains 'Abierto'. The 'Guardar' and 'Cancelar' buttons are at the bottom.

### Listar Guion

- Permite visualizar la información de todos los guiones de la radio que se encuentra registrados en el sistema, además en este apartado se puede modificar la información del guion o a la vez eliminar el guion del sistema.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Guiones**

Id	Fecha Registro	Fecha Aprobación	Diálogo	Estado	Observación	Programa	Tipo de Guión	
1	08/04/2021	08/04/2021	Orden_Pago_No_305379 (2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	<input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
21	02/28/2022	02/28/2022	logo_ciudad.png			ciudad al dia	Abierto	<input type="checkbox"/> Editar <input type="checkbox"/> Eliminar

### Modificar Guión

- De la lista de guiones, seleccionamos el que queremos modificar y damos clic en el botón verde que es el de editar como se muestra a continuación.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Guiones**

Id	Fecha Registro	Fecha Aprobación	Diálogo	Estado	Observación	Programa	Tipo de Guión	
1	08/04/2021	08/04/2021	Orden_Pago_No_305379 (2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	<input type="checkbox"/> Editar <input type="checkbox"/> Eliminar
21	02/28/2022	02/28/2022	logo_ciudad.png			ciudad al dia	Abierto	<input type="checkbox"/> Editar <input type="checkbox"/> Eliminar

- A continuación, nos asomara el formulario con la información del guion y podemos hacer la respectiva modificación. Una vez modificado la información damos en guardar, caso contrario damos clic en el botón Cancelar.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Editar Guión**

Fecha Registro(mm/dd/yyyy) : 02/28/2022  
 Fecha Aprobación (mm/dd/yyyy) : 02/28/2022  
 Diálogo : logo\_ciudad.png  
 Estado Guión : ---  
 Observación :  
 Programa Radial : ciudad al dia  
 Tipo Guión : \*

- Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada, y se puede verificar la modificación en el listar guiones.

### Eliminar Guión

- Para eliminar un guion estando en el listado de los guiones que tiene la radio elegimos el que se desee eliminar y damos clic en el botón Eliminar que se encuentra de color rojo.

Sistema de Programación Radial

Usuarios: 3, Guiones: 2, Programas de Radio: 4, Publicidades Activas: 3

Lista de Guiones

Id	Fecha Registro	Fecha Aprobación	Diálogo	Estado	Observación	Programa	Tipo de Guión	
1	08/04/2021	08/04/2021	Orden_Pago_No_305379 (2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>
21	02/28/2022	02/28/2022	logo_ciudad.png			ciudad al dia	Abierto	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>

- A continuación, nos muestra una advertencia en una ventana emergente donde el productor puede confirmar o cancelar la eliminación de la publicidad.

Sistema de Programación Radial

Usuarios: 3, Guiones: 2, Programas de Radio: 4, Publicidades Activas: 3

Lista de Guiones

Id	Fecha Registro	Fecha Aprobación	Diálogo	Estado	Observación	Programa	Tipo de Guión	
1	08/04/2021	08/04/2021	Orden_Pago_No_305379 (2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>
21	02/28/2022	02/28/2022	logo_ciudad.png			ciudad al dia	Abierto	<input type="checkbox"/> Editar <input type="button" value="Eliminar"/>

Mensaje de la página web  
¿Está seguro que desea eliminar?

## 2. Gestión de Diálogos.

La categoría de Gestionar Diálogos incluye ingresar, modificar y eliminar la información de los diálogos que se emiten en la radio las mismas que se encuentran registradas en el sistema. Además, en esta categoría el locutor va poder subir el archivo del dialogo el mismo que va a ser asignado al guion.

### Crear Diálogos

- Para el ingreso de nuevos diálogos al sistema, primero se debe dar clic en la opción Crear Dialogo y nos asomara un cuadro donde nos pide la información del dialogo que se va a crear.

Sistema de Programación Radial

Usuarios: 3, Guiones: 2, Programas de Radio: 4, Publicidades Activas: 3

Crear Diálogo

Número cabina:

Cargar diálogo:

