



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“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”**

**TRABAJO DE TITULACIÓN**

**TIPO: PROYECTO TÉCNICO**

Presentado para optar por el grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR: JHONATHAN PAÚL CHELA LLUMIGUANO**

**TUTOR: Dr. JULIO ROBERTO SANTILLÁN CASTILLO**

Riobamba – Ecuador

2019

©2019, JHONATHAN PAÚL CHELA LLUMIGUANO

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

El Tribunal del trabajo de titulación certifica que: El trabajo de investigación: Tipo Proyecto Técnico, “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”, de responsabilidad del señor: Jhonathan Paúl Chela Llumiguano, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

**FIRMA**

**FECHA**

Ing. Washington Luna Encalada.

**DECANO DE LA FACULTAD DE  
INFORMÁTICA Y  
ELECTRÓNICA**

Ing. Patricio Moreno Costales.

**DIRECTOR DE LA ESCUELA  
DE INGENIERÍA EN SISTEMAS**

Dr. Julio Santillán Castillo.

**DIRECTOR DE TRABAJO DE  
TITULACIÓN**

Ing. Raúl Rosero Miranda.

**MIEMBRO DEL TRIBUNAL**

Yo, Jhonathan Paúl Chela Llumiguano soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

---

JHONATHAN PAÚL CHELA LLUMIGUANO

## **DEDICATORIA**

El presente trabajo de titulación es dedicado en primer lugar a Dios, quien con su bendición me acompañó durante todo el camino de mis estudios universitarios, también dedico este trabajo a mis padres quienes, con su apoyo incondicional, amor y confianza permitieron culminar mi carrera profesional.

*Jhonathan*

## AGRADECIMIENTO

Soli Deo gloria.

Agradezco en primer lugar a Dios por haberme acompañado con su gracia infinita en el transcurso de este caminar, también quiero expresar mi agradecimiento eterno para mis padres, César y Eva, quienes con su ejemplo de trabajo y honradez me supieron sostener y animar.

De igual manera mi gratitud para la Escuela Superior Politécnica de Chimborazo, a toda la Facultad de Informática y Electrónica en especial a mis profesores de la Escuela de Ingeniería en Sistemas, a mi tutor Dr. Julio Santillán, quienes con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

*Jhonathan*

## ÍNDICE DE ABREVIATURAS

<b>FM</b>	Frecuencia Modulada
<b>SIGUION</b>	Sistema Informático de Gestión de Guion
<b>IEEE</b>	Instituto de Ingeniería Eléctrica y Electrónica
<b>DBMS</b>	Sistema de Administración de Base de Datos
<b>SQL</b>	Structured Query Language
<b>OS</b>	Sistema Operativo
<b>TCP</b>	Protocolo de Control de Transmisión
<b>IP</b>	Protocolo de Internet
<b>LAN</b>	Red de Área Local
<b>ODBC</b>	Open Database Connectivity
<b>JDBC</b>	Java Database Connectivity
<b>SSL</b>	Secure Sockets Layer
<b>BSD</b>	Berkeley Software Distribution
<b>API</b>	Application Programming Interface
<b>MVCC</b>	Multiversion Concurrency Control
<b>JVM</b>	Java Virtual Machine, JVM
<b>JRE</b>	Java Runtime Environment
<b>JIT</b>	Just in Time
<b>JSF</b>	Java Server Face
<b>MVC</b>	Modelo Vista Controlador
<b>XHTML</b>	eXtensible HyperText Markup Language
<b>HTML</b>	HyperText Markup Language
<b>CSS</b>	Hojas de Estilo en Cascada

<b>V&amp;V</b>	Verificación y Validación
<b>ISO</b>	International Organization for Standardization
<b>IEC</b>	Comisión Electrotécnica Internacional
<b>XML-RPC</b>	eXtensible Markup Language -Remote Procedure Call
<b>UML</b>	Unified Modeling Language
<b>IDE</b>	Integrated Development Environment
<b>ISTQB</b>	International Software Testing Qualifications Board



## TABLA DE CONTENIDOS

<b>RESUMEN .....</b>	<b>xviii</b>
<b>ABSTRACT .....</b>	<b>xix</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
 <b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO REFERENCIAL.....</b>	<b>7</b>
<b>1.1. Tecnología de desarrollo .....</b>	<b>7</b>
<b>1.1.1. PostgreSQL.....</b>	<b>7</b>
1.1.1.1. Definición .....	7
1.1.1.2. Arquitectura.....	8
1.1.1.3. Características .....	9
1.1.1.4. Características generales .....	9
1.1.1.5. Características de programación.....	9
1.1.1.6. Características de secuencias SQL .....	9
1.1.1.7. Ventajas .....	10
<b>1.1.2. Lenguaje de programación Java.....</b>	<b>10</b>
1.1.2.1. Origen.....	10
1.1.2.2. Definición .....	11
1.1.2.3. Características .....	11
<b>1.1.3. Java Server Face .....</b>	<b>12</b>
1.1.3.1. Características de JSF.....	12
1.1.3.2. Funcionamiento .....	12
1.1.3.3. Elementos de JSF .....	13
1.1.3.4. Ventajas de usar de JSF .....	14
<b>1.1.4. Bootstrap .....</b>	<b>14</b>

1.1.4.1.	<i>Estructura de un archivo Bootstrap</i> .....	15
1.1.4.2.	<i>Ventajas de usar Bootstrap</i> .....	15
<b>1.2.</b>	<b>Metodología SCRUM</b> .....	<b>16</b>
1.2.1.	<i>Definición</i> .....	16
1.2.2.	<i>El equipo SCRUM</i> .....	16
1.2.3.	<i>Artefactos de SCRUM</i> .....	17
1.2.3.1.	<i>Lista de Producto (Product Backlog)</i> .....	17
1.2.3.2.	<i>Lista de Pendientes del Sprint (Sprint Backlog)</i> .....	17
1.2.3.3.	<i>Incremento</i> .....	17
1.2.4.	<i>Eventos SCRUM</i> .....	17
1.2.5.	<i>Burndwon chart</i> .....	18
<b>1.3.</b>	<b>Modelo V&amp;V de desarrollo de software</b> .....	<b>19</b>
1.3.1.	<i>Verificación</i> .....	20
1.3.2.	<i>Validación</i> .....	20
1.3.3.	<i>Definición del modelo V&amp;V</i> .....	20
<b>1.4.</b>	<b>Pruebas de software</b> .....	<b>22</b>
1.4.1.	<i>Enfoques de pruebas de software</i> .....	23
<b>1.5.</b>	<b>Pruebas de caja negra</b> .....	<b>23</b>
1.5.1.1.	<i>Definición</i> .....	23
1.5.1.2.	<i>Niveles de la aplicación de las pruebas de caja negra</i> .....	24
1.5.1.3.	<i>Tipos de pruebas de caja negra</i> .....	24
1.5.1.4.	<i>Ventajas de las pruebas de caja negra</i> .....	24
<b>1.6.</b>	<b>Pruebas de caja blanca</b> .....	<b>25</b>
1.6.1.1.	<i>Definición</i> .....	25
1.6.1.2.	<i>Niveles de aplicación de las pruebas de caja blanca</i> .....	25
1.6.1.3.	<i>Razones por las que las pruebas unitarias son desarrolladas</i> .....	26
1.6.1.4.	<i>Ventajas de las pruebas de caja blanca</i> .....	26
<b>1.7.</b>	<b>Pruebas unitarias</b> .....	<b>26</b>
1.7.1.1.	<i>Definición</i> .....	27

1.7.1.2.	<i>Características de las pruebas unitarias</i> .....	27
1.7.1.3.	<i>Beneficios de las pruebas unitarias</i> .....	27
<b>1.8.</b>	<b>Cobertura de código</b> .....	<b>28</b>
<b>1.9.</b>	<b>Herramientas para automatizar las pruebas unitarias</b> .....	<b>29</b>
1.9.1.	<i>JUnit</i> .....	29
1.9.2.	<i>Testlink</i> .....	30
1.9.2.1.	<i>Terminología básica</i> .....	30
1.9.2.2.	<i>Ventajas de Testlink</i> .....	31

## **CAPÍTULO II**

<b>2.</b>	<b>MARCO METODOLÓGICO</b> .....	<b>32</b>
<b>2.1.</b>	<b>Tipo de investigación</b> .....	<b>32</b>
<b>2.2.</b>	<b>Métodos y técnicas</b> .....	<b>33</b>
2.2.1.	<i>Métodos aplicados en el trabajo de titulación</i> .....	33
2.2.2.	<i>Técnicas</i> .....	34
<b>2.3.</b>	<b>Fase de planificación del sistema web de gestión de guiones SIGUION</b> .....	<b>34</b>
2.3.1.	<i>Preparación del proyecto</i> .....	34
2.3.2.	<i>Reunión de planificación</i> .....	35
2.3.3.	<i>Equipo SCRUM del proyecto</i> .....	36
2.3.4.	<i>Tipos y roles del proyecto</i> .....	37
2.3.5.	<i>Lista del producto (Product Backlog)</i> .....	37
2.3.6.	<i>Gestión de riesgos</i> .....	41
2.3.6.1.	<i>Categorización de riesgos</i> .....	43
2.3.6.2.	<i>Priorización de riesgos</i> .....	44
2.3.6.3.	<i>Hojas de gestión de riesgo</i> .....	45
<b>2.4.</b>	<b>Fase de desarrollo del sistema web de gestión de guiones SIGUION</b> .....	<b>46</b>
2.4.1.	<i>Estándar de codificación</i> .....	46
2.4.2.	<i>Arquitectura del sistema</i> .....	47
2.4.3.	<i>Diseño de la interfaz de usuario</i> .....	48

2.4.4.	<i>Diseño de la base de datos</i> .....	53
2.4.5.	<i>Diccionario de datos</i> .....	54
2.4.6.	<i>Diagrama de casos de uso</i> .....	55
2.4.6.1.	<i>Procesos del administrador del sistema</i> .....	55
2.4.6.2.	<i>Procesos del director de la radio</i> .....	56
2.4.6.3.	<i>Procesos del locutor de la radio</i> .....	57
2.4.7.	<i>Historias de usuario</i> .....	58
2.4.8.	<i>Historias técnicas</i> .....	59
2.4.9.	<i>Tareas de ingeniería</i> .....	61
2.4.10.	<i>Casos de pruebas unitarias por funcionalidad del sistema</i> .....	61
2.4.11.	<i>Sprint del proyecto</i> .....	63
2.4.11.1.	<i>Sprint 1</i> .....	64
2.4.11.2.	<i>Sprint 2</i> .....	64
2.4.11.3.	<i>Sprint 3</i> .....	66
2.4.11.4.	<i>Sprint 4</i> .....	66
2.4.11.5.	<i>Sprint 5</i> .....	68
2.5.	<b>Fase de cierre e implementación</b> .....	<b>69</b>

### **CAPÍTULO III**

3.	<b>MARCO DE RESULTADOS</b> .....	<b>70</b>
3.1.	<b>Gestión del Proyecto</b> .....	<b>70</b>
3.2.	<b>Evaluación de la exactitud del software</b> .....	<b>71</b>
3.2.1.	<i>Cobertura del código</i> .....	<i>71</i>
3.2.2.	<i>Cobertura de código con JaCoCoverage</i> .....	<i>72</i>
3.2.3.	<i>Resultados de la cobertura de código</i> .....	<i>72</i>
3.2.4.	<i>Criterio de evaluación de la exactitud</i> .....	<i>74</i>
3.3.	<b>Análisis del beneficio del software con respecto al proceso manual</b> .....	<b>75</b>
3.3.1.	<i>Proceso de gestión de guiones</i> .....	<i>75</i>
3.3.2.	<i>Muestra</i> .....	<i>76</i>

3.3.3.	<i>Contraste de normalidad</i> .....	77
3.3.4.	<i>Análisis de datos</i> .....	77
	<b>CONCLUSIONES</b> .....	<b>81</b>
	<b>RECOMENDACIONES</b> .....	<b>82</b>
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b> Características de Java.....	11
<b>Tabla 2-1:</b> Miembros el equipo SCRUM.....	16
<b>Tabla 3-1:</b> Eventos de SCRUM .....	18
<b>Tabla 4-1:</b> Tipos de métricas de cobertura de código .....	28
<b>Tabla 5-1:</b> Terminología básica TestLink.....	30
<b>Tabla 1-2:</b> Reuniones de planificación.....	35
<b>Tabla 2-2:</b> Equipo SCRUM del proyecto SIGUION .....	36
<b>Tabla 3-2:</b> Tipos de usuario y su rol .....	37
<b>Tabla 4-2:</b> Talla de la camiseta del proyecto .....	38
<b>Tabla 5-2:</b> Lista del producto del proyecto .....	38
<b>Tabla 6-2:</b> Gestión de Riesgos de SIGUION.....	41
<b>Tabla 7-2:</b> Parámetros de probabilidad .....	43
<b>Tabla 8-2:</b> Parámetros de impacto .....	43
<b>Tabla 9-2:</b> Rangos de exposición del riesgo .....	44
<b>Tabla 10-2:</b> Priorización de Riesgos .....	44
<b>Tabla 11-2:</b> Hoja de gestión de riesgo .....	45
<b>Tabla 12-2:</b> Estándar de codificación del proyecto.....	47
<b>Tabla 13-2:</b> Simbología UML de SIGUION.....	55
<b>Tabla 14-2:</b> Historia de usuario de SIGUION .....	58
<b>Tabla 15-2:</b> Historia técnica de SIGUION.....	59
<b>Tabla 16-2:</b> Tabla de registro de prueba unitaria .....	60
<b>Tabla 17-2:</b> Tabla de tareas de ingeniería de HU-05 .....	61
<b>Tabla 18-2:</b> Casos de pruebas unitarias por funcionalidad de SIGUION .....	62
<b>Tabla 19-2:</b> Sprint Backlog de SIGUION.....	63
<b>Tabla 20-2:</b> Sprint 1 del Proyecto SIGUION.....	64
<b>Tabla 21-2:</b> Sprint 2 del Proyecto SIGUION.....	65
<b>Tabla 22-2:</b> Sprint 3 del Proyecto SIGUION.....	66
<b>Tabla 23-2:</b> Sprint 4 del Proyecto SIGUION.....	67
<b>Tabla 24-2:</b> Sprint 5 del Proyecto SIGUION.....	68
<b>Tabla 1-3:</b> Resumen de la cobertura de código por requerimientos.....	73
<b>Tabla 2-3:</b> Requisitos de calidad.....	75
<b>Tabla 3-1:</b> Registro de tiempo en la gestión de guiones. ....	76
<b>Tabla 4-3:</b> Diferencia de tiempos promedios entre el sistema y el proceso manual. ....	79

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b> Arquitectura de PostgreSQL.....	8
<b>Figura 2-1:</b> Vista en JSF .....	13
<b>Figura 3-1:</b> Estructura de Bootstrap.....	15
<b>Figura 4-1:</b> Modelo V&V .....	21
<b>Figura 1-2:</b> Diagrama Gantt del Proyecto.....	35
<b>Figura 2-2:</b> Arquitectura de SIGUION .....	48
<b>Figura 3-2:</b> Pantalla de inicio de sesión de SIGUION .....	49
<b>Figura 4-2:</b> Pantalla de gestión del administrador .....	49
<b>Figura 5-2:</b> Pantalla de gestión del director .....	50
<b>Figura 6-2:</b> Pantalla de gestión del locutor .....	50
<b>Figura 7-2:</b> Pantalla de revisión del guion por el locutor.....	51
<b>Figura 8-2:</b> Reporte pdf de un guion aprobado.....	51
<b>Figura 9-2:</b> Panel de Reportes del sistema.....	52
<b>Figura 10-2:</b> Reporte General de guiones Aprobados.....	52
<b>Figura 11-2:</b> Base de datos de SIGUION .....	53
<b>Figura 12-2:</b> Diccionario de la tabla tUsuario.....	54
<b>Figura 13-2:</b> Diccionario de la tabla tClasificación_Programa_Radial .....	54
<b>Figura 14-2:</b> Diagrama del administrador del sistema.....	56
<b>Figura 15-2:</b> Diagrama del director de la entidad .....	57
<b>Figura 16-2:</b> Diagrama del locutor del sistema.....	58
<b>Figura 1-3:</b> Estructura de una prueba unitaria.....	71
<b>Figura 2-3:</b> Cobertura con JaCoCoverage.....	72

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-1:</b> Ejemplo de Burndown chart.....	19
<b>Gráfico 1-3:</b> Burndown chart del proyecto .....	70
<b>Gráfico 2-3:</b> Gráfico de cobertura de código .....	74
<b>Gráfico 3-3:</b> Contraste Shapiro-Wilk de los datos del proyecto SIGUION .....	77



## ÍNDICE DE ANEXOS

**Anexo A:** Historias de Usuario del proyecto SIGUION

**Anexo B:** Gestión de Riesgos del proyecto SIGUION

**Anexo C:** Manual de Usuario

**Anexo D:** Diccionario de datos

**Anexo E:** Historias Técnicas del proyecto SIGUION

**Anexo F:** Planificación de pruebas unitarias por requerimientos del sistema SIGUION

**Anexo G:** Contraste de normalidad Shapiro-Wilks

## **RESUMEN**

El presente trabajo de titulación tuvo como objetivo desarrollar un sistema web utilizando pruebas unitarias en la Radio Crisólito Celestial para brindar una herramienta que automatiza el proceso manual de gestión de guiones de programas de radio que se transmiten en la entidad. Se utilizó el lenguaje de programación Java, la tecnología para sistemas web Java Server Face, el framework de diseño de interfaces responsivas Bootstrap 4 y la base de datos del proyecto fue implementada con PostgreSQL. Siguiendo el ciclo de desarrollo de vida del software en V, junto con la metodología de desarrollo ágil SCRUM, se planificó la ejecución de los requerimientos obtenidos del cliente, de esta manera, con una participación activa del director y todas las personas de la entidad involucradas en la construcción del sistema se consiguió un producto acorde a las necesidades especificadas. Las pruebas unitarias fueron implementadas con JUnit para validar el diseño de los módulos del software garantizando un alto porcentaje en el correcto funcionamiento de las unidades codificadas. Además, se evaluó la exactitud de los casos de prueba mediante la cobertura de código determinando que existe un porcentaje del 98 % con un valor de 0.98 validando que el software está muy por encima del valor de aceptación. Se analizó también el beneficio que el sistema web con respecto al proceso manual tiene, y se determinó una reducción del 53 % de tiempo manual en el proceso de gestión de guiones.

**PALABRAS CLAVE:** <INGENIERÍA DE SOFTWARE>, <GESTIÓN DE GUIONES>, <DESARROLLO DE APLICACIONES WEB>, <MODELO V>, <PRUEBAS UNITARIAS>, <COBERTURA DE CÓDIGO>, <JAVA SERVER FACES (FRAMEWORK)>, <SCRUM (METODOLOGÍA DE DESARROLLO ÁGIL)>.

## **ABSTRACT**

The present degree work has as an aim to develop a web system using unit tests in Radio Crisólito Celestial to provide a tool that automates the manual process of script management of radio programs that is transmitted in the entity. The Java programming language was used, the technology for Java Server Face web systems, the responsive interface design framework Bootstrap 4 and the project database were implemented with PostgreSQL. Following the development cycle of the life of the software in V, together with the development methodology agile SCRUM, the execution of the requirements obtained from the client was planned, in this way, with an active participation of the direct and all the people of the entity involved in the construction of the system was achieved a product according to specific needs. The unit tests were implemented with JUnit to validate the design of the software modules guaranteeing a high percentage in the correct operation of the coded units. In addition, the accuracy of the test cases is evaluated through the code coverage determined that there is a percentage of 98% with a value of 0.98 validating that the software is well above the acceptance value. The benefit that the web system has with respect to the manual process was also analyzed, and a 53% reduction of manual time in the script management process was determined.

**KEYWORDS:** <SOFTWARE ENGINEERING>, <SCRIPT MANAGEMENT>, <WEB APPLICATIONS DEVELOPMENT>, <MODEL V>, <UNIT TESTS>, <CODE COVERAGE>, <JAVA SERVER FACES (FRAMEWORK)>, <SCRUM (AGILE DEVELOPMENT METHODOLOGY)>

## INTRODUCCIÓN

Hoy en día, el avance tecnológico ha permitido que la demanda del desarrollo de software se eleve junto con su complejidad, esto ha hecho que en la mayoría de ocasiones se presente un producto con errores específicos y no descubiertos en las unidades que componen el mismo, por lo tanto, es de vital importancia aplicar un desarrollo basado en pruebas unitarias.

El mundo de las pruebas que se pueden aplicar en los sistemas informáticos se divide en niveles. Este proyecto se centra en el nivel de pruebas unitarias automatizadas que tiene como finalidad comprobar la exactitud de las unidades que componen el todo del software para asegurar la funcionalidad interna del producto.

Por tanto, este documento se encuentra dirigido al desarrollo de un sistema de gestión de guiones de radio basado en pruebas unitarias automatizadas mediante la metodología de desarrollo ágil SCRUM, el cual tiene como objetivo probar las unidades de código del software previo al proceso de integración para la detección de errores, con la finalidad de asegurar en un alto porcentaje la exactitud en la funcionalidad del producto.

El presente documento se encuentra dividido en tres capítulos: El capítulo I se basa en el marco referencial de los temas y herramientas que se usan en este proyecto. El capítulo II se centra en la creación, diseño y desarrollo del sistema de gestión de guiones de radio con pruebas unitarias. Y en el capítulo III se comprueba y se analiza los resultados obtenidos dentro de un entorno de desarrollo basado en casos de pruebas unitarias automatizadas.

## **Antecedentes**

El proceso de desarrollo de software incluye siempre la aplicación de pruebas al producto alcanzado en cada iteración, pero hay que tomar en cuenta el tiempo que involucra el desarrollo y presentación del producto funcional con el porcentaje más alto de eficiencia. Al no verificar el funcionamiento de cada componente o unidad del código que se desarrolla, la etapa de pruebas toma mayor tiempo del planificado para identificar y encontrar el lugar exacto en el cual el código no devuelve el resultado para el cual fue construido. Esto, adiciona también que en las pruebas de funcionalidad a las que el software es sometido, el mismo falle, incrementando la cantidad de unidades del software para la corrección y mantenimiento.

Al no existir un control de unidades de código previo al proceso de integración del software, las pruebas unitarias son la mejor opción a considerar, esto permitiría asegurar el correcto funcionamiento de las unidades del código y también el funcionamiento del software luego del proceso de integración.

Actualmente el proceso de presentación, revisión, análisis y aprobación de los guiones en la radio Crisólito Celestial 93.1 FM, se lo realiza de manera manual, en documentos físicos por parte de cada representante de cada programación.

Al ser una radio estación comunitaria involucra la participación de varias personas y comunicadores comunitarios, los mismos que en el horario en el que no trabajan en la radio estación, se dedican a realizar diferentes labores dentro y fuera de la ciudad y provincia, por lo que en varias ocasiones se ha presentado la dificultad de acudir a las oficinas centrales para presentar los guiones de cada programación.

Esto ha causado el retraso en la aprobación de los guiones de cada programación provocando que los programas salgan al aire sin revisión y aprobación previa por parte del director. Además, el director en su informe mensual de programaciones de la radio no ha podido incluir varios de los guiones que no se han presentado a tiempo.

El desarrollo del sistema web de gestión de guiones de radio denominado “SIGUIÓN”, permite realizar el proceso que involucran la aprobación de guiones en la entidad mencionada, además incluye procesos de reporte que responden a requerimientos establecidos.

## **Formulación del problema**

¿Cómo contribuirá la aplicación de pruebas unitarias en el desarrollo del sistema web para la gestión de guiones en la Radio Crisólito Celestial?

## **Sistematización del problema**

¿Cómo se realiza en la actualidad la gestión de los guiones de cada programación radial en la Radio Crisólito Celestial?

¿Cómo el sistema web permitirá la gestión y el control de los guiones de cada programación radial en la Radio Crisólito Celestial?

¿De qué manera la aplicación de pruebas unitarias automatizadas permitirá verificar el funcionamiento del código del sistema web de gestión de guiones?

¿De qué manera el sistema web mejoraría el proceso de gestión de los guiones de cada programación radial en la Radio Crisólito Celestial?

## **Justificación del trabajo de titulación**

En este apartado se explicarán los fundamentos conceptuales que apoyan al presente desarrollo y la orientación que tomará.

### ***Justificación teórica***

El desarrollo de software basado en pruebas unitarias es el camino que se toma para comprobar las unidades del software que componen el todo del mismo. Este tipo de pruebas en esencia son trozos de código diseñados para validar que una parte específica del sistema desarrollado devuelva los valores exactos para los cuales fue creado.

Además, es importante añadir que para que una prueba sea considerada unitaria debe tener las siguientes características: automatizable, completa, repetible, e independiente.

Este tipo de pruebas se las realiza sobre el código fuente de un software, regularmente es aplicada sobre unidades que componen el todo de un sistema informático. De una manera general y directa sobre las pruebas unitarias se puede decir que se reciben datos de entrada que son pasadas al

proceso de la unidad de código implementada para esperar resultados, una vez obtenido el resultado se puede emitir el criterio de evaluación y verificar si la unidad de código devuelve el valor para el que ha sido creada.

La automatización de pruebas es uno de los aspectos muy importantes a considerar cuando se habla de pruebas unitarias. Para lograr este objetivo se hace uso de software especial separado del producto en estudio, esto para controlar la ejecución de las pruebas implementadas y además realizar la comparación entre los resultados obtenidos y esperados.

En este proyecto para la implementación de pruebas unitarias se utilizará el framework JUnit que permite la ejecución de unidades de código de manera controlada para comprobar el correcto funcionamiento.

Con respecto a la gestión de los casos de pruebas unitarias automatizadas, se utilizará TestLink, una herramienta gratuita que permite organizar planes de prueba. Estos planes permiten a los miembros del equipo ejecutar casos de test y registrar los resultados dinámicamente, generar informes, mantener la trazabilidad con los requerimientos, así como priorizar y asignar tareas.

### ***Justificación aplicativa***

Al ser necesario la revisión y la aprobación de los guiones de las programaciones radiales en la “RADIO CRISÓLITO CELESTIAL 93.1 FM”, se determina que los procesos son llevados manualmente por lo que el proceso de gestión demanda de tiempo y retraso, además es importante notar que esta entidad de radio es una radio – comunitaria, por lo que existen variedad de programas y personas responsables que en el tiempo en el cual no están laborando en la radio se encuentran cumpliendo con sus trabajos dentro y fuera de la ciudad. Por lo tanto, para facilitar la revisión y aprobación de los guiones es necesario desarrollar una aplicación web para agilizar el proceso de gestión alcanzando eficiencia en el manejo de la información, además garantizando la exactitud de respuesta del sistema.

La “RADIO CRISÓLITO CELESTIAL 93.1 FM”, está consciente que el uso de las Tecnologías de la Información y Comunicación son una gran ayuda para realizar su actividad de gestión de guiones, 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 radiales.
- Módulo de revisión de guiones.
- Módulo de reportes.

Este sistema constará de una base de datos, desarrollada con el gestor de base de datos PostgreSQL y almacenada en un servidor de base de datos con conexión a internet, permitiendo gestionar la información con respecto al proceso que se implementará en el sistema web. El servidor de aplicaciones alojará la aplicación desarrollada en el lenguaje Java, utilizando la tecnología Java Server Face junto con el patrón de diseño modelo vista controlador desarrollada bajo la orientación de pruebas unitarias. Además, la interfaz de usuario debe ser web responsiva, por lo que se usará el framework Bootstrap. Permitiendo al usuario la interacción con el sistema desde cualquier dispositivo, en cualquier parte del mundo y en cualquier hora del día.



## **Objetivos**

### **Objetivo general**

Desarrollar un sistema web de gestión de guiones para la Radio Crisólito Celestial utilizando el desarrollo de software orientado a pruebas unitarias con SCRUM, Java Server Face, Bootstrap y PostgreSQL.

### **Objetivos específicos**

- Analizar la aplicación de pruebas unitarias mediante la recopilación de información relevante para su posterior uso en el desarrollo del sistema web.
- Diseñar un sistema web con la metodología SCRUM y pruebas unitarias para la gestión de guiones de la radio Crisólito Celestial.
- Implementar el producto software con las herramientas informáticas PostgreSQL, Java Server Face y Bootstrap para la automatización del proceso de gestión de guiones de radio.
- Evaluar la exactitud del software mediante casos de prueba automatizados con JUnit y TestLink para asegurar la funcionalidad del sistema web.

# CAPÍTULO I

## 1. MARCO TEÓRICO REFERENCIAL

El presente capítulo establece el marco teórico referencial de este trabajo de titulación permitiendo que el lector tenga una idea clara acerca del tema. Se exponen los conceptos, definiciones, características y ventajas, básicos y específicos complementarios de este estudio.

En primer lugar, se definen las tecnologías de desarrollo para la consecución de un sistema web robusto y que cumpla con los requisitos establecidos; se expone el modelo de verificación y validación para aclarar la aplicación de las pruebas unitarias, además, se cita el sustento teórico adecuado sobre este nivel de pruebas objeto de este proyecto. Finalmente se describen las herramientas que permiten la implementación de los casos de test para el análisis de la exactitud del sistema construido.

### 1.1. Tecnología de desarrollo

En este apartado se describe la tecnología de diseño e implementación de la base de datos, así como también el lenguaje de programación, la tecnología de sistemas web junto con el framework de diseño web responsiva. En los siguientes ítems se define cada punto antes mencionado.

#### *1.1.1. PostgreSQL*

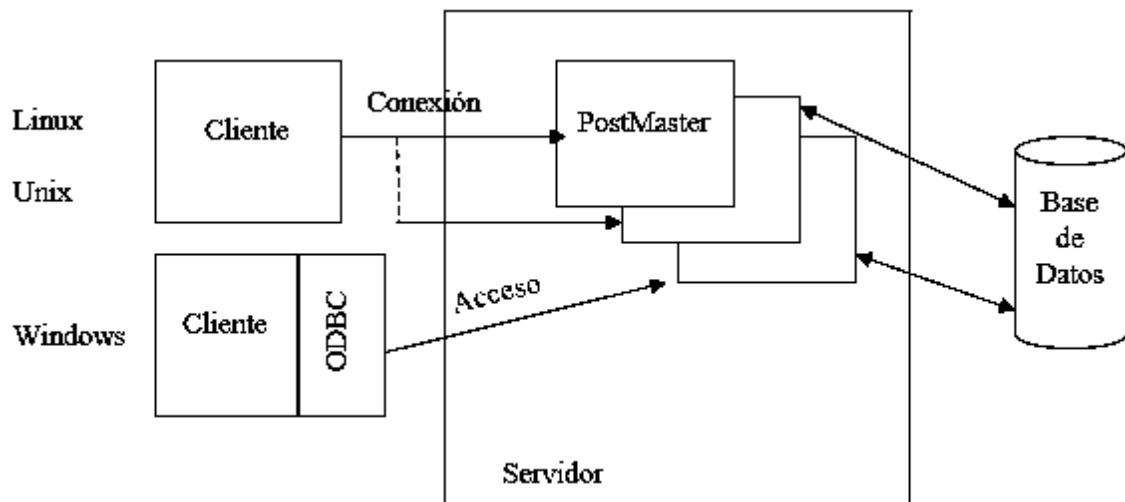
##### *1.1.1.1. Definición*

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)

### 1.1.1.2. 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):



**Figura 1-1:** Arquitectura de PostgreSQL

Fuente: (Niel & Stones, 2005, p. 14)

Realizado por: Chela Jhonathan, 2018.

En la **Figura 1-1** se puede apreciar varios clientes que se conectan al servidor mediante una red, en PostgreSQL, esto necesita ser una red TCP/IP, una red de área local (LAN) o posiblemente el Internet. Cada cliente se conecta al proceso principal del servidor de la base de datos que se denomina PostMaster en la **Figura 1-1**, este crea un nuevo proceso específicamente para las peticiones de servicio de acceso para el cliente.

De esta manera se concentra el manejo de datos en un servidor, en lugar de intentar controlar a muchos clientes que acceden a los mismos datos almacenados en un directorio compartido o un servidor. Esto permite que PostgreSQL, eficientemente mantenga la integridad de datos aun con muchos usuarios simultáneos.

Los programas clientes usan un mensaje específico de protocolo para PostgreSQL, es posible también instalar un software en el cliente que provee una interfaz estándar para que la aplicación trabaje, como el estándar Open Database Connectivity (ODBC) o el estándar Java Database Connectivity (JDBC) 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 PostgreSQL permite una división de labor. Una máquina de servidor bien adaptada para el almacenamiento y acceso a grandes cantidades de datos puede ser usado como un repositorio de datos seguro. Aplicaciones gráficas sofisticadas pueden ser desarrolladas por los clientes, así como también un web – based front – end, puede ser creado para acceder a los datos y devolver resultados a páginas web como un buscador web estándar. (Niel & Stones, 2005, p. 14)

#### *1.1.1.3. Características*

PostgreSQL tiene características muy importantes y que hacen que este DBMS sea único en relación a otros DBMS, por lo que se presenta las características generales, sus características propias de programación y sus características de secuencias SQL.

#### *1.1.1.4. Características generales*

A continuación, se expone las características generales que PostgreSQL tiene:

- Integridad
- Unicode
- Varios métodos de autenticación al sistema
- Acceso encriptado vía SSL
- Licencia BSD

#### *1.1.1.5. Características de programación*

Se puede mencionar tres rasgos de mayor importancia con respecto al desarrollo, que este sistema de gestión de base de datos tiene (Martinez, 2009):

- Se puede incorporar funciones y procedimientos almacenados en diferentes lenguajes de programación.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido).
- Se puede utilizar diferentes interfaces de programación de aplicaciones (API) como, por ejemplo, C, C++, Java, Perl, Net, Phyton, Ruby, entre otros.

#### *1.1.1.6. Características de secuencias SQL*

Las características que PostgreSQL presenta con respecto a las secuencias que se implementan con el lenguaje estructurado de consulta son las siguientes:

- Claves primarias (primary keys) y claves foráneas (foreing key).

- Columnas autoincrementales automáticamente.
- Sub-selects para consultas de datos.
- Consultas recursivas, joins, triggers y la herencia de tablas.

#### *1.1.1.7. Ventajas*

De acuerdo con (POSTGRESQL, 2018), se pueden citar las ventajas que se presentan a continuación con relación a los demás sistemas de gestión de base de datos:

- PostgreSQL implementa un control de concurrencias multiversión (MVCC), Oracle es conocida esta característica como, snapshot isolation.
- Este DBMS permite agregar funciones personalizadas desarrolladas usando diferentes lenguajes de programación como C/C++, Java, etc.
- PostgreSQL es designado para ser extensible, permitiendo definir nuestros propios tipos de datos, tipos de index, lenguajes funcionales entre otros.
- En caso de consulta para resolver distintos problemas o dudas con este gestor de bases de datos, la comunidad de PostgreSQL es muy extensa y ayuda resolviendo muchas interrogantes, también, existen empresas que brindan soporte comercial en caso de ser necesario.

Luego de los puntos abordados con respecto al sistema gestor de base de datos, se define también el lenguaje de alto nivel que se utiliza para la programación de los módulos que componen un sistema web, esto es el estudio de Java.

### **1.1.2. Lenguaje de programación Java**

#### *1.1.2.1. Origen*

El lenguaje de programación Java fue creado como respuesta a la necesidad de un lenguaje capaz de ser compatible con las diferentes arquitecturas físicas; 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 entre los diferentes dispositivos. (Groussard, 2014, pp. 11-12)

Bill Joy, ingeniero de Sun Microsystems, y su equipo de investigadores trabajaban en el “Proyecto Green” que consistía en desarrollar aplicaciones a una amplia variedad de periféricos y sistemas embebidos. Este equipo desarrollo el lenguaje multiplataforma que en principio se conocía como C++-(C++ sin sus defectos), finalmente lo conocieron como Java, este lenguaje nace desde 1991. (Groussard, 2014, pp. 11-12)

### 1.1.2.2. Definición

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

### 1.1.2.3. Características

El lenguaje Java presenta algunas características que lo hacen único, esto ha permitido que su popularidad se mantenga aún en estos tiempos de auge tecnológico. A continuación, se exponen sus características en la **Tabla 1-1**:

**Tabla 1-1:** Características de Java

<b>Propiedad</b>	<b>Breve descripción</b>
<b>Orientado a Objetos</b>	Salvo los tipos de datos primitivos, en Java todo es un objeto, de esta manera se garantiza un mejor dominio de la complejidad, reutilización más sencilla y mayor facilidad de corrección y evolución.
<b>Distribuido</b>	Java implementa los protocolos de red estándar, permitiendo desarrollar aplicaciones cliente / servidor en arquitecturas distribuidas.
<b>Interpretado</b>	Cuando se realiza un programa Java, este no se ejecuta más bien se lo interpreta por la máquina virtual o JVM (Java Virtual Machine). Por lo tanto, en este lenguaje solo hay dos etapas, la compilación y la ejecución. La máquina virtual se encarga de la operación de edición de enlaces en tiempo de ejecución del programa.
<b>Robusto</b>	Java es un lenguaje fuertemente tipado y estricto, ej. la declaración de las variables es obligatoriamente explícita. Con el lenguaje java en el momento de la compilación y de la ejecución se verifica la sintaxis del código para evitar los errores y problemas de compatibilidad.
<b>Securizado</b>	Posee un mecanismo que vigila la seguridad de las aplicaciones de los sistemas, este es el motor de ejecución de Java (JRE).
<b>Independiente de las arquitecturas</b>	Con el lenguaje Java no se produce un código para un tipo de arquitectura específico. El compilador genera un bytecode que es independiente y que es fácil en su interpretación o transformación dinámica en código nativo para aumentar el rendimiento. La máquina virtual se encarga de traducir el bytecode.
<b>Portable</b>	Java es portable gracias a que se trata de un lenguaje interpretado.
<b>Eficaz</b>	Al ser un programa Java interpretado, quiere decir que es lento en relación con un programa nativo, pero Java pone en marcha un proceso de optimización de interpretación de código, llamado JIT (Just in Time) o HotSpot. Este 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++.
<b>Multitarea</b>	Java permite desarrollar aplicaciones que ponen en marcha la ejecución simultánea de varios hilos o procesos ligeros.

Fuente: (Groussard, 2014, pp. 16-19)

Realizado por: Chela Jhonathan, 2018

El lenguaje de programación Java se conjuga de manera compatible con la tecnología de desarrollo de sistemas web robustos, esto es el framework Java Server Face, que permite realizar un buen diseño front – end.

### ***1.1.3. Java Server Face***

Java Server Face (JSF) es un framework modelo, vista, controlador (MVC) basado en el API de Servlets que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante el framework Facelets. Facelets se define en la especificación 2 de JSF como un elemento fundamental que proporciona características de plantillas y de creación de componentes compuestos. (Universidad de Alicante, 2014)

#### ***1.1.3.1. Características de JSF***

Java Server Face proporciona rasgos destacables para desarrollar aplicaciones web, a continuación, se cita sus características: (Universidad de Alicante, 2014)

- Define las interfaces de usuario mediante vistas que agrupan componentes gráficos.
- Mediante los beans gestionados se permite la conexión de los componentes gráficos con los datos de la aplicación.
- Conversión de los datos y la validación automática de la entrada del usuario.
- Navegación entre vistas.
- A partir de la especificación 2.0 un modelo estándar de comunicación Ajax entre la vista y el servidor.

#### ***1.1.3.2. Funcionamiento***

El funcionamiento del framework JSF cumple el proceso que se describe a continuación: (Universidad de Alicante, 2014)

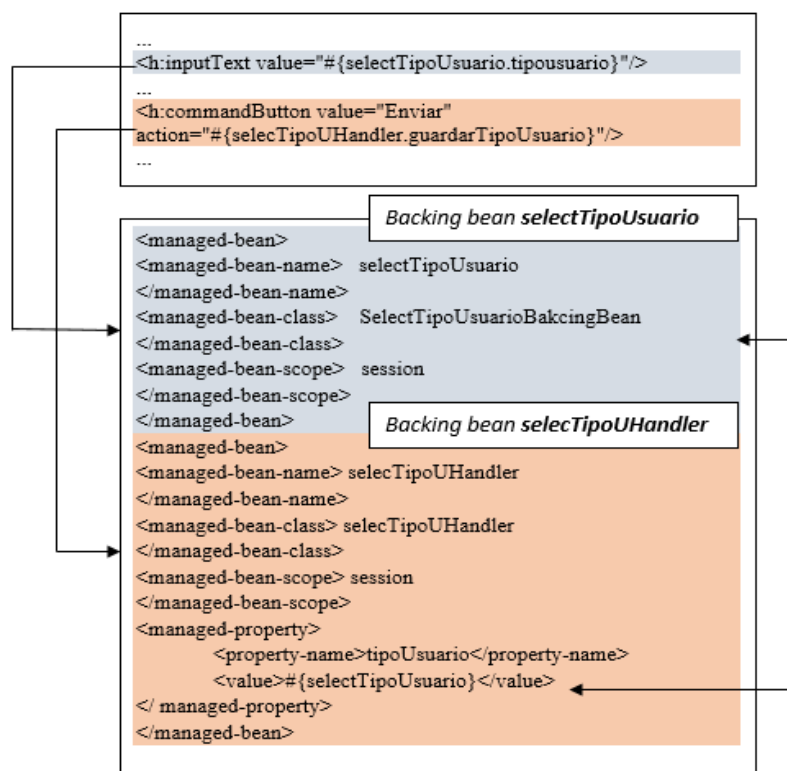
1. Se procesa la página de arriba abajo y se crea un árbol de componentes JSF en forma de objetos instanciados de clases del framework JSF.
2. Se obtienen los valores introducidos por el usuario y se actualizan los beans gestionados con ellos.
3. Se actualizan los componentes con los valores procedentes de las propiedades de los beans gestionados.
4. Se pide a los componentes que se rendericen, generándose el código HTML que se envía de vuelta al navegador como resultado de la petición.

5. El árbol de componentes JSF se guarda en memoria para que posteriores peticiones a la misma página JSF no tengan que crearlo, sino que utilicen el existente.

### 1.1.3.3. Elementos de JSF

Cuándo se habla de una aplicación realizada con JSF, se tiene que tener en cuenta tres elementos importantes, la vista, el modelo y el controlador. A continuación, en los siguientes apartados se expone cada uno de los mismos.

**La vista:** se define mediante páginas con componentes JSF que utilizan beans gestionados para almacenar los datos. Los beans se declaran en el fichero de configuración *faces-config.xml*.



**Figura 2-1:** Vista en JSF

**Fuente:** (Universidad de Alicante, 2014).

**Realizado por:** Chela Jhonathan, 2018.

La **Figura 2-1** muestra las relaciones entre la vista y la configuración, el cuadro de la parte superior representa a la vista (*select-tipoUsuario.jsp*) y el cuadro de la parte inferior representa a la configuración (*faces-config.xml*). El lenguaje de expresiones de JSF se puede utilizar también en el fichero *faces-config.xml* para inicializar los valores de las propiedades de los beans, esto permite acceder de un bean hacia otro. (Universidad de Alicante, 2014)



**El modelo:** constituyen a las propiedades y comportamientos representados en un bean de Java normal, es importante tomar en cuenta que, al referirse a propiedades y comportamientos, son atributos y métodos (getters y setters), etc. (Universidad de Alicante, 2014)

**El controlador:** se define mediante métodos de los beans ligados a acciones de la vista. La acción a ejecutar se define en el código del método y la vista resultante depende de la cadena devuelta y del fichero de configuración *faces-config.xml*. (Universidad de Alicante, 2014)

#### *1.1.3.4. Ventajas de usar de JSF*

Algunas de las ventajas que tiene Java Server Face son (Munoz Simo, 2012):

- El código JSF es similar al estándar HTML, por lo que hace su fácil utilización por desarrolladores y diseñadores web.
- JSF permite introducir javascript en la página lo que acelera la respuesta de la interfaz en el cliente.
- JSF permite desarrollar nuevos componentes a medida, esto hace que sea extensible, mediante APIs se puede modificar el comportamiento y controlar su funcionamiento.
- Las nuevas versiones del framework recogen la funcionalidad de versiones garantizando un porcentaje de compatibilidad alto, esto permite que el mantenimiento de las aplicaciones sea sencillo.

