

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**TESIS DE GRADO**

**“PLATAFORMA MULTICANAL EN LINEA PARA LA GESTION Y  
SEGUIMIENTO DE TICKETS PARA EMPRESAS EN TELECOMUNICACIONES”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POSTULANTE: UNIV. ERICK AGUILAR LOZANO  
TUTOR METODOLÓGICO: LIC. JUAN GONZALO CONTRERAS CANDIA**

**LA PAZ – BOLIVIA**

**2021**

HOJA DE CALIFICACIONES  
UNIVERSIDAD MAYOR DE SAN ANDRES  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMATICA

Tesis de Grado

“PLATAFORMA MULTICANAL EN LINEA PARA LA GESTION Y SEGUIMIENTO  
DE TICKETS PARA EMPRESAS EN TELECOMUNICACIONES”

Presentado por: Erick Aguilar Lozano

Para optar al grado Académico de Licenciado en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral: 85

Nota Literal: Ochenta y Cinco

Ha sido: .....

Director de la carrera de Informática Ph.D. José María Tapia Baltazar

Tutor: Lic. Juan Gonzalo Contreras Candia

Presidente: M. Sc. Morales Rios Menfy

Tribunal: Lic. Huanca Quisbert Carmen R.

Tribunal: Lic. Zeballos Daza Reynaldo J.



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

## **LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## ***Dedicatoria***

*A mi madre Angelica Lozano Cusicanqui y hermano Henry Aguilar Lozano quienes me brindaron todo su amor, cariño, y comprensión y quienes me inculcaron por el buen camino con sus consejos y recomendaciones que me dieron y por el apoyo constante que me brindaron en el largo camino del aprendizaje, nunca tendré con que agradecerles.*

*A mis hermanas Wendy, Ariana, por su paciencia, comprensión y apoyo en cada momento de la vida.*

erickaguilarlozano@gmail.com

## ***Agradecimientos***

*Mis agradecimientos a todas las personas que hicieron posible culminar el presente trabajo de manera satisfactoria.*

*A mi madre Angelica, mis hermanos Henry, Wendy, Ariana, por su amor, confianza y comprensión.*

*A mi tutor Lic. Juan Gonzalo Contreras Candia, quien me brindo su conocimiento y sabiduría para realizar esta tesis hasta su finalización, gracias por su paciencia para revisar una y otra vez este trabajo que es la culminación de varios años de estudio en esta apasionante carrera, gracias.*

*Al Lic. Freddy Miguel Toledo Paz, docente de taller 1 y 2, por brindarme sus conocimientos y consejos para lograr esta tesis.*

*A todos mis amigos por sus recomendaciones y ánimos, gracias por brindarme su amistad, los quiero mucho.*

*Gracias a todos.*

## **RESUMEN**

El Covid-19, además de generar un efecto devastador en las personas y los sistemas de salud, está afectando duramente la economía global, con graves consecuencias para el empleo y los servicios ofrecidos por empresas.

La presente tesis hace referencia al desarrollo de una plataforma multicanal en línea para la gestión y seguimiento de reclamos en base a tickets, con la cual se pretende ayudar a las empresas en telecomunicaciones a realizar una óptima y oportuna asistencia en cuanto a los reclamos de sus clientes sobre los servicios que éstas prestan.

Por parte del cliente, la atención y soporte de los reclamos en la actualidad se limita solo a una llamada telefónica, a un número gratuito, corporativo o línea común, que muchas veces por la excesiva afluencia de llamadas, este medio queda congestionado, lo que ocasiona el descontento y desconformidad del cliente, no quedando como opción asistir a la empresa de manera presencial a realizar el reclamo por razones ya mencionadas.

La plataforma ofrece más canales de atención para la apertura de tickets, entre los cuales tenemos: Telegram (Chat Bots), Gmail (correo electrónico), Central telefónica (atención por menú interactivo de voz u operadores), de esta manera se asegura que el reclamo del cliente sea aperturado y atendido posteriormente.