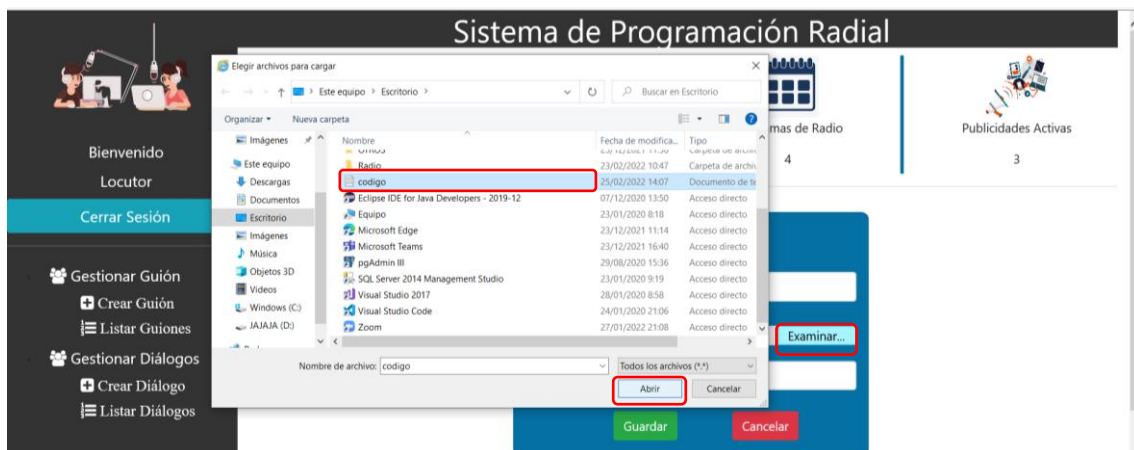
Tiempo min.:

- Para cargar el archivo del dialogo damos clic en examinar y elegimos el archivo que es el dialogo.

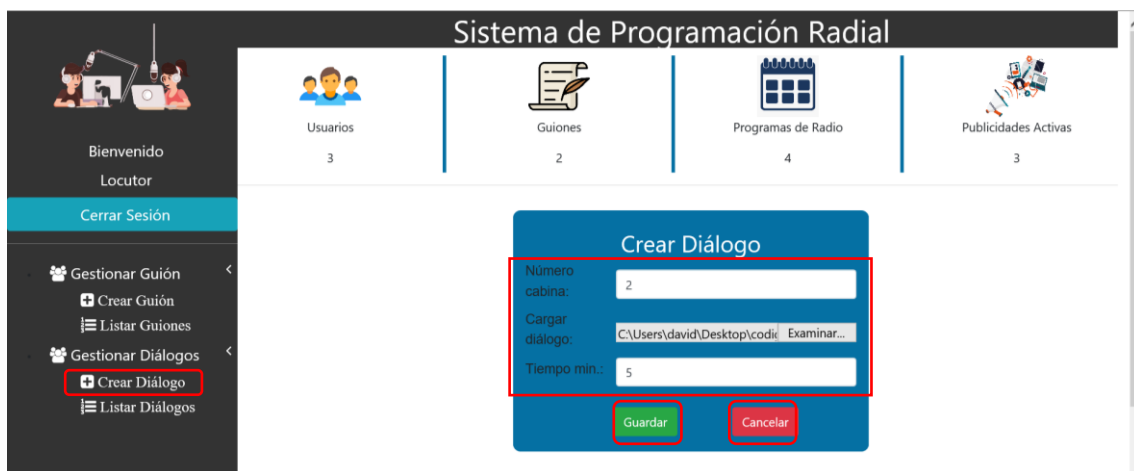


**Nota:** el archivo del dialogo puede ser cualquier tipo de extensión (Word, pdf, txt, etc.)

- Elegimos nuestro archivo del dialogo y damos clic en abrir para cargar el archivo



- Una vez llenado la información que nos pide del dialogo damos clic en guardar, caso contrario damos clic en el botón Cancelar para salir del formulario. Luego de guardar la información el sistema creará el dialogo y se subirá el archivo al sistema y emitirá el mensaje correspondiente indicando la acción realizada.



### Listar Diálogos

- Permite visualizar la información de todos los diálogos de la radio que se encuentra registrados en el sistema, además en este apartado se puede modificar la información del dialogo o a la vez eliminar el dialogo del sistema.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Diálogos**

Id	Cabina	Archivo Diálogo	Tiempo min.	Ver	Editar	Eliminar
51	1	Orden_Pago_No_305379 (2).pdf	5	Ver	Editar	Eliminar
52	1	logo_ciudad.png	3	Ver	Editar	Eliminar
55	2	codigo.txt	2	Ver	Editar	Eliminar
56	2	codigo.txt	5	Ver	Editar	Eliminar

### Ver Archivo

- Estando en listar diálogos para poder ver el archivo que se subió al sistema damos clic en donde dice ver y nos abrirá el archivo que tiene el dialogo.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Diálogos**