En el diseño de interfaces robustas para sistemas web es importante tomar en cuenta también el diseño adaptable de interfaces a dispositivos sobre los cuales un software es invocado, por lo tanto, una de las opciones más acertadas que se tiene es el trabajo con Bootstrap, en los siguientes ítems se realiza la descripción de esta poderosa herramienta para el diseño de interfaces web responsivas.

#### *1.1.4. Bootstrap*

Bootstrap es un producto open source creado por Mark Otto y Jacob Thornton, esta herramienta de software nace como la necesidad de estandarizar el conjunto de herramientas para el trabajo front-end de los ingenieros en la compañía de Twitter. (Spurlock, 2013, p. 1)

El objetivo que Bootstrap persigue es facilitar el diseño web permitiendo la creación de webs adaptables, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla.

#### 1.1.4.1. Estructura de un archivo Bootstrap

Bootstrap tiene una estructura definida que hace posible el funcionamiento en el diseño de las páginas web, a continuación, en la **Figura 3-1** se describe la forma en la que está estructurado este framework:

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

**Figura 3-1:** Estructura de Bootstrap

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

**Realizado por:** Chela Jhonathan, 2018.

La **Figura 3-1** muestra la estructura que tiene el archivo de Bootstrap que incluye las carpetas: css, js, e img. En la carpeta css se incluye las hojas de estilo que hacen posible el diseño de la web; en la carpeta js se encuentra el java script y en la carpeta img se encuentra todos los iconos que se pueden incluir en una página web.

Es importante recalcar 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).

#### 1.1.4.2. Ventajas de usar Bootstrap

Bootstrap presenta ventajas principales y recalcales que a continuación se citan (Alba, 2014):

- Este framework tiene un mantenimiento y servicio de actualización realizados por Twitter.
- Presenta un paquete de elementos web personalizables.
- Incluye Grid system, que permite realizar la maquetación de las columnas.
- Las plantillas son de sencilla adaptación responsiva.
- Utiliza el lenguaje Less, este lenguaje permite enriquecer los estilos (CSS) de la web.
- Consta de documentación, tutoriales y plugins que permiten aprender a utilizar con bastante facilidad.

Además de las tecnologías para el desarrollo de un proyecto se debe realizar también un estudio de la metodología más adecuada que permita definir la planificación y los lineamientos de un trabajo gradual garantizando la funcionalidad y calidad para el cliente, a continuación, se aborda puntos de vital importancia sobre la metodología de desarrollo ágil denominada SCRUM.

## 1.2. Metodología SCRUM

### 1.2.1. Definición

Es una metodología de desarrollo de software ágil y flexible que permite construir un producto incrementalmente en entornos donde los requisitos sufren cambios constantes. El objetivo de SCRUM es generar un proceso conveniente para los proyectos informáticos que se trabajan en equipos de desarrollos pequeños con requisitos poco estables e iteraciones cortas (Laínez Fuentes, 2015, p. 127).

### 1.2.2. El equipo SCRUM

El equipo SCRUM está compuesto por tres roles importantes que hacen posible trabajar y alcanzar el objetivo o meta planificado. A continuación, en la **Tabla 2-1** estos roles son citados y explicados de manera objetiva y breve.

**Tabla 2-1:** Miembros el equipo SCRUM

Nombre	Descripción
<b>Dueño del producto (Product Owner)</b>	Es el responsable de gestionar el listado de trabajo (Product Backlog), expresando claramente cada uno de sus elementos para posteriormente ordenarlos de la manera más conveniente posible para alcanzar los objetivos y misiones establecidos. Además, 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.
<b>Scrum Master</b>	El objetivo fundamental del Scrum Master es asesorar y formar a los miembros del equipo de desarrollo para trabajar de forma auto organizada y con responsabilidad. Además, debe ser capaz de resolver las dificultades que representen un peligro en la ejecución de las tareas en un sprint (iteración o entregable).
<b>Equipo de desarrollo (Development Team)</b>	Está conformado por profesionales que hacen posible la entrega de una parte del producto total funcionando (incremento), que sin ningún problema puede ser ubicado en la etapa de producción al finalizar cada iteración o entregable.

**Fuente:** (Schwaber & Sutherland, 2017, p. 6)

**Realizado por:** Chela Jhonathan, 2018.

### ***1.2.3. Artefactos de SCRUM***

Los artefactos Scrum son elementos físicos que se producen para garantizar la transparencia de información y permitir la oportunidad de la inspección y adaptación.

Dentro de Scrum se consideran tres artefactos que a continuación se describen:

#### ***1.2.3.1. Lista de Producto (Product Backlog)***

Es una lista de ítems que representa los requisitos que generalmente el ente denominado como cliente ha solicitado y marca todo el periodo completo de trabajo. (Requena Mesa, 2018)

Cada uno de los requisitos deben ser descritos incluyendo, la descripción, el orden, la estimación y el valor, estos atributos permiten agrupar y priorizar los elementos de la lista de producto. El equipo de desarrollo es el responsable de la estimación de los elementos de la lista de producto, porque es el equipo quien va a llevar a cabo el trabajo y es quien se compromete a cumplir el objetivo del Sprint. (Ramos Vega, 2017)

#### ***1.2.3.2. Lista de Pendientes del Sprint (Sprint Backlog)***

Este artefacto consiste en un subconjunto de elementos seleccionados más un plan para entregarlos como incremento y conseguir el objetivo del Sprint.

Una lista de pendientes del Sprint está contenida dentro del plan total de lista de producto o Product Backlog, la misma que representa una estimación de funcionalidades que formarán parte de un incremento. Además, mediante el Sprint Backlog se hace visible el trabajo del equipo de desarrollo incluyendo las mejoras a procesos de alta prioridad. (Schwaber & Sutherland, 2017)

#### ***1.2.3.3. Incremento***

El incremento es una pieza de software terminado al final de cada Sprint. De una manera muy clara se puede apreciar que el incremento es la suma de las tareas, casos de uso, historias de usuario y cualquier elemento que se ha desarrollado durante el Sprint y que será puesto a disposición del usuario final. (Roche, 2016)

### ***1.2.4. Eventos SCRUM***

Los eventos de Scrum son considerados como periodos o bloques de tiempo definidos con el objetivo de crear regularidad y minimizar la necesidad de reuniones no definidas. Se considera el Sprint como el evento que contiene a los demás, constituyendo una oportunidad formal para la inspección y adaptación de cualquier cualidad faltante en el desarrollo del producto software. (Schwaber & Sutherland, 2017, p. 9)

A continuación, en la **Tabla 3-1** se citan los cinco eventos que componen Scrum:

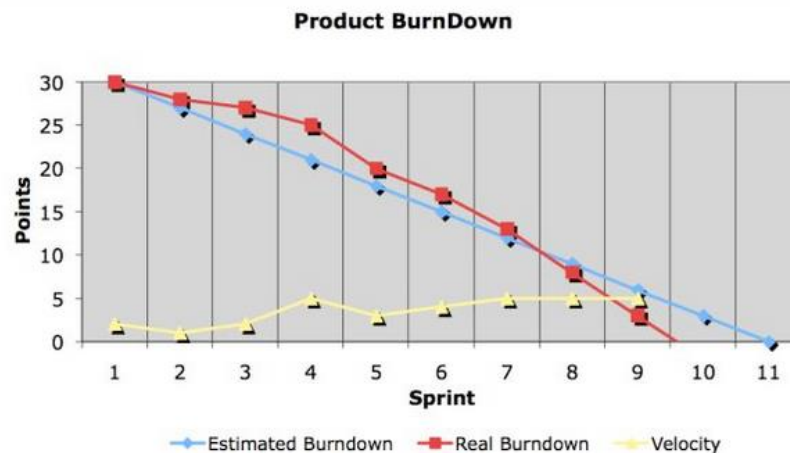
**Tabla 3-1:** Eventos de SCRUM

<b>Evento</b>	<b>Descripción</b>
<b>Sprint</b>	Comprende un periodo máximo de tiempo recomendado de 1 mes, durante la duración de cada Sprint se crea un incremento de producto software, funcional y utilizable, listo para ser puesto en la etapa de producción
<b>Reunión de planificación de Sprint</b>	Es una reunión que se realiza en el inicio de cada Sprint, para planificaciones de 30 días de trabajo esta reunión debe tener una duración de 8 horas. Es importante tener en cuenta que si el Sprint es más corto lógicamente el tiempo es más corto. (Palacios, 2016)
<b>Objetivo del Sprint</b>	Representa la meta que se pretende alcanzar con el equipo de trabajo mediante la implementación de ítems que representan un entregable funcional del producto software en construcción. (Revueltas, 2018)
<b>Scrum Diario</b>	Tiene la finalidad de servir como una guía entre los miembros del equipo de desarrollo para alcanzar el objetivo del Sprint. Esta reunión diaria es corta y el tiempo de duración es de 15 minutos que sirve como un lineamiento para que el equipo de desarrollo crezca y mejore el entendimiento compartido de las cosas más importantes que debe ser realizadas. (Francia, 2017)
<b>Revisión de Sprint</b>	La reunión de revisión del sprint tiene como objetivo mostrar el incremento (entregable) correspondiente terminado sin olvidar la adaptación óptima de la lista de requerimientos para las próximas presentaciones de los avances del proyecto en ejecución. (Ramos Vega, 2017)
<b>Retrospectiva de Sprint</b>	Permite al Equipo Scrum inspeccionarse a sí mismo y permite crear un plan de mejoras que será puesto en marcha inmediatamente en el siguiente Sprint.

Realizado por: Chela Jhonathan, 2018

### **1.2.5. Burndwon chart**

El Burndwon chart es un gráfico que permite monitorear el trabajo pendiente a lo largo del tiempo mostrando la velocidad a la que se completan los objetivos, requisitos o historias de usuarios. Este tipo de grafico permite extrapolar si el equipo de desarrollo podrá completar el trabajo en tiempo estimado. En el (**Gráfico 1-1**) se muestra un ejemplo del gráfico explicado en este apartado.



**Gráfico 1-1:** Ejemplo de Burndown chart

**Fuente:** (Scrum Institute, 2017)

**Realizado por:** Chela Jhonathan, 2018.

El **Gráfico 1-1** permite apreciar la distribución de los ejes, el eje  $x$  u horizontal representa la cantidad de los Sprints que involucra todo el desarrollo del proyecto software, en este caso son 11 Sprints planificados. En el eje  $y$  o vertical se puede apreciar la cantidad de puntos planificados para el proyecto, en este caso son 30 puntos.

A medida que se van completando las historias de usuario que componen cada Sprint se realiza el gráfico de color rojo que muestra el avance real del proyecto, por otra parte, la gráfica de color azul representa el camino ideal que debe seguir el avance del proyecto. El gráfico de color amarillo representa en cierto modo la variación de velocidad que se tiene durante el trabajo del proyecto.

La metodología SCRUM define lineamientos claros que sirven de guía para la construcción de un proyecto web con calidad en cada incremento realizado para un tiempo específico determinado, pero también para la aplicación de pruebas unitarias es necesario seguir un ciclo de vida de desarrollo, de acuerdo al modelo e ítems definidos, el modelo de verificación y validación (V&V) es el que aborda este tipo de pruebas, a continuación, se realiza un estudio acertado sobre el tema.

### 1.3. Modelo V&V de desarrollo de software

El aseguramiento de la calidad incluye la realización de casos de prueba a un producto software, estos casos permiten consolidar un alto porcentaje 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)

En primer lugar, 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.

### ***1.3.1. Verificación***

De acuerdo con la norma ISO/IEC/IEEE 24765: 2010, la verificación es el proceso de evaluación de un sistema o componente para determinar si un producto en cada fase de desarrollo satisface las condiciones impuestas al inicio de la misma. Ésta es una técnica de análisis estático que se realiza sin ejecutar código.

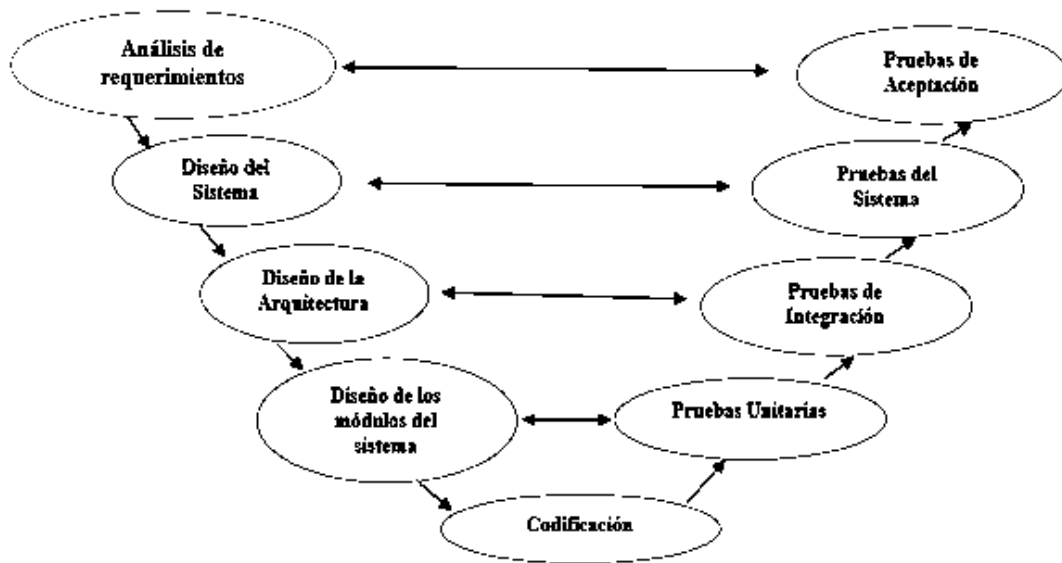
### ***1.3.2. Validación***

Es una técnica de análisis dinámico, funcional y no funcional que involucra realizar pruebas mediante la ejecución de código. La validación es el proceso para evaluar el software que se ejecuta luego de la finalización de una fase en el ciclo de vida de desarrollo, al final permite establecer si se está cumpliendo con las expectativas y los requisitos del cliente. (Mera-Paz, 2016)

### ***1.3.3. Definición del modelo V&V***

El modelo de verificación y validación (V&V) representa un tipo de ciclo de vida de desarrollo de software. Este modelo permite crear una manera secuencial de estados o fases por los cuales la construcción de un sistema informático avanza.

Este estándar se basa en la asociación de la fase de pruebas para cada etapa de desarrollo correspondiente. La siguiente fase de desarrollo solo comienza después de completar la fase anterior, para cada actividad de desarrollo existe una actividad de pruebas que la corresponde. (Pauta Ayabaca & Moscoso Bernal, 2017)



**Figura 4-1:** Modelo V&V

Fuente: (Mera-Paz, 2016)

Realizado por: Chela Jhonathan, 2018.

En la **Figura 4-1**, se expone el modelo de verificación y validación para el desarrollo de software, en la parte izquierda en forma descendente se puede apreciar cada una de las etapas que conlleva a la construcción de un sistema informático, cada fase de la elaboración es verificada con la siguiente fase inmediata y es validada con las pruebas que se exponen en la parte derecha de la figura, a continuación, una explicación de cada etapa (Kumar, 2017):

- **Análisis de requerimientos:** en esta etapa se define los requerimientos y expectativas que el usuario tiene con respecto al software. Para ello, es recomendable el uso de una buena comunicación y el uso del estándar IEEE 830.
- **Diseño del sistema:** esta etapa contiene el diseño del sistema, así como también la configuración completa de comunicación por medio de hardware.
- **Diseño de la arquitectura:** en esta etapa el diseño del sistema es separado en módulos que desempeñan diferentes funciones, además, la transferencia de datos y la comunicación de los módulos internos con el mundo exterior están claramente definidos.
- **Diseño de los módulos del sistema:** esta etapa permite definir o dividir el diseño del sistema en módulos pequeños, también es conocido como diseño detallado o diseño de bajo nivel (Low-Level Design).
- **Codificación:** esta etapa comprende el desarrollo del sistema de acuerdo al diseño establecido, para lo cual se hace uso de un lenguaje de programación y herramientas informáticas que hagan posible la consecución de un software de acuerdo a las especificaciones establecidas.



La fase de pruebas del modelo está expuesta en la parte derecha de la **Figura 4-1**, debe ser entendido cada etapa, en forma ascendente que valida cada una de las fases de desarrollo, a continuación, se presenta una clarificación de cada etapa de pruebas (Kumar, 2017):

- **Pruebas unitarias:** estas pruebas son desarrolladas durante la fase del diseño de módulos del sistema. Los planes de pruebas unitarias son ejecutados para eliminar errores a nivel de unidades de código.
- **Pruebas de integración:** las pruebas de integración son ejecutadas en la fase de diseño de la arquitectura del sistema, después de ejecutar las pruebas unitarias. Estas pruebas permiten verificar la comunicación entre los diferentes módulos del sistema integrado.
- **Pruebas del sistema:** estas pruebas validan la aplicación completa con su funcionalidad, interdependencia y comunicación. Se comprueba los requisitos funcionales y no funcionales de la aplicación desarrollada.
- **Pruebas de aceptación:** Estas pruebas son desarrolladas en un ambiente de usuario final similar a un ambiente de producción, estas pruebas validan si el sistema entregado cumple con los requisitos del usuario y que el sistema está 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. En el capítulo II de este trabajo se expone la documentación necesaria acerca de las historias de usuario y las 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.

#### **1.4. 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.

#### ***1.4.1. 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.

### **1.5. Pruebas de caja negra**

En las pruebas de caja negra el tester prueba una aplicación sin el conocimiento del trabajo interno de la misma. Los datos son ingresados en la aplicación y la salida es comparada con los resultados esperados. Lo que el programa hace con los datos de entrada o como el programa permite la salida de los resultados obtenidos no es un problema para el desarrollador de las pruebas de caja negra. (Ghahrai, 2018)

#### ***1.5.1.1. Definición***

La organización para la certificación de la calidad de software denominada en inglés International Software Testing Qualifications Board (ISTQB), define dos conceptos importantes:

- Pruebas de caja negra: son pruebas ya sean funcionales o no funcionales, sin referencia a la estructura interna del componente o sistema.
- Técnica de diseño de las pruebas de caja negra: es el procedimiento para obtener y/o seleccionar los casos de prueba basado en un análisis de la especificación, ya sean funcionales o no funcionales de un componente o sistema, sin conocer la referencia de su estructura interna.

#### *1.5.1.2. Niveles de la aplicación de las pruebas de caja negra*

Los niveles de aplicación de las pruebas de caja negra son las que a continuación de cita (Software Testing Fundamentals, 2018):

- Pruebas de integración: es el nivel en el cual el software es privado cuando unidades individuales de código son combinados y sometidos a prueba como un grupo.
- Pruebas del sistema: es el nivel en el que el software completo e integrado es probado, el propósito es evaluar el cumplimiento del sistema con los requisitos especificados.
- Pruebas de aceptación: es el nivel en el que el software es probado para su aceptabilidad, el propósito es evaluar el cumplimiento del sistema con los requisitos del negocio y evaluar si es aceptable para la entrega.

#### *1.5.1.3. Tipos de pruebas de caja negra*

Las pruebas de caja negra se pueden catalogar en los siguientes tipos que a continuación se mencionan: (Guru99, 2018)

- Pruebas funcionales: este tipo de pruebas están relacionadas a los requerimientos funcionales del sistema, es realizada por los testers de software.
- Pruebas no funcionales: este tipo de pruebas se relacionan directamente con los requisitos no funcionales, como el rendimiento, la escalabilidad y la usabilidad.
- Pruebas de regresión: este tipo de pruebas se realiza después de haber realizado las correcciones pertinentes, el propósito es verificar si los cambios no han afectado al código existente.

#### *1.5.1.4. Ventajas de las pruebas de caja negra*

Algunas de las ventajas que presenta las pruebas de caja negra, son las que a continuación se mencionan:

- Facilitan la identificación de contradicciones e incertidumbres en las especificaciones funcionales. (Rongala, 2015)

- Las pruebas de caja negra se realizan desde el punto de vista del usuario del sistema. (Rongala, 2015)
- Las pruebas de caja negra evalúan la calidad de la aplicación analizando el resultado real con el resultado esperado. (Nandini, 2017)
- El tester es libre de cualquier presión con respecto al conocimiento de lenguajes de programación específicos para probar la confiabilidad y funcionalidad de una aplicación. (Rongala, 2015)
- El desarrollador y el tester son independientes y no existe ninguna dependencia entre ellos. (Nandini, 2017)

## **1.6. Pruebas de caja blanca**

Las pruebas de caja blanca son llamadas también pruebas estructurales, en este enfoque, las pruebas estructurales deben abarcar completamente la estructura interna del producto software. Las pruebas estructurales son aplicadas a pequeñas unidades como, por ejemplo, subrutinas, o las operaciones asociadas con un objeto. Como lo implica su nombre, el tester puede analizar el código y el resultado que deriva de los casos de prueba. (Agarwal, et al., 2010, p. 173)

### *1.6.1.1. Definición*

Para dar una definición acertada de las pruebas de caja blanca se cita lo que expone la International Software Testing Qualifications Board (ISTQB):

- Pruebas de caja blanca: son pruebas basadas en un análisis de la estructura interna de los componentes o un sistema.
- Técnica de diseño de prueba de caja blanca: es el procedimiento que deriva y/o selecciona los casos de prueba basados en un análisis de la estructura interna de un componente o sistema.

### *1.6.1.2. Niveles de aplicación de las pruebas de caja blanca*

Las pruebas de caja blanca tienen tres niveles de aplicación, a continuación, se describe brevemente cada nivel de aplicación (Guru99, 2018):

- Pruebas unitarias: se prueba los caminos dentro de una unidad de código.
- Pruebas de integración: se prueba los caminos entre las unidades de código.
- Pruebas al sistema: se prueba los caminos entre los subsistemas.

### *1.6.1.3. Razones por las que las pruebas unitarias son desarrolladas*

Las razones principales por las que las pruebas estructurales son desarrolladas son: (Agarwal, et al., 2010, p. 175)

- Para comprobar que todos los caminos en un proceso son correctamente operacionales.
- Todas las decisiones lógicas son ejecutadas por el camino condicional del Sí y el No.
- Todos los bucles son ejecutados con sus valores límites probados.
- Para determinar si las especificaciones de la estructura de datos de entrada se prueban y luego se utilizan para otro procesamiento.
- Para comprobar cada declaración, objetos y funciones en una parte individual. (Guru99, 2018)

### *1.6.1.4. Ventajas de las pruebas de caja blanca*

La aplicación de las pruebas unitarias al desarrollo de un producto software trae ventajas importantes, a continuación, se citan las mismas:

- Ayuda al desarrollador de las pruebas al análisis cuidadoso sobre la implementación de cada caso. (Agarwal, et al., 2010, p. 175)
- Los casos de las pruebas de caja blanca son fáciles de automatizar. (Guru99, 2018)
- Permiten la optimización del código al descubrir los errores ocultos en el producto software. (Agarwal, et al., 2010, p. 175)
- Las pruebas unitarias en la mayoría de casos pueden ser ejecutadas mucho antes de tener disponible la interfaz gráfica del producto software. (Guru99, 2018)

De acuerdo con lo expuesto en el enfoque de pruebas de caja blanca, en el nivel de pruebas unitarias conforme el ciclo de vida de desarrollo de software de verificación y validación, se aborda de manera eficaz este tema en los siguientes puntos.

## **1.7. Pruebas unitarias**

La idea de pruebas unitarias se empieza a tener en cuenta desde el año de 1976, cuando David J. Panzal presentó su trabajo denominado “*Procedures: A New Approach to Software Verification*” en la segunda edición de la conferencia internacional de la ingeniería de software, este trabajo describe al procedimiento de las pruebas como una manera de invocar casos en un módulo respectivo para generar un reporte indicando cuanto al caso de prueba haya fallado. (Gulati & Sharma, 2017, p. 5)

A continuación, se describe la definición más acertada para las pruebas unitarias, las características y los beneficios de trabajar con esta técnica.

#### *1.7.1.1. Definición*

Las pruebas unitarias son una técnica que se usa para comprobar la funcionalidad de una unidad de código. Las pruebas en el nivel de unidades de código son ejecutadas en las clases y los métodos capturando los resultados de los mismos. El propósito de las pruebas unitarias es aislar cada sección del código y verificar si devuelve los resultados correctos. (Dietrich, 2014)

#### *1.7.1.2. Características de las pruebas unitarias*

Las pruebas unitarias cumplen con las siguientes características: (Dietrich, 2014)

- **Automatizable:** los resultados son específicos para cada caso de prueba, pero las pruebas unitarias se pueden automatizar para obtener los resultados de forma individual o en grupos.
- **Legible:** el objetivo de una prueba unitaria es dar a conocer al tester sobre lo que la unidad hace, si la prueba no es legible, el tester no sabrá cuando dicha prueba falle.
- **Rápido:** los códigos de las pruebas unitarias no deben tomar más de 5 minutos en ser creados.
- **Independiente y aislado:** las pruebas unitarias funcionan independientemente de otros casos de prueba, en un ambiente único para cada prueba.
- **Correcto:** una buena prueba unitaria hace exactamente lo que dice que hace, un caso de prueba corresponde a un caso singular y único.
- **Repetible:** en el caso de repetir las pruebas de forma individual o grupal, el resultado debe ser siempre el mismo dando igual el orden en que se realicen los test, las pruebas se almacenan para poder realizar estas repeticiones o poder usarlos en otras ocasiones.

#### *1.7.1.3. Beneficios de las pruebas unitarias*

Los beneficios de realizar las pruebas unitarias a nuestras unidades del software son: (Apiumhub, 2017)

- **Determina las especificaciones:** antes de empezar a codificar los componentes se debe tratar de determinar que va a hacer el componente. Se debe tratar de construir un caso de prueba con todas las posibles entradas y salidas para clarificar el comportamiento esperado del componente.

- Provee una temprana detección de errores: las pruebas unitarias al ser ejecutadas pueden detectar fallas en la primera estancia, además, permiten saber si un componente está completo.
- Mantenimiento de soporte: un conjunto de pruebas unitarias bien escritas sirve como un aumento a la productividad para el equipo, esto debido al aumento de requerimientos todo el tiempo de desarrollo.
- Mejorar el diseño: las pruebas unitarias ayudan a perfeccionar las especificaciones del producto mediante la exposición de fallas en el diseño de la interfaz.
- Documentación del producto: las pruebas unitarias describen como trabaja una pieza de código, de manera sincronizada se guarda el último estado de la especificación de manera sincronizada 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.

### 1.8. Cobertura de código

La cobertura de código es una medida porcentual en las pruebas de software que permite cuantificar el grado en el que el código fuente de un programa ha sido comprobado.

Esta medida ayuda a definir que partes de un código han sido comprobados y que partes no, se considera que una línea de código es cubierta, cuando ha sido ejecutada en la realización de su caso de prueba correspondiente. (López, 2017)

Generalmente, este tipo de medida es utilizada para evaluar la calidad de las pruebas unitarias, por la sencilla razón de que, al efectuar el caso de test respectivo que devuelve un alto valor porcentual de cobertura, se puede afirmar que se tiene cierta certeza sobre el correcto funcionamiento de las unidades funcionales de la aplicación. (López, 2017)

En la **Tabla 4-1** se puede apreciar las diferentes métricas de cobertura de código:

**Tabla 4-1:** Tipos de métricas de cobertura de código

Tipo de métrica	Descripción
Instrucciones	Es la unidad más pequeña, las instrucciones de código de byte, la cobertura permite tener la noción de la cantidad de código ejecutado o no.
Ramas	Este tipo de métrica permite contar el total de ramificaciones caminos que existen en un método y determina el número de ramas ejecutadas o no. Por ejemplo, las ramificaciones de una sentencia switch.

Complejidad ciclomática	Representa el número mínimo de caminos que pueden generar todas las rutas posibles en un método.
Líneas	Para saber que una línea de código ha sido comprobada, como mínimo una instrucción perteneciente a la línea debió ser ejecutado.
Métodos	Los métodos al contener instrucciones en el instante de cobertura de código, el método se considera ejecutado y se devuelve un valor, cuando como mínimo una instrucción fue cubierto.
Clases	Para obtener un valor de cobertura de una clase, como mínimo uno de sus métodos debió ser ejecutado.

Fuente: (López, 2017)

Realizado por: Chela Jhonathan, 2018.

En la **Tabla 4-1** se puede observar los tipos de métricas con los cuales la cobertura de código puede ser tratado, se puede obtener un valor porcentual de acuerdo a las instrucciones, ramas, la complejidad ciclomática, las líneas de código, los métodos y las clases.

Para las pruebas unitarias conforme con cada unidad de código en el presente proyecto, el valor porcentual de cobertura de código corresponderá a la cantidad de líneas de código.

El objeto de estudio de este trabajo de titulación es la aplicación de pruebas unitarias en el diseño modular del sistema conforme a lo expuesto en el modelo de verificación y validación, para la automatización de las pruebas unitarias se exponen en los siguientes puntos las tecnologías adecuadas.

## **1.9. Herramientas para automatizar las pruebas unitarias**

### **1.9.1. JUnit**

JUnit es un framework de código abierto para realizar pruebas unitarias en Java. Es una herramienta muy útil para los desarrolladores en lenguaje Java cuando se trata de escribir y ejecutar pruebas repetibles. (Pandya, 2017)

Este framework fue desarrollado por Kent Beck y Erich Gamma, basado en SUnit, un framework de pruebas unitarias escrito en Smalltalk. La versión de JUnit fue desarrollada en 1997 y se ha convertido en el estándar adoptado por diferentes lenguajes y muchas herramientas. (Gulati & Sharma, 2017)

Esta herramienta se puede incorporar a diferentes entornos integrados de desarrollo de software, entre los cuales se destaca: Elclipse y Netbeans.



JUnit incluye las siguientes características: (JUnit, 2018)

- Afirmaciones para probar los resultados esperados.
- Elementos de prueba para compartir datos comunes de las pruebas.
- Ejecutores de los casos de prueba automatizados.

Hacer uso de este framework conlleva las siguientes ventajas: (Pandya, 2017)

- Los errores son detectados en una etapa temprana permitiendo así que el código sea más confiable.
- La utilidad es elevada de este framework en entornos de trabajo basado en pruebas.
- Las pruebas unitarias obligan a los desarrolladores a leer el código más que su escritura.
- El uso del framework permite escribir un código más legible confiable y libre de errores lo que implica mayor confianza durante el desarrollo.

### **1.9.2. Testlink**

TestLink es una herramienta gratuita que permite crear y gestionar casos de pruebas y organizarlos en planes.

Estos planes de prueba permiten a los miembros de un equipo de desarrollo ejecutar casos de prueba y registrar los resultados dinámicamente, generar informes, mantener la trazabilidad con los requerimientos, así como priorizar y asignar tareas. (Terra, 2013)

#### **1.9.2.1. Terminología básica**

Para utilizar TestLink es importante conocer la terminología que se maneja en esta herramienta, en la **Tabla 5-1** se citan los términos y su breve descripción.

**Tabla 5-1:** Terminología básica TestLink

<b>Término</b>	<b>Definición</b>
<b>Caso de prueba (Test case)</b>	Describe una prueba a través de los resultados esperados y los pasos (acciones, escenarios). Los Test case son la pieza fundamental de TestLink.
<b>Suite de casos de prueba (Test Suite)</b>	Organiza los casos de prueba en unidades lógicas, estos también eran conocidos como componentes y categorías.
<b>Plan de pruebas (Test plan)</b>	Se crea cuando se desea ejecutar casos de prueba. Puede estar compuesto por uno o varios proyectos de prueba.
<b>Usuarios (Users)</b>	Cada usuario tiene un papel que define las características disponibles de TestLink que puede utilizar.
<b>Proyecto de prueba (Test project)</b>	Es un componente que siempre existirá en TestLink y puede ser sometido a muchas versiones diferentes. Un proyecto de prueba

	incluye; pruebas de especificación con casos de prueba, requerimientos y palabras claves. Todos los usuarios dentro de un proyecto, tienen un perfil definido.
--	--

**Fuente:** (QAustral, 2010)

**Realizado por:** Chela Jhonathan, 2018

#### *1.9.2.2. Ventajas de Testlink*

Los siguientes puntos describen las ventajas que TestLink presenta: (Terrera, 2013)

- Testlink soporta múltiples proyectos.
- Permite la exportación e importación de casos de prueba de una manera fácil.
- Testlink es fácil de integrar con varias herramientas de gestión de defectos.
- Para los casos de prueba su ejecución es automatizada a través de XML-RPC.
- Los casos de prueba pueden ser filtrados fácilmente por versiones, palabras clave e id de caso de prueba.
- Permite fácilmente asignar los casos de prueba a múltiples usuarios.
- Se puede fácilmente generar el plan de prueba y los informes de prueba en varios formatos.
- Se puede asignar credenciales y roles a múltiples usuarios.

## CAPÍTULO II

### 2. MARCO METODOLÓGICO

#### 2.1. Tipo de investigación

##### a) Investigación exploratoria

La investigación exploratoria pretende dar una visión general o aproximado respecto a un tema determinado, para Ibarra (2011), este tipo de investigación responde a las preguntas, para qué, cuál es el problema y qué se podría investigar, sobre un problema planteado.

En el presente proyecto, se ha logrado establecer el mejor panorama sobre la aplicación de pruebas unitarias automatizadas en el desarrollo de software. Las pruebas de caja blanca han sido determinadas para su ejecución en la parte interna del código del proyecto mientras que las pruebas de caja negra permitieron verificar y validar las funcionalidades del sistema.

##### b) Investigación descriptiva

Este tipo de investigación se centra en la búsqueda de la especificación de propiedades importantes de personas, grupos, comunidades y otros aspectos que se incluya en el tema tratado.

Se utilizó este método para poder definir los usuarios involucrados en el presente proyecto, así como también los roles, entidades y funcionalidades propias de este trabajo de titulación.

##### c) Investigación aplicada

Este tipo de investigación permite hacer uso de un conocimiento adquirido o recopilado para alcanzar un objetivo o un fin específico. Una vez realizada la formulación del marco teórico relevante al desarrollo de software con pruebas unitarias automatizadas, se procedió a la construcción del sistema de gestión de guiones de radio denominado, SIGUIÓN. En este proyecto se aplica, fundamentos teóricos junto con la ayuda de tecnologías y herramientas que han generado como producto un sistema con gran exactitud en sus funcionalidades y unidades de código.

## **2.2. Métodos y técnicas**

### **2.2.1. Métodos aplicados en el trabajo de titulación**

#### a) Método analítico

Este tipo de método permite descomponer un todo en partes para poder estudiar las causas, la naturaleza y los efectos de un ente de investigación. (Ruiz Limón, 2006)

Este método ha permitido identificar las partes denominadas unidades de código de cada funcionalidad del sistema para poder diseñar y ejecutar las pruebas unitarias automatizadas de caja blanca.

#### b) Método sintético

Este método permite unir o sumar partes para formular un todo, en el presente proyecto esto ha permitido juntar partes o características para lograr definir funcionalidades concretas que posee el sistema SIGUION, además, sobre estas funcionalidades las pruebas de caja negra han sido ejecutadas.

El planteamiento de los entregables o productos incrementales (Sprint) de este proyecto han sido desarrollados siguiendo la orientación del método sintético.

#### c) Método deductivo

Este método plantea la obtención de conclusiones específicas a partir de leyes o principios generales, en el desarrollo de SIGUION, el planteamiento de diseños específicos de pruebas unitarias dentro de cada requisito funcional identificado para el proyecto, así como también, las formulaciones de conclusiones de los resultados obtenidos han sido posible por medio de la deducción.

#### d) Método inductivo

Plantea la obtención de conclusiones generales a partir de hechos particulares, este método fue utilizado para la formulación de conclusiones a los resultados obtenidos de las pruebas de caja negra en el presente proyecto.

#### e) Metodología SCRUM

Es una metodología de desarrollo ágil de software, generalmente se usa para proyectos que comprenden un ciclo de vida de 6 meses. Esta metodología propone la construcción de un producto de manera incremental, los mismos que son planificados cada cierto periodo de tiempo.

Al utilizar esta metodología se garantiza la identificación de riesgos y errores en cada incremento construido, además de la retrospectiva continua durante el proceso de desarrollo. Al finalizar el último sprint planificado se garantiza la entrega de un producto funcional e integrado.

### **2.2.2. Técnicas**

**Observación:** Esta técnica permitirá percibir los resultados de las pruebas unitarias en el desarrollo del sistema de gestión de guiones de radio “SIGUION”.

**Entrevista:** Dirigida a los directivos y personas que laboran en la Radio Crisólito Celestial 93.1 FM para obtener la información sobre el proceso de gestión de guiones de las programaciones radiales en la entidad.

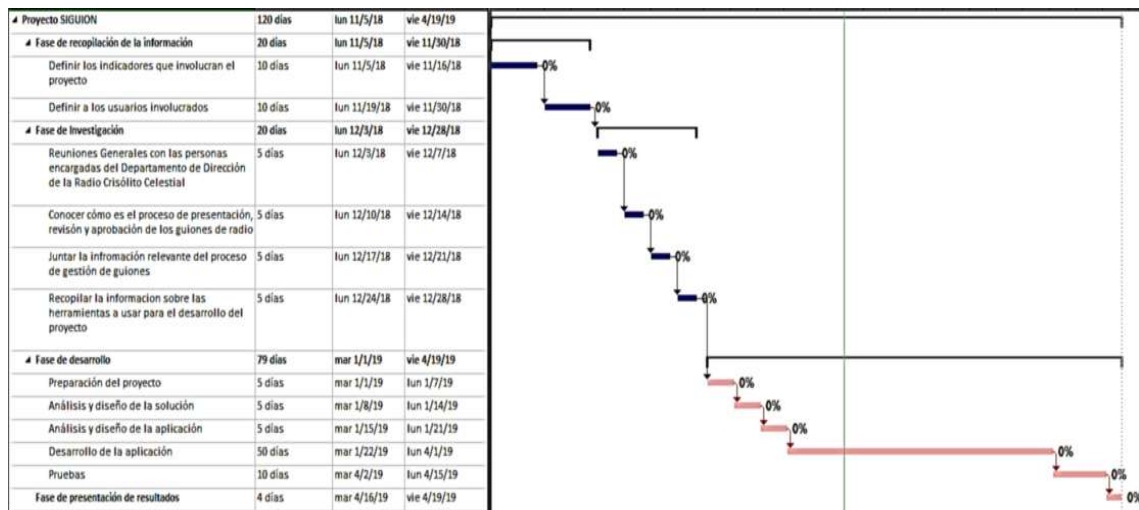
**Análisis Documental:** Con esta técnica fue posible obtener información de fuentes bibliográficas, como, libros, foros, documentos científicos y tesis relacionadas al proyecto, la misma que servirá para fundamentar teóricamente y establecer la guía metodológica del presente proyecto.

## **2.3. Fase de planificación del sistema web de gestión de guiones SIGUION**

### **2.3.1. Preparación del proyecto**

Para el desarrollo del sistema web de gestión de guiones de radio (SIGUION) con pruebas unitarias, es necesario un análisis de ingeniería en el cual se pueda representar las distantes fases que un proyecto de software requiere.

Este resultado del análisis de cada fase del proyecto con su respectivo tiempo de duración es representado, a continuación, en en la **Figura 1-2:**



**Figura 1-2:** Diagrama Gantt del Proyecto

**Realizado por:** Chela Jhonathan, 2019.

En la **Figura 1-2** se describe 4 procesos que involucran el desarrollo del proyecto, la fase de recopilación de la información, la fase de investigación, la fase de desarrollo y por último la fase de presentación de resultados, estas fases marcan el camino que junto a la metodología SCRUM permitirán obtener un producto software desarrollado con pruebas unitarias.

### 2.3.2. Reunión de planificación

Para el establecimiento de los requerimientos del sistema se establece una serie de reuniones de planificación con las personas involucradas en el proyecto software. En estos encuentros se establecen los requisitos del sistema junto con los tiempos de entrega para cada incremento, en la **Tabla 1-2** se expone el desarrollo de las actividades.

**Tabla 1-2:** Reuniones de planificación

Fecha	Asistentes	Actividades	Resultados
05-11-2018	Director de la Radio Ab. Arnulfo Manobanda Desarrollador del producto Jhonathan Chela	Recopilación de información del proceso de gestión de guiones en la radio	Determinación de los módulos del sistema.
12-11-2018	Director de la Radio Ab. Arnulfo Manobanda Locutor de la Radio Lic. Ángel Aucatoma Desarrollador del producto Jhonathan Chela	Definición de las necesidades y alcance del sistema.	Descripción del listado de requerimientos del sistema.
19-11-2018	Director de la Radio Ab. Arnulfo Manobanda Desarrollador del producto	Exposición de los requerimientos y el alcance del sistema.	Aceptación de los requerimientos y el alcance del sistema.

	Jhonathan Chela		alcance establecido para el sistema.
--	-----------------	--	--------------------------------------

**Realizado por:** Chela Jhonathan, 2019.

El alcance del sistema web de gestión de guiones establece como alcance los módulos que se detallan a continuación:

- Módulo de autenticación de usuarios: este módulo permite la autenticación del usuario del sistema, esto es posible a las credenciales de inicio de sesión establecidas para cada rol.
- Módulo de gestión de usuarios: este módulo permite el ingreso, modificación y eliminación de usuarios en el sistema, para gestionar las tareas correspondientes en el proceso de gestión de guiones de la radio.
- Módulo de gestión de programas radiales: este módulo permite el registro, actualización, y suspensión de los programas radiales emitidos en la entidad radial.
- Módulo de gestión de guiones: este módulo hace posible guardar los guiones de una programación radial, también permite ingresar notas de revisión para los guiones, la respectiva aprobación o no aprobación de los mismos.
- Módulo de reportes: este módulo permite realizar los diferentes reportes siguiendo el formato establecido y especificado en la lista del producto (Product Backlog).

### 2.3.3. Equipo SCRUM del proyecto

De acuerdo a lo establecido en la metodología de desarrollo de software ágil, los roles definidos claramente son, el Dueño del producto, el Scrum Master y el Desarrollador de la aplicación.

En la **Tabla 2-2** se puede apreciar las personas y los roles que involucran del desarrollo de SIGUIÓN:

**Tabla 2-2:** Equipo SCRUM del proyecto SIGUIÓN

Persona/Cargo	Rol	Responsabilidad
Director de la Radio Crisólito Celestial 93.1 FM	Dueño del producto.	Definir los ítems que representan las especificaciones del sistema.
Dr. Julio Santillán	Scrum Master	Responsable de la asesoría y seguimiento del proyecto.
Jhonathan Chela	Programador y tester del proyecto.	Codificar, diseñar y ejecutar las pruebas unitarias en el software.

**Realizado por:** Chela Jhonathan, 2018.

### 2.3.4. Tipos y roles del proyecto

Para el desarrollo del sistema SIGUION se logró definir tres tipos de roles de usuarios, los mismos que permitirán cumplir con los requerimientos establecidos, cada tipo de usuario cumple con funciones específicas, en la **Tabla 3-2** se detalla cada tipo de usuario y el rol que cumple.