La plataforma posee un sistema que ayuda a realizar la gestión y seguimiento de los tickets, para el desarrollo del sistema se aplicó la metodología Scrum y complementando con la metodología de ingeniería Web U.W.E. basada en U.M.L. (Lenguaje Unificado de Modelado). El sistema se desarrolló con la estructura de una API (Application Programming Interfaces), es

decir que posee funciones y procedimientos que le permiten interactuar con los canales de atención mediante formatos de texto JSON (JavaScript Object Notation), lo cual permite una sencilla integración con los canales ya mencionados ya que estos poseen estructuras API's de igual manera con niveles de seguridad y autenticación de alto nivel como OAuth v1, v2 y JWT (Json Web Token).

Se utilizo el Framework Codeigniter3 para el desarrollo del sistema, también un gestor de base de datos en MySQL para el almacenamiento de la información.

También mencionar que se implementaron notificaciones PUSH, tecnologías correspondientes a Aplicaciones Web Progresivas.

La plataforma fue implementada en un Servidor Centos7 con la utilización de Asterisk para la implementación de la central telefónica.

Finalmente, la plataforma concluida e implementada ha cumplido con todos los objetivos propuestos en un principio, es decir, se ha logrado optimizar la gestión y seguimiento de tickets para empresas en telecomunicaciones.

## **ABSTRACT**

Covid-19, in addition to generating a devastating effect on people and health systems, is severely affecting the global economy, with serious consequences for employment and the services offered by companies.

This thesis refers to the development of an online multichannel platform for the management and follow-up of claims based on tickets, with which it is intended to help telecommunications companies to provide optimal and timely assistance regarding their claims. clients about the services they are providing.

On the part of the client, the attention and support of the claims at present is limited only to a telephone call, to a toll-free number, corporate or common line, which many times due to the excessive influx of calls, this medium is congested, which causes dissatisfaction and dissatisfaction of the client, not being an option to attend the company in person to make the claim for reasons already mentioned.

The platform offers more service channels for the opening of tickets, among which we have: Telegram (Telegram Bot), Gmail (email), Telephone exchange (attention by interactive voice menu or operators), in this way it is ensured that the customer's claim is opened and attended later.

The platform has a system that helps to carry out the management and monitoring of tickets, for the development of the system the Scrum methodology was applied and complementing with the U.W.E. Web engineering methodology. based on U.M.L. (Unified Modeling Language). The system was developed with the structure of an API (Application

Programming Interfaces), that is, it has functions and procedures that allow it to interact with service channels through JSON (JavaScript Object Notation) text formats, which allows easy integration with the channels already mentioned since they have API structures in the same way with levels of security and high-level authentication such as OAuth v1, v2 and JWT (Json Web Token).

The Codeigniter3 Framework was used for the development of the system, also a database manager in MySQL for the storage of information.

Also mention that PUSH notifications were implemented, technologies corresponding to Progressive Web Applications.

The platform was implemented on a Centos7 Server with the use of Asterisk for the implementation of the telephone exchange.

Finally, the completed and implemented platform has met all the objectives proposed at the beginning, that is, it has been possible to optimize the management and monitoring of tickets for telecommunications companies.