Id	Cabina	Archivo Diálogo	Tiempo min.	Ver	Editar	Eliminar
51	1	Orden_Pago_No_305379 (2).pdf	5	Ver	Editar	Eliminar
52	1	logo_ciudad.png	3	Ver	Editar	Eliminar
55	2	codigo.txt	2	Ver	Editar	Eliminar
56	2	codigo.txt	5	Ver	Editar	Eliminar

### Modificar Dialogo

- De la lista de diálogos que se encuentra en el sistema, seleccionamos el que queremos modificar y damos clic en el botón verde que es el de editar como se muestra a continuación.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Lista de Diálogos**

Id	Cabina	Archivo Diálogo	Tiempo min.	Ver	Editar	Eliminar
51	1	Orden_Pago_No_305379 (2).pdf	5	Ver	Editar	Eliminar
52	1	logo_ciudad.png	3	Ver	Editar	Eliminar
55	2	codigo.txt	2	Ver	Editar	Eliminar
56	2	codigo.txt	5	Ver	Editar	Eliminar

- A continuación, nos asomara el formulario con la información del dialogo y podemos hacer la respectiva modificación. Una vez modificado la información damos en guardar, caso contrario damos clic en el botón Cancelar.

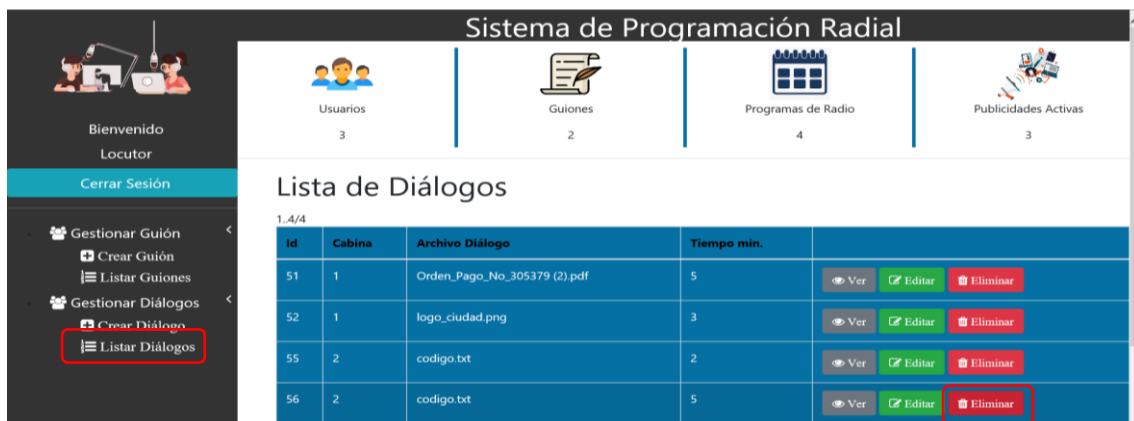




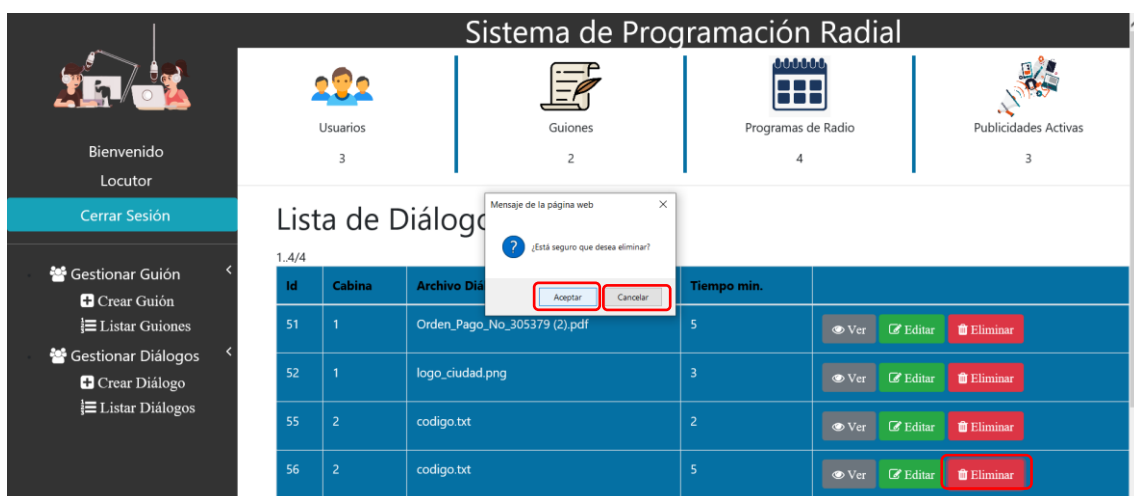
- Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada, y se puede verificar la modificación en el listar diálogos.