**Tabla 3-2:** Tipos de usuario y su rol

<b>Tipo de Usuario</b>	<b>Rol</b>
Administrador del sistema	<ul style="list-style-type: none"><li>- Gestionar usuarios del sistema.</li><li>- Gestionar los programas radiales.</li></ul>
Director de la radio	<ul style="list-style-type: none"><li>- Visualizar el listado de guiones por fecha.</li><li>- Revisar un guion de la radio para su posterior aprobación.</li><li>- Aprobar un guion de un programa radial.</li><li>- Cancelar el guion de una programación radial.</li><li>- Ingresar sugerencias de cambios a los guiones de la radio.</li><li>- Reporte en lista de guiones No aprobados.</li><li>- Reporte en lista de guiones aprobados.</li><li>- Reporte en lista de guiones revisados.</li><li>- Reporte de usuarios.</li><li>- Reporte de programas radiales.</li><li>- Reporte en lista de guiones No aprobados por fecha de registro.</li><li>- Reporte en lista de guiones aprobados por fecha de registro.</li></ul>
Locutor	<ul style="list-style-type: none"><li>- Gestionar los guiones del programa radial.</li><li>- Visualizar la sugerencia hecha al guion por el director.</li><li>- Realizar cambios al guion.</li><li>- Reporte en formato pdf. de un guion radial.</li></ul>

**Realizado por:** Chela Jhonathan, 2018.

### 2.3.5. Lista del producto (Product Backlog)

La lista del producto contiene el conjunto de historias de usuario en las que se detalla los requerimientos establecidos para el desarrollo del sistema SIGUION. Para realizar el establecimiento definitivo de los requerimientos del sistema se realizó varias reuniones con el cliente analizando las necesidades de la institución radial. Para conocer las tablas de las historias de usuario, ver el **ANEXO A**.

Para poder definir la lista de producto que contiene los requerimientos funcionales y técnicos se utilizó puntos de estimación conforme a la técnica denominada “Talla de la camiseta”.

De acuerdo a la técnica de estimación de puntos antes mencionada se definen las categorías con las letras: XL, L, S, XS, XXS, XXXS, donde cada talla significa la duración del Sprint o fracción del mismo, adicionalmente cada punto estimado está representado por una hora de trabajo.



**Tabla 4-2:** Talla de la camiseta del proyecto

Tallas	Puntos Estimados
XL	60
L	30
S	15
XS	12
XXS	6
XXXXS	3

Realizado por: Chela Jhonathan, 2019.

En la **Tabla 4-2** se puede apreciar cada talla con su respectiva equivalencia, es importante recalcar que cada semana de trabajo representa cinco días con seis horas laborables.

A continuación, en la tabla **Tabla 5-2** se expone la lista del producto que involucra el desarrollo del proyecto SIGUION. Se establece las nomenclaturas **HT** para historias técnicas y **HU** para las historias de usuario, cada ítem tiene su estimación y prioridad.

**Tabla 5-2:** Lista del producto del proyecto

Id	Lista del Producto	Puntos Estimados	Prioridad
<b>HT-01</b>	Como desarrollador necesito realizar una recopilación de requerimientos del sistema.	L	Alta.
<b>HT-02</b>	Como desarrollador necesito realizar la planificación de los entregables del sistema.	S	Alta.
<b>HT-03</b>	Como desarrollador necesito definir las historias de usuario del sistema.	S	Alta.
<b>HT-04</b>	Como desarrollador necesito realizar la planificación de las pruebas unitarias automatizadas.	L	Alta.
<b>HT-05</b>	Como desarrollador necesito diseñar la arquitectura del sistema.	XXS	Alta.
<b>HT-06</b>	Como desarrollador necesito diseñar las interfaces de usuario.	XXS	Alta.
<b>HT-07</b>	Como desarrollador necesito diseñar la base de datos.	XS	Alta.
<b>HT-08</b>	Como desarrollador necesito realizar la implementación de la base de datos.	XXS	Alta.
<b>HT-09</b>	Como desarrollador necesita realizar la conexión a la base de datos.	XXS	Alta.
<b>HU-01</b>	Como usuario necesito autenticar mis credenciales para ingresar al sistema.	XS	Alta.
<b>HT-10</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-01	XXS	Alta.
<b>HU-02</b>	Como administrador del sistema necesito ingresar nuevos usuarios al sistema.	S	Media.
<b>HT-11</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-02	XXS	Alta.

<b>HU-03</b>	Como administrador del sistema necesito ingresar programas radiales al sistema.	S	Media.
<b>HT-12</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-03	XXS	Alta.
<b>HU-04</b>	Como locutor necesito ingresar guiones de programas radiales.	S	Media.
<b>HT-13</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-04	XXS	Alta.
<b>HU-05</b>	Como director necesito visualizar el guion de un programa radial.	XS	Media.
<b>HT-14</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-05	XXS	Alta.
<b>HU-06</b>	Como director necesito realizar la aprobación del guion de un programa radial.	XS	Media.
<b>HT-15</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-06	XXS	Alta.
<b>HT-16</b>	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	XXS	Alta.
<b>HT-17</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-16.	XXS	Alta.
<b>HU-07</b>	Como director necesito ingresar el estado de No Aprobado a un guion del programa radial.	XXS	Baja.
<b>HT-18</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-07	XXS	Alta.
<b>HU-08</b>	Como director necesito ingresar el estado de Revisado a un guion del programa radial.	S	Media.
<b>HT-19</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-08	XXS	Alta.
<b>HU-09</b>	Como director necesito ingresar las sugerencias respectivas al guion de un programa de la radio.	S	Media.
<b>HT-20</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-09	XXS	Alta.
<b>HU-10</b>	Como locutor necesito visualizar las sugerencias del guion de un programa radio.	S	Alta.
<b>HT-21</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-10	XXS	Alta.
<b>HT-22</b>	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	XXS	Alta.
<b>HT-23</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-22.	XXS	Alta.
<b>HU-11</b>	Como locutor necesito modificar el guion de un programa de radio.	S	Media.
<b>HT-24</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-11	XXS	Alta.

<b>HU-12</b>	Como locutor necesito realizar los respectivos cambios al guion de un programa de radio.	XS	Media.
<b>HT-25</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-12	XXS	Alta.
<b>HU-13</b>	Como locutor necesito visualizar el estado del guion.	S	Media.
<b>HT-26</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-13	XXS	Alta.
<b>HU-14</b>	Como administrador del sistema necesito modificar los programas radiales.	S	Baja.
<b>HT-27</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-14	XXS	Alta.
<b>HU-15</b>	Como administrador del sistema necesito eliminar los programas radiales.	S	Baja.
<b>HT-28</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-15	XXS	Alta.
<b>HU-16</b>	Como administrador del sistema necesito modificar la información de los usuarios.	S	Media.
<b>HT-29</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-16	XXS	Alta.
<b>HT-30</b>	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	XXS	Alta.
<b>HT-31</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-30.	XXS	Alta.
<b>HU-17</b>	Como administrador del sistema necesito eliminar los usuarios ingresados.	S	Media.
<b>HT-32</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-17.	XXS	Alta.
<b>HU-18</b>	Como locutor necesito eliminar un guion del programa radial.	XS	Baja.
<b>HT-33</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-18.	XXS	Alta.
<b>HU-19</b>	Como director necesito realizar el reporte de guiones no aprobados en la semana en formato pdf.	S	Media.
<b>HT-34</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-19	XXS	Alta.
<b>HU-20</b>	Como director necesito realizar el reporte de los guiones aprobados en la semana en formato pdf.	S	Media.
<b>HT-35</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-20	XXS	Alta.
<b>HU-21</b>	Como director necesito realizar el reporte de los guiones revisados en la semana en formato pdf.	S	Media.
<b>HT-36</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-21	XXS	Alta.
<b>HU-22</b>	Como director del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	XXS	Alta.

<b>HT-37</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-22	S	Media.
<b>HU-23</b>	Como director del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	XXS	Alta.
<b>HT-38</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-23	S	Media.
<b>HT-39</b>	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	XXS	Alta.
<b>HT-40</b>	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-39	XXS	Alta.

**Realizado por:** Chela Jhonathan, 2019.

### 2.3.6. Gestión de riesgos

La gestión de riesgos es el proceso mediante el cual se realiza un análisis de eventos futuros que tienen un grado de probabilidad e impacto que afecta de manera positiva o negativa al proyecto en el que se está trabajando.

El análisis de riesgos permite realizar un plan de gestión de los mismos con el objetivo de prevenir y corregir de tal modo que el proyecto se desarrolle de una buena manera.

Dentro de la identificación de los riesgos en los proyectos de software se encuentran tres, a continuación, el nombre de cada uno:

- **RN:** Riesgo de Negocio.
- **RP:** Riesgo del proyecto.
- **RT:** Riesgo Técnico.

**Tabla 6-2:** Gestión de Riesgos de SIGUION

<b>Id</b>	<b>Riesgo</b>	<b>Categoría</b>	<b>Consecuencia</b>
<b>R1</b>	Falta de compromiso por parte del usuario en el establecimiento de las bases para el desarrollo del sistema.	RN	<ul style="list-style-type: none"> <li>• No satisfacer las necesidades del proyecto.</li> <li>• Mal diseño del proyecto.</li> <li>• Paro temporal del desarrollo del sistema.</li> <li>• No cumplir con la funcionalidad del sistema.</li> <li>• Información inconsistente en el desarrollo.</li> </ul>
<b>R2</b>	Recolección de requerimientos incompletos o ambiguos.	RT	<ul style="list-style-type: none"> <li>• Mal diseño del proyecto.</li> <li>• Re-planificación periódica del sistema.</li> </ul>

			<ul style="list-style-type: none"> <li>• No cumplir con las funcionalidades del sistema.</li> <li>• Sistema funcional pero que no cumple con el objetivo establecido.</li> </ul>
<b>R3</b>	Modificación continua de requerimientos.	RP	<ul style="list-style-type: none"> <li>• Re-planificación constante de la lista de producto.</li> <li>• Funcionalidades ambiguas sobre el sistema.</li> <li>• Paro temporal en el desarrollo del sistema.</li> <li>• Retraso en el proyecto.</li> </ul>
<b>R4</b>	Incorrecta definición de la base de datos.	RT	<ul style="list-style-type: none"> <li>• Paro temporal en el desarrollo del sistema.</li> <li>• Retraso en el cumplimiento de los incrementos del proyecto.</li> <li>• Pérdida de información.</li> </ul>
<b>R5</b>	Subestimación del tamaño de la aplicación.	RT	<ul style="list-style-type: none"> <li>• Incremento de prórrogas en el tiempo de entrega.</li> <li>• Insatisfacción del cliente.</li> <li>• Incumplimiento con los plazos de entrega.</li> <li>• Re-planificación constante del proyecto.</li> </ul>
<b>R6</b>	Subestimación de costos en el presupuesto del proyecto.	RT	<ul style="list-style-type: none"> <li>• Pérdidas económicas.</li> <li>• Cancelación del proyecto.</li> </ul>
<b>R7</b>	Diseño de interfaces incompleto.	RT	<ul style="list-style-type: none"> <li>• Retraso en la entrega del proyecto.</li> <li>• Re-planificación del tiempo de entrega.</li> <li>• Insatisfacción del cliente.</li> </ul>
<b>R8</b>	Falta de especificación de la arquitectura lógica del sistema.	RT	<ul style="list-style-type: none"> <li>• Incremento en el tiempo de desarrollo.</li> <li>• Retraso en la entrega del proyecto.</li> <li>• Re-planificación del proyecto.</li> </ul>
<b>R9</b>	Retrasos en el cumplimiento de tiempos del sprint.	RP	<ul style="list-style-type: none"> <li>• Insatisfacción en el cliente.</li> <li>• Incremento del tiempo total para culminar el proyecto.</li> </ul>
<b>R10</b>	Falta de conocimiento y experiencia sobre las tareas asignadas y las herramientas a utilizar.	RT	<ul style="list-style-type: none"> <li>• Retraso en el proyecto.</li> <li>• Re-planificación en el proyecto.</li> </ul>
<b>R11</b>	Pérdida de backups o archivos de respaldo del proyecto.	RP	<ul style="list-style-type: none"> <li>• Falta de respaldo del sistema en caso de pérdida total del proyecto.</li> <li>• Cancelación del proyecto.</li> </ul>
<b>R12</b>	Alcance de las pruebas unitarias automatizadas no definidas correctamente.	RT	<ul style="list-style-type: none"> <li>• No probar la funcionalidad completa de un ítem planificado.</li> <li>• Volver a realizar las pruebas.</li> </ul>

			<ul style="list-style-type: none"> <li>• Consumir más tiempo de lo planificado para las pruebas.</li> </ul>
<b>R13</b>	Inmadurez en la priorización sobre la ejecución de pruebas.	RT	<ul style="list-style-type: none"> <li>• Re-planificación de las pruebas dentro del proyecto.</li> <li>• Paro temporal en el desarrollo del proyecto.</li> </ul>
<b>R14</b>	Excesivo uso del tiempo en la reparación de defectos de las pruebas unitarias automatizadas.	RT	<ul style="list-style-type: none"> <li>• Paro temporal en el desarrollo del proyecto.</li> <li>• Retraso en la entrega del proyecto.</li> </ul>
<b>R15</b>	Retraso en la realización de pruebas por errores después del despliegue del sistema.	RT	<ul style="list-style-type: none"> <li>• Paro temporal en el desarrollo del proyecto.</li> <li>• Retraso en la entrega del proyecto.</li> <li>• Falta de integración de las unidades que componen una versión del sistema.</li> </ul>

**Realizado por:** Chela Jhonathan, 2019.

### 2.3.6.1. Categorización de riesgos

Luego del análisis de los posibles riesgos que el proyecto involucra es de vital importancia la categorización de los mismos, se utilizó los parámetros de determinación de la probabilidad de ocurrencia, el impacto y la exposición del riesgo, a continuación, se detalla cada uno de ellos:

#### **Determinación de la probabilidad:**

Para determinar la probabilidad de ocurrencia del riesgo se utilizó los parámetros que se exponen en la **Tabla 7-2:**

**Tabla 7-2:** Parámetros de probabilidad

<b>Rango de probabilidades</b>	<b>Descripción</b>	<b>Valor</b>
1%-33%	Baja	1
34%-67%	Media	2
68%-99%	Alta	3

**Realizado por:** Chela Jhonathan, 2019.

#### **Determinación del impacto:**

Para determinar el impacto del riesgo se utilizó los parámetros que se exponen en la **Tabla 8-2:**

**Tabla 8-2:** Parámetros de impacto

<b>Rango de probabilidades</b>	<b>Descripción</b>	<b>Valor</b>
1%-33%	Baja	1
34%-67%	Media	2
68%-99%	Alta	3

**Realizado por:** Chela Jhonathan, 2019.

### Determinación de la exposición del riesgo:

Para determinar la exposición de los riesgos en el proyecto se utilizó los intervalos que se describen en la **Tabla 9-2**.

**Tabla 9-2:** Rangos de exposición del riesgo

Rango de probabilidades	Valor	Color
Alta	$\geq 6$	Rojo
Media	3-5	Amarillo
Baja	1-2	Verde

Realizado por: Chela Jhonathan, 2019.

#### 2.3.6.2. Priorización de riesgos

La priorización de riesgos es el proceso en el cual se ordena de manera jerárquica los riesgos determinados en el proyecto en desarrollo. Este proceso es realizado con el objetivo de tener una noción cuales son los eventos futuros que pueden poner en riesgo el avance del proyecto.

En la **Tabla 10-2** se puede apreciar el orden en el cual los riesgos identificados para el proyecto, se realiza un orden de los riesgos con mayor valor de probabilidad impacto y exposición, de mayor a menor son identificados con el color rojo, amarillo y verde respectivamente.

**Tabla 10-2:** Priorización de Riesgos

Id_Riesgo	Probabilidad			Impacto		Exposición	
	%	Prob.	Valor	Impacto	Valor	Exposición	Valor
R2	30	BAJA	1	MEDIA	2	ALTA	8
R9	50	MEDIA	2	ALTA	3	ALTA	6
R12	60	MEDIA	2	ALTA	3	ALTA	6
R13	70	ALTA	3	ALTA	3	ALTA	8
R15	30	BAJA	1	ALTA	3	ALTA	6
R3	30	BAJA	1	ALTA	3	MEDIA	4
R4	25	BAJA	1	ALTA	3	MEDIA	5
R5	20	BAJA	1	ALTA	3	MEDIA	4
R8	33	BAJA	1	ALTA	3	MEDIA	4

<b>R10</b>	50	MEDIA	2	ALTA	3	MEDIA	5
<b>R11</b>	50	MEDIA	2	ALTA	3	MEDIA	4
<b>R1</b>	30	BAJA	1	MEDIA	2	BAJA	2
<b>R6</b>	20	BAJA	1	BAJA	1	BAJA	2
<b>R7</b>	25	BAJA	1	MEDIA	2	BAJA	2
<b>R14</b>	25	BAJA	1	ALTA	3	BAJA	2

Realizado por: Chela Jhonathan, 2019.

### 2.3.6.3. Hojas de gestión de riesgo

Las hojas de gestión de riesgo son productos formales establecidos para cada riesgo identificado, este documento permite realizar la reducción, supervisión y gestión respectiva.

**Tabla 11-2:** Hoja de gestión de riesgo

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R2		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Media <b>Valor:</b> 2	<b>Exposición:</b> Alta <b>Valor:</b> 8	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Recolección de requerimientos incompletos o ambiguos.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Falta de compromiso con el cliente para las reuniones de recolección de información. Información incompleta en los requisitos expuestos. Búsqueda de información parcial con respecto a las necesidades del sistema.			
<b>Consecuencias:</b> Mal diseño del proyecto. Re-planificación periódica del sistema. No cumplir con las funcionalidades del sistema.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Revisión de la documentación necesaria para definir correctamente el alcance de los requisitos. Revisión continua de los requerimientos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Revisión continua de los requerimientos con los stakeholders involucrados en el proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

Realizado por: Chela Jhonathan, 2019.

En la **Tabla 11-2** se muestra el modelo de la tabla de gestión de riesgos que se utiliza en este proyecto, a continuación, se expone cada uno de los ítems de la tabla:

- **Id del Riesgo:** este ítem permite identificar el riesgo, en este caso el riesgo se identifica con la letra mayúscula R y el número correspondiente.



- Fecha: describe la fecha en la cual se está realizando la gestión del riesgo.
- Probabilidad: describe la categoría y el valor de la probabilidad de ocurrencia de un riesgo.
- Impacto: describe la categoría y el valor del impacto que tiene el riesgo identificado.
- Exposición: describe el valor de la exposición del riesgo identificado y descrito respectivamente.
- Prioridad: describe el nivel de prioridad establecido para gestionar el riesgo identificado.
- Descripción: expone la identificación del nombre del riesgo respectivo.
- Refinamiento: en este apartado se expone las causas y las consecuencias que conllevan el riesgo identificado.
- Reducción: describe los protocolos establecidos para reducir el riesgo.
- Supervisión: describe argumentos para realizar la supervisión del riesgo.
- Gestión: describe el protocolo correspondiente para realizar la gestión del riesgo respectivo.
- Estado Actual: determina si el riesgo está actualmente en fase de reducción iniciada, fase de supervisión iniciada o gestionando el riesgo.
- Responsable: expone el nombre del responsable de la gestión del riesgo.

Todas las hojas de gestión de los riesgos identificados se adjuntan en el **ANEXO B**.

## **2.4. Fase de desarrollo del sistema web de gestión de guiones SIGUION**

### ***2.4.1. Estándar de codificación***

En la fase de desarrollo del sistema, es importante fundamentar los pilares y las bases sobre las cuales el desarrollo del proyecto debe seguir una trayectoria correcta y ordenada. Uno de los aspectos fundamentales a considerar siempre es el estándar de codificación del proyecto.

El estándar de codificación permite establecer un orden en la escritura del código del sistema, esto facilita la lectura y futuros posibles cambios que el mismo puede sufrir, ya sea para implementar nuevas funcionalidades o actualizar funcionalidades existentes.

El estándar CamelCase es el más utilizado como guía de codificación de sistemas software, este estándar existe en sus dos variantes UpperCamelCase, y lowerCamelCase. Para este proyecto se utiliza la segunda variante expuesta.

Esta variante en si establece como lineamiento general que un nombre empieza con una letra minúscula y en el caso requerido de identificación la primera letra debe distinguirse en mayúscula, en la **Tabla 12-2** se exponen los lineamientos generales para de este proyecto.

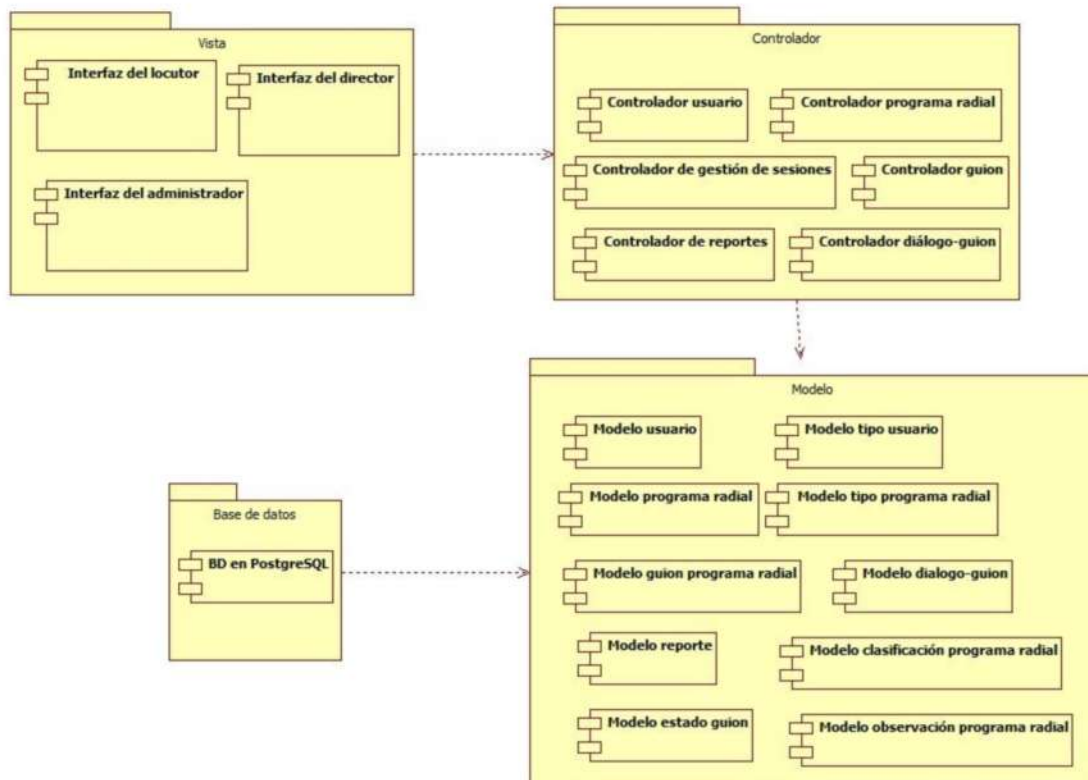
**Tabla 12-2:** Estándar de codificación del proyecto

<b>Categoría</b>	<b>Definición</b>	<b>Ejemplo</b>
Métodos Funciones Clases	Se nombran cada una de las categorías clasificadas en este ítem de la siguiente manera: la primera letra que identifican a la categoría seguido de la segunda palabra que empieza con una letra mayúscula.	Método: autenticarUsuario() Función: listarUsuarioporCedula() Clase: claseUsuario
Variables	Dentro del estándar lowerCamelCase, las variables deben empezar con una la primera letra en minúsculas seguidos del identificativo correspondiente con la primera letra en mayúscula.	int idUsuario string nombreUsuario
Paquetes	La misma propuesta para nombrar a los paquetes que serán usados e invocados en el caso que se los requiera.	<paqueteAccesodatos>
Botones y otros similares	Para los botones o elementos similares, el nombre empieza con la primera letra en minúscula, pero utilizando acrónimos que identifiquen al elemento seguido de la segunda palabra que inicia con mayúscula.	bntActualizarUsuario lblNombreUsuario cbxProgramaRadial

**Realizado por:** Chela Jhonathan, 2019.

#### **2.4.2. Arquitectura del sistema**

El sistema SIGUION está construido basado en la arquitectura cliente – servidor, debido a que es una aplicación web, además se sigue el patrón modelo, vista, controlador (MVC), con modelo de programación en n capas para permitir la escalabilidad del sistema.



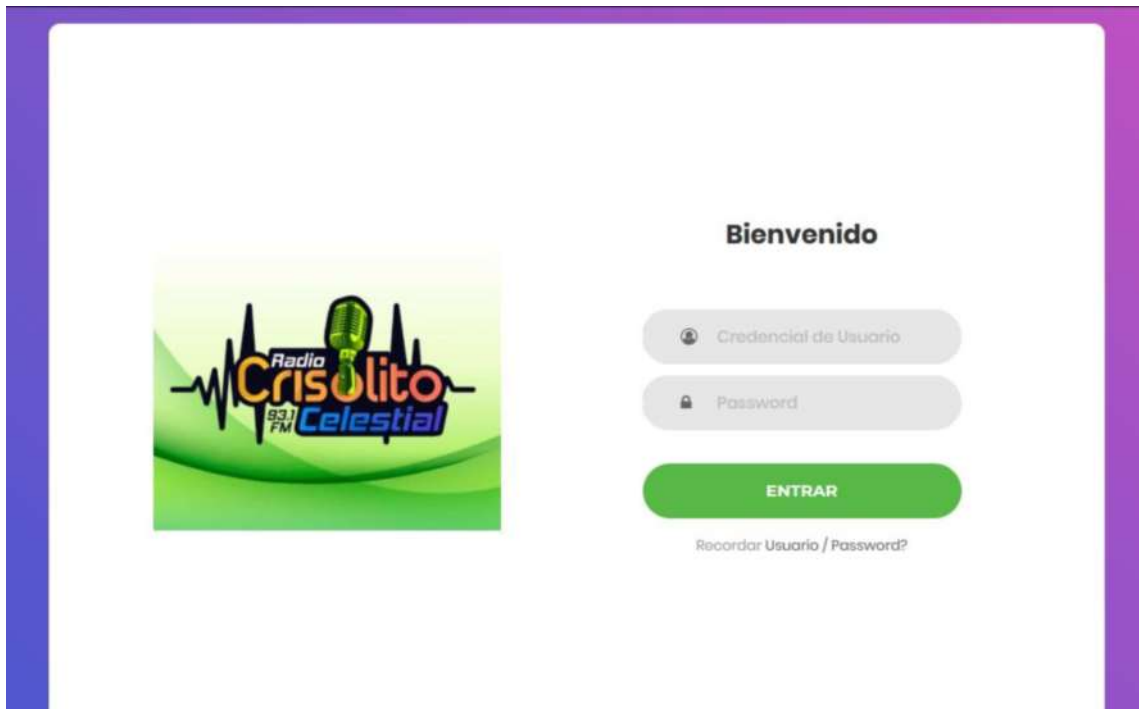
**Figura 2-2:** Arquitectura de SIGUIÓN

**Realizado por:** Chela Jhonathan, 2019.

La **Figura 2-2** expone el esquema general del sistema web de gestión de guiones para la radio Crisólito Celestial, este sistema consta de una base de datos almacenada en un servidor de base de datos con conexión a internet, permitiendo gestionar la información con respecto al proceso que se implementará en el sistema web. El servidor de aplicaciones alojará la aplicación desarrollada en el lenguaje Java, esto hace referencia al controlador y modelo del sistema, utilizando la tecnología Java Server Face junto con el framework Bootstrap la vista del sistema cumple con el requisito de web responsiva, adaptable a cualquier dispositivo inteligente, smartphones, tablets, computadoras, etc. Esto permite, al usuario la interacción con el sistema desde cualquier dispositivo, en cualquier parte del mundo y en cualquier hora del día.

### **2.4.3. Diseño de la interfaz de usuario**

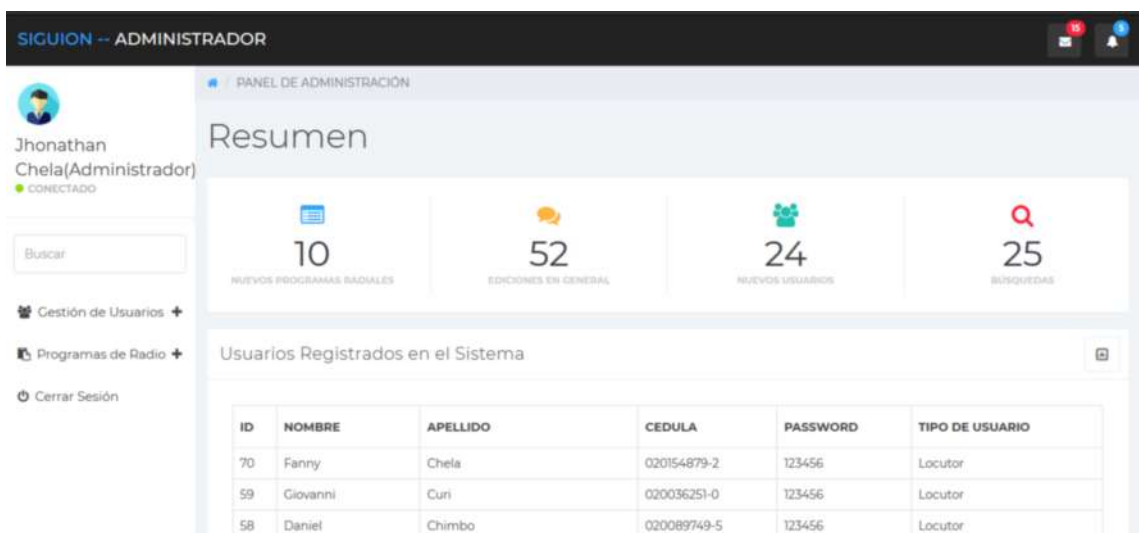
La interfaz de usuario es el conjunto de elementos gráficos (colores, iconos, botones, entre otros) que hace posible la interacción del usuario con el sistema web, permitiendo realizar tareas y cumplir objetivos para los cuales el sistema web ha sido construido.



**Figura 3-2:** Pantalla de inicio de sesión de SIGUION

**Realizado por:** Chela Jhonathan, 2019.

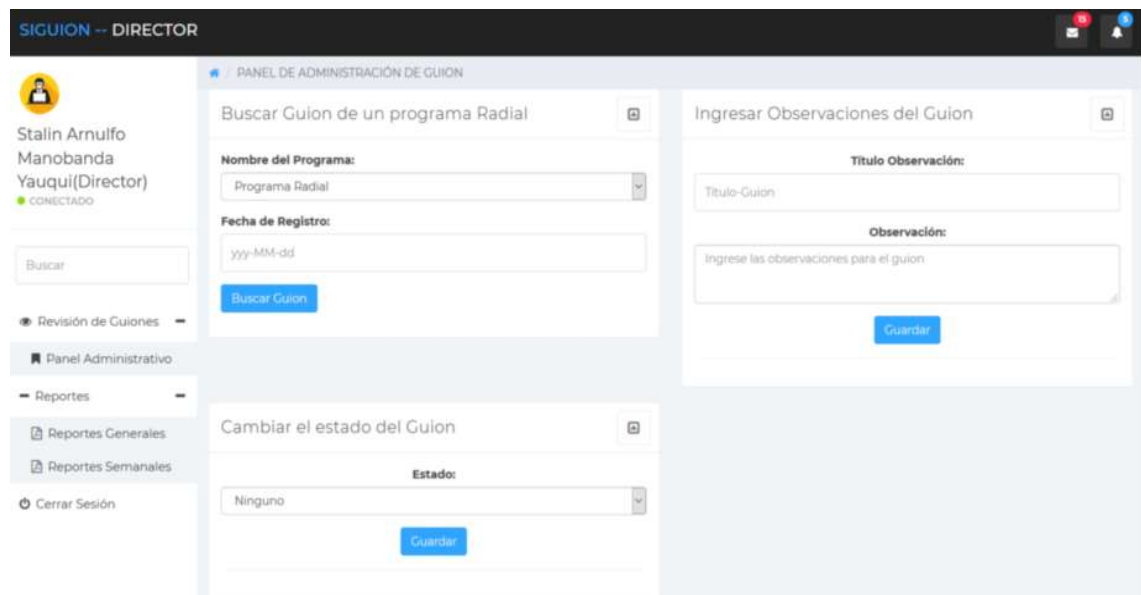
La **Figura 3-2** muestra la pantalla de inicio de sesión al sistema de gestión de guiones, esta es la pantalla principal y por defecto que aparecerá al ingresar a la página de inicio. Para ingresar al sistema debe digitar sus correspondientes credenciales, que, al ser verificadas, se tendrá el acceso a diferentes paneles de acuerdo al rol de usuario.



**Figura 4-2:** Pantalla de gestión del administrador

**Realizado por:** Chela Jhonathan, 2019.

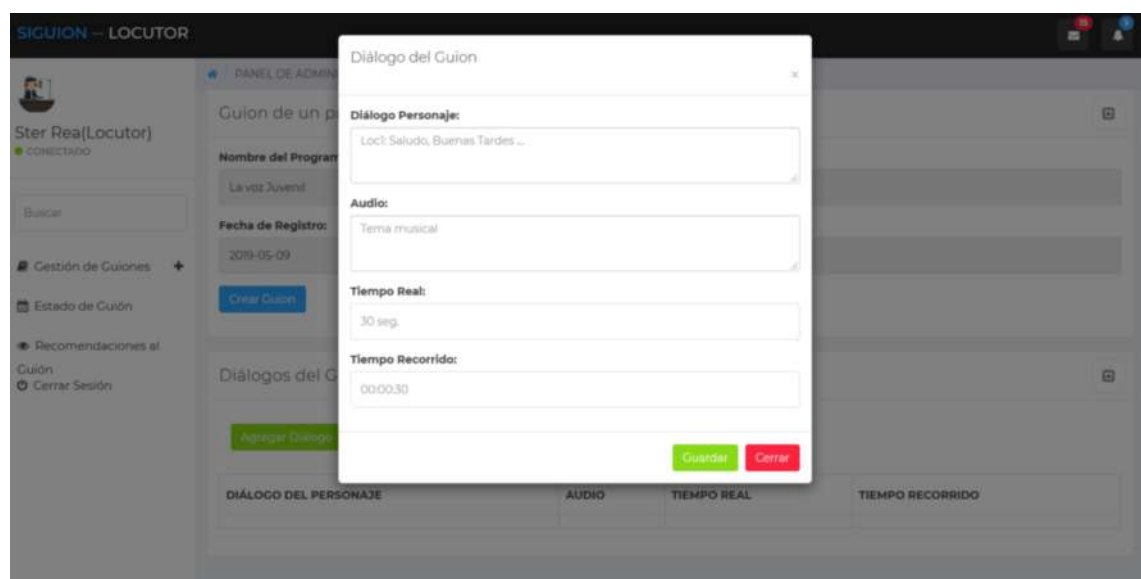
La **Figura 4-2** muestra la pantalla con las opciones propias del administrador del sistema, se puede ver la opción de ingresar usuarios al sistema, esta pantalla permite registrar usuarios nuevos, mediante una interacción con modales característicos de bootstrap.



**Figura 5-2:** Pantalla de gestión del director

**Realizado por:** Chela Jhonathan, 2019.

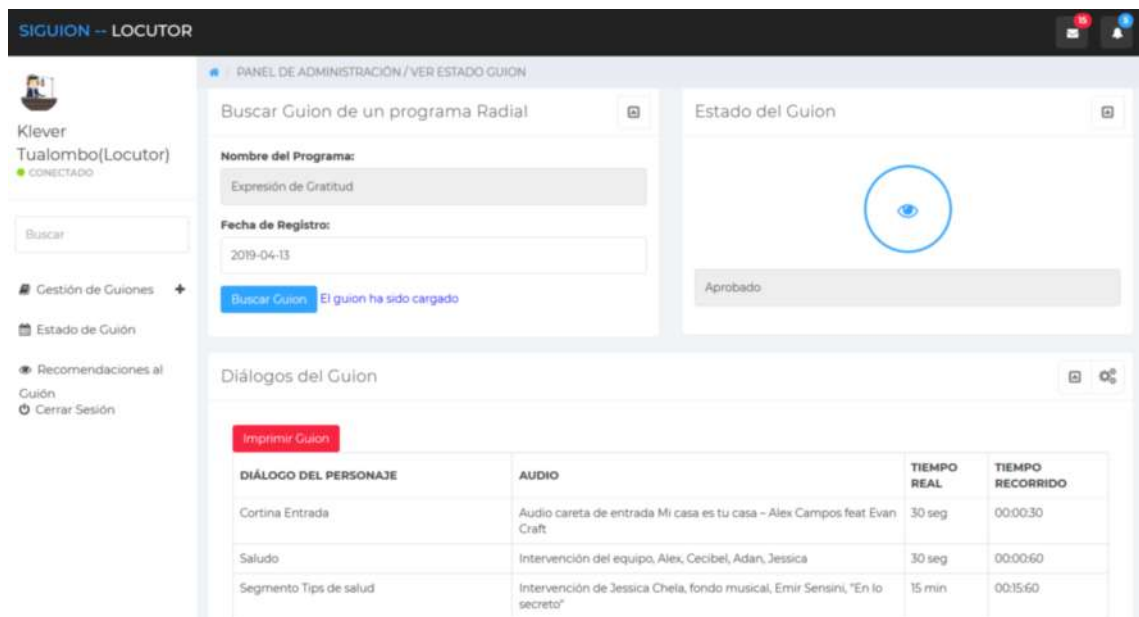
La **Figura 5-2** muestra la pantalla de administración con las opciones respectivas que pertenecen al usuario denominado director. El panel de administración permite la gestión de guiones de radio, primero, se parecían cuadros de opciones, buscar un guion, ingresar observaciones de un guion y cambiar el estado del guion.



**Figura 6-2:** Pantalla de gestión del locutor

**Realizado por:** Chela Jhonathan, 2019.

En la **Figura 6-2** se puede apreciar la pantalla del sistema de gestión para el usuario denominado Locutor, se puede observar el modelo diseñado para ingresar guiones de radio, además en la parte izquierda se puede apreciar las opciones propias de locutor de la entidad radial.



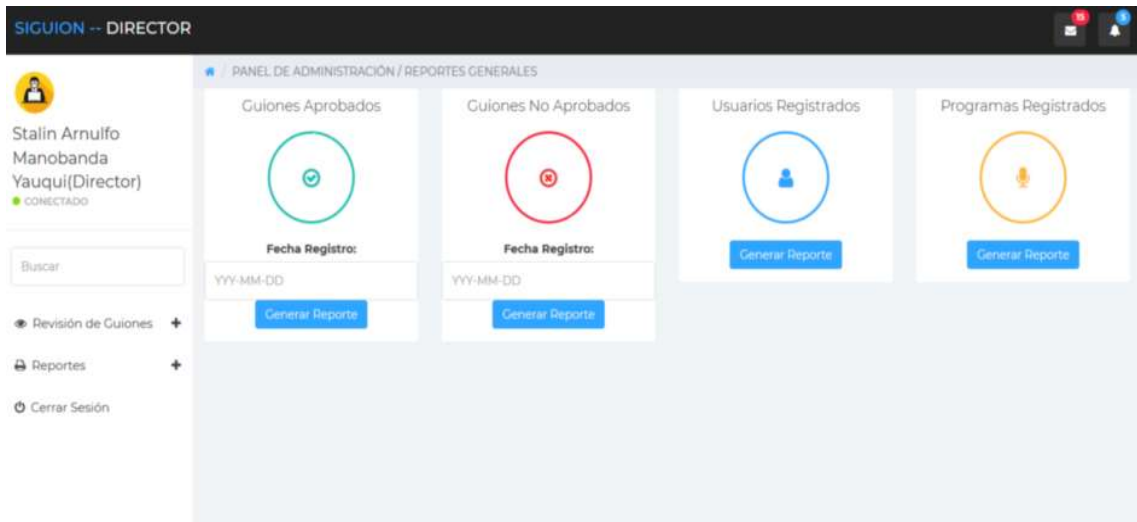
**Figura 7-2:** Pantalla de revisión del guion por el locutor

Realizado por: Chela Jhonathan, 2019.



**Figura 8-2:** Reporte pdf de un guion aprobado

Realizado por: Chela Jhonathan, 2019.



**Figura 9-2:** Panel de Reportes del sistema

**Realizado por:** Chela Jhonathan, 2019.

Nombre del Programa	Estado	Clasificación	Descripción	Tipo	Descripción	Día Transmisión	Hora Transmisión	Fecha Registro	Fecha Aprobación
Expresión de Gratitud	Aprobado	A	Apto para todo público	E	Entretención	Domingo	18:00 - 20:00	2019-04-13	2019-04-30

Abg Arnulfo Manobanda  
Director

Fecha de emisión del reporte: 5/9/19 1:30 PM

**Figura 10-2:** Reporte General de guiones Aprobados

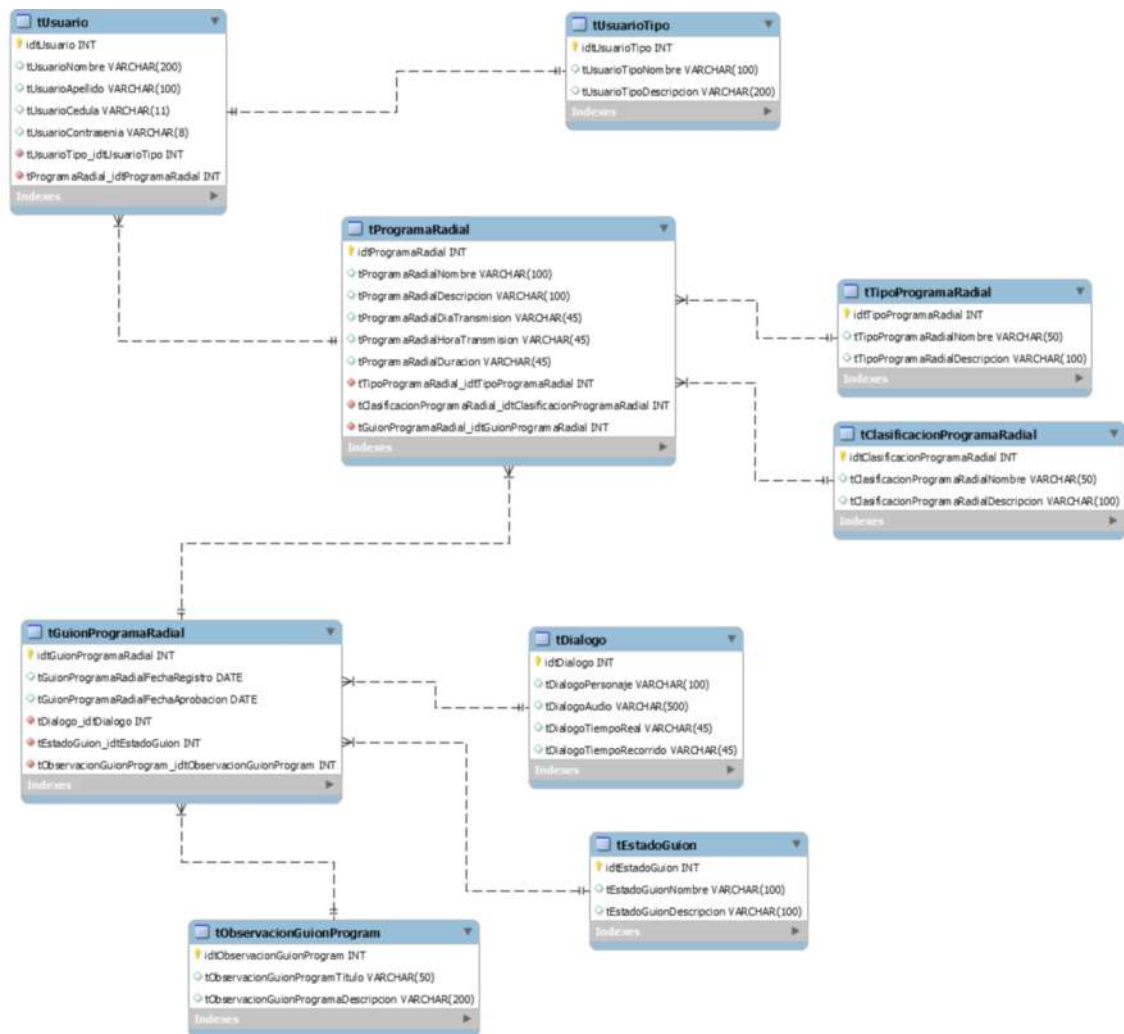
**Realizado por:** Chela Jhonathan, 2019.