## INDICE

	P.
<b>CAPITULO I.....</b>	<b>1</b>
<b>MARCO INTRODUCTORIO .....</b>	<b>1</b>
<b>1.1. Introducción .....</b>	<b>1</b>
<b>1.2. Estado del Arte .....</b>	<b>3</b>
<b>1.3. Marco Teórico .....</b>	<b>7</b>
<b>1.4. Planteamiento del Problema .....</b>	<b>23</b>
<b>1.4.1. Problema General .....</b>	<b>24</b>
<b>1.4.2. Problemas Específicos .....</b>	<b>24</b>
<b>1.5. Objetivos .....</b>	<b>24</b>
<b>1.5.1. Objetivo General .....</b>	<b>24</b>
<b>1.5.2. Objetivos Específicos .....</b>	<b>24</b>
<b>1.6. Hipótesis .....</b>	<b>25</b>
<b>1.7. Justificación .....</b>	<b>25</b>
<b>1.7.1. Justificación Social .....</b>	<b>25</b>
<b>1.7.2. Justificación Técnica .....</b>	<b>26</b>
<b>1.7.3. Justificación Económica .....</b>	<b>27</b>
<b>1.8. Límites y Alcances.....</b>	<b>28</b>
<b>1.8.1. Delimitación Temática.....</b>	<b>28</b>
<b>1.8.2. Delimitación Espacial .....</b>	<b>29</b>
<b>1.8.3. Delimitación Temporal.....</b>	<b>29</b>
<b>1.9. Metodología .....</b>	<b>29</b>
<b>1.9.1. Metodología de Investigación.....</b>	<b>29</b>
<b>1.9.2. Ingeniería .....</b>	<b>30</b>
<b>1.10. Aporte.....</b>	<b>30</b>
<b>CAPITULO 2 .....</b>	<b>31</b>
<b>MARCO TEORICO .....</b>	<b>31</b>
<b>2.1. Introducción .....</b>	<b>31</b>
<b>2.1.1. La atención y soporte al cliente en Bolivia.....</b>	<b>31</b>
<b>2.2. Ingeniería de software .....</b>	<b>32</b>

<b>2.2.1 Metodología ágil de desarrollo de software</b> .....	33
<b>2.2.2 SCRUM</b> .....	34
2.2.2.1. Gestión de la evolución del Proyecto. ....	34
2.2.2.2. Roles. ....	35
2.2.2.3. Artefactos.....	38
2.2.2.4. Eventos. ....	40
<b>2.3. Desarrollo Web basada en U.M.L. (U.W.E.)</b> .....	<b>43</b>
<b>2.3.1. Lenguaje de Modelado de datos U.M.L.</b> .....	43
<b>2.3.2. Ingeniería Web basada en U.M.L. (U.W.E.)</b> .....	48
2.3.2.1. Fases de Desarrollo.....	48
<b>2.4. Patrón de diseño</b> .....	<b>50</b>
<b>2.4.1. Modelo vista controlador MVC</b> .....	51
<b>2.5. Tecnologías</b> .....	<b>52</b>
<b>2.5.1. Tecnologías Web</b> .....	52
2.5.1.1. PHP.....	52
2.5.1.2. Codeigniter 3 Framework.....	53
2.5.1.3. JavaScript.....	55
2.5.1.4. JQuery.....	56
2.5.1.5. Google APIs. ....	57
2.5.1.6. Aplicación Web Progresiva (PWA). ....	59
<b>2.5.2. Software, Protocolos y aplicaciones de comunicación</b> .....	60
2.5.2.1. OpenSIPS.....	60
2.5.2.2. Asterisk PBX. ....	62
2.5.2.3. Protocolo SIP.....	65
2.5.2.4. Telegram Bots.....	66
<b>2.6. Gestor de Base de datos</b> .....	<b>69</b>
<b>2.6.1. MySQL</b> .....	69
<b>2.7. Calidad de software</b> .....	<b>70</b>
<b>2.7.1. Norma ISO 9126</b> .....	70
2.7.1.1 Funcionalidad. ....	71
2.7.1.2. Confiabilidad. ....	72
2.7.1.3. Usabilidad.....	72
2.7.1.4. Eficiencia.....	73