### Eliminar Diálogos

- Para eliminar un diálogo estando en el listado de los diálogos que tiene la radio elegimos el que se desee eliminar y damos clic en el botón Eliminar que se encuentra de color rojo.



- A continuación, nos muestra una advertencia en una ventana emergente donde el locutor puede confirmar o cancelar la eliminación del diálogo.



- Luego de la eliminación el sistema emitirá el mensaje correspondiente indicando la acción realizada, y se puede verificar en el listar diálogos.



## ➤ Rol del Coordinador de la Radio.

Este perfil permite realizar tareas como la revisión de guiones y la gestión de reportes de todas las categorías que son: usuarios, programas de radio, guiones y publicidad.

- Al ingresar las credenciales correctas pertenecientes al Coordinador, el sistema muestra la siguiente pantalla.

Id	Fecha de Registro	Fecha de Aprobación	Diálogo	Estado	Observación	Programa	Tipo Guión	
1	08/04/2021	08/04/2021	Orden_Pago_(2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	Revisar Ver
21	02/27/2022	02/27/2022	logo_ciudad.pi			ciudad al dia	Abierto	Revisar Ver

Este perfil consta en la parte superior izquierda del sistema se tiene el nombre, el tipo de usuario que está en el sistema, el logo del Coordinador y el botón de cerrar sesión. Debajo de esto se encuentran las dos categorías de trabajo propias del perfil del coordinador. En el centro se encuentra el área de trabajo donde se va a visualizar la información según la categoría que se seleccione, y además en la parte superior se encuentra la cantidad de usuarios, guiones, programas de radio y publicidad que tiene el sistema.

En la categoría de revisar guiones el coordinador puede revisar los guiones donde puede escribir sus recomendaciones y el cual será reflejados al locutor. De la misma manera puede hacer los reportes generales de los usuarios, programas de radio, guiones y de publicidades, estos reportes se pueden generar en dos formatos en pdf y en Excel.

### 1. Revisión de Guiones

#### Revisar Guiones.

- Para revisar los guiones damos clic en la categoría de Revisar Guiones, primero debemos ver el archivo del dialogo que tiene el guion y según eso para dar las observaciones. Para ver el archivo damos clic en Ver y se nos va a abrir el archivo.

Id	Fecha de Registro	Fecha de Aprobación	Diálogo	Estado	Observación	Programa	Tipo Guión	
1	08/04/2021	08/04/2021	Orden_Pago_(2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	Revisar Ver
21	02/27/2022	02/27/2022	logo_ciudad.pi			ciudad al dia	Abierto	Revisar Ver

- Una vez revisado el archivo del dialogo procedemos a revisar el guion para eso damos clic en revisar y se nos va a abrir la información del guion.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Revisión de Guiones**

Id	Fecha de Registro	Fecha de Aprobación	Diálogo	Estado	Observación	Programa	Tipo Guión	
1	08/04/2021	08/04/2021	Orden_Pago_(2).pdf	Registrado	PROBADO	Programa informativo2	Abierto	Revisar
21	02/27/2022	02/27/2022	logo_ciudad.pi			ciudad al dia	Abierto	Revisar

- Según lo que el coordinador decida puede llenar las observaciones del guion el cual será reflejado en el perfil del locutor, donde el coordinador puede guardar la revisión del guion o cancelar la revisión.

**Sistema de Programación Radial**

Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3

**Revisar Guión**

Fecha de Registro(mm/dd/yyyy) : 02/27/2022  
 Fecha de Aprobación (mm/dd/yyyy) : 02/27/2022  
 Diálogo : logo\_ciudad.png  
 Estado Guión : Rechazado  
 Observación : no tiene sentido  
 Programa Radial : ciudad al dia  
 Tipo Guión : Abierto

Guardar | Cancelar

## 2. Reportes

### Reporte General de Usuarios.

- Para hacer los reportes generales de los usuarios que están registrados en el sistema debemos hacer clic en donde dice usuarios y se nos va a abrir el listado de los usuarios.

**Sistema de Programación Radial**


Usuarios: 3 | Guiones: 2 | Programas de Radio: 4 | Publicidades Activas: 3


**Reporte de Usuarios**


Id	Nombre	Apellido	Cédula	Contraseña	Tipo de Usuario
11	Coordinador	Lopez	1002345678	123	Coordinador
22	Locutor	Romero	1234567890	123	Locutor
8	Productor	Perez	1231234509	123	Productor


- Si el coordinador desea puede utilizar los filtros de búsqueda si se desea una búsqueda personalizada, caso contrario damos clic en los iconos de pdf o de Excel según se desee y así se nos genera el archivo con la información de los usuarios.