En este proyecto denominado SIGUIÓN, las especificaciones del cliente han sido la construcción de una interfaz adaptable a cualquier dispositivo por el cual el sistema pueda ser invocado, por lo que el diseño es web responsive basado en el framework bootstrap, dichas interfaces se pueden visualizar a continuación, las pantallas restantes se encuentran en el manual de usuario del ANEXO C.

#### 2.4.4. Diseño de la base de datos

La base de datos permite almacenar grandes cantidades de datos, por lo general para sistemas como SIGUIÓN, las bases de datos más utilizadas son las relacionales, este tipo de estructura es una abstracción que ha sido implementada físicamente a través de un sistema de gestión.

La base de datos de SIGUIÓN luego de realizar el análisis y diseño ha sido posible la implementación física en el sistema PostgreSQL, con un total de 10 tablas principales, las mismas que son expuestas en la **Figura 11-2**.



**Figura 11-2:** Base de datos de SIGUIÓN

Realizado por: Chela Jhonathan, 2019.



### 2.4.5. Diccionario de datos

El diccionario de datos muestra las características propias de cada tabla involucrada en la base de datos para el proyecto SIGUION. Es importante recalcar que el mismo fue generado automáticamente por el *plugin* generador de MySQL Workbench.

Cada tabla contiene los siguientes campos:

- Column name: hace referencia al nombre de las columnas de una tabla.
- DataType: permite ver el tipo de dato que contiene una tabla.
- PK: indica si es una clave primaria.
- NN: hace referencia a si el campo es permitido que guarde un valor nulo.
- UQ: determina que un valor debe ser único.
- BIN: indica si el valor del campo es binario.
- UN: tipo de dato sin firmar.
- ZF: indica que contiene la característica de que si es un valor numérico se tiene que llenar con 0.
- AI: determina si el valor es autoincremental.
- Default: indica algún valor por defecto.
- Comment: permite ver si existe algún comentario propio para el campo.

Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
idtUsuario	INT	✓	✓							
tUsuarioNombre	VARCHAR(200)									
tUsuarioApellido	VARCHAR(100)									
tUsuarioCedula	VARCHAR(11)									
tUsuarioContrasenia	VARCHAR(8)									
tUsuarioTipo_idtUsuarioTipo	INT		✓							

**Figura 12-2:** Diccionario de la tabla tUsuario

**Realizado por:** Chela Jhonathan, 2019.

La **Figura 12-2** muestra el diccionario de datos de la tabla Usuario, indicado los nombres de las columnas con los tipos y las especificaciones antes descritas.

Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
idtClasificacionProgramaRadial	INT	✓	✓							
tClasificacionProgramaRadialNombre	VARCHAR(50)									
tClasificacionProgramaRadialDescripcion	VARCHAR(100)									

**Figura 13-2:** Diccionario de la tabla tClasificación\_Programa\_Radial

**Realizado por:** Chela Jhonathan, 2019.


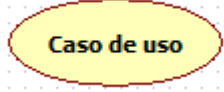

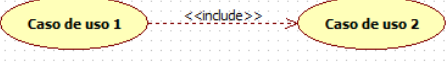

El diccionario de datos de este presente proyecto puede ser visualizado en el **ANEXO D**.

#### 2.4.6. Diagrama de casos de uso

En este proyecto se describe los procesos que los tres roles principales de SIGUION cumple, para este efecto se hace uso del lenguaje de modelado unificado (UML). Específicamente en este apartado se presenta los diagramas de caso de uso de los usuarios del sistema, administrador, director y locutor.

El diagrama de caso de uso describe las acciones que SIGUION debe cumplir, a continuación, se expone la simbología básica utilizada.

**Tabla 13-2:** Simbología UML de SIGUION

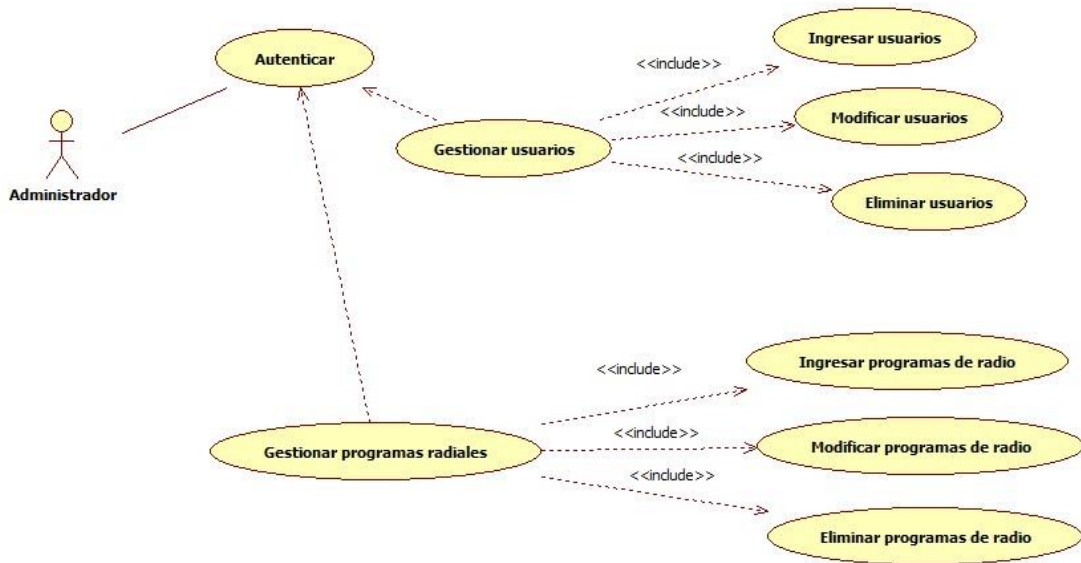
Símbolo	Nombre	Descripción
	Actor	Permite definir y representar a los roles de usuario involucrados en el sistema.
	Caso de uso	Representa una acción que el sistema realiza.
	Relación de asociación	Representa la relación de asociación entre el actor y un caso de uso.
	Relación de inclusión	Permite describir que una acción o caso de uso incluye también otra acción.
	Relación de dependencia	Describe que un proceso o caso de usos depende de una acción que lo precede.

Realizado por: Chela Jhonathan, 2019.

##### 2.4.6.1. Procesos del administrador del sistema

Este diagrama de casos de uso describe el comportamiento que el sistema SIGUION debe tener cuando el administrador del sistema sea permitido a acceder.

En la **Figura 14-2** se expone el diagrama de caso de uso del administrador del sistema.



**Figura 14-2:** Diagrama del administrador del sistema

Realizado por: Chela Jhonathan, 2019.

En la **Figura 14-2** se expone las acciones de gestión de usuarios y gestión de programas radiales que dependen en sentido común primeramente del proceso de autenticación de usuario. La gestión de cualquier categoría incluye, el proceso de ingreso, modificación y eliminación de información.

#### 2.4.6.2. Procesos del director de la radio

En este diagrama de casos de uso se describe las acciones que el rol de usuario denominado Director debe cumplir.

En la **Figura 15-2** se presenta las acciones propias en este módulo del sistema, tales como, aprobar el guion de una programación radial y las tareas de reportes contempladas dentro de este rol.



**Figura 15-2:** Diagrama del director de la entidad

Realizado por: Chela Jhonathan, 2019.

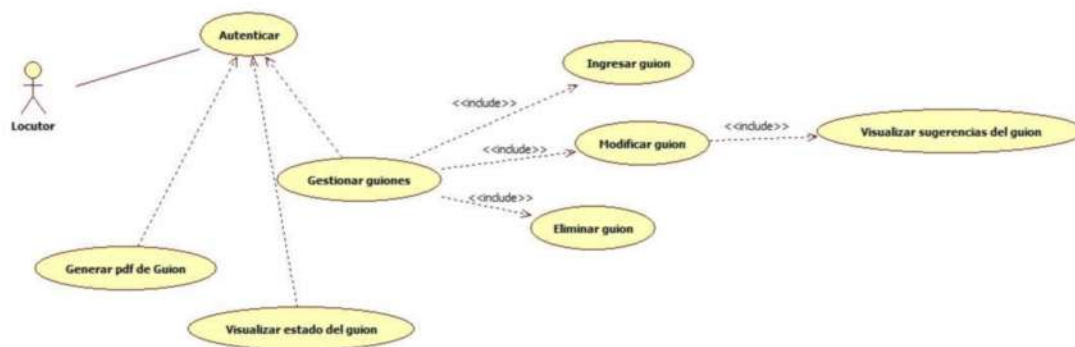
La **Figura 15-2** permite percibir las acciones propias que se realiza con este tipo de rol de usuario, todo el proceso de aprobación de un guion es descrito, así como también el proceso de reportes que son de utilidad para el Director de la radio.

#### 2.4.6.3. Procesos del locutor de la radio

Este diagrama de casos de uso describe todas las acciones que contemplan el modulo perteneciente al usuario denominado Locutor.

El locutor es la persona encargada de realizar las programaciones radiales, por lo tanto, también es el responsable de gestionar el guion de su programa.

La **Figura 16-2** visualiza el proceso de gestión del guion, así como también, la visualización de las sugerencias hechas a un guion por el director y la visualización del estado del guion, si este está aprobado o no.



**Figura 16-2:** Diagrama del locutor del sistema

Realizado por: Chela Jhonathan, 2019.

### 2.4.7. Historias de usuario

Las historias de usuario representan los requisitos definidos por el dueño del producto (*Product Owner*), está descrito en lenguaje natural utilizando la recomendación presentada por *Scrum*. Cada historia de usuario permite ver en forma detallada una funcionalidad o trabajo técnico a realizar en el desarrollo de un proyecto software.

En las tablas que se presentan y se ordenan las historias, se procede a describir en primer lugar, el tipo de historia de usuario, sea de funcionalidad o técnica. Constan, además, el ID de identificación, una breve descripción de la historia, que usuario necesita la funcionalidad o tarea, el sprint en cual están siendo ejecutadas, puntos de estimación y puntos reales, así como también la prioridad y el riesgo en el desarrollo. Por último, se presentan una respectiva descripción y observaciones respectivas para cada historia de usuario.

En la **Tabla 14-2** se puede apreciar la documentación de una historia de usuario que involucra el desarrollo del proyecto SIGUION.

**Tabla 14-2:** Historia de usuario de SIGUION

<b>Historia de Usuario</b>	
<b>ID:</b> HU-05	<b>Descripción:</b> Como director necesito visualizar el guion de un programa radial.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6

<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>director</i> necesito visualizar el guion de un programa radial.	
<b>Observaciones:</b> El sistema debe permitir visualizar los datos de un guion de un programa radial.	

**Realizado por:** Chela Jhonathan, 2019.

El desarrollo del proyecto SIGUION, involucra 23 historias de usuario que representan las funcionalidades del sistema. Ver **ANEXO A**.

#### 2.4.8. Historias técnicas

Las historias técnicas fueron definidas para poder realizar la planificación y el seguimiento documentado de los casos de prueba unitaria que han sido ejecutados en las unidades de código que comprenden el proyecto.

En la **Tabla 15-2** se puede apreciar la documentación de una historia técnica correspondiente a la historia de usuario HU-05.

**Tabla 15-2:** Historia técnica de SIGUION

<b>Historia Técnica</b>	
<b>ID:</b> HT-14	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-05.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-05.	
<b>Observaciones:</b> Cada prueba unitaria automatizada debe ser realizada con el framework JUnit y ejecutada para probar su funcionalidad.	

**Realizado por:** Chela Jhonathan, 2019.

Adicional a la tabla de descripción de la historia técnica también se ha elaborado una tabla que describe el alcance y resultado del caso de prueba unitaria para la respectiva historia de usuario.

**Tabla 16-2:** Tabla de registro de prueba unitaria

<b>Id</b>	<b>PU22 - PU24</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-05
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion</li> <li>• Cargar diálogos del guion</li> </ul>
<b>Fecha</b>	25-01-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible visualizar el guion de un programa radial.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten que un guion sea visualizado.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idtprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial,     fecharegistro);     assertNotNull(result); } /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");     int idtguionprogramaradial = 6;     List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;();     result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial);     assertNotNull(result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como director</li> <li>2. Ubicarse en el panel de administración</li> <li>3. En la caja de texto ingresar el criterio de búsqueda del guion</li> <li>4. Clic en Buscar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	25-01-2019.
<b>Observaciones</b>	Ninguna.

Realizado por: Chela Jhonathan, 2019.

En la **Tabla 16-2** se puede apreciar la estructura de la documentación del caso de prueba **PU22 - PU24**, aplicado en la historia de usuario HU-05. Se presenta una breve descripción de las unidades de código, así como también, la fecha, la funcionalidad, los datos de entrada, el resultado esperado, el código generado por JUnit y el resultado obtenido; este caso es una prueba con

resultado de éxito. Los casos de pruebas unitarias de este proyecto se encuentran debidamente documentados en el **ANEXO E**.

#### **2.4.9. Tareas de ingeniería**

Tareas de ingeniería son actividades que en su totalidad conforman una historia de usuario, estas tareas se documentan en una tabla en la misma que se muestra el estado en el que se encuentran, con la finalidad de dar a conocer al equipo de desarrollo el proceso continuo de avance de una historia de usuario. En la **Tabla 17-2** se puede visualizar las tareas de ingeniería de la historia de usuario HU-05.

**Tabla 17-2:** Tabla de tareas de ingeniería de HU-05

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-05.- Como director necesito visualizar el guion de un programa radial.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-05.- Creación de la clase guion radial con los atributos correspondientes.
		T02HU-05.-Creación del método que permita visualizar el guion en la capa lógica de negocios la clase InProgramaRadial()
		T03HU-05.-Creación del método que permita visualizar el guion en la clase adProgramaRadial() en el paquete de acceso a datos.
		T04HU-05.- Integrar con la interfaz gráfica correspondiente para visualizar un guion
		T05HU-05.- Diseño de mensajes de información.
		T06HU-05.- Pruebas de aceptación.

**Realizado por:** Chela Jhonathan, 2019.

La **Tabla 17-2** presenta la documentación de las tareas de ingeniería que conforman la historia de usuario HU-05, en la misma que se detalla una descripción indicando a que historia de usuario corresponde, el estado en el que se encuentran cada una de las actividades programadas. Ver **ANEXO A**.

#### **2.4.10. Casos de pruebas unitarias por historias de usuario del sistema**

Los casos de pruebas unitarias comprendidos en el desarrollo de SIGUION han sido analizados y formulados para su posterior ejecución y análisis de resultados. Se estableció 104 casos de pruebas unitarias correspondientes a 55 funcionalidades del sistema, todas las funcionalidades que han sido desarrolladas pertenecen a las 23 historias de usuario identificadas en este proyecto.

A continuación, en la **Tabla 18-2** se presenta un ejemplo de la planificación de los casos de pruebas unitarias:



**Tabla 18-2:** Casos de pruebas unitarias por requerimiento de SIGUION

<b>Requerimiento</b>	<b>Funcionalidad</b>	<b>Casos de prueba</b>	<b>Cantidad de líneas de código</b>	<b>% de Cobertura de código</b>	<b>Cantidad de líneas de código cubiertas</b>
HU-01	F1, F2	PU1, PU2, PU3	52	99%	52
HU-02	F1, F2	PU4, PU5, PU6, PU7	57	99%	56
HU-03	F1, F2, F3, F4	PU8, PU9, PU10, PU11, PU12, PU13, PU14, PU15	115	99%	114
HU-04	F1, F2, F3	PU16, PU17, PU18, PU19, PU20, PU21	111	98%	109
HU-05	F1, F2	PU22, PU23, PU24	59	97%	57
HU-06	F1, F2, F3	PU25, PU26, PU27, PU28, PU29, PU30	130	96%	125
HU-07	F1, F2, F3	PU31, PU32, PU33, PU34, PU35	112	96%	108
HU-08	F1, F2, F3	PU36, PU37, PU38, PU39, PU40	112	96%	108
HU-09	F1, F2, F3	PU41, PU42, PU43, PU44, PU45	88	98%	86
HU-10	F1, F2	PU46, PU47, PU48	53	98%	52
HU-11	F1, F2, F3	PU49, PU50, PU51, PU52, PU53, PU54	122	98%	120
HU-12	F1, F2, F3, F4	PU55, PU56, PU57, PU58, PU59, PU60	111	95%	105
HU-13	F1	PU61, PU62, PU63	56	96%	54
HU-14	F1, F2, F3, F4, F5	PU64, PU65, PU66, PU67, PU68, PU69, PU70, PU71, PU72, PU73	154	96%	148
HU-15	F1, F2, F3	PU74, PU75, PU76, PU77, PU78, PU79	90	98%	88
HU-16	F1, F2	PU80, PU81, PU82, PU83	61	98%	60
HU-17	F1, F2	PU84, PU85, PU86, PU87	61	98%	60
HU-18	F1, F2, F3	PU88, PU89, PU90, PU91, PU92, PU93	116	98%	114
HU-19	F1	PU94, PU95	51	100%	51
HU-20	F1	PU96, PU97, PU98	70	99%	69
HU-21	F1	PU99, PU100	51	99%	50
HU-22	F1	PU101, PU102	54	99%	53

HU-23	F1	PU103, PU104	54	99%	53
-------	----	-----------------	----	-----	----

**Realizado por:** Chela Jhonathan, 2019.

En la **Tabla 18-2** se puede apreciar un resumen organizado en columnas, en cada fila se describe una historia de usuario que involucra una cantidad determinada de funciones y sus pruebas unitarias respectivamente, además, se presenta un total general de las líneas de código que componen cada caso junto con el valor del porcentaje de cobertura. Finalmente, en la última columna se puede apreciar la cantidad de líneas de código cubierta al ejecutar un test unitario. Los casos de prueba unitaria por requerimientos se encuentran detallados en el **ANEXO F**.

#### **2.4.11. Sprint del proyecto**

Luego de haber realizado el respectivo análisis y definido la lista de producto o product backlog para el proyecto, se establece un plan de entregas para el proyecto (Sprint Backlog).

En el plan de entregas se detalla el alcance de las historias de usuarios y las historias técnicas determinando la fecha de inicio y la fecha en la cual se culmina cada incremento. Además. Se establece la cantidad de horas planificadas para cada sprint.

Es importante recalcar que cada sprint está planificado en un lapso aproximado de tres semanas, con un horario de trabajo de 6 horas diarias.

En la **Tabla 19-2** se expone el Sprint Backlog del proyecto SIGUION.

**Tabla 19-2:** Sprint Backlog de SIGUION

<b>Sprint</b>	<b>Descripción</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	01/01/2019	14 /01/2019	90
Sprint 2	HU-01, HT-10, HU-02, HT-11, HU-03, HT-12, HU-04, HT-13, HU-05, HT-14, HU-06, HT-15 HT-16, HT-17	15 /01/2019	31/01/ 2019	100
Sprint 3	HU-07, HT-18, HU-08, HT-19, HU-09, HT-20, HU-10, HT-21, HU-11, HT-22, HT-23, HT-24	01/02/2019	15/02/2019	100
Sprint 4	HU-12, HT-25, HU-13, HT-26, HU-14, HT-27, HU-15, HT-28, HU-16, HT-29, HU-17, HT-30, HT-31, HT-32	18/02/2019	07/03/2019	100
Sprint 5	HU-18, HT-33, HU-19, HT-34, HU-20, HT-35, HU-21, HT-36, HU-22, HT-37, HU- 23, HT-38, HT-39, HT-40	08/03/2019	22/03/2019	100

**Realizado por:** Chela Jhonathan, 2019.

En la **Tabla 19-2** se puede apreciar los campos Sprint, que hace mención a los nombres de cada sprint planificado; Descripción, que determina los códigos de historias de usuario e historias

técnicas que contiene cada iteración, los campos en donde se registran la fecha de inicio, fecha de finalización y las horas estimadas.

#### 2.4.11.1. *Sprint 1*

Este sprint contiene la planificación de las historias de usuario técnicas, este tipo de historias de usuario fueron planificadas con el objetivo de establecer las bases sólidas sobre las cuales se desarrolló el proyecto de construcción de SIGUION con pruebas unitarias.

Entre las especificaciones que se determinan en la **Tabla 20-2** se encuentran, la identificación y especificación de los requerimientos necesarios, el diseño de la base de datos, el establecimiento de los estándares para el proyecto, el plan de pruebas, entre otro. A continuación, se detalla cada historia técnica (HT) que pertenece a este sprint:

**Tabla 20-2:** Sprint 1 del Proyecto SIGUION

<b>Sprint 1</b>				
<b>Fecha Inicio:</b> 01/01/2019	<b>Fecha Fin:</b> 14 /01/2019	<b>Esfuerzo Estimado (H):</b> 90		<b>Esfuerzo Real (H):</b> 85
<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 una recopilación de requerimientos del sistema.	10	Análisis	Jhonathan Chela
HT-02	Como desarrollador necesito realizar la planificación de los entregables del sistema.	15	Análisis	Jhonathan Chela
HT-03	Como desarrollador necesito definir las historias de usuario del sistema.	15	Análisis	Jhonathan Chela
HT-04	Como desarrollador necesito realizar la planificación de las pruebas unitarias automatizadas.	15	Análisis	Jhonathan Chela
HT-05	Como desarrollador necesito diseñar la arquitectura del sistema.	6	Diseño	Jhonathan Chela
HT-06	Como desarrollador necesito diseñar las interfaces de usuario.	6	Diseño	Jhonathan Chela
HT-07	Como desarrollador necesito diseñar la base de datos.	6	Diseño	Jhonathan Chela
HT-08	Como desarrollador necesito realizar la implementación de la base de datos.	6	Desarrollo	Jhonathan Chela
HT-09	Como desarrollador necesita realizar la conexión a la base de datos.	6	Desarrollo	Jhonathan Chela

**Realizado por:** Chela Jhonathan, 2019.

#### 2.4.11.2. *Sprint 2*

En este segundo sprint de planificación inicia el desarrollo de los incrementos funcionales del sistema, en la planificación constan historias técnicas (HT) e historias de usuario (HU). Existe

una totalidad de 6 historias de usuario que representan la implementación de funcionalidades junto con 8 historias técnicas que representan a las pruebas unitarias automatizadas en esta primera iteración.

A continuación, en la **Tabla 21-2** se detalla el alcance del primer entregable del proyecto SIGUION junto con la planificación de las pruebas unitarias del sistema.

**Tabla 21-2:** Sprint 2 del Proyecto SIGUION.

<b>Sprint 2</b>				
<b>Fecha Inicio:</b> 15 /01/2019	<b>Fecha Fin:</b> 31/01/ 2019	<b>Esfuerzo Estimado (H):</b> 100	<b>Esfuerzo Real (H):</b> 111	
<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 para ingresar al sistema.	6	Desarrollo	Jhonathan Chela
HT-10	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-01	6	Test	Jhonathan Chela
HU-02	Como administrador del sistema necesito ingresar nuevos usuarios al sistema.	12	Desarrollo	Jhonathan Chela
HT-11	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-02	6	Test	Jhonathan Chela
HU-03	Como administrador del sistema necesito ingresar programas radiales al sistema.	12	Desarrollo	Jhonathan Chela
HT-12	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-03	6	Test	Jhonathan Chela
HU-04	Como locutor necesito ingresar guiones de programas radiales.	15	Desarrollo	Jhonathan Chela
HT-13	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-04	6	Test	Jhonathan Chela
HU-05	Como director necesito visualizar el guion de un programa radial.	12	Desarrollo	Jhonathan Chela
HT-14	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-05	6	Test	Jhonathan Chela
HU-06	Como director necesito realizar la aprobación del guion de un programa radial.	6	Desarrollo	Jhonathan Chela
HT-15	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-06	6	Test	Jhonathan Chela
HT-16	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	6	Desarrollo	Jhonathan Chela
HT-17	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-16.	6	Test	Jhonathan Chela

**Realizado por:** Chela Jhonathan, 2019.

### 2.4.11.3. *Sprint 3*

Este tercer sprint detalla 4 historias de usuario junto con 6 historias técnicas, este conjunto marca un nuevo avance del proyecto SIGUION. Este sprint al ser terminado e integrado al sprint anterior determina las funcionalidades casi en su totalidad de los usuarios del sistema denominados, director y locutor.

A continuación, en la **Tabla 22-2** se presenta la planificación de las historias de usuario e historias técnicas de este avance.

**Tabla 22-2:** Sprint 3 del Proyecto SIGUION

<b>Sprint 3</b>				
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>	<b>Esfuerzo Estimado (H):</b>	<b>Esfuerzo Real (H):</b>	
01/02/2019	15/02/2019	100	93	
<b>Pila del sprint</b>				
<b>ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HU-07	Como director necesito ingresar el estado de No Aprobado a un guion del programa radial.	12	Desarrollo	Jhonathan Chela
HT-18	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-07	6	Test	Jhonathan Chela
HU-08	Como director necesito ingresar el estado de Revisado a un guion del programa radial.	6	Desarrollo	Jhonathan Chela
HT-19	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-08	6	Test	Jhonathan Chela
HU-09	Como director necesito ingresar las sugerencias respectivas al guion de un programa de la radio.	15	Desarrollo	Jhonathan Chela
HT-20	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-09	6	Test	Jhonathan Chela
HU-10	Como locutor necesito visualizar las sugerencias del guion de un programa radio.	15	Desarrollo	Jhonathan Chela
HT-21	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-10	6	Test	Jhonathan Chela
HT-22	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	15	Desarrollo	Jhonathan Chela
HT-23	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-22.	6	Test	Jhonathan Chela

**Realizado por:** Chela Jhonathan, 2019.

### 2.4.11.4. *Sprint 4*

En este sprint se continúa con el desarrollo de las funcionalidades y pruebas unitarias automatizadas de SIGUION, en esta planificación constan 6 historias de usuario y 8 historias

técnicas. Las funcionalidades implementadas pertenecen a los usuarios del sistema denominados, administrador y locutor.

En la **Tabla 23-2** se exponen de manera detallada las historias técnicas y las historias de usuario correspondientes.

**Tabla 23-2:** Sprint 4 del Proyecto SIGUION

<b>Sprint 4</b>				
<b>Fecha Inicio:</b> 18/02/2019	<b>Fecha Fin:</b> 07/03/2019	<b>Esfuerzo Estimado (H):</b> 100	<b>Esfuerzo Real (H):</b> 114	
<b>Pila del sprint</b>				
<b>ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HU-11	Como locutor necesito modificar el guion de un programa de radio.	6	Desarrollo	Jhonathan Chela
HT-24	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-11	6	Test	Jhonathan Chela
HU-12	Como locutor necesito realizar los respectivos cambios al guion de un programa de radio.	12	Desarrollo	Jhonathan Chela
HT-25	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-12	6	Test	Jhonathan Chela
HU-13	Como locutor necesito visualizar el estado del guion.	6	Desarrollo	Jhonathan Chela
HT-26	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-13	6	Test	Jhonathan Chela
HU-14	Como administrador del sistema necesito modificar los programas radiales.	12	Desarrollo	Jhonathan Chela
HT-27	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-14	6	Test	Jhonathan Chela
HU-15	Como administrador del sistema necesito eliminar los programas radiales.	12	Desarrollo	Jhonathan Chela
HT-28	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-15	6	Test	Jhonathan Chela
HU-16	Como administrador del sistema necesito modificar la información de los usuarios.	12	Desarrollo	Jhonathan Chela
HT-29	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-16	6	Test	Jhonathan Chela
HT-30	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	12	Desarrollo	Jhonathan Chela
HT-31	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-30.	6	Test	Jhonathan Chela

Realizado por: Chela Jhonathan, 2019.

#### 2.4.11.5. *Sprint 5*

Este sprint marca la finalización del desarrollo del proyecto SIGUIÓN, consta de 7 historias de usuario junto con 9 historias técnicas. En este sprint se logra cumplir con la totalidad de las funcionalidades para los tres roles de usuarios del sistema definidos, administrador, locutor, director.

En la **Tabla 24-2** se presenta el detalle de las historias planificadas para este incremento, además una vez finalizada la planificación se procede a el estudio y análisis de resultados para su posterior presentación.

**Tabla 24-2:** Sprint 5 del Proyecto SIGUIÓN.

<b>Sprint 5</b>				
<b>Fecha Inicio:</b> 08/03/2019	<b>Fecha Fin:</b> 22/03/2019	<b>Esfuerzo Estimado (H):</b> 100	<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>
HU-17	Como administrador del sistema necesito eliminar los usuarios ingresados.	12	Desarrollo	Jhonathan Chela
HT-32	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-17.	6	Test	Jhonathan Chela
HU-18	Como locutor necesito eliminar un guion del programa radial.	6	Desarrollo	Jhonathan Chela
HT-33	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-18.	6	Test	Jhonathan Chela
HU-19	Como director necesito realizar el reporte de guiones no aprobados en la semana en formato pdf.	12	Desarrollo	Jhonathan Chela
HT-34	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-19	6	Test	Jhonathan Chela
HU-20	Como director necesito realizar el reporte de los guiones aprobados en la semana en formato pdf.	12	Desarrollo	Jhonathan Chela
HT-35	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-20	6	Test	Jhonathan Chela
HU-21	Como director necesito realizar el reporte de los guiones revisados en la semana en formato pdf.	12	Desarrollo	Jhonathan Chela
HT-36	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-21	6	Test	Jhonathan Chela
HU-22	Como administrador del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	6	Desarrollo	Jhonathan Chela
HT-37	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-22	6	Test	Jhonathan Chela

HU-23	Como administrador del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	6	Desarrollo	Jhonathan Chela
HT-38	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-23	6	Test	Jhonathan Chela
HT-39	Como desarrollador necesito presentar el avance respectivo y realizar una retrospectiva sobre el proyecto.	6	Desarrollo	Jhonathan Chela
HT-40	Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HT-39	6	Test	Jhonathan Chela

**Realizado por:** Chela Jhonathan, 2019.

## 2.5. Fase de cierre e implementación

Para la implementación del sistema de gestión de guiones denominado SIGUION se corrige todos los errores que se presentan en la fase Beta del producto. En esta etapa final se prueban las funcionalidades del producto en un periodo de siete días, con un total de 20 usuarios del sistema, 1 usuario corresponde al rol de Director de la entidad de radio, 1 usuario corresponde al rol de Administrador del sistema y 18 usuarios corresponden al rol de Locutor de la entidad. Para más información ver el Capítulo 3 dónde se analiza el beneficio de los procesos manuales que fueron automatizados en el sistema.

SIGUION fue desplegado en un servidor de aplicaciones GlassFish Server 4.1.1, la base de datos del sistema fue implementada un servidor de base de datos con la versión 11.3 de PostgreSQL.

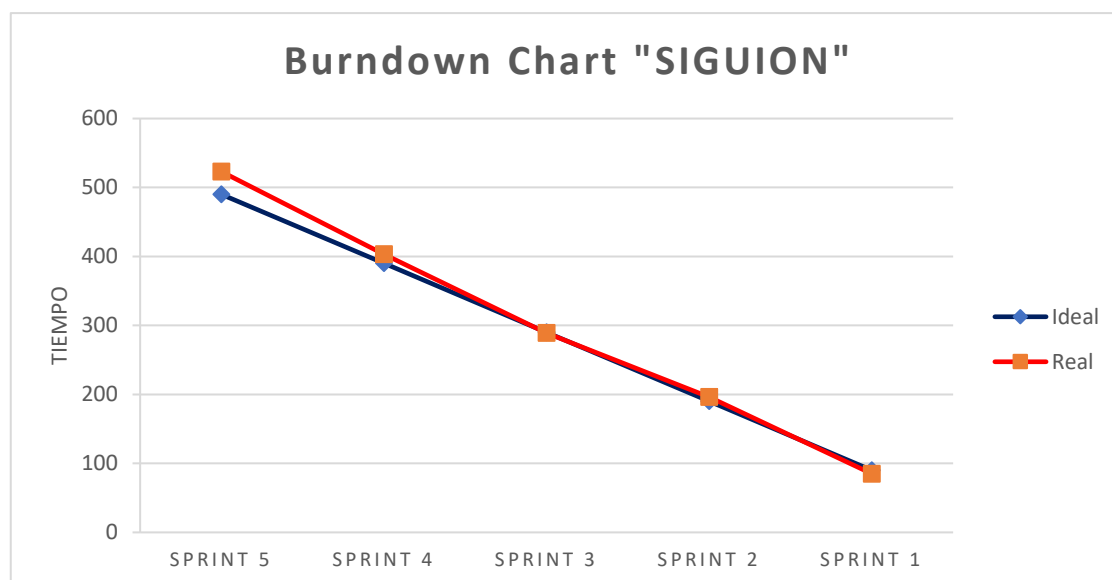


## CAPÍTULO III

### 3. MARCO DE RESULTADOS

#### 3.1. Gestión del Proyecto

SCRUM permite monitorear la velocidad del proyecto mediante un gráfico denominado Burndown Chart. En este gráfico se traza la línea de la velocidad planificada o estimada para el proyecto y la línea de la velocidad real, estas dos líneas permiten saber si existieron retrasos o adelantos en el tiempo de entrega para cada incremento planificado.



**Gráfico 1-3:** Burndown chart del proyecto

**Realizado por:** Chela Jhonathan, 2019.

En el **Gráfico 1-3**, se puede apreciar para el proyecto SIGUION, la línea de la velocidad ideal de color azul y la línea de la velocidad real de color rojo. Se han planificado un total de 5 Sprints con una duración de 490 horas estimadas.

En el Sprint 1 y el Sprint 3, existe un adelanto en la entrega de tiempo, mientras que en los Sprints 2, 4 y 5 hay una variación del tiempo estimado, esto debido al alcance de cada Sprint y a la cantidad de pruebas unitarias ejecutadas en estos periodos planificados.

### 3.2. Evaluación de la exactitud del software.

Debido a que se plantea como objetivo a cumplir, el evaluar la exactitud del software mediante casos de prueba automatizados para asegurar la funcionalidad del sistema. En este proyecto se pretende asegurar la calidad de software, específicamente uno de los atributos, la funcionalidad interna, mediante pruebas unitarias aplicadas a los módulos o unidades que componen el sistema SIGUION.

Para esto se formula la cobertura de código como métrica para validar el funcionamiento de cada unidad de código de acuerdo al caso.

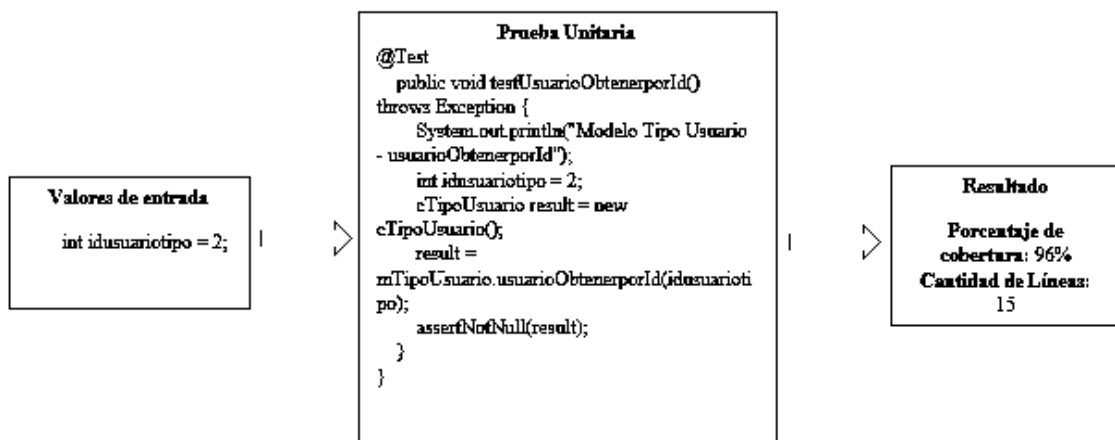
De acuerdo con la norma ISO/IEC 25012, uno de los atributos de calidad de datos inherente del software es la exactitud, la misma que hace referencia al grado en el que los datos representan el valor del atributo deseado de un concepto en un contexto de uso específico.

#### 3.2.1. Cobertura del código

La cobertura de código es una medida porcentual en las pruebas de software, esta determina el grado en que el código fuente de un programa ha sido comprobado. De acuerdo a cada caso de prueba unitaria ejecutada se puede conocer la cantidad de las partes del código que no han sido comprobadas y las que ya fueron probadas.

De acuerdo al diseño de las pruebas unitarias presentadas en el ANEXO E , se ha utilizado la cobertura por líneas de código, esta métrica permite validar la funcionalidad elegida del software.

Los casos de prueba corresponden a la siguiente estructura:



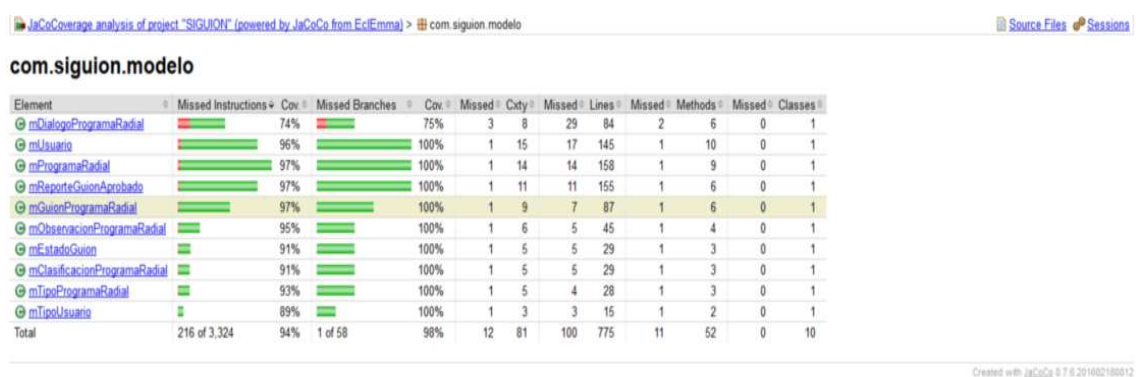
**Figura 1-3:** Estructura de una prueba unitaria

Realizado por: Chela Jhonathan, 2019.

En la **Figura 1-3**, se explica la estructura que siguen las pruebas unitarias desarrolladas, en cada caso de prueba existe un valor o valores que luego al ser ejecutados se verifica las líneas de código de la unidad probada, el resultado esperado es la cantidad de líneas de código de la unidad y el porcentaje que se ha cubierto de acuerdo a los valores devueltos por la función.

### 3.2.2. Cobertura de código con JaCoCoverage

El proyecto SIGUIÓN fue desarrollado en el IDE Netbeans 8.2, para cubrir todos los casos de prueba de los 23 requerimientos funcionales, se utilizó el plugin JaCoCoverage el mismo que ejecuta las pruebas unitarias del proyecto y presenta una tabla de resultados.



**Figura 2-3:** Cobertura con JaCoCoverage

Realizado por: Chela Jhonathan, 2019.

La **Figura 2-3**, muestra el resumen de resultados que presenta el plugin JaCoCoverage luego de haber ejecutado todos los casos de prueba, en el mismo se puede observar el porcentaje de cobertura de código y las líneas que han sido cubiertas en cada unidad definida.

### 3.2.3. Resultados de la cobertura de código

Como se ha expuesto en el Capítulo II, el proyecto consta de un total de 23 historias de usuario con 104 casos de pruebas unitarias que fueron desarrollados. A continuación, se presenta una tabla de resumen de los resultados de la cobertura de código.

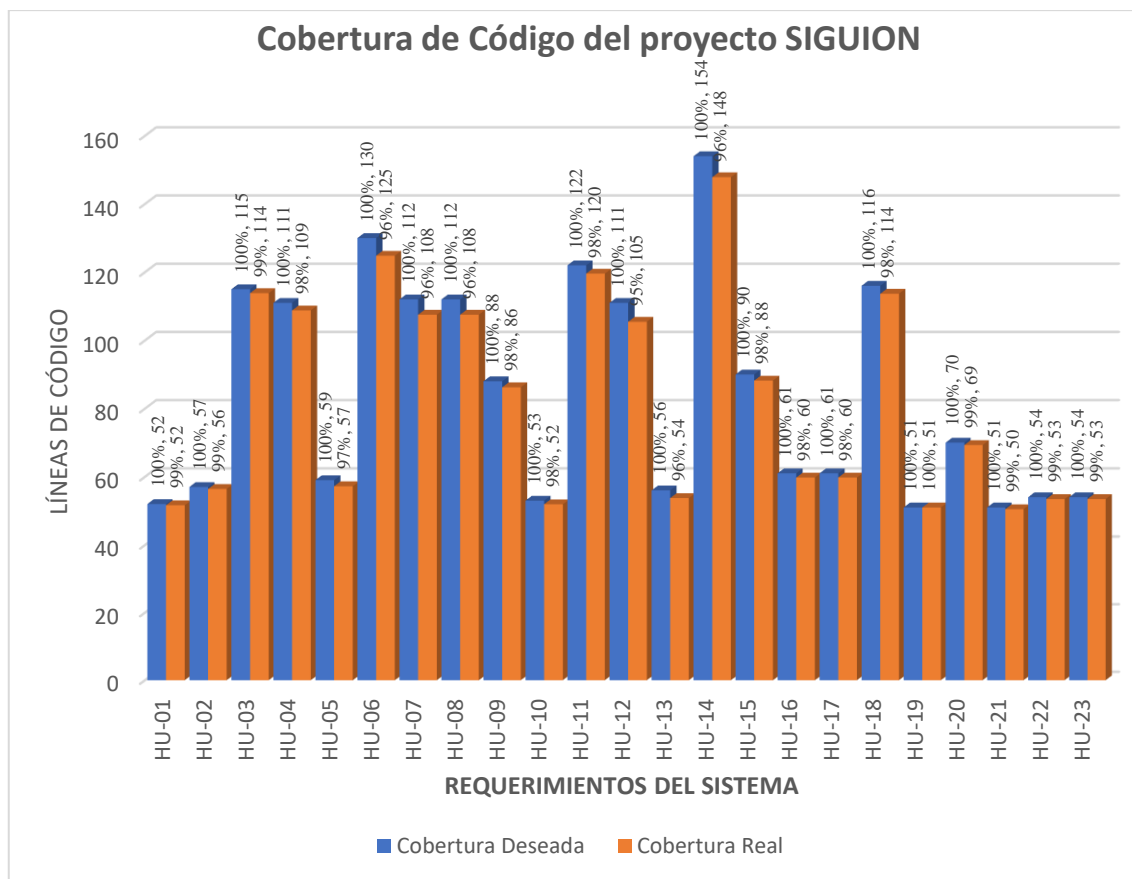
**Tabla 1-3:** Resumen de la cobertura de código por requerimientos

Requerimientos	Líneas de Código	Cobertura Deseado	Cantidad Real (Líneas de código)	Porcentaje Real
HU-01	52	100%	52	99%
HU-02	57	100%	56	99%
HU-03	115	100%	114	99%
HU-04	111	100%	109	98%
HU-05	59	100%	57	97%
HU-06	130	100%	125	96%
HU-07	112	100%	108	96%
HU-08	112	100%	108	96%
HU-09	88	100%	86	98%
HU-10	53	100%	52	98%
HU-11	122	100%	120	98%
HU-12	111	100%	105	95%
HU-13	56	100%	54	96%
HU-14	154	100%	148	96%
HU-15	90	100%	88	98%
HU-16	61	100%	60	98%
HU-17	61	100%	60	98%
HU-18	116	100%	114	98%
HU-19	51	100%	51	100%
HU-20	70	100%	69	99%
HU-21	51	100%	50	99%
HU-22	54	100%	53	99%
HU-23	54	100%	53	99%
<b>Promedio % de Cobertura</b>				<b>98%</b>

Realizado por: Chela Jhonathan, 2019.

La **Tabla 1-3**, representa un resumen de la cobertura de código presentada en el **ANEXO F**, se describe la cantidad de líneas de código y el porcentaje de cobertura junto con las líneas que han sido comprobadas.

En el **Gráfico 2-3** se presenta la cobertura de código en cada historia de usuario, la barra de color naranja representa la cobertura real del código y la barra de color azul la cobertura deseada.



**Gráfico 2-3:** Gráfico de cobertura de código

**Realizado por:** Chela Jhonathan, 2019.

Luego de haber representado la cobertura del código del proyecto en un gráfico de barras se puede analizar claramente que en relación a la cobertura deseada se tiene un valor alto de cobertura real.

En el **Gráfico 2-3**, se representan los valores que se adjuntan en la **Tabla 1-3**, por lo que de manera aproximada se puede decir que el código total ha sido comprobado en un 98%. Todos los casos de pruebas unitarias representados han sido ejecutados y devueltos con un valor de éxito.

### 3.2.4. Criterio de evaluación de la exactitud

Para la evaluación de la exactitud se ha tomado en cuenta un valor de dominio de aceptación, para esto se basa en lo que (González Pinzón & González Sanabria, 2013) exponen en su trabajo con respecto a la métrica de exactitud.

A continuación, en la **Tabla 2-3**, se exponen tres características de calidad, sus respectivas subcaracterísticas, las métricas y el valor del nivel mínimo requerido.

**Tabla 2-3:** Requisitos de calidad

Característica	Subcaracterística	Métrica	Nivel Mínimo Requerido
Funcionalidad	Consistencia	$(C1+C2+C3+C4)/4$	0,95
	Compleitud	$(C5+C6+C7)/3$	0,95
	Precisión	$(P1+P2+P3)/3$	0,95
	<b>Exactitud</b>	<b>E1</b>	<b>0,55</b>
	Actualidad	No Aplica	---
Usabilidad	Entendibilidad	$(U1+U2+U3)/3$	0,95
	Manejabilidad	M1	0,55
Mantenibilidad	Facilidad de cambio	R1	0,70

**Fuente:** (González Pinzón & González Sanabria, 2013).

**Realizado por:** Chela Jhonathan, 2019

De manera general conforme a la forma del cálculo de las métricas, es importante aclarar qué; para el cálculo de la consistencia, se definen cuatro atributos, los mismos que son representados por la letra C junto con el número. Con respecto a la definición de la métrica de la completitud se identifica sus atributos con la letra C y la numeración del 5 – 7. De manera similar se expone para las subcaracterísticas; entendibilidad, manejabilidad, facilidad de cambio, se definen con las letras U, M, R respectivamente, con una numeración del 1 – 3.

La exactitud es definida con la letra E y el número 1, este indica el valor del nivel mínimo requerido, este es 0,55.

De acuerdo a los datos de cobertura para el proyecto expuesto en la **Tabla 1-3**, se tiene que el valor del porcentaje promedio es 98%, por lo que se procede a la transformación en el dominio de 0 a 1, se tiene un valor de 0,98. Por lo tanto, como el nivel mínimo requerido expuesto para la aceptación de exactitud conforme a la **Tabla 2-3** es menor al valor promedio de cobertura, entonces, las funcionalidades de las unidades de código del proyecto SIGUION tienen exactitud.

### **3.3. Análisis del beneficio del software con respecto al proceso manual**

Para determinar si existe un beneficio del sistema SIGUION con respecto al proceso manual que se realiza en la gestión de guiones, se realiza un análisis estadístico para refutar o aceptar lo planteado, en los siguientes puntos se presenta el proceso.

#### **3.3.1. Proceso de gestión de guiones**

Para determinar si existe un beneficio con respecto al tiempo en el proceso de gestión de guiones en la Radio Crisólito Celestial se plantea los siguientes aspectos:

**Objeto de Experimentación:** Sistema de Gestión de Guiones SIGUION para la Radio Crisólito Celestial.

**Sujetos de Experimentación:** Director de la entidad y locutores.

### 3.3.2. Muestra

En este proceso de análisis estadístico se emplea una muestra igual a 20 usuarios, los mismos que representan a las personas involucradas en la población, los datos representan los tiempos en segundos del proceso de gestión de guiones.

**Tabla 3-1:** Registro de tiempo en la gestión de guiones.

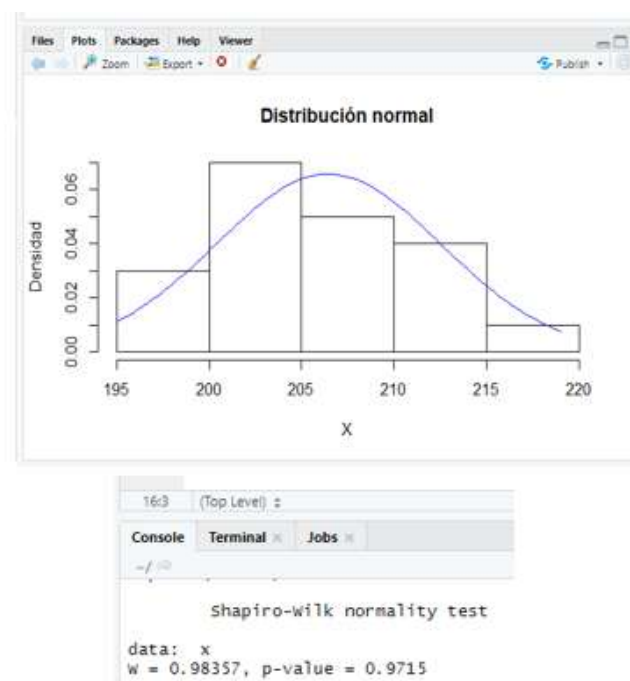
Datos de registro de datos del proceso de gestión de Guiones									
Datos de Usuario		SIGUION (Segundos)				Proceso Manual (Segundos)			
No.	Usuario	Ingresar	Revisar	Modificar	Promedio	Ingresar	Revisar	Modificar	Promedio
1	Manobanda Stalin	280	66	277	208	789	150	395	445
2	Aucatoma Angel	270	65	289	208	798	145	399	447
3	Chela Alfredo	290	67	278	212	768	160	384	437
4	Quille Magaly	275	66	275	205	792	155	396	448
5	Rea Oswaldo	265	65	285	205	763	156	382	434
6	Tenelema Marcia	280	66	295	214	756	145	378	426
7	Quicaliquin Wilmer	297	60	299	219	754	156	377	429
8	Hinojosa Hermes	260	62	288	203	765	153	383	434
9	Tualombo Klever	265	63	274	201	789	159	395	448
10	Manobanda Sergio	280	62	286	209	782	145	391	439
11	Rea Esther	265	60	284	203	793	140	397	443
12	Arellano David	267	66	283	205	794	156	397	449
13	Pilco Emilio	278	60	278	205	786	157	393	445
14	Chimbo Daniel	298	67	279	215	753	158	377	429
15	Curi Giovanni	288	69	268	208	798	159	399	452
16	Vásquez César	256	62	267	195	751	160	376	429
17	Chela Fanny	245	60	285	197	798	161	399	453
18	Manya Jairo	289	60	287	212	790	158	395	448
19	Ninabanda Noé	278	66	254	199	789	156	395	447
20	Yallico Rodolfo	268	65	286	206	799	149	400	449

Realizado por: Chela Jhonathan, 2019.

En la **Tabla 3-1** se puede apreciar el registro de tiempos en segundos del proceso de gestión de guiones, este proceso involucra, ingresar, revisar y realizar las respectivas modificaciones a un guion. Para el proceso de análisis se realiza un promedio de los tiempos, en la tabla se representan los valores en las columnas de color celeste.

### 3.3.3. *Contraste de normalidad*

En el proceso de análisis estadísticos para poder concluir si existe o no un beneficio en la gestión de guiones, es importante comprobar si los datos proceden de una población con distribución normal. Para este fin se utiliza el contraste de normalidad de Shapiro-Wilk, que se aplica siempre y cuando el tamaño de la población es menor a 50.



**Gráfico 3-3:** Contraste Shapiro-Wilk de los datos del proyecto SIGUION

**Realizado por:** Chela Jhonathan, 2019.

En el **Gráfico 3-3** se presenta la gráfica de los resultados, al realizar el contraste de normalidad, se obtiene el estadístico de contraste  $W$  igual a 0.98357, y la probabilidad de contraste  $p$  igual a 0.9715, siendo superior al nivel de significancia 0.05 se asume que la distribución de la muestra es normal, por lo tanto, se puede aplicar la estadística paramétrica. Ver **ANEXO G**.

### 3.3.4. *Análisis de datos*

Dado que el espacio muestral es 20 y la población tiene una distribución normal, para el análisis estadístico se utiliza la distribución  $T$  de Student.



Para esto se plantea la siguiente interrogante a responder, ¿Existe una diferencia significativa entre los tiempos promedios del proceso manual y el sistema SIGUIÓN en la gestión de guiones de radio? En el análisis se utiliza un nivel de significancia del 1%, entonces, se plantea las siguientes hipótesis:

- **H<sub>0</sub>**: No hay diferencia significativa entre los tiempos promedios del proceso manual y al usar el sistema en la gestión de guiones.
- **H<sub>1</sub>**: Hay diferencia significativa entre los tiempos promedios del proceso manual y al usar el sistema en la gestión de guiones.

Se aplica la prueba de dos colas:

$$\begin{array}{l}
 H_0 = \mu_D = 0 \\
 H_1 = \mu_D \neq 0
 \end{array}
 \left\{ \begin{array}{l}
 1) \mu_D > 0, \text{ cuando } \mu_S > \mu_M \\
 \text{o} \\
 2) \mu_D < 0, \text{ cuando } \mu_S < \mu_M
 \end{array} \right.$$

$$\alpha = 1\%$$

$$\text{Tamaño de la muestra} = n = 20$$

$$\alpha/2 = 0.01/2 = 0.005$$

$$\text{Grados de libertad} = v = 20 - 1 = 19$$

$$t_{\alpha/2} = t_{0.005} = 2.861;$$

Determinación de la región crítica:

- Área de la aceptación de la hipótesis nula = entre los valores [-2.86093; 2.86093]
- Área de rechazo de la hipótesis nula = a la izquierda de -2.86093 y a la derecha de 2.86093.

Se procede a utilizar el estadístico de contraste de prueba t-Student:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Donde:

$\bar{x}$  = media muestral.

**S** = desviación estándar muestral.

**n**: = tamaño de la muestra.

$$\mu_0 = 0$$

Reemplazando valores:

$$t = \frac{235.1 - 0}{12.33/\sqrt{20}} = \frac{235.1}{2.76} = 85.18$$

El valor de t calculado es 85.18, este valor es mayor al valor crítico t de la tabla que es 2.86093, por lo cual se acepta la hipótesis alterna (**H<sub>1</sub>**), dado que se determina que cae en la región de rechazo.

Entonces, de acuerdo con **H<sub>1</sub>**, existe una diferencia significativa entre los tiempos promedios del proceso manual y al usar el sistema, en la de gestión de guiones.

Para verificar que la diferencia de medias es a favor del sistema SIGUION ( $\mu_D < 0$ ,  $\mu_S < \mu_M$ ), se calcula la diferencia de tiempos promedios entre SIGUION y el proceso manual, a continuación, en la **Tabla 4-3** se aprecian las cifras:

**Tabla 4-3:** Diferencia de tiempos promedios entre el sistema y el proceso manual.

<b>Tiempo SIGUION (Segundos)</b>	<b>Tiempo Manual (Segundos)</b>	<b>Diferencia (Segundos)</b>
208	445	-237
208	447	-239
212	437	-226
205	448	-242
205	434	-229
214	426	-213
219	429	-210
203	434	-230
201	448	-247
209	439	-230
203	443	-240
205	449	-244
205	445	-240
215	429	-215
208	452	-244
195	429	-234
197	453	-256
212	448	-236
199	447	-247
206	449	-243
<b>Promedio (<math>\mu_S</math>): 206.45</b>	<b>Promedio (<math>\mu_M</math>): 441.55</b>	<b><math>\mu_D = -235.1</math></b>
<b>Media</b>		<b>235.1</b>
<b>Desviación Estándar</b>		<b>12.3284011</b>

Realizado por: Chela Jhonathan, 2019.

En la **Tabla 4-3**, se puede apreciar los siguientes valores:

$$\mu_S = 206.45 \text{ s (Media de SIGUIÓN)}$$

$$\mu_M = 441.55 \text{ s (Media del proceso manual)}$$

$$\mu_D = \mu_S - \mu_M = 206.45 - 441.55 = - 235.1 \text{ s (valor de la diferencia)}$$

Entonces:

$\mu_D < 0$  porque  $\mu_S < \mu_M$ , esto satisface la segunda opción propuesta de la hipótesis nula planteada.

Teniendo en cuenta que la gestión de guiones toma un total de 441.55 s en el proceso manual y al usar SIGUIÓN un total de 206.45 s, la diferencia entre estos tiempos es de 235.1 s, se puede determinar por una regla de tres simple el porcentaje en el que se reduce el tiempo manual, a continuación, el cálculo:

$$PR = \frac{(235.1 * 100)}{441.55} = 53.2442532 \%$$

Por lo tanto, al aceptar **H1**, con  $\mu_D < 0$  y  $\mu_S < \mu_M$  se puede decir que existe beneficio de disminución de tiempo con respecto al proceso manual en un 53%.

## CONCLUSIONES

- Se realizó el análisis de la información con respecto al uso de pruebas unitarias y se determinó que el modelo del ciclo de vida del desarrollo de software en V implementa casos de prueba unitarias en el diseño de los módulos del sistema.
- Después de realizar la recolección de la información sobre el proceso de gestión de guiones en la Radio Crisolito Celestial, se logró realizar el diseño del sistema web planificando un total de 5 Sprints que abarcan 23 historias de usuario y 40 historias técnicas, 30 representan a las pruebas unitarias; todo esto se gestionó con la metodología SCRUM.
- Para el sistema denominado SIGUION, ha sido posible implementar una base de datos relacional en PostgreSQL, además, se ha desarrollado el software siguiendo el patrón de arquitectura modelo, vista, controlador; el modelo y el controlador del sistema fueron codificados en el lenguaje Java, así como también la vista del proyecto ha sido desarrollada con JSF y Bootstrap permitiendo tener una interfaz web responsiva.
- Según la evaluación de la exactitud en el sistema SIGUION tomando como métrica la cobertura del código de las unidades planificadas, se tiene que para un total de 104 casos de prueba que pertenecen a las 23 historias de usuario respectivamente, la cobertura es del 98% y el valor del nivel de cobertura es de 0,98. Al comparar el valor del nivel mínimo de aceptación que es el 0,55 con el valor de cobertura se concluye que existe un alto porcentaje de exactitud que asegura la funcionalidad interna para el proyecto SIGUION.
- La implementación del sistema informático SIGUION ha permitido optimizar el tiempo en el proceso de gestión de guiones, luego del análisis se ha podido determinar que el tiempo manual se reduce en un 53%, entonces, se concluye que existe un beneficio de tiempo con la utilización del software.

## RECOMENDACIONES

- Para la consecución de un producto software con el más alto porcentaje de exactitud en las unidades de código se recomienda realizar la planificación y la implementación del conjunto de pruebas unitarias necesarias para lograr el objetivo planteado.
- Para el correcto funcionamiento del sistema, se deben utilizar los navegadores actualizados debido a que existen elementos css de la interfaz que no son compatibles con versiones no actualizadas de los navegadores web.
- Para asegurar una completa cobertura del código con la herramienta JaCoCoverage las pruebas unitarias deben tener los valores necesarios inicializados de tal modo que todas ellas tengan el valor de éxito.
- Para la evaluar la funcionalidad interna del software con respecto a la calidad de datos inherente, se recomienda utilizar el estándar ISO 25012 que aborda entre una de sus métricas la exactitud.
- En un estudio futuro, para determinar en un mayor porcentaje los errores en las unidades de código y en el software, se recomienda implementar las pruebas de mutación debido a que aseguran la calidad de las pruebas al crear casos en los cuales un test unitario debería fallar.

## BIBLIOGRAFÍA

**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: 10 de Diciembre de 2018.] Disponible en: <https://xn--diseocreativo-lkb.com/developar-web-bootstrap/>.

**APIUMHUB.** *Beneficios de las pruebas unitarias*. [en línea]. 2017. [Consulta: 14 de Diciembre de 2018.] Disponible en: <https://apiumhub.com/es/tech-blog-barcelona/beneficios-de-las-pruebas-unitarias/>.

**DESAI, Avadhut.** *What is a Product Owner* [en línea]. 2018. [Consulta: 15 de Enero de 2019.] Disponible en: <https://www.quora.com/Scrum-product-development-What-is-a-Product-Owner>.

**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: 25 de Enero de 2019.] Disponible en: <https://www.scrum.org/resources/blog/el-daily-scrum-no-es-una-reunion-de-estado>.

**GONZÁLEZ PINZÓN, MIGUEL FERNANDO; & GONZÁLEZ SANABRIA, JUAN SEBASTIÁN.** "Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación". *Revista Facultad de Ingeniería*, vol. 22, n°35 (2013) , (Colombia) pp. 113-125.

**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.

**HERRERA, Dario.** *Características de un gran equipo Scrum*. [en línea]. 2018. [Consulta: 16 de Enero de 2019.] Disponible en: <https://darioherrera.com/caracteristicas-gran-equipo-scrum/>.

**JUNIT.** *JUnit 4*. [en línea]. 2018. [Consulta: 18 de Diciembre de 2018.] 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: 18 de Diciembre de 2018.] Disponible en: <https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/>.

**LAÍNEZ 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: 5 de 12 de 2018.] 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.

**MONFERRER AGUT, Raúl.** *Especificación de Requisitos Software según el estándar de IEEE 830*. [en línea]. 2001. [Consulta: 10 de Diciembre de 2018.] Disponible en: [http://zeus.inf.ucv.cl/~bcrawford/AULA\\_ICI\\_3242/ERS\\_IEEE830.pdf](http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI_3242/ERS_IEEE830.pdf).

**MUNOZ SIMO, José Manuel.** *JSF, características principales, ventajas y puntos a destacar*. [en línea]. 2012. [Consulta: 10 de Diciembre de 2018.] 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: 30 de Enero de 2019.] Disponible en: <https://jeronimopalacios.com/scrum/>.

**PANDYA, Rahul.** *Quora, What is JUnit?* [en línea]. 2017. [Consulta: 18 de Diciembre de 2018.] 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.

**PÉREZ, Alejandro.** *Scrum Master, qué es y qué no es?* [en línea]. 2017. [Consulta: 15 de Enero de 2019.] Disponible en : <http://www.ceolevel.com/scrum-master-que-es-y-que-no-es>.

**POSTGRESQL.** *PostgreSQL Tutorial*. [en línea]. 2018. [Consulta: 4 de Diciembre de 2018.] Disponible en: <http://www.postgresqltutorial.com/what-is-postgresql/>.

**QAUSTRAL.** *Manual de TestLink*. Córdoba, Argentina : QAustral, 2010.

**RAMOS VEGA, Cristina.** *Los eventos en Scrum*. [en línea]. 2017. [Consulta: 30 de Enero de 2019.] Disponible en: <https://cristinaramosvega.com/los-eventos-scrum/>.

**REQUENA MESA, Abraham.** *Qué es un sprint en scrum?* [en línea]. 2018. [Consulta: 18 de Enero de 2019.] Disponible en: <https://openwebinars.net/blog/que-es-un-sprint-scrum/>.

**RETAMOSA SANTOS, Ana.** *SCRUM, Aplicación del método ágilen lagestión de proyectos*. [en línea]. 2015. [Consulta: 30 de Enero de 2019.] Disponible en: [https://e-archivo.uc3m.es/bitstream/handle/10016/26132/TFG\\_Ana\\_Retamosa\\_Santos.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/26132/TFG_Ana_Retamosa_Santos.pdf).



**REVUELTAS, Pedro.** *El poder del sprint goal.* [en línea]. 2018. [Consulta: 24 de Enero de 2019.] 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>.

**RUIZ LIMÓN, Ramón.** *El Método Analítico.* [en línea] 2006. [Consulta: 4 de Abril de 2019.] Disponible en: <http://www.eumed.net/libros-gratis/2007a/257/7.1.htm>.

**SARA CLIP.** *Eventos en Scrum.* [en línea] 2017. [Consulta: 17 de Enero de 2019.] Disponible en: <https://www.saraclip.com/eventos-en-scrum/>.

**SCHWABER, KEN; & SUTHERLAND, JEFF.** *Guía de Scrum.* [en línea]. 2017. [Consulta: 15 de Enero de 2019.] Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>.

**SCRUM ORG.** *A Better Way Of Building Products.* [en línea]. 2013. [Consulta: 14 de Enero de 2019.] Disponible en: <https://www.scrum.org/resources/what-is-scrum>.

**SCRUM EXPERT.** *The ScrumMaster Job Description.* [en línea]. 2017. [Consulta: 15 de Enero de 2019.] Disponible en: <http://www.scrumexpert.com/knowledge/scrummaster-job-description/>.

**SCRUM INSTITUTE.** *Scrum Burndown Chart.* [en línea]. 2017. [Consulta: 31 de Enero de 2019.] Disponible en: [https://www.scrum-institute.org/Burndown\\_Chart.php](https://www.scrum-institute.org/Burndown_Chart.php).

**SOFTWARE TESTING FUNDAMENTALS.** *System Testing.* [en línea]. 2018. [Consulta: 15 de 12 de 2018.] Disponible en: <http://softwaretestingfundamentals.com/system-testing/>.

**SOMMERVILLE, Ian.** *Ingeniería del Software*. 7ª ed. Madrid-España : Pearson, 2005, pp. 108-110.

**SPURLOCK, Jake.** *Responsive Web Development, Bootstrap*. 1ª ed. California-USA : O'Reilly, 2013, pp. 1-10.

**TERRERA, Gustavo.** *Testlink – Autoaprendizaje – Parte I*. [en línea]. 2013. [Consulta: 18 de Diciembre de 2018.] Disponible en: <https://testingbaires.com/testlink-autoaprendizaje-parte-i/>.

**UNIVERSIDAD DE ALICANTE.** *El MVC en JavaServer Faces*. [en línea]. 2014. [Consulta: 7 de Diciembre de 2018.] Disponible en: <http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion02-apuntes.html#Resumen+de+los+elementos+de+JSF>.

**UNIVERSIDAD ICESI.** *Proceso de la Ingeniería de Requisitos*. [en línea]. 2008. [Consulta: 25 de Noviembre de 2018.] Disponible en: <https://danielvn7.wordpress.com/2008/03/27/proceso-de-la-ingenieria-de-requisitos/>.

## ANEXOS

### Anexo A: Historias de Usuario del proyecto SIGUION

Historia de Usuario	
<b>ID:</b> HU-01	<b>Descripción:</b> Como usuario necesito autenticar mis credenciales para ingresar al sistema.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>usuario</i> necesito autenticar mis credenciales para ingresar al sistema.	
<b>Observaciones:</b> El sistema debe permitir ingresar a las ventanas correspondientes de los roles, administrador, locutor, director.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar credenciales correctas y el sistema debe permitir el ingreso correcto.</li> <li>• Ingresar credenciales incorrectas y el sistema debe emitir el mensaje respectivo.</li> <li>• Presionar el boton ingresar sin conexión a la base de datos y el sistema debe emitir el mensaje respectivo.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-01.- Como director necesito visualizar el guion de un programa radial.		
Pendientes	En proceso	Terminadas
		T01HU-01.- Creación de la clase Usuario con los métodos de login y cerrar sesion.
		T02HU-01.-Creación del método que permita el ingreso al sistema en el controlador del sistema.
		T03HU-01.-Creación del método que permite ingresar y cerrar sesión en el modelo del sistema.
		T04HU-01.- Integrar con la interfaz gráfica correspondiente para autenticar a los usuarios.
		T05HU-01.- Diseño de mensajes de información.
		T06HU-01.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-02	<b>Descripción:</b> Como administrador del sistema necesito ingresar nuevos usuarios al sistema.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>administrador</i> del sistema necesito ingresar la información de los usuarios.	
<b>Observaciones:</b> El sistema debe permitir ingresar el registro de información de nuevos usuarios.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar datos nuevos de un usuario del sistema y el sistema debe guardar la información.</li> <li>• No ingresar datos en el formulario de ingreso y el sistema debe emitir el mensaje correspondiente.</li> <li>• Guardar información sin conexión a la base de datos, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-02.- Como administrador del sistema necesito ingresar nuevos usuarios al sistema.		
Pendientes	En proceso	Terminadas
		T01HU-02.- Creación de las clases necesarias para poder ingresar la información de usuarios nuevos al sistema.
		T02HU-02.-Creación del método que permita el ingreso de usuarios al sistema en el controlador.
		T03HU-02.-Creación del método que permite ingresar información de nuevos usuarios en el modelo.
		T04HU-02.- Integrar con la interfaz gráfica correspondiente para ingresar la nueva información.
		T05HU-02.- Diseño de mensajes de información.
		T06HU-02.- Pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>ID:</b> HU-03	<b>Descripción:</b> Como administrador del sistema necesito ingresar programas radiales al sistema.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>administrador</i> del sistema necesito ingresar la información de los programas radiales.	
<b>Observaciones:</b> El sistema debe permitir ingresar el registro de información de nuevos programas radiales.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar datos nuevos de un programa radial y el sistema debe guardar la información.</li> <li>• No ingresar datos en el formulario de ingreso y el sistema debe emitir el mensaje correspondiente.</li> <li>• Guardar información sin conexión a la base de datos, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-03.- Como administrador del sistema necesito ingresar programas radiales al sistema.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-03.- Creación de las clases necesarias para poder ingresar la información de usuarios nuevos al sistema.
		T02HU-03.-Creación del método que permita el ingreso de usuarios al sistema en el controlador.
		T03HU-03.-Creación del método que permite ingresar información de nuevos usuarios en el modelo.
		T04HU-03.- Integrar con la interfaz gráfica correspondiente para ingresar la nueva información.
		T05HU-03.- Diseño de mensajes de información.
		T06HU-03.- Pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>ID:</b> HU-04	<b>Descripción:</b> Como locutor necesito ingresar guiones de programas radiales.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 15
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 15
<b>Descripción:</b> Como <i>locutor</i> necesito ingresar la información de los guiones de los programas radiales.	
<b>Observaciones:</b> El sistema debe permitir ingresar el registro de información de un guion radial por programa radial.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar datos nuevos de un guion y el sistema debe guardar la información.</li> <li>• No ingresar datos en el formulario de ingreso y el sistema debe emitir el mensaje correspondiente.</li> <li>• Guardar información sin conexión a la base de datos, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-04.- Como locutor necesito ingresar guiones de programas radiales.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-04.- Creación de las clases necesarias para poder ingresar la información de un guion de programa radial.
		T02HU-04.-Creación del método que permita el ingreso de un guion en el controlador del sistema.
		T03HU-04.-Creación del método que permite ingresar información de un guion de programa de radio en el modelo del sistema.
		T04HU-04.- Integrar con la interfaz gráfica correspondiente para ingresar la nueva información.
		T05HU-04.- Diseño de mensajes de información.
		T06HU-04.- Pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>ID:</b> HU-05	<b>Descripción:</b> Como director necesito visualizar el guion de un programa radial.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>director</i> necesito visualizar el guion de un programa radial.	
<b>Observaciones:</b> El sistema debe permitir consultar la información de un guion de programa radial y visualizar.	
<b>Pruebas de Aceptación:</b>	
<ul style="list-style-type: none"> <li>• Ingresar la fecha de registro del guion y el sistema debe consultar la información requerida.</li> <li>• No ingresar datos en el formulario de consulta y el sistema debe emitir el mensaje correspondiente.</li> </ul>	

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-06.- Como director necesito visualizar el guion de un programa radial.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-05.- Creación de las clases necesarias para poder consultar la información de un guion por fecha de registro.
		T02HU-05.-Creación del método que permita la consulta de un guion de programa de radio en el controlador del sistema.
		T03HU-05.-Creación del método que permite consultar la información de un guion de programa de radio en el modelo del sistema.
		T04HU-05.- Integrar con la interfaz gráfica correspondiente para visualizar la información almacenada de un guion.
		T05HU-05.- Diseño de mensajes de información.
		T06HU-05.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-06	<b>Descripción:</b> Como director necesito realizar la aprobación del guion de un programa radial.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>director</i> necesito realizar la aprobación del guion de un programa radial.	
<b>Observaciones:</b> El sistema debe permitir ingresar el estado de Aprobado a un guion de programa radial consultado.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>Ingresar el estado del guion Aprobado y el sistema debe guardar la información.</li> <li>Sin conexión a la base de datos ingresar el estado del guion Aprobado y el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-06.- Como director necesito realizar la aprobación del guion de un programa radial.		
Pendientes	En proceso	Terminadas
		T01HU-06.- Creación de las clases necesarias para poder ingresar nuevos estados a un guion de programa de radio.
		T02HU-06.-Creación del método, en el controlador del sistema, que permita el ingreso de un nuevo estado a un guion.
		T03HU-06.- Creación del método, en el modelo del sistema, que permita el ingreso de un nuevo estado a un guion.
		T04HU-06.- Integrar con la interfaz gráfica correspondiente para ingresar el nuevo estado de un guion.
		T05HU-06.- Diseño de mensajes de información.
		T06HU-06.- Pruebas de aceptación.



Historia de Usuario	
<b>ID:</b> HU-07	<b>Descripción:</b> Como director necesito ingresar el estado de No Aprobado a un guion del programa radial.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>director</i> necesito ingresar el estado de No Aprobado a un guion del programa radial.	
<b>Observaciones:</b> El sistema debe permitir ingresar el estado de No Aprobado a un guion de programa radial consultado.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar el estado del guion No Aprobado y el sistema debe guardar la información.</li> <li>• Sin conexión a la base de datos ingresar el estado del guion No Aprobado y el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-07.- Como director necesito ingresar el estado de No Aprobado a un guion del programa radial.		
Pendientes	En proceso	Terminadas
		T01HU-07.- Creación de las clases necesarias para poder ingresar nuevos estados a un guion de programa de radio.
		T02HU-07.-Creación del método, en el controlador del sistema, que permita el ingreso de un nuevo estado a un guion.
		T03HU-07.- Creación del método, en el modelo del sistema, que permita el ingreso de un nuevo estado a un guion.
		T04HU-07.- Integrar con la interfaz gráfica correspondiente para ingresar el nuevo estado de un guion.
		T05HU-07.- Diseño de mensajes de información.
		T06HU-07.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-08	<b>Descripción:</b> Como director necesito ingresar el estado de Revisado a un guion del programa radial.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>director</i> necesito ingresar el estado de Revisado a un guion del programa radial.	
<b>Observaciones:</b> El sistema debe permitir ingresar el estado de Revisado a un guion de programa radial consultado.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar el estado del guion Revisado y el sistema debe guardar la información.</li> <li>• Sin conexión a la base de datos ingresar el estado del guion Revisado y el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-08.- Como director necesito ingresar el estado de Revisado a un guion del programa radial.		
Pendientes	En proceso	Terminadas
		T01HU-08.- Creación de las clases necesarias para poder ingresar nuevos estados a un guion de programa de radio.
		T02HU-08.-Creación del método, en el controlador del sistema, que permita el ingreso de un nuevo estado a un guion.
		T03HU-08.- Creación del método, en el modelo del sistema, que permita el ingreso de un nuevo estado a un guion.
		T04HU-08.- Integrar con la interfaz gráfica correspondiente para ingresar el nuevo estado de un guion.
		T05HU-08.- Diseño de mensajes de información.
		T06HU-08.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-09	<b>Descripción:</b> Como director necesito ingresar las sugerencias respectivas al guion de un programa de la radio.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 15
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 15
<b>Descripción:</b> Como <i>director</i> necesito ingresar las sugerencias respectivas al guion de un programa de la radio.	
<b>Observaciones:</b> El sistema debe permitir ingresar sugerencias a un guion de programa de radio.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>Ingresar las sugerencias respectivas al guion, el sistema debe emitir el mensaje que confirma que los datos han sido guardados.</li> <li>Sin conexión a la base de datos ingresar las sugerencias de un guion, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-09.- Como director necesito ingresar las sugerencias respectivas al guion de un programa de la radio.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-09.- Creación de las clases necesarias para poder ingresar las sugerencias a un guion de programa de radio.
		T02HU-09.-Creación del método, en el controlador del sistema, que permita el ingreso de sugerencias a un guion.
		T03HU-09.- Creación del método, en el modelo del sistema, que permita el ingreso de sugerencias a un guion.
		T04HU-09.- Integrar con la interfaz gráfica correspondiente para ingresar las sugerencias de un guion.
		T05HU-09.- Diseño de mensajes de información.
		T06HU-09.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-10	<b>Descripción:</b> Como locutor necesito visualizar las sugerencias del guion de un programa radio.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 15
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 15
<b>Descripción:</b> Como <i>locutor</i> necesito visualizar las sugerencias de guion de un programa de la radio.	
<b>Observaciones:</b> El sistema debe permitir visualizar las sugerencias de un guion de programa de radio.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar la fecha de registro de un guion, el sistema debe visualizar las sugerencias de un programa radial.</li> <li>• Ingresar una fecha de registro de guion incorrecta, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-10.- Como locutor necesito visualizar las sugerencias del guion de un programa radio.		
Pendientes	En proceso	Terminadas
		T01HU-10.- Creación de las clases necesarias para poder visualizar las sugerencias del guion.
		T02HU-10.-Creación del método, en el controlador del sistema, que permita el consultar las sugerencias de un guion.
		T03HU-10.- Creación del método, en el modelo del sistema, que permita el consultar las sugerencias de un guion.
		T04HU-10.- Integrar con la interfaz gráfica correspondiente para visualizar las sugerencias de un guion.
		T05HU-10.- Diseño de mensajes de información.
		T06HU-10.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-11	<b>Descripción:</b> Como locutor necesito modificar el guion de un programa de radio.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>locutor</i> necesito modificar el guion de un programa de radio.	
<b>Observaciones:</b> El sistema debe modificar el guion de un programa de radio.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar los datos modificados para el guion, el sistema debe emitir el mensaje de que los datos han sido guardados.</li> <li>• Ingresar los datos modificados para el guion, el sistema sin conexión a la base de datos, debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-11.- Como locutor necesito modificar el guion de un programa de radio.		
Pendientes	En proceso	Terminadas
		T01HU-11.- Creación de las clases necesarias para poder actualizar la información de un programa radial.
		T02HU-11.-Creación del método, en el controlador del sistema, que permita el actualizar la información de un guion.
		T03HU-11.- Creación del método, en el modelo del sistema, que permita el actualizar la información de un guion.
		T04HU-11.- Integrar con la interfaz gráfica correspondiente para modificar la información de un guion.
		T05HU-11.- Diseño de mensajes de información.
		T06HU-11.- Pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>ID:</b> HU-12	<b>Descripción:</b> Como locutor necesito realizar los respectivos cambios al guion de un programa de radio.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>locutor</i> necesito realizar los respectivos cambios al guion de un programa de radio.	
<b>Observaciones:</b> El sistema debe actualizar el guion de un programa de radio.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar los datos modificados para el guion, el sistema debe emitir el mensaje de que los datos han sido guardados.</li> <li>• Ingresar los datos modificados para el guion, el sistema sin conexión a la base de datos, debe emitir el mensaje correspondiente.</li> </ul>	

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-12.- Como locutor necesito realizar los respectivos cambios al guion de un programa de radio.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-12.- Creación de las clases necesarias para poder actualizar la información de un programa radial.
		T02HU-12.- Creación del método, en el controlador del sistema, que permita el actualizar la información de un guion.
		T03HU-12.- Creación del método, en el modelo del sistema, que permita el actualizar la información de un guion.
		T04HU-12.- Integrar con la interfaz gráfica correspondiente para modificar la información de un guion.
		T05HU-12.- Diseño de mensajes de información.
		T06HU-12.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-13	<b>Descripción:</b> Como locutor necesito visualizar el estado del guion.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>locutor</i> necesito visualizar el estado del guion.	
<b>Observaciones:</b> El sistema debe visualizar el estado correspondiente a un guion.	
<b>Pruebas de Aceptación:</b>	
<ul style="list-style-type: none"> <li>• Ingresar la fecha de registro del guion, el sistema debe visualizar el estado en el que se encuentra el guion consultado.</li> <li>• Ingresar la fecha de registro del guion incorrecta, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-13.- Como locutor necesito visualizar el estado del guion.		
Pendientes	En proceso	Terminadas
		T01HU-13.- Creación de las clases necesarias para poder visualizar la información del estado de un programa radial.
		T02HU-13.-Creación del método, en el controlador del sistema, que permita visualizar el estado de un guion.
		T03HU-13.- Creación del método, en el modelo del sistema, que permita visualizar el estado de un guion.
		T04HU-13.- Integrar con la interfaz gráfica correspondiente para poder visualizar el estado de un guion.
		T05HU-13.- Diseño de mensajes de información.
		T06HU-13.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-14	<b>Descripción:</b> Como administrador del sistema necesito modificar los programas radiales.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>administrador</i> del sistema necesito modificar los programas radiales.	
<b>Observaciones:</b> El sistema debe permitir modificar la información de los programas radiales.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>Ingresar en el formulario de actualización de la información de un programa radial la nueva información y presionar en el boton guardar, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-14.- Como administrador del sistema necesito modificar los programas radiales.		
Pendientes	En proceso	Terminadas
		T01HU-14.- Creación de las clases necesarias para poder actualizar la información de los programas radiales.
		T02HU-14.-Creación del método, en el controlador del sistema, que permita actualizar la información de los programas radiales.
		T03HU-14.- Creación del método, en el modelo del sistema, que permita actualizar la información de los programas radiales.
		T04HU-14.- Integrar con la interfaz gráfica correspondiente para poder actualizar el estado de un guion.
		T05HU-14.- Diseño de mensajes de información.
		T06HU-14.- Pruebas de aceptación.



Historia de Usuario	
<b>ID:</b> HU-15	<b>Descripción:</b> Como administrador del sistema necesito eliminar los programas radiales.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>administrador</i> del sistema necesito eliminar los programas radiales.	
<b>Observaciones:</b> El sistema debe permitir eliminar la información de los programas radiales.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>Consultar la información del programa radial, dar clic en el boton Eliminar, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-15.- Como administrador del sistema necesito eliminar los programas radiales.		
Pendientes	En proceso	Terminadas
		T01HU-15.- Creación de las clases necesarias para poder eliminar la información de los programas radiales.
		T02HU-15.-Creación del método, en el controlador del sistema, que permita eliminar la información de los programas radiales.
		T03HU-15.- Creación del método, en el modelo del sistema, que permita eliminar la información de los programas radiales.
		T04HU-15.- Integrar con la interfaz gráfica correspondiente para poder actualizar el estado de un guion.
		T05HU-15.- Diseño de mensajes de información.
		T06HU-15.- Pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>ID:</b> HU-16	<b>Descripción:</b> Como administrador del sistema necesito modificar la información de los usuarios.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>administrador</i> del sistema necesito modificar la información de los usuarios.	
<b>Observaciones:</b> El sistema debe permitir modificar la información de los usuarios registrados en el sistema.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Ingresar en el formulario de actualización de información de usuarios, los campos requeridos, dar clic en el boton guardar, el sistema debe emitir el mensaje de confirmación de la acción.</li> <li>• Dejar espacios en blanco en el formulario de actualización, dar clic en el boton, guarda el sistema debe emitir el mensaje respectivo.</li> </ul>	

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-16.- Como administrador del sistema necesito modificar la información de los usuarios.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-16.- Creación de las clases necesarias para poder modificar la información de los usuarios.
		T02HU-16.-Creación del método, en el controlador del sistema, que permita modificar la información de los usuarios.
		T03HU-16.- Creación del método, en el modelo del sistema, que permita modificar la información de los usuarios.
		T04HU-16.- Integrar con la interfaz gráfica correspondiente para poder actualizar la información de los usuarios.
		T05HU-16.- Diseño de mensajes de información.
		T06HU-16.- Pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>ID:</b> HU-17	<b>Descripción:</b> Como administrador del sistema necesito eliminar los usuarios ingresados.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>administrador</i> del sistema necesito eliminar los usuarios ingresados.	
<b>Observaciones:</b> El sistema debe permitir eliminar la información de los usuarios registrados.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Buscar por el número de cedula de identidad la información de un usuario, al dar clic el boton eliminar, el sistema debe emitir un mensaje que confirme la acción.</li> <li>• Sin conexión a la base de datos, buscar el usuario y dar clic en el boton eliminar, el sistema debe emitir el mensaje correspondiente.</li> </ul>	

<b>Tareas de ingeniería</b>		
<b>Descripción:</b> HU-17.- Como administrador del sistema necesito eliminar los usuarios ingresados.		
<b>Pendientes</b>	<b>En proceso</b>	<b>Terminadas</b>
		T01HU-17.- Creación de las clases necesarias para poder eliminar la información de los usuarios.
		T02HU-17.-Creación del método, en el controlador del sistema, que permita eliminar la información de los usuarios.
		T03HU-17.- Creación del método, en el modelo del sistema, que permita eliminar la información de los usuarios.
		T04HU-17.- Integrar con la interfaz gráfica correspondiente para poder eliminar la información de los usuarios.
		T05HU-17.- Diseño de mensajes de información.
		T06HU-17.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-18	<b>Descripción:</b> Como locutor necesito eliminar un guion del programa radial.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>locutor</i> necesito eliminar un guion del programa radial.	
<b>Observaciones:</b> El sistema debe permitir eliminar el guion de un programa radial.	
<b>Pruebas de Aceptación:</b>	
<ul style="list-style-type: none"> <li>• Buscar el guion por fecha de registro, dar clic en eliminar guion, el sistema debe emitir el mensaje de confirmación de la acción.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-18.- Como <i>locutor</i> necesito eliminar un guion del programa radial.		
Pendientes	En proceso	Terminadas
		T01HU-18.- Creación de las clases necesarias para poder eliminar la información del guion correspondiente.
		T02HU-18.-Creación del método, en el controlador del sistema, que permita eliminar la información que pertenece a un guion consultado.
		T03HU-18.- Creación del método, en el modelo del sistema, que permita eliminar la información que pertenece a un guion consultado.
		T04HU-18.- Integrar con la interfaz gráfica correspondiente para poder eliminar la información que pertenece a un guion consultado.
		T05HU-18.- Diseño de mensajes de información.
		T06HU-18.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-19	<b>Descripción:</b> Como director necesito realizar el reporte de guiones no aprobados en la semana en formato pdf.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>director</i> necesito realizar el reporte de guiones No Aprobados en la semana en formato pdf.	
<b>Observaciones:</b> El sistema debe permitir exportar en formato pdf un reporte semanal de guiones No Aprobados.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Dar clic en generar reporte de Guiones No Aprobados, el sistema debe generar un reporte pdf con una lista de guiones No Aprobados.</li> <li>• Sin conexión a la base de datos, dar clic en generar reporte de Guiones No Aprobados, el sistema debe emitir el respectivo mensaje.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-19.- Como <i>director</i> necesito realizar el reporte de guiones No Aprobados en la semana en formato pdf.		
Pendientes	En proceso	Terminadas
		T01HU-19.- Creación de las clases necesarias para poder consultar los guiones que corresponden a la semana con el estado de No Aprobado.
		T02HU-19.-Creación del método, en el controlador del sistema, que permita consultar una lista de guiones con el estado de No Aprobados.
		T03HU-19.- Creación del método, en el modelo del sistema, que permita generar la consulta del listado de guiones No aprobados en la semana.
		T04HU-19.- Integrar con la interfaz gráfica correspondiente para poder generar el reporte requerido.
		T05HU-19.- Diseño de mensajes de información.
		T06HU-19.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-20	<b>Descripción:</b> Como director necesito realizar el reporte de los guiones aprobados en la semana en formato pdf.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>director</i> necesito realizar el reporte de los guiones Aprobados en la semana en formato pdf.	
<b>Observaciones:</b> El sistema debe permitir exportar en formato pdf un reporte semanal de guiones Aprobados.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Dar clic en generar reporte de Guiones Aprobados, el sistema debe generar un reporte pdf con una lista de guiones Aprobados.</li> <li>• Sin conexión a la base de datos, dar clic en generar reporte de Guiones Aprobados, el sistema debe emitir el respectivo mensaje.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-20.- Como <i>director</i> necesito realizar el reporte de guiones Aprobados en la semana en formato pdf.		
Pendientes	En proceso	Terminadas
		T01HU-20.- Creación de las clases necesarias para poder consultar los guiones que corresponden a la semana con el estado de Aprobado.
		T02HU-20.-Creación del método, en el controlador del sistema, que permita consultar una lista de guiones con el estado de Aprobados.
		T03HU-20.- Creación del método, en el modelo del sistema, que permita generar la consulta del listado de guiones Aprobados en la semana.
		T04HU-20.- Integrar con la interfaz gráfica correspondiente poder generar el reporte requerido.
		T05HU-20.- Diseño de mensajes de información.
		T06HU-20.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-21	<b>Descripción:</b> Como director necesito realizar el reporte de los guiones revisados en la semana en formato pdf.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 12
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 12
<b>Descripción:</b> Como <i>director</i> necesito realizar el reporte de los guiones Revisados en la semana en formato pdf.	
<b>Observaciones:</b> El sistema debe permitir exportar en formato pdf un reporte semanal de guiones Revisados.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>• Dar clic en generar reporte de Guiones Revisados, el sistema debe generar un reporte pdf con una lista de guiones Revisados.</li> <li>• Sin conexión a la base de datos, dar clic en generar reporte de Guiones Revisados, el sistema debe emitir el respectivo mensaje.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-21.- Como <i>director</i> necesito realizar el reporte de guiones Revisados en la semana en formato pdf.		
Pendientes	En proceso	Terminadas
		T01HU-21.- Creación de las clases necesarias para poder consultar los guiones que corresponden a la semana con el estado de Revisado.
		T02HU-21.- Creación del método, en el controlador del sistema, que permita consultar una lista de guiones con el estado de Revisados.
		T03HU-21.- Creación del método, en el modelo del sistema, que permita generar la consulta del listado de guiones Revisados en la semana.
		T04HU-21.- Integrar con la interfaz gráfica correspondiente poder generar el reporte requerido.
		T05HU-21.- Diseño de mensajes de información.
		T06HU-21.- Pruebas de aceptación.