**Sistema de Programación Radial**

  
 Usuarios  
3

  
 Guiones  
2

  
 Programas de Radio  
4

  
 Publicidades Activas  
3

**Reporte de Usuarios**


PDF
Excel


Id	Nombre	Apellido	Cédula	Contraseña	Tipo de Usuario
11	Coordinador	Lopez	1002345678	123	Coordinador
22	Locutor	Romero	1234567890	123	Locutor
8	Productor	Perez	1231234509	123	Productor


**Reporte General de Programas de Radio.**


- Para hacer los reportes generales de los programas de radios que están registrados en el sistema debemos hacer clic en donde dice programas y se nos va a abrir el listado de los programas de la radio que están en el sistema.

**Sistema de Programación Radial**

  
 Usuarios  
3

  
 Guiones  
2

  
 Programas de Radio  
4

  
 Publicidades Activas  
3


**Reporte de Programas**


PDF
Excel


Id	Nombre	Descripción	Día	Hora	Duración	Clasificación	Tipo	Locutor
2	Programa informativo2	Es un programa informativo	Lunes-Viernes	16:00	2:00:00	B	Deportivo	Productor
9	prueba	es la prueba	lunes - viernes	01:00	01:00:00	B	Deportivo	Locutor
12	ciudad al dia	todo lo de la ciudad	lunes - viernes	11:00 - 13:00	00:05:00	A	Entretenimiento	Coordinador
13	ciudad al dia	para todo riobamba	lunes - viernes	16:00 - 17:00	00:10:00	pueba	Entretenimiento	Productor


- Si el coordinador desea puede utilizar los filtros de búsqueda si se desea una búsqueda personalizada, caso contrario damos clic en los iconos de pdf o de Excel según se desee y así se nos genera el archivo con la información de los programas.

**Sistema de Programación Radial**

  
 Usuarios  
3

  
 Guiones  
2

  
 Programas de Radio  
4

  
 Publicidades Activas  
3

**Reporte de Programas**

PDF
Excel

Id	Nombre	Descripción	Día	Hora	Duración	Clasificación	Tipo	Locutor
2	Programa informativo2	Es un programa informativo	Lunes-Viernes	16:00	2:00:00	B	Deportivo	Productor
9	prueba	es la prueba	lunes - viernes	01:00	01:00:00	B	Deportivo	Locutor
12	ciudad al dia	todo lo de la ciudad	lunes - viernes	11:00 - 13:00	00:05:00	A	Entretenimiento	Coordinador
13	ciudad al dia	para todo riobamba	lunes - viernes	16:00 - 17:00	00:10:00	pueba	Entretenimiento	Productor

**Reporte General de Guiones.**

- Para hacer los reportes generales de los guiones que están registrados en el sistema debemos hacer clic en donde dice guiones y se nos va a abrir el listado de los guiones que están registrados en el sistema.

Id	Fecha Registro	Fecha Aprobación	Diálogo	Estado	Observación	Programa	Tipo Guión
1	08/04/2021	08/04/2021	Orden_Pago_No_31 (2).pdf	Registrado	PROBADO	Programa informativo2	Abierto
21	02/27/2022	02/27/2022	logo_ciudad.png	Rechazado	no tiene sentido	ciudad al dia	Abierto

- Si el coordinador desea puede utilizar los filtros de búsqueda si se desea una búsqueda personalizada, caso contrario damos clic en los iconos de pdf o de Excel según se desee y así se nos genera el archivo con la información de los guiones que están registrados en el sistema.

### **Reporte General de la Publicidad.**

- Para hacer los reportes generales de la publicidad que están registrados en el sistema debemos hacer clic en donde dice Publicidad y se nos va a abrir el listado de las publicidades que están registradas en el sistema.

Id	Nombre	Descripción	Fecha Inicio	Fecha Fin	Estado	Horario	Programa Radial
26	Anuncios	todos los del municipio	02/12/2022	02/13/2022	Terminado	lunes - sabado 13:00 - 16:00	ciudad al dia
28	sddd	sss	02/04/2022	02/04/2022	Activo	lunes - domingo 13:00	ciudad al dia
29	ventas	todas las ventas	02/01/2022	02/04/2022	Activo	lunes - viernes 06:00 - 07:00 - 10:00	Programa informativo2

- Si el coordinador desea puede utilizar los filtros de búsqueda si se desea una búsqueda personalizada, caso contrario damos clic en los iconos de pdf o de Excel según se desee y así se nos genera el archivo con la información de las publicidades que están en el sistema.