Historia de Usuario	
<b>ID:</b> HU-22	<b>Descripción:</b> Como director del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>director</i> del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	
<b>Observaciones:</b> El sistema debe permitir exportar en formato pdf un reporte de guiones aprobados que coincida con una fecha de registro.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>Ingresar una fecha de registro, dar clic en el boton Generar Reporte Guion Aprobado, en caso de tener registro de información, el sistema debe generar el reporte pdf, caso contrario, el sistema debe emitir el mensaje respectivo.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-22.- Como <i>director</i> del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.		
Pendientes	En proceso	Terminadas
		T01HU-22.- Creación de las clases necesarias para poder consultar los guiones que corresponden con el estado de Aprobado a partir de una fecha de registro.
		T02HU-22.-Creación del método, en el controlador del sistema, que permita consultar una lista de guiones con el estado de Aprobado a partir de una fecha de registro.
		T03HU-22.- Creación del método, en el modelo del sistema, que permita consultar una lista de guiones con el estado de Aprobado a partir de una fecha de registro.
		T04HU-22.- Integrar con la interfaz gráfica correspondiente poder generar el reporte requerido.
		T05HU-22.- Diseño de mensajes de información.
		T06HU-22.- Pruebas de aceptación.



Historia de Usuario	
<b>ID:</b> HU-23	<b>Descripción:</b> Como director del sistema necesito realizar el reporte de los guiones no aprobados que coincidan con una fecha de registro.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Director.	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>director</i> del sistema necesito realizar el reporte de los guiones no aprobados que coincidan con una fecha de registro.	
<b>Observaciones:</b> El sistema debe permitir exportar en formato pdf un reporte de guiones no aprobados que coincida con una fecha de registro.	
<b>Pruebas de Aceptación:</b> <ul style="list-style-type: none"> <li>Ingresar una fecha de registro, dar clic en el boton Generar Reporte Guion No Aprobado, en caso de tener registro de información, el sistema debe generar el reporte pdf, caso contrario, el sistema debe emitir el mensaje respectivo.</li> </ul>	

Tareas de ingeniería		
<b>Descripción:</b> HU-23.- Como <i>director</i> del sistema necesito realizar el reporte de los guiones no aprobados que coincidan con una fecha de registro.		
Pendientes	En proceso	Terminadas
		T01HU-23.- Creación de las clases necesarias para poder consultar los guiones que corresponden con el estado de No Aprobado a partir de una fecha de registro.
		T02HU-23.- Creación del método, en el controlador del sistema, que permita consultar una lista de guiones con el estado de No Aprobado a partir de una fecha de registro.
		T03HU-23.- Creación del método, en el modelo del sistema, que permita consultar una lista de guiones con el estado de No Aprobado a partir de una fecha de registro.
		T04HU-23.- Integrar con la interfaz gráfica correspondiente poder generar el reporte requerido.
		T05HU-23.- Diseño de mensajes de información.
		T06HU-23.- Pruebas de aceptación.

**Anexo B:** Gestión de Riesgos del proyecto SIGUIÓN.

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R1		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Media <b>Valor:</b> 2	<b>Exposición:</b> Baja <b>Valor:</b> 2	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Falta de compromiso por parte del usuario en el establecimiento de las bases para el desarrollo del sistema.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Falta de compromiso con el cliente para las reuniones de recolección de información. Información incompleta en los requisitos expuestos. Búsqueda de información parcial con respecto a las necesidades del sistema.			
<b>Consecuencias:</b> No satisfacer las necesidades del proyecto. Mal diseño del proyecto. Información inconsistente para el desarrollo.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Revisión de la documentación necesaria para definir correctamente el alcance de los requisitos. Revisión continua de los requerimientos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Revisión continua de los requerimientos con los stakeholders involucrados en el proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R2		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Media <b>Valor:</b> 2	<b>Exposición:</b> Alta <b>Valor:</b> 8	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Recolección de requerimientos incompletos o ambiguos.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Falta de compromiso con el cliente para las reuniones de recolección de información. Información incompleta en los requisitos expuestos. Búsqueda de información parcial con respecto a las necesidades del sistema.			
<b>Consecuencias:</b> Mal diseño del proyecto. Re-planificación periódica del sistema. No cumplir con las funcionalidades del sistema.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Revisión de la documentación necesaria para definir correctamente el alcance de los requisitos. Revisión continua de los requerimientos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Revisión continua de los requerimientos con los stakeholders involucrados en el proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R3		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Media <b>Valor:</b> 4	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Modificación continua de requerimientos.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Estudio detallado ambiguo de requerimientos debido a la falta de madurez en el proceso. Definición deficiente de los requerimientos del sistema.			
<b>Consecuencias:</b> Re-planificación periódica del sistema. Retrasos en el tiempo de entrega de los incrementos de software. No cumplir con las funcionalidades del sistema.			
<b>REDUCCIÓN:</b> Revisión del estudio de los requerimientos del sistema. Corrección de los requerimientos mal definidos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Revisión continua de los requerimientos con los stakeholders involucrados en el proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada <input type="checkbox"/>			
Fase de Supervisión iniciada <input checked="" type="checkbox"/>			
Gestionando el riesgo <input type="checkbox"/>			
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R4		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Media <b>Valor:</b> 5	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Incorrecta definición de la base de datos.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Recolección de información incompleta. Mala definición de requerimientos del sistema. Mala comprensión de los procesos ha automatizar.			
<b>Consecuencias:</b> Mal diseño de la base de datos. Re-construcción de la base de datos. No cumplir con las funcionalidades del sistema.			
<b>REDUCCIÓN:</b> Reunión con los stakeholders para entender de mejor manera los procesos ha automatizar. Modificar los puntos de la definición de la base de datos incorrectos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Cumplir el protocolo de gestión establecido.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada <input type="checkbox"/>			
Fase de Supervisión iniciada <input checked="" type="checkbox"/>			
Gestionando el riesgo <input type="checkbox"/>			
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R5		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Media <b>Valor:</b> 4	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Subestimación del tamaño de la aplicación.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Desconocimiento del verdadero alcance del proyecto. Mala planificación de los incrementos del sistema. Estudio del tamaño del proyecto deficiente.			
<b>Consecuencias:</b> Incremento en el tiempo de planificación para las tareas de desarrollo. Retrasos en la finalización de los incrementos. Incremento en el valor del esfuerzo y recursos necesarios para el proyecto.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Negociar asignación de recursos necesarios para cumplir con las metas establecidas. Re-planificación del proyecto.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Ejecutar el protocolo de reducción del riesgo.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada <input type="checkbox"/>			
Fase de Supervisión iniciada <input checked="" type="checkbox"/>			
Gestionando el riesgo <input type="checkbox"/>			
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R6		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Baja <b>Valor:</b> 1	<b>Exposición:</b> Baja <b>Valor:</b> 2	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Subestimación de costos en el presupuesto del proyecto.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Mala estimación de recursos necesarios para el desarrollo del sistema. Información incompleta en los requisitos expuestos. Cambio constante de necesidades de recursos para el desarrollo del proyecto.			
<b>Consecuencias:</b> Perdidas económicas. Cancelación del proyecto.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Revisión de la documentación necesaria para definir correctamente el alcance de los requisitos. Revisión continua de los requerimientos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Revisión continua de los requerimientos con los stakeholders involucrados en el proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada <input type="checkbox"/>			
Fase de Supervisión iniciada <input checked="" type="checkbox"/>			
Gestionando el riesgo <input type="checkbox"/>			
<b>RESPONSABLE:</b> Jhonathan Chela			

HOJA DE GESTIÓN DEL RIESGO			
<b>ID. DEL RIESGO:</b> R7		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Media <b>Valor:</b> 2	<b>Exposición:</b> Baja <b>Valor:</b> 2	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Diseño de interfaces incompleto.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Falta de madurez para el uso de tecnologías propias para el diseño de la interfaz. Asignación de prioridad al trabajo de diseño de interfaces insuficiente. Especificaciones incompletas de las interfaces necesarias.			
<b>Consecuencias:</b> Retraso en la entrega del proyecto. Re-planificación del tiempo de entrega. Insatisfacción del cliente.			
<b>REDUCCIÓN:</b> Revisión continua de las especificaciones de las interfaces. Capacitación en las tecnologías de diseño de interfaces desconocidas.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Ejecución de los pasos de ejecución establecidos. Monitoreo constante de las debilidades encontradas en el proceso de diseño de interfaces.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

HOJA DE GESTIÓN DEL RIESGO			
<b>ID. DEL RIESGO:</b> R8		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Media <b>Valor:</b> 4	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Falta de especificación de la arquitectura lógica del sistema.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Mal análisis del funcionamiento del sistema. Especificación de las necesidades del sistema ambigua o insuficiente. Mala definición de requerimientos.			
<b>Consecuencias:</b> Incremento en el tiempo de desarrollo. Retraso en la entrega del proyecto. Re-planificación del proyecto.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Revisión de la documentación necesaria para definir correctamente la arquitectura lógica del sistema. Monitoreo continuo en el proceso de especificación de la arquitectura lógica del sistema.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Ejecución de los procesos de reducción establecidos.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

HOJA DE GESTIÓN DEL RIESGO			
<b>ID. DEL RIESGO:</b> R9		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Media <b>Valor:</b> 2	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Alta <b>Valor:</b> 6	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Retrasos en el cumplimiento de tiempos del sprint.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Falta de compromiso con el cumplimiento de los objetivos del sprint. El tiempo asignado para el proceso de desarrollo del sprint es utilizado en resolver problemas de sprints anteriores. Procesos inmaduros con respecto a la tecnología de desarrollo del sistema.			
<b>Consecuencias:</b> Insatisfacción en el cliente. Incremento del tiempo total para culminar el proyecto.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto. Negociar una re-planificación de tiempos de entrega de los incrementos de los productos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Ejecución de los lineamientos establecidos de acuerdo a la metodología de desarrollo establecida.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada <input type="checkbox"/>			
Fase de Supervisión iniciada <input checked="" type="checkbox"/>			
Gestionando el riesgo <input type="checkbox"/>			
<b>RESPONSABLE:</b> Jhonathan Chela			

HOJA DE GESTIÓN DEL RIESGO			
<b>ID. DEL RIESGO:</b> R10		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Media <b>Valor:</b> 2	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Media <b>Valor:</b> 5	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Falta de conocimiento y experiencia sobre las tareas asignadas y las herramientas a utilizar.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Falta de conocimiento del lenguaje de programación. Inexperiencia en el uso de tecnologías de desarrollo especificadas para el proyecto.			
<b>Consecuencias:</b> Retraso en el proyecto. Re-planificación en el proyecto.			
<b>REDUCCIÓN:</b> Planificar sesiones de capacitación rápidas con respecto a las tecnologías utilizadas en el desarrollo de proyecto.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Ejecución de los pasos de reducción planificados.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada <input type="checkbox"/>			
Fase de Supervisión iniciada <input checked="" type="checkbox"/>			
Gestionando el riesgo <input type="checkbox"/>			
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R11		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Media <b>Valor:</b> 2	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Media <b>Valor:</b> 4	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Pérdida de backups o archivos de respaldo del proyecto.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> No existe un protocolo de backup para el proyecto. Inexperiencia en el uso de herramientas para realizar respaldos del proyecto.			
<b>Consecuencias:</b> Falta de respaldo del sistema en caso de pérdida total del proyecto. Cancelación del proyecto.			
<b>REDUCCIÓN:</b> Implementar tecnología de backup para el proyecto en caso de no existir alguno. Actualizar las herramientas y lineamientos de respaldo del proyecto si existe un sistema de backup antiguo u obsoleto.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el monitoreo continuo del proceso de respaldo del proyecto.			
<b>GESTIÓN:</b> Aplicar los lineamientos y protocolos establecidos para el respaldo del proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R12		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Media <b>Valor:</b> 2	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Alta <b>Valor:</b> 6	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Alcance de las pruebas unitarias automatizadas no definidas correctamente.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Desconocimiento de la programación de las pruebas unitarias. Inmadurez en el uso de pruebas unitarias en un proyecto. Pruebas unitarias mal construidas.			
<b>Consecuencias:</b> No probar la funcionalidad completa de un ítem planificado. Volver a realizar las pruebas. Consumir más tiempo de lo planificado para las pruebas.			
<b>REDUCCIÓN:</b> Aplicar mejores prácticas sobre la realización de pruebas unitarias en el desarrollo del sistema. Definir el alcance de las pruebas unitarias correctamente y de manera priorizada.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control del alcance de las pruebas unitarias construidas.			
<b>GESTIÓN:</b> Aplicar los ítems de reducción y supervisión del riesgo planificado.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R13		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Alta <b>Valor:</b> 3	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Alta <b>Valor:</b> 8	<b>Prioridad:</b> Alta <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Inmadurez en la priorización sobre la ejecución de pruebas.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Mala priorización de los casos de pruebas unitarias a ejecutar. Excesivos casos de pruebas unitarias a automatizar, incrementando el tiempo de entrega del producto.			
<b>Consecuencias:</b> Re-planificación de las pruebas dentro del proyecto. Paro temporal en el desarrollo del proyecto			
<b>REDUCCIÓN:</b> Priorización de casos de prueba para unidades de código funcionales del sistema.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de la ejecución de las pruebas unitarias establecidas.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Revisión continua de los requerimientos con los stakeholders involucrados en el proyecto.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			

<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R14		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Baja <b>Valor:</b> 2	<b>Prioridad:</b> Media <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Excesivo uso del tiempo en la reparación de defectos de las pruebas unitarias automatizadas.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b> Planificación deficiente para la generación y ejecución de los casos de pruebas unitarias.			
<b>Consecuencias:</b> Paro temporal en el desarrollo del proyecto. Retraso en la entrega del proyecto.			
<b>REDUCCIÓN:</b> Planificar nuevas reuniones con el dueño del producto para la re-planificación de tiempo de entrega. Revisión de la documentación necesaria para definir correctamente el alcance de los requisitos. Revisión continua de los requerimientos.			
<b>SUPERVISIÓN</b> Seguir los ítems de reducción planificados. Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b> Reuniones y socialización con el dueño del producto. Priorización de los casos de pruebas unitarias.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b> Jhonathan Chela			



<b>HOJA DE GESTIÓN DEL RIESGO</b>			
<b>ID. DEL RIESGO:</b> R15		<b>FECHA:</b> 06/01/2019	
<b>Probabilidad:</b> Baja <b>Valor:</b> 1	<b>Impacto:</b> Alta <b>Valor:</b> 3	<b>Exposición:</b> Alta <b>Valor:</b> 6	<b>Prioridad:</b> Alta <b>Valor:</b> 3
<b>DESCRIPCIÓN:</b> Retraso en la realización de pruebas por errores después del despliegue del sistema.			
<b>REFINAMIENTO:</b>			
<b>Causas:</b>			
Errores de funcionalidad en las unidades de código no solucionados.			
Errores de sintaxis en el código de las unidades de software que no son identificados y solucionados.			
<b>Consecuencias:</b>			
Paro temporal en el desarrollo del proyecto.			
Retraso en la entrega del proyecto.			
Falta de integración de las unidades que componen una versión del sistema.			
<b>REDUCCIÓN:</b>			
Identificación temprana de los errores en las unidades de código antes del proceso de integración.			
Monitorización de la ejecución de las pruebas unitarias.			
<b>SUPERVISIÓN</b>			
Seguir los ítems de reducción planificados.			
Realizar el control de los requerimientos establecidos.			
<b>GESTIÓN:</b>			
Ejecutar los pasos de reducción y supervisión planificados para este riesgo.			
<b>ESTADO ACTUAL:</b>			
Fase de reducción iniciada	<input type="checkbox"/>		
Fase de Supervisión iniciada	<input checked="" type="checkbox"/>		
Gestionando el riesgo	<input type="checkbox"/>		
<b>RESPONSABLE:</b>			
Jhonathan Chela			

**SISTEMA WEB DE GESTIÓN DE GUIONES PARA LA RADIO CRISÓLITO  
CELESTIAL 93.1 FM DE LA CIUDAD DE GUARANDA.**

**SIGUION**

**V.1.1**

**APLICACIÓN WEB**

**ÁREA DE DESARROLLO**

**2019**

**MANUAL DE USUARIO**

**Introducción**

El presente documento es un registro del funcionamiento del sistema web de gestión de guiones realizado para la Radio Crisólito Celestial 93.1FM.

Este proyecto es liberado en su versión 1.1, y ha sido desarrollado con tecnología OpenSource, se ha utilizado el lenguaje JAVA, una base de datos construida en POSTGRESQL, la tecnología de JSF y para el diseño de una interfaz responsiva el framework Bootstrap.

Para el uso del sistema web denominado SIGUION, se define tres tipos de roles de usuario que el sistema acepta, estos son: administrador del sistema, director de la entidad y locutor de la radio.

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, luego se aborda las funcionalidades propias del administrador del sistema; además en la parte final se aborda las funciones que el sistema permite al director y locutor de la entidad.

## 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 como se muestra en la imagen que se adjunta:




La pantalla de ingreso al sistema consta de un formulario con una imagen que identifica la entidad de radio, y dos cajas de texto que permiten ingresar las credenciales, en este caso, la credencial de usuario es especificado por parte del administrador del sistema de la entidad y la contraseña que el usuario asigna al sistema.

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 correspondientes, ejemplo:



## Perfil del Administrador del sistema

El perfil del administrador del sistema permite realizar tareas como: la gestión de usuarios del sistema, la gestión de programas radiales que se transmiten en la entidad. Una vez ingresado las credenciales correctas y el sistema de paso al perfil de administrador del sistema, se puede observar el área de trabajo que se adjunta en la imagen:



The screenshot shows the 'SIGUION -- ADMINISTRADOR' dashboard. The user is identified as 'Jhonathan Chela(Administrador)' and is 'CONECTADO'. The dashboard features a 'Resumen' section with four key metrics: 10 NUEVOS PROGRAMAS RADIALES, 52 EDICIONES EN GENERAL, 24 NUEVOS USUARIOS, and 25 Búsquedas. Below this is a table titled 'Usuarios Registrados en el Sistema' with the following data:

ID	NOMBRE	APELLIDO	CEDULA	PASSWORD	TIPO DE USUARIO
70	Fanny	Chela	020154879-2	123456	Locutor
59	Giovanni	Curi	020036251-0	123456	Locutor
58	Daniel	Chimbo	020089749-5	123456	Locutor

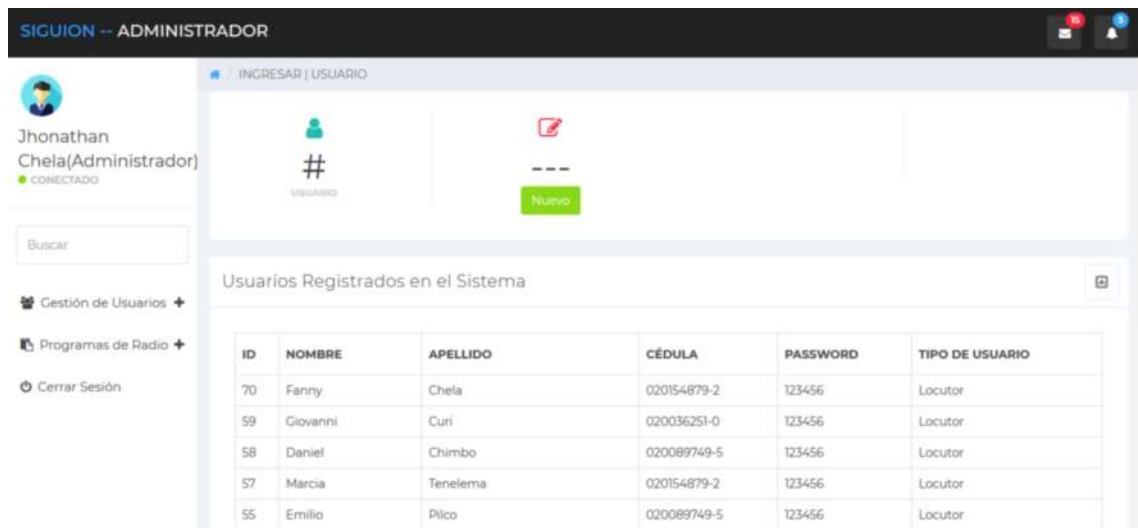
En la parte izquierda derecha del sistema se tiene el nombre y el tipo de usuario que está en el sistema, además, en la parte vertical inferior se encuentran las dos grandes categorías de trabajo propias del perfil del administrador del sistema.

En la parte derecha, en el área de trabajo, se puede ver un resumen de las últimas actualizaciones realizadas en el sistema, además de un cuadro usuarios registrados en el sistema.

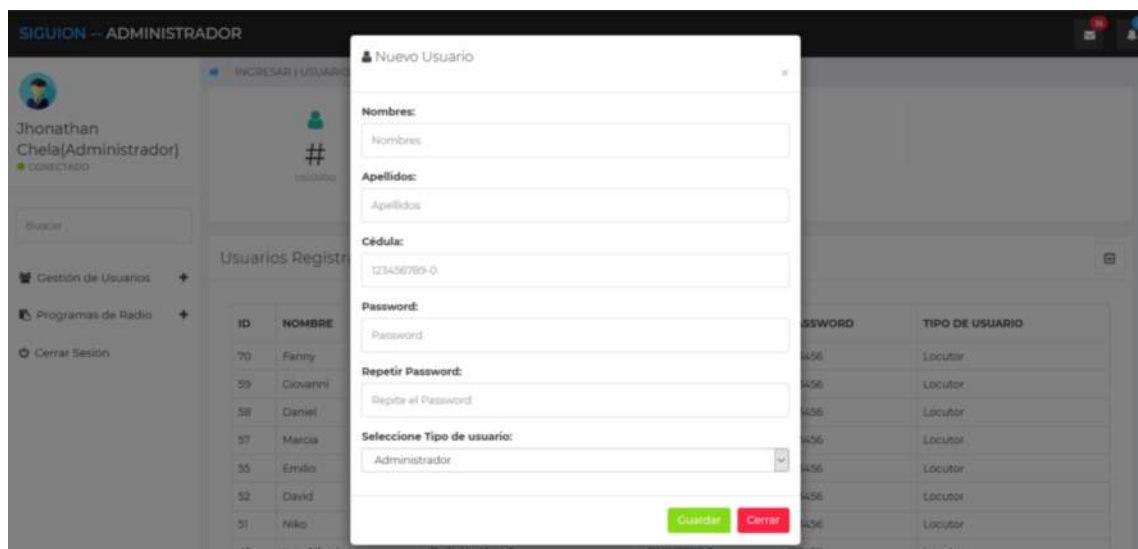
### Gestión de Usuarios.

La tarea de gestión de usuarios, incluye el ingreso de nueva información de usuarios, modificación del registro de usuarios y la eliminación de los usuarios.

Para el ingreso de nuevo usuarios en el sistema, primero se debe dar clic en la opción Gestión de Usuarios – Ingresar Usuario



Luego se debe dar clic en el boton Nuevo, y en el formulario de ingreso de información deben ser llenados todos los campos, todos estos son obligatorios.



Para que los datos sean almacenados en el sistema, se debe dar clic en el boton Guardar, si no se quiere ingresar la información se debe dar clic en el boton Cerrar para salir del formulario de ingreso.

Luego de guardar la información el sistema emitirá el mensaje correspondiente indicando la acción realizada.

Si en el formulario no se ingresa un campo, el sistema emite mensajes como los que se adjuntan en la siguiente imagen:

Nuevo Usuario

Nombres:  
Nombres  
Nombre requerido

Apellidos:  
Apellidos  
Apellido requerido

Cédula:  
123456789-0  
C.I. requerido

Password:  
Password  
Password requerido

Repetir Password:  
Repite el Password  
El Password no coincide

Seleccione Tipo de usuario:  
Administrador

Guardar Cerrar

## Modificación de usuarios.

De manera similar al ingreso de usuarios nuevos, dar clic en Gestión de Usuarios – Modificar Usuarios, el sistema redirige al panel correspondiente, como el que se adjunta en la imagen:

SIGUION -- ADMINISTRADOR

EDITAR | USUARIO

123456789-9  
Buscar

Modificar Usuario

Nombres:  
Nombres

Apellidos:  
Apellidos

Cédula:  
123456789-0

Password:  
Password

Jhonathan Chela(Administrador)  
CONECTADO

Buscar

Gestión de Usuarios -

- Ingresar Usuarios
- Modificar Usuarios
- Eliminar Usuarios

Programas de Radio +

Cerrar Sesión

En el campo de búsqueda se debe ingresar el número de C.I. del usuario a buscar y dar clic en el boton Buscar, en caso de ingresar un C.I. incorrecto o dejar el campo vacío el sistema emitirá el mensaje correspondiente.

Los datos consultados se presentan en el formulario que se adjunta en la parte inferior que se denomina Modificar Usuario, en cada campo es necesario no dejar vacío, es importante notar que todos los campos son obligatorios, una vez terminado la modificación de datos, dar clic en Guardar, como se muestra en la imagen que se adjunta:

### Modificar Usuario

**Nombres:**

**Apellidos:**

**Cédula:**

**Password:**

**Repetir Password:**

**Seleccione Tipo de usuario:**

[Guardar](#)

Una vez realizada la actualización de los datos del usuario consultado, el sistema emitirá el mensaje correspondiente, y se puede verificar la acción revisando la tabla de registro que se adjunta en el área de trabajo, siempre el primer campo pertenece al usuario actualizado.

EDITAR | USUARIO

  
 #  
USUARIO

[Buscar](#)

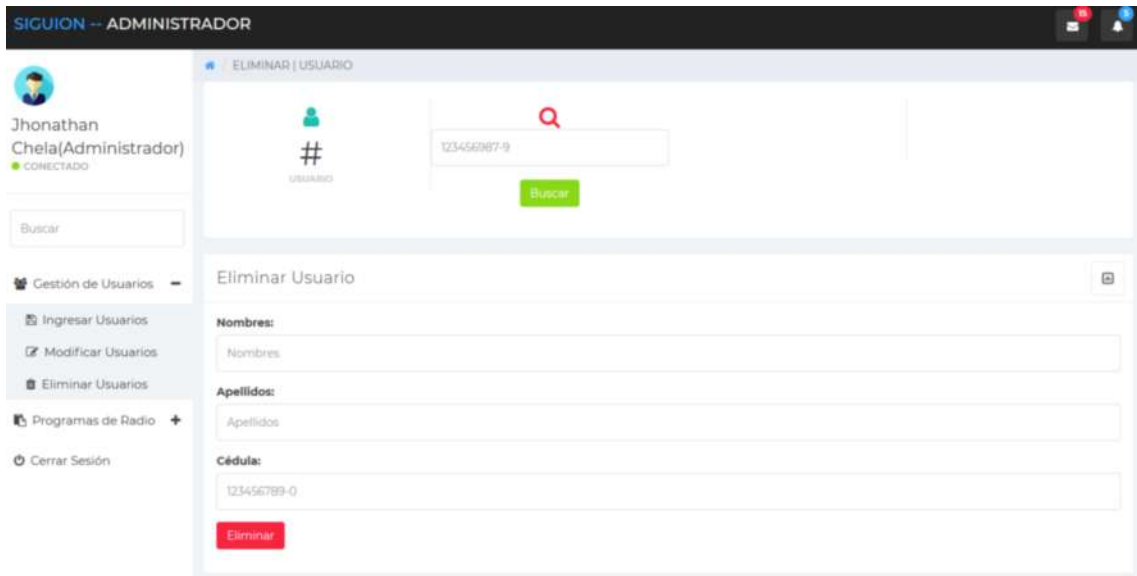
Datos Actualizados Correctamente

### Usuarios Registrados en el Sistema

ID	NOMBRE	APELLIDO	CÉDULA	PASSWORD	TIPO DE USUARIO
70	Fanny	Chela	020154879-2	123456	Locutor
59	Giovanni	Curi	020036251-0	123456	Locutor
58	Daniel	Chimbo	020089749-5	123456	Locutor
57	Marcia	Tenelema	020154879-2	123456	Locutor
55	Emilio	Pilco	020089749-5	123456	Locutor
52	David	Arellano	021548796-3	123456	Locutor
51	Niko	Evans	021589745-0	123456	Locutor
45	Luis Alfredo	Chela Ninabanda	020147896-3	123456	Locutor
29	Ángel Pedro	Aucatorna Tandapilco	020227978-2	123456	Locutor

## Eliminar Usuario

Para acceder al panel de trabajo de la opción Eliminar Usuario, se debe dar clic en Gestión de Usuarios – Eliminar Usuarios, a continuación, aparece una pantalla de gestión como la que se adjunta en la imagen:



The screenshot displays the 'SIGUIÓN -- ADMINISTRADOR' interface. On the left, a sidebar shows the user profile 'Jhonathan Chela(Administrador)' and a menu with options: 'Gestión de Usuarios', 'Ingresar Usuarios', 'Modificar Usuarios', 'Eliminar Usuarios', 'Programas de Radio', and 'Cerrar Sesión'. The main content area is titled 'ELIMINAR | USUARIO'. It features a search bar with a magnifying glass icon and a green 'Buscar' button. Below the search bar, there is a form titled 'Eliminar Usuario' with three input fields: 'Nombres', 'Apellidos', and 'Cédula'. The 'Cédula' field contains the value '123456789-0'. A red 'Eliminar' button is positioned at the bottom of the form.

Al igual que el panel para modificar usuarios, primero se debe ingresar la credencial de C.I. correspondiente al usuario que se desea eliminar, en el formulario que se adjunta se puede observar los nombres, apellidos y la cédula del usuario.

Para eliminar el usuario simplemente se debe dar clic en el botón Eliminar, el sistema emitirá el respectivo mensaje confirmando la acción realizada.

## Gestión de Programas Radiales

La opción de gestión de programas radiales incluye, ingresar, modificar y eliminar la información de los programas que se emiten en la entidad.

### Ingresar Programa Radial.

Para ingresar a la opción, dar clic en Programa de Radio – Ingresar Programas y el sistema redirige el área de trabajo a los formularios correspondientes:



**SIGUION -- ADMINISTRADOR**

Jhonathan Chela(Administrador) CONECTADO

INGRESAR | PROGRAMA RADIAL

PROGRAMA RADIAL **Nuevo**

Programas Radiales Registrados en el Sistema

ID	NOMBRE	DESCRIPCIÓN	DÍA DE TRANSMISIÓN	HORA TRANSMISIÓN	DURACIÓN EN HORAS	TIPO DE PROGRAMA	CLASIFICACIÓN	REPRESENTANTE
20	Infancia con voz	Infancia con voz	Miércoles	18h30-19h30	1 HORA	Entretención	A	LuisEvans
18	Pura Fe	Pura Fe	Sábado	10h00-12h00	2 HORAS	Entretención	A	NikoEvans
10	Mujer de Fe	Mujer de Fe	Sábado	10h00 - 12h00	2 HORAS	Formativo, Educativo, Cultural	A	MagalyQuille
9	Hombro a Hombro con el Agricultor	Hombro a Hombro con el Agricultor, programa de educación agrícola	Jueves	16h00 - 17h30	1 HORA 30 MINUTOS	Formativo, Educativo, Cultural	A	Noe Ninabanda
8	La voz Juvenil	La voz Juvenil	Martes	20h00 - 21h30	1 HORA 30 MINUTOS	Entretención	A	SterRea

La pantalla de ingreso de un nuevo programa incluye una tabla de registro de programas radiales ingresados en el sistema, en esta tabla se puede ir observando y verificando el ingreso de la información de un nuevo programa radial.

Para ingresar nuevos datos, dar clic en el botón Nuevo, y en el formulario ingresar los campos requeridos:

**SIGUION -- ADMINISTRADOR**

Jhonathan Chela(Administrador) CONECTADO

INGRESAR | PROGRAMA RADIAL

PROGRAMA RADIAL **Nuevo**

Programas Radiales Registrados en el Sistema

Nombre de la Programación:  
Nombre del Programa

Descripción:  
Breve reseña

Día de Transmisión:  
Lunes

Hora de Transmisión:  
20h00

Tiempo de Duración:  
1 HORA

Tipo de Programa:  
Tipo de Programa

Clasificación:  
Clasificación del Programa

Representante:  
Representante

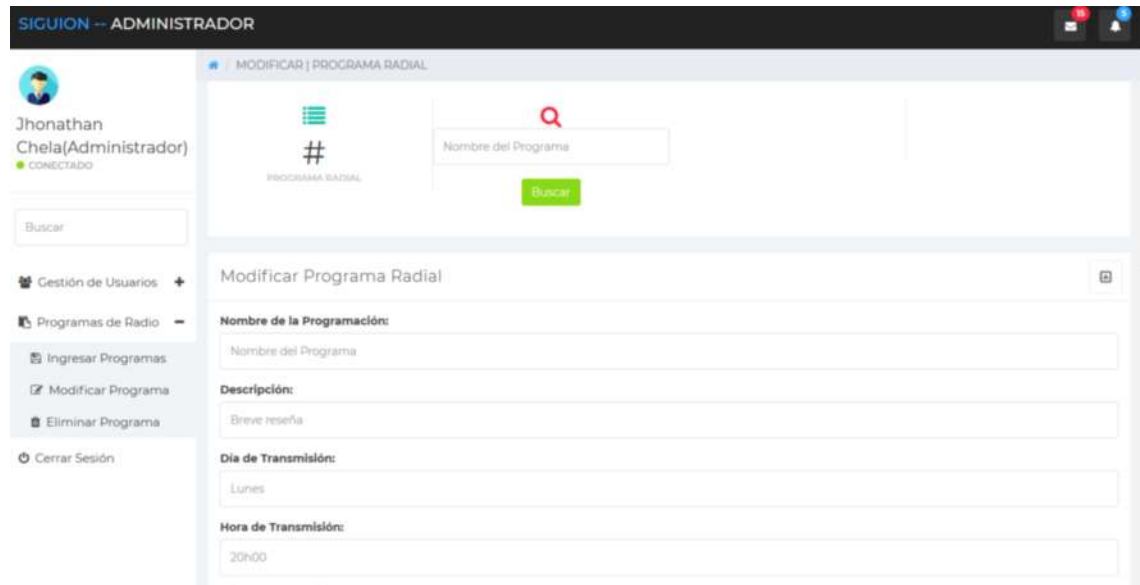
**Guardar** **Cerrar**

ID	NOMBRE	DESCRIPCIÓN	DÍA DE TRANSMISIÓN	HORA TRANSMISIÓN	DURACIÓN EN HORAS	TIPO DE PROGRAMA	CLASIFICACIÓN	REPRESENTANTE
20	Infancia con voz	Infancia con voz	Miércoles	18h30-19h30	1 HORA	Entretención	A	LuisEvans
18	Pura Fe	Pura Fe	Sábado	10h00-12h00	2 HORAS	Entretención	A	NikoEvans
10	Mujer de Fe	Mujer de Fe	Sábado	10h00 - 12h00	2 HORAS	Formativo, Educativo, Cultural	A	MagalyQuille
9	Hombro a Hombro con el Agricultor	Hombro a Hombro con el Agricultor, programa de educación agrícola	Jueves	16h00 - 17h30	1 HORA 30 MINUTOS	Formativo, Educativo, Cultural	A	Noe Ninabanda
8	La voz Juvenil	La voz Juvenil	Martes	20h00 - 21h30	1 HORA 30 MINUTOS	Entretención	A	SterRea

Es importante notar que todos los campos son necesarios y obligatorios, si se desea salir y no realizar ninguna acción se debe dar clic en el botón Cerrar, si se desea guardar la información dar clic en el botón Guardar y el sistema emitirá el mensaje correspondiente de confirmación de la acción.

## Modificar Programas de Radio.

Para ingresar a la opción de actualización de información de los usuarios, se debe dar clic en Programas de Radio – Modificar Programa, el sistema mostrará una pantalla como la que se adjunta a continuación:



The screenshot shows the 'MODIFICAR | PROGRAMA RADIAL' page in the SIGUION ADMINISTRADOR system. The user is logged in as Jhonathan Chela (Administrador). The page features a search bar for the program name and a 'Buscar' button. Below the search bar is the 'Modificar Programa Radial' form, which includes fields for 'Nombre de la Programación', 'Descripción', 'Día de Transmisión', and 'Hora de Transmisión'. The left sidebar contains navigation options: 'Gestión de Usuarios', 'Programas de Radio', 'Ingresar Programas', 'Modificar Programa', 'Eliminar Programa', and 'Cerrar Sesión'.

En el campo de texto que indica que se ingrese el nombre de programa, se debe ingresar este criterio, dar clic en el boton Buscar, si la información ingresada es correcta, el sistema mostrara la información requerida en el formulario denominado Modificar Programa Radial.



This image shows a detailed view of the 'Modificar Programa Radial' form. It includes the following fields and options:

- Nombre de la Programación:** Input field with 'Nombre del Programa'.
- Descripción:** Input field with 'Breve reseña'.
- Día de Transmisión:** Input field with 'Lunes'.
- Hora de Transmisión:** Input field with '20h00'.
- Tiempo de Duración:** Input field with '1 HORA'.
- Tipo de Programa:** Dropdown menu with 'Tipo de Programa'.
- Clasificación:** Dropdown menu with 'Clasificación del Programa'.
- Representante:** Dropdown menu with 'Representante'.
- Guardar:** Green button to save the changes.

Se debe ingresar en el formulario la información necesaria a actualizar, y para guardar los datos, dar clic en el boton Guardar, el sistema emitirá el mensaje correspondiente. Además, se puede

verificar el registro del proceso en el formulario de la tabla de registro de programas de radio que se adjunta en el área de trabajo:

Programas Radiales Registrados en el Sistema

ID	NOMBRE	DESCRIPCIÓN	DÍA DE TRANSMISIÓN	HORA TRANSMISIÓN	DURACIÓN EN HORAS	TIPO DE PROGRAMA	CLASIFICACIÓN	REPRESENTANTE
20	Infancia con voz	Infancia con voz	Miércoles	18h30-19h30	1 HORA	Entretenimiento	A	LuisEvans
18	Pura Fe	Pura Fe	Sábado	10h00-12h00	2 HORAS	Entretenimiento	A	NikoEvans
10	Mujer de Fe	Mujer de Fe	Sábado	10h00 - 12h00	2 HORAS	Formativo, Educativo, Cultural	A	MagalyQuille
9	Hombro a Hombro con el Agricultor	Hombro a Hombro con el Agricultor, programa de educación agrícola	Jueves	16h00 - 17h30	1 HORA 30 MINUTOS	Formativo, Educativo, Cultural	A	Noe Ninabanda
8	La voz Juvenil	La voz Juvenil	Martes	20h00 - 21h30	1 HORA 30 MINUTOS	Entretenimiento	A	SterRea
7	Noticiero Kichwa	Primer Noticiero en el idioma Kichwa	Lunes - Viernes	07h00 - 08h30, 12h30 - 14h00, 18h00 - 18h30	3 HORAS	Informativo, Noticias	A	Ángel PedroAucotoma Tandapilco
6	Expresión de Gratitud	Expresión de Gratitud	Domingo	18h00 - 20h00	2 HORAS	Entretenimiento	A	KleverTualombo

### Eliminar Programa Radial.

Para ingresar a esta opción, dar clic en Programas de Radio – Eliminar Programa, el sistema muestra la pantalla que se adjunta a continuación:

The screenshot shows the 'SIGUIÓN -- ADMINISTRADOR' interface. On the left is a sidebar with the user profile 'Jhonathan Chela(Administrador)' and a menu with options: 'Gestión de Usuarios', 'Programas de Radio', 'Ingresar Programas', 'Modificar Programa', 'Eliminar Programa', and 'Cerrar Sesión'. The main content area is titled 'ELIMINAR | PROGRAMA RADIAL' and contains a search bar with a magnifying glass icon and a '# PROGRAMAS RADIALES' label. Below the search bar is a 'Buscar' button. The main form is titled 'Eliminar Programa Radial' and includes the following fields: 'Nombre de la Programación' (with a search icon), 'Descripción' (with a brief description icon), 'Día de Transmisión' (with a dropdown menu showing 'Lunes'), 'Hora de Transmisión' (with a dropdown menu showing '20h00'), and 'Tiempo de Duración'.

El criterio de búsqueda de la información es el nombre del programa de radio, por lo que se debe ingresar en el campo que se requiere esta categoría, y dar clic en el boton Buscar, enseguida en el formulario que se adjunta se mostrarán los datos del programa radial:

Eliminar Programa Radial 🗑️

**Nombre de la Programación:**

Infancia con voz

**Descripción:**

Infancia con voz

**Día de Transmisión:**

Miércoles

**Hora de Transmisión:**

18h30-19h30

**Tiempo de Duración:**

1 HORA

**Eliminar**

Para terminar la acción, se debe dar clic en el boton Eliminar, y el sistema emitirá el mensaje correspondiente confirmando la acción.

Se puede verificar la eliminación de un programa radial en la tabla de registros que se adjunta en el área de trabajo de la opción que se está abordando:

Programas Radiales Registrados en el Sistema 🗑️

ID	NOMBRE	DESCRIPCIÓN	DÍA DE TRANSMISIÓN	HORA TRANSMISIÓN	DURACIÓN EN HORAS	TIPO DE PROGRAMA	CLASIFICACIÓN	REPRESENTANTE
20	Infancia con voz	Infancia con voz	Miércoles	18h30-19h30	1 HORA	Entretenimiento	A	LuisEvans
18	Pura Fe	Pura Fe	Sábado	10h00-12h00	2 HORAS	Entretenimiento	A	NikoEvans
10	Mujer de Fe	Mujer de Fe	Sábado	10h00 - 12h00	2 HORAS	Formativo, Educativo, Cultural	A	MagalyQuille
9	Hombro a Hombro con el Agricultor	Hombro a Hombro con el Agricultor, programa de educación agricola	Jueves	16h00 - 17h30	1 HORA 30 MINUTOS	Formativo, Educativo, Cultural	A	Noe Ninabanda
8	La voz Juvenil	La voz Juvenil	Martes	20h00 - 21h30	1 HORA 30 MINUTOS	Entretenimiento	A	SterRea
7	Noticiero Kichwa	Primer Noticiero en el idioma Kichwa	Lunes - Viernes	07h00 - 08h30, 12h30 - 14h00, 18h00 - 18h30	3 HORAS	Informativo, Noticias	A	Ángel PedroAucatoma Tandapilco
6	Expresión de Gratitud	Expresión de Gratitud	Domingo	18h00 - 20h00	2 HORAS	Entretenimiento	A	KieverTualombo

**Salir de sistema**

Para salir del sistema, se debe dar clic en la opción, Cerrar Sesión que se adjunta en todos los perfiles de usuario definidos, el sistema regresará al inicio o a la pantalla de autenticación de usuarios:

SIGUION -- ADMINISTRADOR

Eliminar

Jhonathan Chela(Administrador)  
● CONECTADO

Buscar

Gestión de Usuarios +

Programas de Radio +

Cerrar Sesión

Programas Radiales Registrados en el Sistema

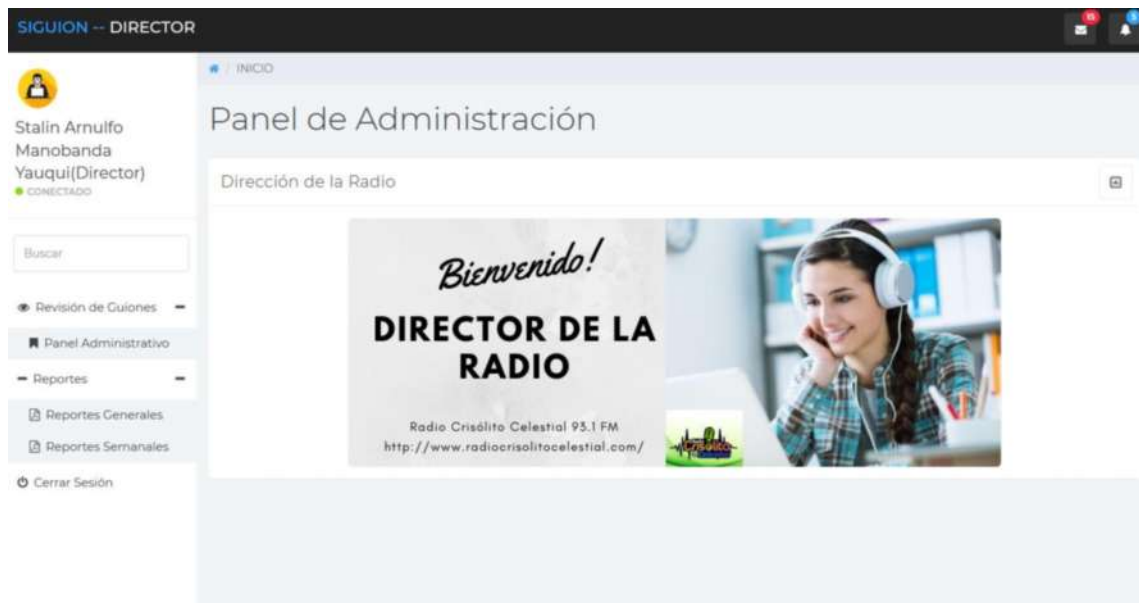
ID	NOMBRE	DESCRIPCIÓN	DÍA DE TRANSMISIÓN	HORA TRANSMISIÓN	DURACIÓN EN HORAS	TIPO DE PROGRAMA	CLASIFICACIÓN	REPRESENTANTE
20	Infancia con voz	Infancia con voz	Miércoles	18h30-19h30	1 HORA	Entretenimiento	A	LuisEvans
18	Pura Fe	Pura Fe	Sábado	10h00-12h00	2 HORAS	Entretenimiento	A	NikoEvans
10	Mujer de Fe	Mujer de Fe	Sábado	10h00 - 12h00	2 HORAS	Formativo, Educativo, Cultural	A	MagalyQuille
9	Hombro a Hombro con el Agricultor	Hombro a Hombro con el Agricultor, programa de educación agrícola	Jueves	16h00 - 17h30	1 HORA 30 MINUTOS	Formativo, Educativo, Cultural	A	Noe Ninabanda
8	La voz Juvenil	La voz Juvenil	Martes	20h00 - 21h30	1 HORA 30 MINUTOS	Entretenimiento	A	SterRea
7	Noticiero Kichwa	Primer Noticiero en el idioma Kichwa	Lunes - Viernes	07h00 - 08h30, 12h30 - 14h00, 18h00 - 18h30	3 HORAS	Informativo, Noticias	A	Ángel PedroAucatoma Tandapilco
6	Expresión de Gratitud	Expresión de Gratitud	Domingo	18h00 - 20h00	2 HORAS	Entretenimiento	A	KieverTualombo

### Perfil del Director de la entidad

Al ingresar las credenciales correctas del Director de la entidad, el sistema permite el acceso a la pantalla de trabajo correspondiente, esta pantalla consta de dos categorías, cada una con subcategorías definitivas de trabajo:

- Revisión de Guiones
  - Panel Administrativo
- Reportes
  - Reportes Generales
  - Reportes Semanales

A continuación, en la imagen se adjunta lo descrito anteriormente:

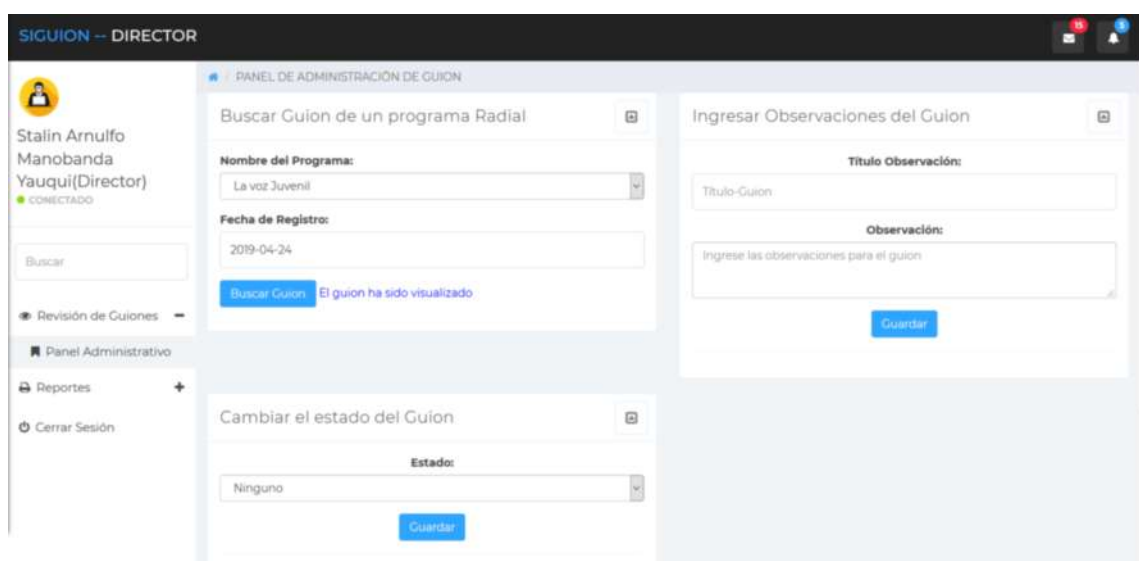


### Revisión de Guiones

Este panel de administración está diseñado con las tareas administrativas propias que el director de la entidad debe tener con respecto a la gestión de guiones de programas de radio.

Los guiones de programas de radio deben ser buscados por el nombre del programa y por la fecha de registro respectivamente, es necesario visualizar la estructura del guion, para poder realizar las observaciones necesarias en el caso que sea necesario, también, se debe cambiar el estado de un guion, ya sea si este esté Revisado, Aprobado o No Aprobado.

A continuación, se adjunta la pantalla principal del panel de administración de revisión de guiones:



En el formulario Buscar Guion de un Programa Radial, se debe elegir el nombre del programa radial e ingresar una fecha de registro, si el registro existe el sistema emitirá el mensaje

correspondiente, además, el sistema visualizará el guion en el formulario de guion que se adjunta en la pantalla:

DIÁLOGO DEL PERSONAJE	AUDIO	TIEMPO REAL	TIEMPO RECORRIDO
Cortina Entrada	Audio careta de entrada Mi casa es tu casa – Alex Campos feat Evan Craft	30 seg	00:00:30
Saludo	Intervención del equipo, Alex, Cecibel, Adan, Jessica	30 seg	00:00:60
Segmento Tips de salud	Intervención de Jessica Chela, fondo musical, Emir Sensini, "En lo secreto"	15 min	00:15:60
Segmento de comentario y recepción de llamadas telefónicas	Intervención de los integrantes de la programación, complacencia de temas musicales	15 min	00:30:60
Segmento CHOCOLATE PARA EL ALMA	Intervención de Adan Tualombo, con una charla motivacional, basada en la Biblia	15 min	00:45:60
Música orgánica	Reproducir los temas musicales peticionados por los oyentes	15 min	00:60:60
Parte final de la programación	Palabras de despedida por cada integrante de la programación, Alex, Adan, Cecibel, Jessica	40 seg	00:61:00

Para ingresar las observaciones respectivas al guion revisado, el usuario tiene que dirigirse al formulario Ingresar Observaciones del Guion, como se adjunta en la siguiente imagen:

Formulario "Ingresar Observaciones del Guion" con los siguientes campos:

- Titulo Observación:** Campo de texto con el valor "Titulo-Guion".
- Observación:** Campo de texto con el placeholder "Ingrese las observaciones para el guion".
- Guardar:** Botón de acción.

El título de la observación y la observación son campos que no deben estar vacíos, el sistema, al dar clic en el boton Guardar emitirá un mensaje correspondiente, si los campos no estan vacíos el mensaje confirmará que la observación fue agregada, caso contrario, el sistema visualizará el mensaje que indique que los campos están vacíos.

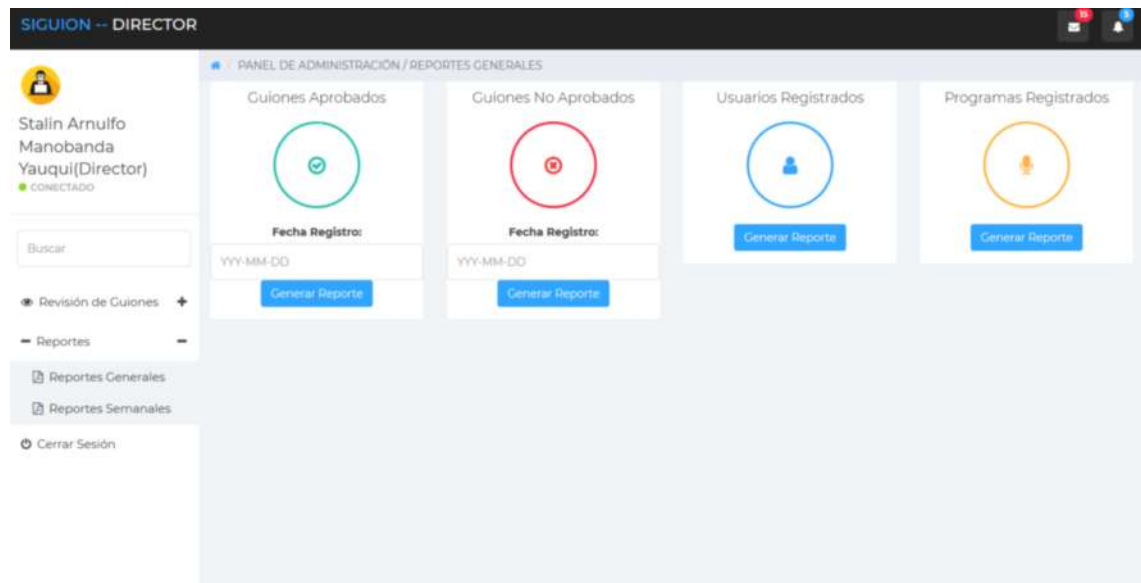
Formulario "Cambiar el estado del Guion" con un menú desplegable "Estado:" que muestra las siguientes opciones:

- Ninguno
- Aprobado
- No Aprobado
- Revisado

Para ingresar un nuevo estado a un guion consultado, en el formulario Cambiar estado del Guion se debe seleccionar el nuevo estado y se debe dar clic en el boton Guardar, el sistema emitirá un mensaje confirmando la acción.

## Reportes Generales

Para ingresar a esta opción se debe dar clic en Reportes – Reportes Generales, el sistema muestra la pantalla que a continuación se adjunta:



En este panel de trabajo, el sistema permite realizar un reporte de guiones aprobados y no aprobados ingresando una fecha de registro, además, de un reporte en formato pdf de todos los usuarios y programas radiales registrados en el sistema.

En las imágenes siguientes se cita un ejemplo de reporte general por fecha de registro de guiones aprobados almacenados en el sistema:



Una vez ingresado la fecha de registro dar clic en el botón Generar Reporte, y el sistema permite exportar el reporte en formato pdf, de la lista de los guiones que coinciden con la fecha de registro ingresada, en la imagen se muestra un ejemplo.





Radio Crisólito Celestial 93.1 FM

Reporte de Guiones Aprobados

Parroquia Veintimilla - Ciudad de Guaranda - Provincia de Bolívar

Nombre del Programa	Estado	Clasificación	Descripción	Tipo	Descripción	Día Transmisión	Hora Transmisión	Fecha Registro
Expresión de Gratitud	Aprobado	A	Apto para todo publico	E	Entretenimiento	Domingo	18h00 - 20h00	2019-04-13

Abg. Arnulfo Manobanda  
Director

Fecha de emisión del reporte:

5/22/19 12:44 PM

De igual manera si se da clic sobre el boton Generar Reporte en Usuarios Registrados, se obtiene un reporte en formato pdf de todos los usuarios almacenados en la base de datos:

Usuarios Registrados



Generar Reporte



Radio Cristalito Celestial 93.1 FM

Reporte de Personas Participantes en la entidad

Parroquia Veintimilla - Ciudad de Guaranda - Provincia de Bolívar

Nombres	Apellidos	Cédula	Tipo de Usuario
Jhonathan	Chela	020158336-6	Administrador
Stalin Arnulfo	Manobanda Yauqui	020178189-5	Director
Kiever	Tualombo	020136549-8	Locutor
Ster	Rea	020789563-7	Locutor
Noe	Ninabanda	020201465-9	Locutor
Megely	Quille	020154879-2	Locutor
Cristina	Yallico Chariguaman	020197189-2	Locutor
Angel Pedro	Aucatorna Tandaplico	020227978-2	Locutor
Luis Alfredo	Chela Ninabanda	020147896-3	Locutor
Niko	Evans	021589745-0	Locutor
David	Arellano	021548796-3	Locutor
Emilio	Pilco	020089749-5	Locutor
Marcia	Tenelema	020154879-2	Locutor
Daniel	Chimbo	020089749-5	Locutor
Giovanni	Curi	020036251-0	Locutor

Ab. Arnulfo Manobanda

6/21/19 3:52 AM

Para generar el reporte de programas radiales, de la misma manera, se procede a dar clic en el boton Generar Reportes del apartado Programas Radiales.

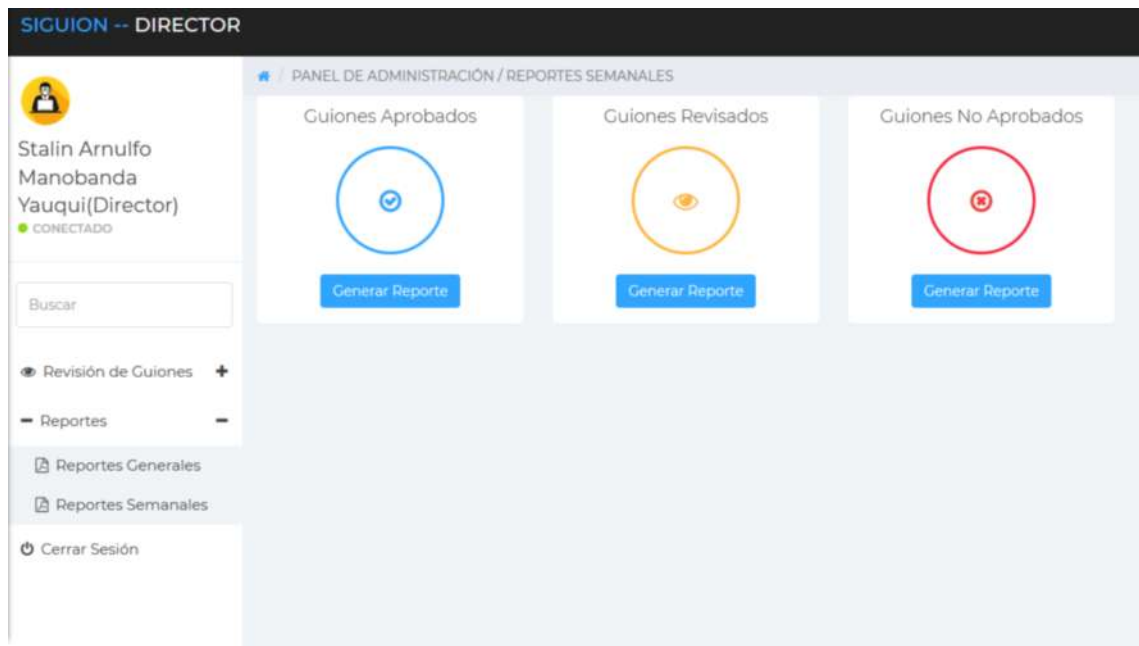
Programas Registrados



Generar Reporte

### Generar Reportes semanales.

Para ingresar a esta opción se debe dar clic en Reportes – Reportes semanales, a continuación, el sistema muestra la pantalla que se adjunta:

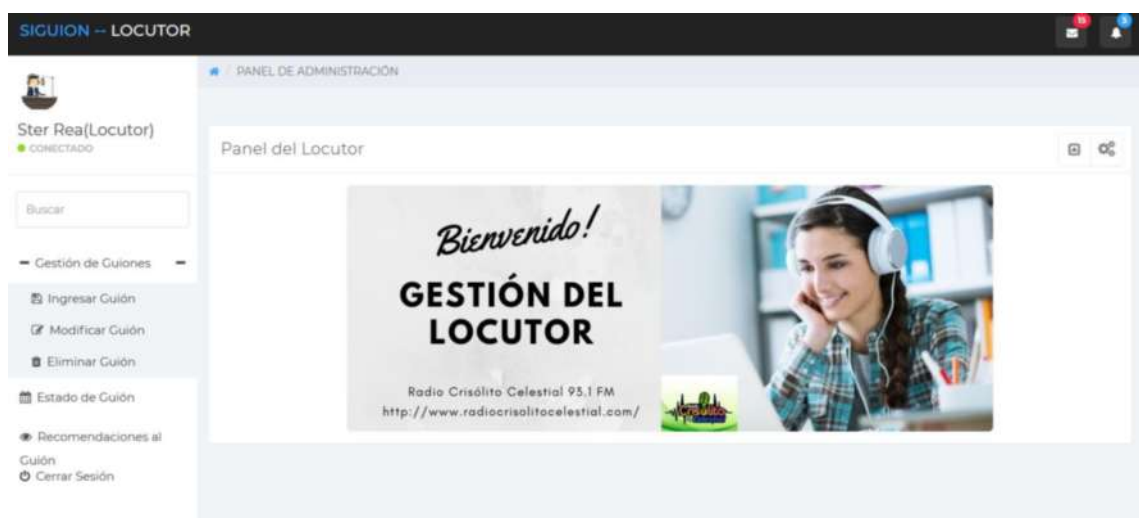


De igual manera que los reportes anteriores, al dar clic en Generar Reporte de cada apartado, el sistema exporta un documento pdf, con la información consultada.

### Perfil del Locutor de la Radio

Este perfil permite realizar tareas como la gestión de un guion de programa de radio, revisar el estado de un guion, observar las recomendaciones que un guion tiene, e imprimir un guion radial.

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



**Ingresar un guion.**

Para ingresar un guion de un programa radial, dar clic en la opción Gestión de Guiones – Ingresar Guion.

The screenshot shows the 'SIGUIÓN -- LOCUTOR' administration interface. On the left is a sidebar with the user profile 'Ster Rea(Locutor)' and 'CONECTADO', a search bar, and navigation links for 'Gestión de Guiones', 'Estado de Guion', 'Recomendaciones al Guion', and 'Cerrar Sesión'. The main content area is titled 'PANEL DE ADMINISTRACIÓN / INGRESAR GUION' and contains a form for 'Guion de un programa Radial'. The form has two input fields: 'Nombre del Programa:' with the value 'La voz Juvenil' and 'Fecha de Registro:' with the value '2019-05-22'. Below these fields is a blue 'Crear Guion' button. Underneath the form is a section titled 'Diálogos del Guion' which includes a green 'Agregar Diálogo' button and a table with the following headers: 'DIÁLOGO DEL PERSONAJE', 'AUDIO', 'TIEMPO REAL', and 'TIEMPO RECORRIDO'.

El sistema muestra el área de trabajo del sistema en el que se pueden apreciar dos partes importantes para realizar el ingreso de un guion.

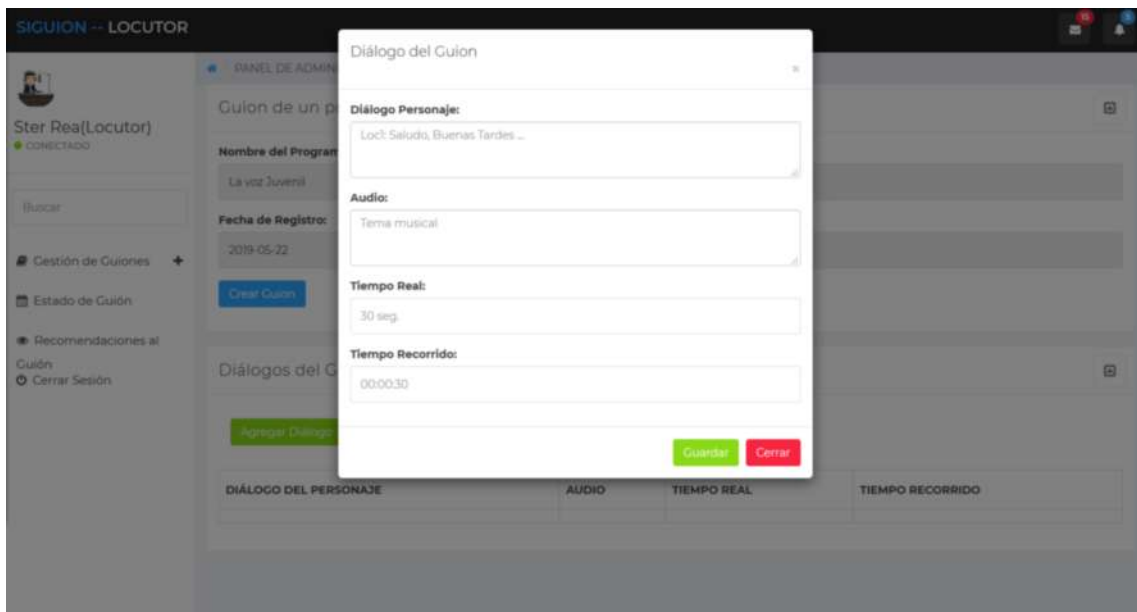
En primer lugar, se puede ver en el formulario Guion de un programa Radial, el nombre del programa del cual el usuario es encargado y la fecha actual en la que se está registrando el guion.

Se debe dar clic en crear el guion, el sistema mostrará un mensaje confirmando la acción.

Después, se debe dar clic en el botón Agregar Diálogo,

This is a close-up of the 'Diálogos del Guion' section from the previous screenshot. It features a green 'Agregar Diálogo' button at the top. Below the button is a table with the following headers: 'DIÁLOGO DEL PERSONAJE', 'AUDIO', 'TIEMPO REAL', and 'TIEMPO RECORRIDO'. The table body is currently empty.

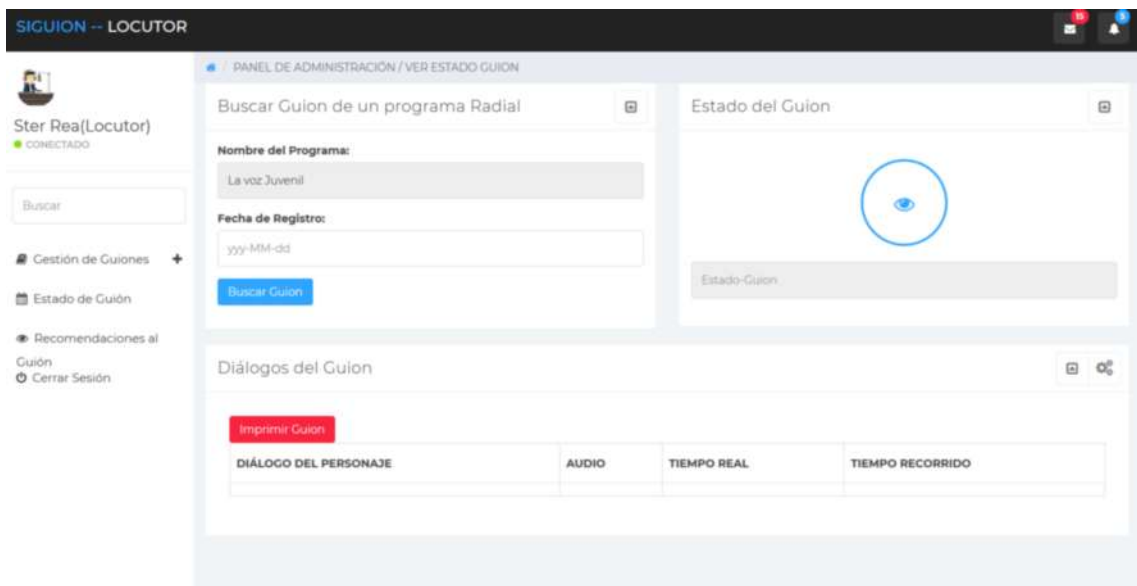
Enseguida aparece un formulario con las opciones propias del estándar que se utiliza para ingresar un guion. Se debe ingresar los campos requeridos y dar clic en el botón Guardar, enseguida se grafica en el formulario Diálogos del Guion, los diálogos que se van ingresando.



Si se desea cerrar el formulario de ingreso de guiones, dar clic en el boton Cerrar.

### Revisar el estado de un guion.

Para ingresar a esta opción, dar clic en Estado de Guión, el sistema muestra la siguiente pantalla:



Esta pantalla permite realizar la consulta del guion ingresado a partir de una fecha de registro, para lo cual, el usuario debe ingresar en el formulario Buscar Guión de un programa Radial, una fecha de registro y dar clic en el boton Buscar Guión.

Si los registros existen de acuerdo a los datos ingresados, el sistema muestra en el formulario Estado del Guión el estado correspondiente y grafica el guion que se ha ingresado en el formulario Diálogos del Guión. A continuación, un ejemplo de lo explicado:

**SIGUIÓN -- LOCUTOR**

PANEL DE ADMINISTRACIÓN / VER ESTADO GUIÓN

Buscar Guión de un programa Radial

Nombre del Programa:  
Expresión de Gratitud

Fecha de Registro:  
2019-04-13

Buscar Guión El guion ha sido cargado

Estado del Guión

Aprobado

Diálogos del Guión

Imprimir Guión

DIÁLOGO DEL PERSONAJE	AUDIO	TIEMPO REAL	TIEMPO RECORRIDO
Cortina Entrada	Audio careta de entrada Mi casa es tu casa – Alex Campos feat Evan Craft	30 seg	00:00:30
Saludo	Intervención del equipo, Alex, Cecibel, Adan, Jessica	30 seg	00:00:60
Segmento Tips de salud	Intervención de Jessica Chela, fondo musical, Emir Sensini, "En lo secreto"	15 min	00:15:60

El panel de observación del guion también tiene la opción de imprimir el guion, siempre y cuando el estado del guion sea Aprobado. Observar el ejemplo:

Radio Crisolito Celestial 93.1 FM

Guion Aprobado

Parroquia Veintimilla - Ciudad de Guaranda - Provincia de Bolivar

Programa: Expresión de Gratitud

Fecha de impresión: 5/22/19 1:56 PM

Diálogo Persona	Diálogo Audio	Tiempo Real	Tiempo Recorrido
Cortina Entrada	Audio careta de entrada Mi casa es tu casa – Alex Campos feat Evan Craft	30 seg	00:00:30
Saludo	Intervención del equipo, Alex, Cecibel, Adan, Jessica	30 seg	00:00:60
Segmento Tips de salud	Intervención de Jessica Chela, fondo musical, Emir Sensini, "En lo secreto"	15 min	00:15:60
Segmento de comentario y recepción de llamadas telefónicas	Intervención de los integrantes de la programación, complacencia de temas musicales	15 min	00:30:60
Segmento CHOCOLATE PARA EL ALMA	Intervención de Adan Tualombo, con una charla motivacional, basada en la Biblia	15 min	00:45:60
Música orgánica	Reproducir los temas musicales peticionados por los oyentes	15 min	00:60:60
Parte final de la programación	Palabras de despedida por cada integrante de la programación, Alex, Adan, Cecibel, Jessica	40 seg	00:61:00

### Revisar las recomendaciones realizados a un guion.

Para poder ingresar al panel de trabajo que permite revisar las observaciones realizadas al guion, se debe dar clic en la opción Recomendaciones al Guión, ubicado en la parte izquierda de la pantalla.

Luego se debe ingresar una fecha de registro y si los registros existen el sistema visualizará las observaciones y el guion consultado, caso contrario se emitirá un mensaje correspondiente a la acción realizada.

DIÁLOGO DEL PERSONAJE	AUDIO	TIEMPO REAL	TIEMPO RECORRIDO
Cortina Entrada	Audio careta de entrada Mi casa es tu casa - Alex Campos feat Evan Craft	30 seg	00:00:30
Saludo	Intervención del equipo, Alex, Cecibel, Adan, Jessica	30 seg	00:00:60
Segmento Tips de salud	Intervención de Jessica Chela, fondo musical, Emir Sensini, "En lo secreto"	15 min	00:15:60
Segmento de comentario y recepción de llamadas telefónicas	Intervención de los integrantes de la programación, complacencia de temas musicales	15 min	00:30:60
Segmento CHOCOLATE PARA EL ALMA	Intervención de Adan Tualombo, con una charla motivacional, basada en la Biblia	15 min	00:45:60
Música orgánica	Reproducir los temas musicales peticionados por los oyentes	15 min	00:60:60
Parte final de la programación	Palabras de despedida por cada integrante de la programación, Alex, Adan, Cecibel, Jessica	40 seg	00:61:00







## Anexo E: Historias Técnicas del proyecto SIGUION

Historia Técnica	
<b>ID:</b> HT-01	<b>Descripción:</b> Como desarrollador necesito realizar una recopilación de requerimientos del sistema.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 10
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar una recopilación de requerimientos del sistema.	
<b>Observaciones:</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>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 15
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 10
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar la planificación de los entregables del sistema.	
<b>Observaciones:</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.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-03	<b>Descripción:</b> Como desarrollador necesito definir las historias de usuario del sistema.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 15
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 10
<b>Descripción:</b> Como <i>desarrollador</i> necesito definir las historias de usuario del sistema.	
<b>Observaciones:</b> Las historias de usuario deben priorizarse utilizando la talla de la camiseta.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-04	<b>Descripción:</b> Como desarrollador necesito realizar la planificación de las pruebas unitarias automatizadas.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 15
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 10
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar la planificación de las pruebas unitarias automatizadas.	
<b>Observaciones:</b> Las pruebas unitarias deben ser consideradas como historias técnicas en la planificación de cada Sprint.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-05	<b>Descripción:</b> Como desarrollador necesito diseñar la arquitectura del sistema
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito diseñar la arquitectura del sistema	
<b>Observaciones:</b> La arquitectura del sistema debe ser diseñada de tal modo que se contemple una solución en n capas y escalable.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-06	<b>Descripción:</b> Como desarrollador necesito diseñar las interfaces de usuario
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito diseñar las interfaces de usuario	
<b>Observaciones:</b> La interfaz completa del sistema debe ser amigable con el usuario, además debe ser web responsiva, debido a que será utilizado en un navegador web, en cualquier dispositivo móvil capaz de invocar al software.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-07	<b>Descripción:</b> Como desarrollador necesito diseñar la base de datos
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito diseñar la base de datos	
<b>Observaciones:</b> La base de datos debe ser implementada en POSTGRESQL.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-08	<b>Descripción:</b> Como desarrollador necesito realizar la implementación de la base de datos.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar la implementación de la base de datos.	
<b>Observaciones:</b> La base de datos debe ser implementada en POSTGRESQL.	

<b>Historia Técnica</b>	
<b>ID:</b> HT-09	<b>Descripción:</b> Como desarrollador necesita realizar la conexión a la base de datos.
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 1
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesita realizar la conexión a la base de datos.	
<b>Observaciones:</b> La base de datos debe ser implementada en POSTGRESQL.	

## Gestión de casos de pruebas unitarias.

<b>Historia Técnica</b>	
<b>ID:</b> HT-10	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-01
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-01	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Caso de Prueba PU1 - PU3</b>	
<b>Id</b>	<b>PU1 - PU3</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-01
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Iniciar Sesión</li> <li>• Cerrar Sesión</li> </ul>
<b>Fecha</b>	16-01-2019.
<b>Funcionalidad / Característica</b>	Con la funcionalidad del Inicio de Sesión y Cerrar Sesión, el usuario puede ingresar al sistema y salir del sistema SIGUION.
<b>Datos / Acciones 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.</li> <li>• Cerrar Sesión: se redirige a la página inicial, dando por terminada la sesión iniciada.</li> </ul>
<b>Código</b>	<pre>/**  * Test of getLogin method, of class loginControlador.  */ @Test public void testGetLogin() {     System.out.println("getLogin");     loginControlador instance = new loginControlador();     UIComponent expectedResult = null;     UIComponent result = instance.getLogin(); }  /**  * Test of setLogin method, of class loginControlador.  */ @Test public void testSetLogin() {</pre>

	<pre> System.out.println("setLogin"); UIComponent login = null; loginControlador instance = new loginControlador(); instance.setLogin(login);  } /**  * Test of cerrarSession method, of class loginControlador.  */ @Test public void testCerrarSession() { System.out.println("cerrarSession"); loginControlador instance = new loginControlador(); String expectedResult = ""; String result = instance.cerrarSession(); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<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</li> <li>2. En los campos indicados ingresar las credenciales correspondientes</li> <li>3. Dar clic en el botón "Iniciar sesión"</li> <li>4. Para finalizar la sesión, dar clic en el botón "Cerrar Sesión"</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	16-01-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-11	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-02
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-02	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	



Caso de Prueba PU4 - PU7	
<b>Id</b>	PU4 - PU7
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-02
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Cargar Usuarios Registrados en el sistema</li> <li>• Insertar Usuario Nuevo</li> </ul>
<b>Fecha</b>	18-01-2019.
<b>Funcionalidad / Característica</b>	Con la funcionalidad de ingresar usuarios nuevos al sistema, el software permite registrar información y almacenarla.
<b>Datos / Acciones de Entrada</b>	Nombre de usuario, Apellido de usuario, C.I. de usuario, Contraseña, Tipo de Usuario
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of insertarNuevoUsuario method, of class mUsuario.  */ @Test public void testInsertarNuevoUsuario() throws Exception {     System.out.println("Modelo Usuario ---- insertarNuevoUsuario");     cUsuario objUsuario = new cUsuario();      objUsuario.setTusuariNombre("Luis");     objUsuario.setTusuariApellido("Evans");     objUsuario.setTusuariCedula("020089749-5");     objUsuario.setTusuariContrasenia("123456");     cTipoUsuario obj = new cTipoUsuario();     obj.setIdusuariotipo(3);     objUsuario.setObjTipoUsuario(obj);      boolean expResult = true;     boolean result = mUsuario.insertarNuevoUsuario(objUsuario);     assertEquals(expResult, result);  } /**  * Test of obtenerTodos method, of class mUsuario.  */ @Test public void testObtenerTodos() throws Exception {     System.out.println("Modelo Usuario --- obtenerTodos");     List&lt;cUsuario&gt; result = new ArrayList&lt;&gt;();     result = mUsuario.obtenerTodos();     assertNotNull(result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como administrador</li> <li>2. Ubicarse en el panel de ingreso de usuarios</li> <li>3. Clic en el boton de Nuevo</li> <li>4. Ingresar los campos correspondientes</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	18-01-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-12	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-03
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-03	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU8 - PU15</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-03
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Cargar Programas Radiales Registrados en el sistema</li> <li>• Cargar Tipo de Programación</li> <li>• Cargar Usuario</li> <li>• Insertar Nuevo Programa Radial.</li> </ul>
<b>Fecha</b>	21-01-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar la funcionalidad de ingreso de nuevos programas radiales.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que definen a un programa radial.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerTodos method, of class mProgramaRadial.  */ @Test public void testObtenerTodos() {     System.out.println("Modelo Programa Radial --- obtenerTodos");     List&lt;cProgramaradial&gt; result = new ArrayList&lt;&gt;();     result = mProgramaRadial.obtenerTodos();     assertNotNull(result); } /**  * Test of obtenerTodos method, of class mTipoProgramaRadial.  */ @Test public void testObtenerTodos() {     System.out.println("Modelo Tipo Programa Radial - obtenerTodos");     List&lt;cTipoProgramaRadial&gt; result = new ArrayList&lt;&gt;();     result = mTipoProgramaRadial.obtenerTodos(); } </pre>

	<pre> assertNotNull(result); } /**  * Test of obtenerTodos method, of class mUsuario.  */ @Test public void testObtenerTodos() throws Exception {     System.out.println("Modelo Usuario --- obtenerTodos");     List&lt;cUsuario&gt; result = new ArrayList&lt;&gt;();     result = mUsuario.obtenerTodos();     assertNotNull(result); } /**  * Test of insertarNuevoPrograma method, of class mProgramaRadial.  */ @Test public void testInsertarNuevoPrograma() throws Exception {     System.out.println("Modelo Programa Radial --- insertarNuevoPrograma");     cProgramaradial objProgramaRadial = new cProgramaradial();      objProgramaRadial.setTprogramaradialnombre("Infancia con voz");     objProgramaRadial.setTprogramaradialdescripcion("Infancia con voz");      objProgramaRadial.setTprogramaradialdiatransmision("Miércoles");     objProgramaRadial.setTprogramaradialhoratransmision("18h30-19h30");     objProgramaRadial.setTprogramaradialduracion("1 HORA");     cTipoProgramaRadial objtipo = new cTipoProgramaRadial();     objtipo.setIdtipoprogramaradial(3);     objProgramaRadial.setObjTipoProgramaRadial(objtipo);     cClasificacionProgramaRadial objClasif = new cClasificacionProgramaRadial();     objClasif.setIdtclasificacionprogramaradial(1);     objProgramaRadial.setObjClasificacionProgramaRadial(objClasif);     cUsuario objU = new cUsuario();     objU.setIdtusuario(55);     objProgramaRadial.setObjUsuario(objU);      boolean expResult = true;     boolean result = mProgramaRadial.insertarNuevoPrograma(objProgramaRadial);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como administrador</li> <li>2. Ubicarse en el panel de ingreso de programas radiales</li> <li>3. Clic en el boton de Nuevo</li> <li>4. Ingresar los campos correspondientes</li> <li>5. Clic en el botón Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	21-01-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-13	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-04
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-04.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU16 - PU21</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-04
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Crear un guion</li> <li>• Agregar diálogo al guion</li> <li>• Cargar diálogos del guion</li> </ul>
<b>Fecha</b>	23-01-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar la funcionalidad de ingreso de un guion a un programa de radio correspondiente.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que definen a un guion de programa radial.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of insertarNuevoGuion method, of class mGuionProgramaRadial.  */ @Test public void testInsertarNuevoGuion() throws Exception {     System.out.println("Modelo Guion --- insertarNuevoGuion");     cGuionProgramaRadial objGuionPrograma = new cGuionProgramaRadial();      objGuionPrograma.setTguionprogramaradialfecharegistro("2019-05-14");     objGuionPrograma.setTguionprogramaradialfechaaprobacion("1999-01-01");      cEstadoGuion objEstado = new cEstadoGuion();     objEstado.setIdtestadoguion(1);     objGuionPrograma.setObjEstadoguion(objEstado);      cProgramaradial objPrograma = new cProgramaradial();     objPrograma.setIdtprogramaradial(20);     objGuionPrograma.setObjProgramaradial(objPrograma);      objGuionPrograma.setTguionprogramaradialBandera(777); </pre>

	<pre> boolean expResult = true; boolean result = mGuionProgramaRadial.insertarNuevoGuion(objGuionPrograma); assertEquals(expResult, result);  } /**  * Test of insertarDialogoProgramaRadial method, of class  * mDialogoProgramaRadial.  */ @Test public void testInsertarDialogoProgramaRadial() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- insertarDialogoProgramaRadial");     cDialogo objDialogo = new cDialogo();      objDialogo.setTdialogopersonaje("Introducción Infancia con Voz");     objDialogo.setTdialogoaudio("Saludo a la audiencia por los niños locutores");     objDialogo.setTdialogotiemporeal("30seg.");     objDialogo.setTdialogotiemporecorrido("00:00:30");      cGuionProgramaRadial objguion = new cGuionProgramaRadial();     objguion.setIdtguionprogramaradial(38);      objDialogo.setObjGuionProgramaRadial(objguion);      boolean expResult = true;     boolean result = mDialogoProgramaRadial.insertarDialogoProgramaRadial(objDialogo);     assertEquals(expResult, result);  } /**  * Test of obtenerGuionPorId method, of class mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorId() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorId");     int idtguionprogramaradial = 33;     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorId(idtguionprogramaradial);     assertNotNull(result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como locutor</li> <li>2. Ubicarse en el panel de registro de guiones</li> <li>3. Clic en el botón Crear Guion</li> <li>4. Clic en el botón Ingresar Diálogo</li> <li>5. Clic en el botón Guardar Diálogo</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	23-01-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-14	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-05
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-05.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU22 - PU24</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-05
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion</li> <li>• Cargar diálogos del guion</li> </ul>
<b>Fecha</b>	25-01-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible visualizar el guion de un programa radial.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten que un guion sea visualizado.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idtprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial, fecharegistro);     assertNotNull(result); } /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");     int idtguionprogramaradial = 6; </pre>

	<pre>List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;(); result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial); assertNotNull(result); }</pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como director</li> <li>2. Ubicarse en el panel de administración</li> <li>3. En la caja de texto ingresar el criterio de búsqueda del guion</li> <li>4. Clic en Buscar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	25-01-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-15	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-06
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 2
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-06.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU25 - PU30</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-06
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion</li> <li>• Cargar diálogos del guion</li> <li>• Ingresar el estado <i>Aprobado</i> a un guion.</li> </ul>
<b>Fecha</b>	29-01-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible actualizar el estado de un guion.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten que el estado de un guion sea actualizado.

<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idtprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial, fecharegistro);     assertNotNull(result); } /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");     int idtguionprogramaradial = 6;     List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;();     result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial);     assertNotNull(result); } /**  * Test of actualizarGuionProgramaRadial method, of class  * mGuionProgramaRadial.  */ @Test public void testActualizarGuionProgramaRadial() throws Exception {     System.out.println("Modelo --- actualizarGuionProgramaRadial");     cGuionProgramaRadial objGuionPrograma = new cGuionProgramaRadial();     objGuionPrograma.setIdtguionprogramaradial(33);     objGuionPrograma.setTguionprogramaradialfecharegistro("2019-04-24");     objGuionPrograma.setTguionprogramaradialfechaaprobacion("2019-04-25");      cEstadoGuion objEstado = new cEstadoGuion();     objEstado.setIdtestadoguion(2);     objGuionPrograma.setObjEstadoguion(objEstado);      cProgramaradial objPrograma = new cProgramaradial();     objPrograma.setIdtprogramaradial(8);     objGuionPrograma.setObjProgramaradial(objPrograma);      objGuionPrograma.setTguionprogramaradialBandera(580);      boolean expResult = true;     boolean result = mGuionProgramaRadial.actualizarGuionProgramaRadial(objGuionPrograma);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como director</li> <li>2. Ubicarse en el panel de gestión de guiones</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en el botón Buscar</li> <li>5. En el panel de estado de guion elegir el estado aprobado</li> <li>6. Clic en Guardar</li> </ol>



<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	29-01-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-18	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-07
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-07.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU31 - PU35</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-07
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion</li> <li>• Cargar diálogos del guion</li> <li>• Ingresar el estado <i>No Aprobado</i> a un guion.</li> </ul>
<b>Fecha</b>	04-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible actualizar el estado de un guion.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten que el estado de un guion sea actualizado
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre>/**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();</pre>

	<pre>         result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial,         fecharegistro);         assertNotNull(result);     }     /**      * Test of obtenerDialogos method, of class mDialogoProgramaRadial.      */     @Test     public void testObtenerDialogos() throws Exception {         System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");         int idtguionprogramaradial = 6;         List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;();         result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial);         assertNotNull(result);     }     /**      * Test of actualizarGuionProgramaRadial method, of class      * mGuionProgramaRadial.      */     @Test     public void testActualizarGuionProgramaRadial() throws Exception {         System.out.println("Modelo --- actualizarGuionProgramaRadial");         cGuionProgramaRadial objGuionPrograma = new cGuionProgramaRadial();         objGuionPrograma.setIdtguionprogramaradial(33);         objGuionPrograma.setTguionprogramaradialfecharegistro("2019-04-24");         objGuionPrograma.setTguionprogramaradialfechaaprobacion("2019-04-25");          cEstadoGuion objEstado = new cEstadoGuion();         objEstado.setIdtestadoguion(3);         objGuionPrograma.setObjEstadoguion(objEstado);          cProgramaradial objPrograma = new cProgramaradial();         objPrograma.setIdtprogramaradial(8);         objGuionPrograma.setObjProgramaradial(objPrograma);          objGuionPrograma.setTguionprogramaradialBandera(580);          boolean expResult = true;         boolean result =         mGuionProgramaRadial.actualizarGuionProgramaRadial(objGuionPrograma);         assertEquals(expResult, result);     } } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como director</li> <li>2. Ubicarse en el panel de gestión de guiones</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en el botón Buscar</li> <li>5. En el panel de estado de guion elegir el estado no aprobado</li> <li>6. Clic en Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	04-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-19	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-08
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-08.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU31 - PU35</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-08
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion</li> <li>• Cargar diálogos del guion</li> <li>• Ingresar el estado <i>Revisado</i> a un guion.</li> </ul>
<b>Fecha</b>	06-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible actualizar el estado de un guion.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten que el estado de un guion sea actualizado
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idtprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial, fecharegistro);     assertNotNull(result); } /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos"); </pre>

	<pre> int idtguionprogramaradial = 6; List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;(); result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial); assertNotNull(result); } /**  * Test of actualizarGuionProgramaRadial method, of class  * mGuionProgramaRadial.  */ @Test public void testActualizarGuionProgramaRadial() throws Exception {     System.out.println("Modelo --- actualizarGuionProgramaRadial");     cGuionProgramaRadial objGuionPrograma = new cGuionProgramaRadial();     objGuionPrograma.setIdtguionprogramaradial(33);     objGuionPrograma.setTguionprogramaradialfecharegistro("2019-04-24");     objGuionPrograma.setTguionprogramaradialfechaaprobacion("2019-04-25");      cEstadoGuion objEstado = new cEstadoGuion();     objEstado.setIdtestadoguion(4);     objGuionPrograma.setObjEstadoguion(objEstado);      cProgramaradial objPrograma = new cProgramaradial();     objPrograma.setIdtprogramaradial(8);     objGuionPrograma.setObjProgramaradial(objPrograma);      objGuionPrograma.setTguionprogramaradialBandera(580);      boolean expResult = true;     boolean result = mGuionProgramaRadial.actualizarGuionProgramaRadial(objGuionPrograma);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como director</li> <li>2. Ubicarse en el panel de gestión de guiones</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en el botón Buscar</li> <li>5. En el panel de estado de guion elegir el revisado</li> <li>6. Clic en Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	06-02-2019.
<b>Observaciones</b>	Ninguna.

Historia Técnica	
<b>ID:</b> HT-20	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-09
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-09.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU41 - PU45</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-09
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion</li> <li>• Cargar diálogos del guion</li> <li>• Insertar Observación del Guion del Programa Radial</li> </ul>
<b>Fecha</b>	08-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible ingresar las observaciones para un programa de radio.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten que las observaciones respectivas sean ingresadas para un guion de programa radial.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idtprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial, fecharegistro);     assertNotNull(result); } /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos"); </pre>

	<pre> int idtguionprogramaradial = 6; List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;(); result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial); assertNotNull(result); } /**  * Test of insertarObservacion method, of class mObservacionProgramaRadial.  */ @Test public void testInsertarObservacion() throws Exception {     System.out.println("Modelo Observ.Programa Radial - insertarObservacion");     cObservacionGuionPrograma objObservacion = new cObservacionGuionPrograma();     objObservacion.setTobservacionguionprogramatitulo("Test 1");     objObservacion.setTobservacionprogramadescripcion("Test 1 -- desc");     cGuionProgramaRadial obj = new cGuionProgramaRadial();     obj.setIdtguionprogramaradial(6);     objObservacion.setObjGuionProgramaRadial(obj);     boolean expResult = true;     boolean result = mObservacionProgramaRadial.insertarObservacion(objObservacion);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como director</li> <li>2. Ubicarse en el panel de gestión de guiones</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en el botón Buscar</li> <li>5. En el panel de observaciones de guion, ingresar los campos correspondientes</li> <li>6. Clic en Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	08-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-21	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-10
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 3
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-10.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU46 - PU48</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-10
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Cargar Observación del Guion</li> <li>• Cargar Diálogos del Guion</li> </ul>
<b>Fecha</b>	13-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible visualizar las observaciones para un programa de radio.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permiten visualizar las observaciones respectivas de un guion de programa radial.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerObserbacionPorId method, of class  * mObservacionProgramaRadial.  */ @Test public void testObtenerObserbacionPorId() throws Exception {     System.out.println("Modelo Observ.Programa Radial - obtenerObserbacionPorId");     int idtguionprogramaradial = 6;     cObservacionGuionPrograma result = new cObservacionGuionPrograma();     result = mObservacionProgramaRadial.obtenerObserbacionPorId(idtguionprogramaradial);     assertNotNull(result); }  /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");     int idtguionprogramaradial = 6; </pre>

	<pre>List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;(); result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial); assertNotNull(result); }</pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como locutor</li> <li>2. Ubicarse en el panel de consulta de sugerencias de un guion</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en Buscar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	13-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-24	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-11
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-11.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU49 - PU54</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-11
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar un guion de programa radial por nombre y fecha de registro</li> <li>• Cargar Guion</li> <li>• Modificar Guion</li> </ul>
<b>Fecha</b>	13-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible visualizar las observaciones para un programa de radio.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite actualizar un guion.



<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerGuionPorId method, of class mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorId() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorId");     int idtguionprogramaradial = 33;     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorId(idtguionprogramaradial);     assertNotNull(result); }  /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");     int idtguionprogramaradial = 6;     List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;();     result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial);     assertNotNull(result); }  /**  * Test of actualizarGuionProgramaRadial method, of class  * mGuionProgramaRadial.  */ @Test public void testActualizarGuionProgramaRadial() throws Exception {     System.out.println("Modelo --- actualizarGuionProgramaRadial");     cGuionProgramaRadial objGuionPrograma = new cGuionProgramaRadial();     objGuionPrograma.setIdtguionprogramaradial(33);     objGuionPrograma.setTguionprogramaradialfecharegistro("2019-04-24");     objGuionPrograma.setTguionprogramaradialfechaaprobacion("2019-04-25");      cEstadoGuion objEstado = new cEstadoGuion();     objEstado.setIdtestadoguion(2);     objGuionPrograma.setObjEstadoguion(objEstado);      cProgramaradial objPrograma = new cProgramaradial();     objPrograma.setIdtprogramaradial(8);     objGuionPrograma.setObjProgramaradial(objPrograma);      objGuionPrograma.setTguionprogramaradialBandera(580);      boolean expResult = true;     boolean result = mGuionProgramaRadial.actualizarGuionProgramaRadial(objGuionPrograma);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como locutor</li> <li>2. Ubicarse en el panel de consulta de sugerencias de un guion</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en Buscar</li> <li>5. Clic en modificar guion</li> <li>6. Ingresar las modificaciones necesarias al guion</li> <li>7. Clic en Guardar</li> </ol>

<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	13-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-25	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-12
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-12.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU55 - PU60</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-12
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar Guion de un Programa Radial por Nombre y Fecha de Registro</li> <li>• Cargar Observaciones del Guion</li> <li>• Cargar Diálogo del Guion</li> <li>• Modificar Diálogo Programa Radial</li> </ul>
<b>Fecha</b>	18-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible modificar el guion de un programa radial.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite actualizar el guion de un programa radial.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre>/**  * Test of obtenerGuionPorId method, of class mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorId() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorId");     int idtguionprogramaradial = 33;     cGuionProgramaRadial result = new cGuionProgramaRadial();     result = mGuionProgramaRadial.obtenerGuionPorId(idtguionprogramaradial);     assertNotNull(result); }</pre>

	<pre> }  /**  * Test of obtenerDialogos method, of class mDialogoProgramaRadial.  */ @Test public void testObtenerDialogos() throws Exception {     System.out.println("Modelo Dialogo Guion Programa --- obtenerDialogos");     int idtguionprogramaradial = 6;     List&lt;cDialogo&gt; result = new ArrayList&lt;&gt;();     result = mDialogoProgramaRadial.obtenerDialogos(idtguionprogramaradial);     assertNotNull(result); }  /**  * Test of obtenerObserbacionPorId method, of class  * mObservacionProgramaRadial.  */ @Test public void testObtenerObserbacionPorId() throws Exception {     System.out.println("Modelo Observ.Programa Radial - obtenerObserbacionPorId");     int idtguionprogramaradial = 6;     cObservacionGuionPrograma result = new cObservacionGuionPrograma();     result = mObservacionProgramaRadial.obtenerObserbacionPorId(idtguionprogramaradial);     assertNotNull(result); }  /**  * Test of actualizarGuionProgramaRadial method, of class  * mGuionProgramaRadial.  */ @Test public void testActualizarGuionProgramaRadial() throws Exception {     System.out.println("Modelo --- actualizarGuionProgramaRadial");     cGuionProgramaRadial objGuionPrograma = new cGuionProgramaRadial();     objGuionPrograma.setIdtguionprogramaradial(33);     objGuionPrograma.setTguionprogramaradialfecharegistro("2019-04-24");     objGuionPrograma.setTguionprogramaradialfechaaprobacion("2019-04-25");      cEstadoGuion objEstado = new cEstadoGuion();     objEstado.setIdtestadoguion(2);     objGuionPrograma.setObjEstadoguion(objEstado);      cProgramaradial objPrograma = new cProgramaradial();     objPrograma.setIdtprogramaradial(8);     objGuionPrograma.setObjProgramaradial(objPrograma);      objGuionPrograma.setTguionprogramaradialBandera(580);      boolean expResult = true;     boolean result = mGuionProgramaRadial.actualizarGuionProgramaRadial(objGuionPrograma);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como locutor</li> <li>2. Ubicarse en el panel de consulta de sugerencias de un guion</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en Buscar</li> <li>5. Clic en modificar guion</li> </ol>

	6. Ingresar las modificaciones necesarias al guion 7. Clic en Guardar
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	18-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-26	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-13
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-13.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU61 - PU63</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-13
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Cargar Diálogos del Guion</li> </ul>
<b>Fecha</b>	18-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible visualizar el estado del guion.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite visualizar el estado del guion.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre>/**  * Test of obtenerGuionPorIdFechaRegistro method, of class  * mGuionProgramaRadial.  */ @Test public void testObtenerGuionPorIdFechaRegistro() throws Exception {     System.out.println("Modelo Guion --- obtenerGuionPorIdFechaRegistro");     int idtprogramaradial = 6;     String fecharegistro = "2019-04-13";     cGuionProgramaRadial result = new cGuionProgramaRadial();</pre>

	<pre>         result = mGuionProgramaRadial.obtenerGuionPorIdFechaRegistro(idtprogramaradial,         fecharegistro);         assertNotNull(result);     }      /**      * Test of obtenerObserbacionPorId method, of class      * mObservacionProgramaRadial.      */     @Test     public void testObtenerObserbacionPorId() throws Exception {         System.out.println("Modelo Observ. Programa Radial - obtenerObserbacionPorId");         int idtguionprogramaradial = 6;         cObservacionGuionPrograma result = new cObservacionGuionPrograma();         result =         mObservacionProgramaRadial.obtenerObserbacionPorId(idtguionprogramaradial);         assertNotNull(result);     } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como locutor</li> <li>2. Ubicarse en el panel de consulta de estado de un guion</li> <li>3. Ingresar el campo de criterio de búsqueda del guion</li> <li>4. Clic en Buscar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	18-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-27	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-14
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-14.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU64 - PU73</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-14
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar un programa radial por el nombre.</li> <li>• Cargar Tipo de Programa.</li> <li>• Cargar Usuario.</li> <li>• Ingresar los nuevos datos del usuario.</li> <li>• Cargar Listado de Programas Radiales registrados en el sistema.</li> </ul>
<b>Fecha</b>	20-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible modificar la información de los programas radiales.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite modificar la información de los programas radiales almacenados.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerProgramaPorNombre method, of class mProgramaRadial.  */ @Test public void testObtenerProgramaPorNombre() throws Exception {     System.out.println("Modelo Programa Radial --- obtenerProgramaPorNombre");     String nombrePrograma = "La voz Juvenil";     cProgramaradial result = new cProgramaradial();     result = mProgramaRadial.obtenerProgramaPorNombre(nombrePrograma);     assertNotNull(result); } /**  * Test of obtenerTodos method, of class mTipoProgramaRadial.  */ @Test public void testObtenerTodos() {     System.out.println("Modelo Tipo Programa Radial - obtenerTodos");     List&lt;cTipoProgramaRadial&gt; result = new ArrayList&lt;&gt;();     result = mTipoProgramaRadial.obtenerTodos();     assertNotNull(result); } /**  * Test of obtenerTodos method, of class mUsuario.  */ @Test public void testObtenerTodos() throws Exception {     System.out.println("Modelo Usuario --- obtenerTodos");     List&lt;cUsuario&gt; result = new ArrayList&lt;&gt;();     result = mUsuario.obtenerTodos();     assertNotNull(result); } /**  * Test of actualizarProgramaPorNombre method, of class mProgramaRadial.  */ @Test public void testActualizarProgramaPorNombre() throws Exception {     System.out.println("Modelo Programa Radial --- actualizarProgramaPorNombre");     cProgramaradial objProgramaRadial = new cProgramaradial();      objProgramaRadial.setIdprogramaradial(18);     objProgramaRadial.setTprogramaradialnombre("Pura Fe");     objProgramaRadial.setTprogramaradialdescripcion("Pura Fe");     objProgramaRadial.setTprogramaradialdiatransmision("Sábado");     objProgramaRadial.setTprogramaradialhoratransmision("10h00-12h00");     objProgramaRadial.setTprogramaradialduracion("2 HORAS");     cTipoProgramaRadial objtipo = new cTipoProgramaRadial();     objtipo.setIdtipoprogramaradial(3);     objProgramaRadial.setObjTipoProgramaRadial(objtipo);     cClasificacionProgramaRadial objClasif = new cClasificacionProgramaRadial(); </pre>

	<pre> objClasif.setIdclasificacionprogramaradial(1); objProgramaRadial.setObjClasificacionProgramaRadial(objClasif); cUsuario objU = new cUsuario(); objU.setIdusuario(51); objProgramaRadial.setObjUsuario(objU); boolean expResult = true; boolean result = mProgramaRadial.actualizarProgramaPorNombre(objProgramaRadial); assertEquals(expResult, result);  } /**  * Test of obtenerTodos method, of class mProgramaRadial.  */ @Test public void testObtenerTodos() {     System.out.println("Modelo Programa Radial --- obtenerTodos");     List&lt;cProgramaradial&gt; result = new ArrayList&lt;&gt;();     result = mProgramaRadial.obtenerTodos();     assertNotNull(result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como administrador</li> <li>2. Ubicarse en el panel de modificar programas radiales</li> <li>3. Ingresar el campo de criterio de búsqueda</li> <li>4. Clic en Buscar</li> <li>5. En el panel de datos de programas radiales, modificar los campos correspondientes</li> <li>6. Clic en Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	20-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-28	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-15
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-15.	

**Observaciones:** Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.

<b>Id</b>	<b>PU74 - PU79</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-15
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar un programa radial por el nombre</li> <li>• Eliminar Programa Radial</li> <li>• Cargar Listado de Programas Radiales registrados en el sistema.</li> </ul>
<b>Fecha</b>	22-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible eliminar la información de los programas radiales.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite modificar la información de los programas radiales almacenados.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerProgramaPorNombre method, of class mProgramaRadial.  */ @Test public void testObtenerProgramaPorNombre() throws Exception {     System.out.println("Modelo Programa Radial --- obtenerProgramaPorNombre");     String nombrePrograma = "La voz Juvenil";     cProgramaradial result = new cProgramaradial();     result = mProgramaRadial.obtenerProgramaPorNombre(nombrePrograma);     assertNotNull(result); } /**  * Test of obtenerTodos method, of class mTipoProgramaRadial.  */ @Test public void testObtenerTodos() {     System.out.println("Modelo Tipo Programa Radial - obtenerTodos");     List&lt;cTipoProgramaRadial&gt; result = new ArrayList&lt;&gt;();     result = mTipoProgramaRadial.obtenerTodos();     assertNotNull(result); } /**  * Test of obtenerTodos method, of class mUsuario.  */ @Test public void testObtenerTodos() throws Exception {     System.out.println("Modelo Usuario --- obtenerTodos");     List&lt;cUsuario&gt; result = new ArrayList&lt;&gt;();     result = mUsuario.obtenerTodos();     assertNotNull(result); } /**  * Test of eliminarProgramaPorNombre method, of class mProgramaRadial.  */ @Test public void testEliminarProgramaPorNombre() throws Exception {     System.out.println("Modelo Programa Radial --- eliminarProgramaPorNombre");     cProgramaradial objProgramaRadial = new cProgramaradial();     objProgramaRadial.setIdprogramaradial(18);     objProgramaRadial.setObjUsuario(objU);     boolean expResult = true;     boolean result = mProgramaRadial.eliminarProgramaPorNombre(objProgramaRadial);     assertEquals(expResult, result); </pre>



	<pre> } /**  * Test of obtenerTodos method, of class mProgramaRadial.  */ @Test public void testObtenerTodos() {     System.out.println("Modelo Programa Radial --- obtenerTodos");     List&lt;cProgramaradial&gt; result = new ArrayList&lt;&gt;();     result = mProgramaRadial.obtenerTodos();     assertNotNull(result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como administrador</li> <li>2. Ubicarse en el panel de eliminar programas radiales</li> <li>3. Ingresar el campo de criterio de búsqueda</li> <li>4. Clic en Buscar</li> <li>5. En el panel de datos de programas radiales, visualizar el programa a eliminar</li> <li>6. Clic en Eliminar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	22-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-29	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-16
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 4
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-16.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU80 - PU83</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-16
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar Usuario por Cédula</li> <li>• Modificar Usuario</li> </ul>
<b>Fecha</b>	25-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible modificar la información de los usuarios registrados en el sistema.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite modificar la información de los usuarios registrados en el sistema.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerUsuarioporId method, of class mUsuario.  */ @Test public void testObtenerUsuarioporId() throws Exception {     System.out.println(" Modelo Usuario --- obtenerUsuarioporId");     int idusuario = 51;     cUsuario result = new cUsuario();     result = mUsuario.obtenerUsuarioporId(idusuario);     assertNotNull(result); } /**  * Test of actualizarUsuarioPorCedula method, of class mUsuario.  */ @Test public void testActualizarUsuarioPorCedula() throws Exception {     System.out.println("Modelo Usuario --- actualizarUsuarioPorCedula");     cUsuario objUsuario = new cUsuario();     objUsuario.setIdusuario(51);     objUsuario.setTusuarionombre("Niko");     objUsuario.setTusuarioapellido("Evans");     objUsuario.setTusuariocedula("021589745-0");     objUsuario.setTusuariocontrasenia("123456");     cTipoUsuario obj = new cTipoUsuario();     obj.setIdusuarionombre(3);     objUsuario.setObjTipoUsuario(obj);      boolean expectedResult = true;     boolean result = mUsuario.actualizarUsuarioPorCedula(objUsuario);     assertEquals(expectedResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como administrador</li> <li>2. Ubicarse en el panel de modificar datos del usuario</li> <li>3. Ingresar el campo de criterio de búsqueda</li> <li>4. Clic en Buscar</li> <li>5. En el panel de datos de usuario, modificar los campos requeridos</li> <li>6. Clic en Guardar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	25-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-32	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-17
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-17.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU84 - PU87</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-17
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar Usuario por Cédula</li> <li>• Eliminar Usuario</li> </ul>
<b>Fecha</b>	27-02-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible eliminar la información de los usuarios registrados en el sistema.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite eliminar la información de los usuarios registrados en el sistema.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.
<b>Código</b>	<pre> /**  * Test of obtenerUsuarioPorId method, of class mUsuario.  */ @Test public void testObtenerUsuarioPorId() throws Exception {     System.out.println(" Modelo Usuario --- obtenerUsuarioPorId");     int idusuario = 51;     cUsuario result = new cUsuario();     result = mUsuario.obtenerUsuarioPorId(idusuario);     assertNotNull(result); } /**  * Test of eliminarUsuarioPorId method, of class mUsuario.  */ @Test public void testEliminarUsuarioPorId() throws Exception {     System.out.println("Modelo Usuario --- eliminarUsuarioPorId");     int idusuario = 54;     boolean expResult = true;     boolean result = mUsuario.eliminarUsuarioPorId(idusuario);     assertEquals(expResult, result); } </pre>

<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como administrador</li> <li>2. Ubicarse en el panel de eliminar datos del usuario</li> <li>3. Ingresar el campo de criterio de búsqueda</li> <li>4. Clic en Buscar</li> <li>5. En el panel de datos de usuario, revisar el usuario a eliminar</li> <li>6. Clic en Eliminar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	27-02-2019.
<b>Observaciones</b>	Ninguna.

<b>Historia Técnica</b>	
<b>ID:</b> HT-33	<b>Descripción:</b> Como desarrollador necesito realizar las pruebas unitarias automatizadas para la HU-18
<b>Modificación de historia de usuario:</b>	
<b>Usuario:</b> Desarrollador	<b>Sprint Asignado:</b> 5
<b>Prioridad en el Negocio:</b> Media (Alta / Media / Baja)	<b>Puntos Estimados:</b> 6
<b>Riesgo en el Desarrollo:</b> Medio (Alto / Medio / Bajo)	<b>Puntos Reales:</b> 6
<b>Descripción:</b> Como <i>desarrollador</i> necesito realizar las pruebas unitarias automatizadas para la HU-18.	
<b>Observaciones:</b> Las pruebas unitarias deben ser generadas con JUNIT y se debe inicializar los valores necesarios para que el test no falle.	

<b>Id</b>	<b>PU84 - PU87</b>
<b>Caso de Prueba</b>	Pruebas unitarias que corresponden a la funcionalidad de la Historia de Usuario HU-18
<b>Descripción</b>	Los casos de prueba presentados, pertenecen a las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Buscar guion del programa por Nombre de programación y fecha de registro.</li> <li>• Cargar diálogos del guion</li> <li>• Eliminar Guion</li> </ul>
<b>Fecha</b>	08-03-2019.
<b>Funcionalidad / Característica</b>	Estos casos de prueba unitaria permiten probar las unidades de código que hacen posible eliminar la información de los usuarios registrados en el sistema.
<b>Datos / Acciones de Entrada</b>	Deben ser inicializados en la prueba unitaria los atributos que permite eliminar la información de los usuarios registrados en el sistema.
<b>Resultado Esperado</b>	La prueba unitaria debe ser exitosa.

<b>Código</b>	<pre> /**  * Test of obtenerUsuarioPorId method, of class mUsuario.  */ @Test public void testObtenerUsuarioPorId() throws Exception {     System.out.println(" Modelo Usuario --- obtenerUsuarioPorId");     int idusuario = 51;     cUsuario result = new cUsuario();     result = mUsuario.obtenerUsuarioPorId(idusuario);     assertNotNull(result); } /**  * Test of eliminarUsuarioPorId method, of class mUsuario.  */ @Test public void testEliminarUsuarioPorId() throws Exception {     System.out.println("Modelo Usuario --- eliminarUsuarioPorId");     int idusuario = 54;     boolean expResult = true;     boolean result = mUsuario.eliminarUsuarioPorId(idusuario);     assertEquals(expResult, result); } </pre>
<b>Dependencias con otros casos de Prueba</b>	No.
<b>Información para el seguimiento</b>	
<b>Pasos de ejecución</b>	<ol style="list-style-type: none"> <li>1. Ingresar al sistema como locutor</li> <li>2. Ubicarse en el panel de eliminar guion de programa radial</li> <li>3. Ingresar el campo de criterio de búsqueda</li> <li>4. Clic en Buscar</li> <li>5. En el panel de datos de guion, revisar el guion de programa radial</li> <li>6. Clic en Eliminar</li> </ol>
<b>Estado</b>	Éxito.
<b>Última Fecha de Estado</b>	08-03-2019.
<b>Observaciones</b>	Ninguna.

**Anexo F:** Planificación de pruebas unitarias por requerimientos del sistema SIGUIÓN

REQUERIMIENTO	FUNCIONALIDADES	MÉTODOS		LINEAS DE CÓDIGO	% DE COBERTURA DE CÓDIGO	TOTAL		
		Nº CASO	NOMBRE					
HU-01.- Como usuario necesito autenticar mis credenciales para ingresar al sistema.	F1.- Iniciar Sesión	PU1	Guion Controlador login()	28	100 %	52 99%		
		PU2	Modelo Usuario obtenerUsuarioporCedula(String)	18	98%			
	F2.- Cerrar Sesión	PU3	Guion Controlador cerrarSession()	6	100%			
HU-02.- Como administrador del sistema necesito ingresar nuevos usuarios al sistema.	F1.- Cargar Usuarios Registrados en el sistema	PU4	Usuario Controlador cargarUsuario()	5	100%	57 99%		
		PU5	Modelo Usuario obtenerTodos()	18	100%			
	F2.- Insertar Usuario Nuevo	PU6	Usuario Controlador insertarNuevoUsuario()	18	100%			
		PU7	Modelo Usuario insertarNuevoUsuario()	16	98%			
HU-03.- Como administrador del sistema necesito ingresar programas radiales al sistema.	F1.- Cargar Programas Radiales Registrados en el sistema.	PU8	Programa Radial Controlador cargaProgramaRadial()	6	100%	115 99%		
		PU9	Modelo Programa Radial obtenerTodos()	19	100%			
	F2.- Insertar Usuario Nuevo.	PU10	Programa Radial Controlador insertarNuevoProgramaRadial()	20	98%			
		PU11	Modelo Programa Radial insertarNuevoPrograma()	19	98%			
	F3.- Cargar Tipo de Programación.	PU12	Programa Radial Controlador getLstTipoProgramaItem()	10	98%			
		PU13	Modelo Tipo Programa Radial obtenerTodos()	13	98%			
	F4.- Carga Usuario	PU14	Programa Radial Contralador getLstUsuarioItem()	10	100%			
		PU15	Modelo Usuario obtenerTodos()	18	100%			
	HU-04.- Como locutor necesito ingresar guiones de programas radiales.	F1.- Crear un guion.	PU16	Guion Controlador crearGuionPrograma()	25		100%	111 98%
			PU17	Modelo Guion Programa Radial insertarNuevoGuion()	16		98%	
F2.- Agregar diálogo al guion.		PU18	Guion Controlador insertar_Dialogo_Guion_Creado()	19	100%			
		PU19	Modelo Diálogo Programa Radial insertarDialogoProgramaRadial()	16	98%			
F3.- Cargar los diálogos del guion.		PU20	Guion Controlador cargarDialogos_Guion()	17	98%			
		PU21	Modelo Guion Controlador obtenerGuionPorIdFechaRegistro()	18	96%			
HU-05.- Como director necesito visualizar el guion de un programa radial.		F1.- Buscar Guion.	PU22	Guion Controlador buscarGuionPrograma_CargarDialogos()	19	98%	59 97%	
	PU23		Modelo Guion obtenerGuionPorIdFechaRegistro()	19	96%			
	F2.- Cargar diálogos del guion.	PU24	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%			

<b>HU-06.-</b> Como director necesito realizar la aprobación del guion de un programa radial.	<b>F1.-</b> Buscar Guion.	PU25	Guion Controlador buscarGuionPrograma_CargarDialogos()	19	98%	130 96%
		PU26	Modelo Guion obtenerGuionPorIdFechaRegistro()	18	96%	
	<b>F2.-</b> Cargar diálogos del guion	PU27	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
	<b>F3.-</b> Ingresar el estado Aprobado al guion	PU28	Guion Controlador cambiarEstadoGuion()	40	96%	
		PU29	Modelo Estado Guion obtenerEstadoPorId()	14	95%	
		PU30	Modelo Guion Programa Radial actualizarGuionProgramaRadial()	18	100%	
<b>HU-07.-</b> Como director necesito ingresar el estado de No Aprobado a un guion del programa radial.	<b>F1.-</b> Buscar Guion	PU31	Guion Controlador buscarGuionPrograma_CargarDialogos()	19	98%	112 96%
		PU32	Modelo Guion obtenerGuionPorIdFechaRegistro()	18	96%	
	<b>F2.-</b> Cargar diálogos del guion	PU33	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
	<b>F3.-</b> Ingresar el estado No Aprobado al guion	PU34	Guion Controlador cambiarEstadoGuion()	40	96%	
		PU35	Modelo Estado Guion obtenerEstadoPorId()	14	95%	
<b>HU-08.-</b> Como director necesito ingresar el estado de Revisado a un guion del programa radial.	<b>F1.-</b> Buscar Guion	PU36	Guion Controlador buscarGuionPrograma_CargarDialogos()	19	98%	112 96%
		PU37	Modelo Guion obtenerGuionPorIdFechaRegistro()	18	96%	
	<b>F2.-</b> Cargar diálogos del guion	PU38	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
	<b>F3.-</b> Ingresar el estado Revisado al guion	PU39	Guion Controlador cambiarEstadoGuion()	40	96%	
		PU40	Modelo Estado Guion obtenerEstadoPorId()	14	95%	
<b>HU-09.-</b> Como director necesito ingresar las sugerencias respectivas al guion de un programa de la radio.	<b>F1.-</b> Buscar Guion	PU41	Guion Controlador buscarGuionPrograma_CargarDialogos()	19	98%	88 98%
		PU42	Modelo Guion obtenerGuionPorIdFechaRegistro()	18	96%	
	<b>F2.-</b> Cargar diálogos del guion	PU43	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
	<b>F3.-</b> Insertar Observación del Guion del Programa Radial	PU44	Guion Controlador ingresarObservacion_GuionProgramaRadial()	16	100%	
		PU45	Modelo Observación Programa Radial insertarObservacion()	14	100%	
<b>HU-10.-</b> Como locutor necesito visualizar las sugerencias del guion de un programa radio.	<b>F1.-</b> Cargar Observación del Guion	PU46	Guion Controlador cargarObservacion_Guion()	17	98%	53 98%
		PU47	Modelo Observación obtenerObservacionPorId()	15	100%	
	<b>F2.-</b> Cargar Diálogos del Guion	PU48	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
<b>HU-11.-</b> Como locutor necesito modificar el guion de un programa de radio.	<b>F1.-</b> Buscar un guion de programa radial por nombre y fecha de registro	PU49	Guion Controlador buscarGuionProgramaPorNombreFechaR()	19	96%	122 98%
		PU50	Modelo Programa Radial obtenerProgramaPorNombre()	20	98%	
	<b>F2.-</b> Cargar Guion	PU51	Guion Controlador cargarDialogoGuion()	17	100%	
		PU52	Modelo Diálogo Programa Radial obtenerDialogoGuion()	21	96%	
	<b>F3.-</b> Modificar Guion	PU53	Guion Controlador modificarGuionPrograma()	27	100%	
		PU54	Modelo Diálogo Radial actualizarGuion()	18	100%	

<b>HU-12.-</b> Como locutor necesito realizar los respectivos cambios al guion de un programa de radio.	<b>F1.-</b> Buscar Guion de un Programa Radial por Nombre y Fecha de Registro	PU55	Guion Controlador buscarGuionFechaRegistro()	21	96%	111 95%
		PU56	Modelo Guion Programa buscarGuionFechaRegistro()	17	96%	
	<b>F2.-</b> Cargar Observaciones del Guion	PU57	Modelo Observación Programa Radial obtenerObservacionPorId()	15	95%	
	<b>F3.-</b> Cargar Diálogo del Guion	PU58	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
	<b>F4.-</b> Modificar Diálogo Programa Radial	PU59	Guion Controlador actualizarDialogoGuion()	20	96%	
		PU60	Modelo Diálogo Programa Radial actualizarDialogoProgramaRadialFR()	17	91%	
<b>HU-13.-</b> Como locutor necesito visualizar el estado del guion.	<b>F1.-</b> Cargar Diálogos del Guion	PU61	Guion Controlador cargarDialogos_Guion()	17	96%	56 96%
		PU62	Modelo Guion Programa Radial obtenerGuionPorIdFechaRegistro()	18	96%	
		PU63	Modelo Diálogo Programa Radial obtenerDialogos()	21	96%	
<b>HU-14.-</b> Como administrador del sistema necesito modificar los programas radiales.	<b>F1.-</b> Buscar un programa radial por el nombre	PU64	Programa Radial Controlador buscarProgramaPorNombre()	16	96%	154 98%
		PU65	Modelo Programa Radial obtenerProgramaPorNombre()	20	96%	
	<b>F2.-</b> Cargar Tipo de Programa	PU66	Programa Radial Controlador getLstTipoProgramaItem()	10	100%	
		PU67	Modelo Tipo Programa Radial obtenerTodos()	13	98%	
	<b>F3.-</b> Cargar Usuario	PU68	Programa Radial Controlador getLstUsuarioItem()	10	100%	
		PU69	Modelo Usuario obtenerTodos()	18	96%	
	<b>F4.-</b> Ingresar los nuevos datos del usuario	PU70	Programa Radial Controlador modificarProgramaNombre()	21	98%	
		PU71	Modelo Programa Radial actualizarProgramaPorNombre()	21	98%	
	<b>F5.-</b> Cargar Listado de Programas Radiales registrados en el sistema.	PU72	Programa Radial Controlador cargaProgramaRadial()	6	100%	
		PU73	Modelo Programa Radial obtenerTodos()	19	100%	
<b>HU-15.-</b> Como administrador del sistema necesito eliminar los programas radiales.	<b>F1.-</b> Buscar un programa radial por el nombre	PU74	Programa Radial Controlador buscarProgramaPorNombre()	16	96%	90 98%
		PU75	Modelo Programa Radial obtenerProgramaPorNombre()	20	96%	
	<b>F2.-</b> Eliminar Programa Radial	PU76	Programa Radial Controlador eliminarProgramaRadialId()	20	96%	
		PU77	Modelo Programa Radial eliminarProgramaRadialPorId()	9	100%	
	<b>F3.-</b> Cargar Listado de Programas Radiales registrados en el sistema.	PU78	Programa Radial Controlador cargaProgramaRadial()	6	100%	
		PU79	Modelo Programa Radial obtenerTodos()	19	100%	
<b>HU-16.-</b> Como administrador del sistema necesito modificar la información de los usuarios.	<b>F1.-</b> Buscar Usuario por Cédula	PU80	Usuario Controlador buscarUsuarioCedula()	6	100%	61 98%
		PU81	Modelo Usuario buscarUsuarioCedula()	18	96%	
	<b>F2.-</b> Modificar Usuario	PU82	Usuario Controlador modificarUsuariosCedula()	19	98%	
		PU83	Modelo Usuario actualizarUsuarioPorCedula()	18	97%	
		PU84	Usuario Controlador buscarUsuarioCedula()	6	100%	



<b>HU-17.-</b> Como administrador del sistema necesito eliminar los usuarios ingresados.	<b>F1.-</b> Buscar Usuario por Cédula	PU85	Modelo Usuario buscarUsuarioCedula()	18	96%	98%
	<b>F2.-</b> Eliminar Programa Radial	PU86	Programa Radial eliminarProgramaRadialId()	19	98%	
		PU87	Modelo Programa Radial eliminarProgramaRadialPorId ()	18	96%	
<b>HU-18.-</b> Como locutor necesito eliminar un guion del programa radial.	<b>F1.-</b> Buscar guion del programa por Nombre de programación y fecha de registro.	PU88	Programa Radial Controlador buscarGuionNombreFecha()	16	100%	116 98%
		PU89	Modelo Programa Radial obtenerGuionNombreFecha()	22	96%	
	<b>F2.-</b> Cargar diálogos del guion	PU90	Programa Radial Controlador cargarDiálogos()	17	96%	
		PU91	Modelo Programa Radial obtenerDiálogos()	21	100%	
	<b>F3.-</b> Eliminar Guion	PU92	Programa Radial Controlador eliminarGuionProgramaRadial()	18	100%	
		PU93	Modelo Programa Radial eliminarGuion()	22	96%	
	<b>HU-19.-</b> Como director necesito realizar el reporte de guiones no aprobados en la semana en formato pdf.	<b>F1.-</b> Cargar Reporte Guion No Aprobado	PU94	Reporte Controlador cargarReporteGuionNoAprobado()	21	
PU95			Modelo Reporte Guion obtenerGuionNoAprobadoListaTodos()	30	100%	
<b>HU-20.-</b> Como director necesito realizar el reporte de los guiones aprobados en la semana en formato pdf.	<b>F1.-</b> Cargar Reporte Guion Aprobado	PU96	Reporte Controlador cargarGuionAprobado()	20	100%	70 99%
		PU97	Reporte Controlador cargarReporteGuionAprobado()	20	100%	
		PU98	Modelo Reporte Guion obtenerGuionAprobadoListaTodos()	30	98%	
<b>HU-21.-</b> Como director necesito realizar el reporte de los guiones revisados en la semana en formato pdf.	<b>F1.-</b> Cargar Reporte Guion Revisado	PU99	Reporte Controlador cargarReporteGuionRevisado()	21	100%	51 99%
		PU100	Modelo Reporte Guion obtenerGuionRevisadoListaTodos()	30	98%	
<b>HU-22.-</b> Como administrador del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	<b>F1.-</b> Cargar Reporte de Guion Aprobado por Fecha de Registro	PU101	Reporte Controlador cargarGuionAprobado_FechaRegistro()	22	100%	54 99%
		PU102	Modelo Reporte Guion Aprobado obtenerGuionAprobadoFRListaTodos()	32	98%	
<b>HU-23.-</b> Como administrador del sistema necesito realizar el reporte de los guiones aprobados que coincidan con una fecha de registro.	<b>F1.-</b> Cargar Reporte Guion No aprobado por Fecha de Registro	PU103	Reporte Controlador cargarGuionNoAprobado_FechaRegistro()	22	100%	54 99%
		PU104	Modelo Reporte Guion obtenerGuionNoAprobadoFRListaTodos()	32	98%	

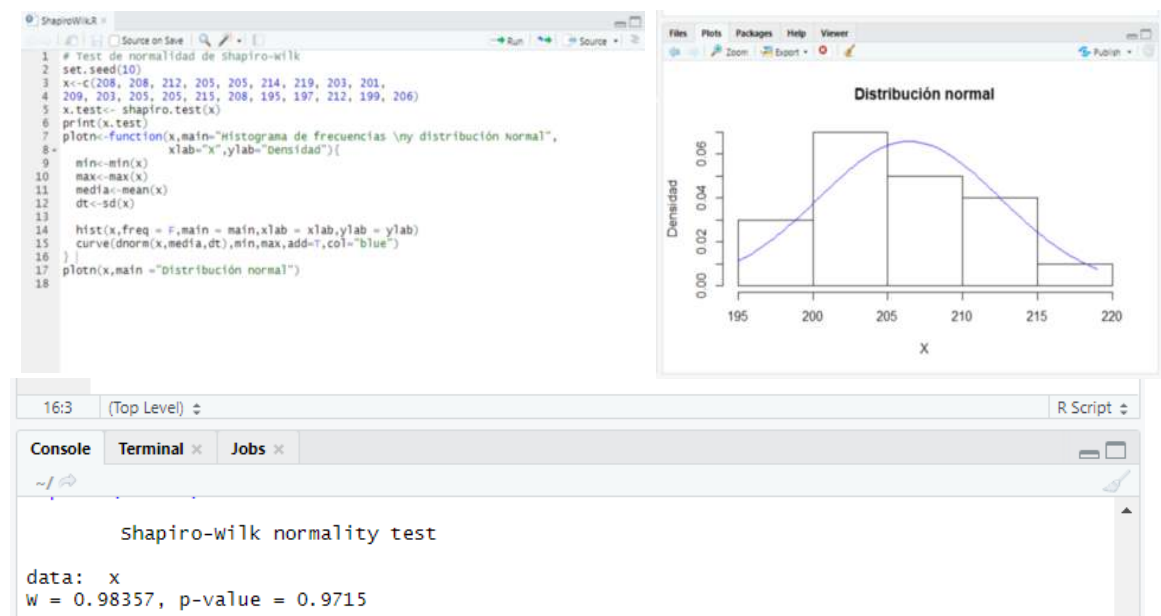
## Anexo G: Contraste de normalidad Shapiro-Wilks

Para determinar si el conjunto de datos (208, 208, 212, 205, 205, 214, 219, 203, 201, 209, 203, 205, 205, 215, 208, 195, 197, 212, 199, 206) procede de una población con una distribución normal se procede a realizar la prueba de contraste de normalidad Shapiro-Wilks, donde se plantea una hipótesis nula y una alternativa donde:

**H<sub>0</sub>**: La muestra procede de una población normal.

**H<sub>1</sub>**: La muestra no procede de una población normal.

Para realizar la prueba se realiza el contraste de la normalidad con la herramienta estadística R Studio, como se adjunta en los siguientes gráficos.



El gráfico muestra, el estadístico de contraste (W) con un valor de 0.98357, y una probabilidad de contraste (p) igual a 0.9715, este último representa una probabilidad crítica muy grande por lo que no tenemos suficiente evidencia para rechazar la hipótesis nula.