2.7.1.5. Mantenibilidad.....	73
2.7.1.6. Portabilidad.....	74
<b>2.8. Seguridad .....</b>	<b>74</b>
<b>2.8.1. Firewall Linux (Shorewall).....</b>	<b>74</b>
<b>2.8.2. Características funcionales de Shorewall.....</b>	<b>75</b>
<b>2.8.3. ¿Cuándo conviene usar Shorewall?.....</b>	<b>75</b>
<b>CAPITULO 3 .....</b>	<b>77</b>
<b>MARCO APLICATIVO .....</b>	<b>77</b>
<b>3.1. Introducción .....</b>	<b>77</b>
<b>3.2. Identificación de roles Scrum .....</b>	<b>77</b>
<b>3.3. Análisis de Requerimientos.....</b>	<b>78</b>
<b>3.3.1. Requisitos Globales.....</b>	<b>79</b>
<b>3.3.2. Requisitos Elementales .....</b>	<b>79</b>
<b>3.3.3. Requisitos Funcionales .....</b>	<b>82</b>
<b>3.3.4. Requisitos de Navegación.....</b>	<b>84</b>
<b>3.3.5. Modelado de casos de uso.....</b>	<b>85</b>
<b>3.4. Sprint 0 – Preparación del proyecto.....</b>	<b>99</b>
<b>3.4.1. Modelo Entidad-Relación.....</b>	<b>101</b>
<b>3.4.2. Modelo Relacional.....</b>	<b>101</b>
<b>3.4.3. Modelo Conceptual.....</b>	<b>102</b>
<b>3.5. Desarrollo de los Sprints .....</b>	<b>104</b>
<b>3.5.1. Sprint 1.....</b>	<b>104</b>
3.5.1.1. Integración entre el sistema y los canales de atención .....	105
3.5.1.2. Modelo de Navegación.....	108
3.5.1.3. Modelo de Presentación.....	111
<b>3.5.2. Sprint 2.....</b>	<b>123</b>
3.5.2.1. Modelo de Navegación.....	124
3.5.2.2. Modelo de Presentación.....	125
<b>3.5.3. Sprint 3.....</b>	<b>132</b>
3.5.3.1. Modelo de Navegación.....	133
3.5.3.2. Modelo de Presentación.....	134

<b>CAPITULO 4 .....</b>	<b>143</b>
<b>DEMOSTRACION DE LA HIPOTESIS .....</b>	<b>143</b>
<b>4.1. Introducción .....</b>	<b>143</b>
<b>4.2. Comprobación de hipótesis .....</b>	<b>143</b>
<b>4.3. Planteamiento de la hipótesis nula y alternativa.....</b>	<b>145</b>
<b>4.4. Interpretación de resultados .....</b>	<b>149</b>
<b>CAPITULO 5 .....</b>	<b>150</b>
<b>CALIDAD Y SEGURIDAD .....</b>	<b>150</b>
<b>5.1. Calidad .....</b>	<b>150</b>
<b>5.1.1. Funcionalidad .....</b>	<b>150</b>
<b>5.1.2. Confiabilidad .....</b>	<b>155</b>
<b>5.1.3. Usabilidad .....</b>	<b>157</b>
<b>5.1.4. Mantenibilidad .....</b>	<b>157</b>
<b>5.1.5. Portabilidad .....</b>	<b>158</b>
<b>5.1.6. Resultados .....</b>	<b>159</b>
<b>5.2. Seguridad .....</b>	<b>160</b>
<b>5.2.1. Autenticación de Usuarios.....</b>	<b>160</b>
<b>5.2.2. Firewall Linux (Shorewall) .....</b>	<b>161</b>
<b>CAPITULO 6 .....</b>	<b>164</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>164</b>
<b>6.1. Conclusiones .....</b>	<b>164</b>
<b>6.2. Recomendaciones .....</b>	<b>165</b>
<b>BIBLIOGRAFIA.....</b>	<b>166</b>
<b>ANEXOS .....</b>	<b>169</b>
<b>ANEXO A: ÁRBOL DE PROBLEMAS.....</b>	<b>170</b>

## INDICE DE FIGURAS

Figura 1. Esquema de trabajo de un Servidor.....	8
Figura 2. Esquema de trabajo de un Servidor SIP Proxy.....	17
Figura 3. Comparación de Metodologías de Desarrollo de Software .....	33
Figura 4. Descripción de los Roles Scrum (Webipedia Scrum).....	38
Figura 5. Descripción resumida de los Eventos Scrum .....	43
Figura 6. Interrelación entre los elementos del patrón MCV .....	52
Figura 7. Diagrama del flujo de una aplicación en CodeIgniter .....	55
Figura 8. Contenido de la arquitectura del núcleo de OpenSIPS .....	62
Figura 9. Esquema general de los componentes de Elastix .....	65
Figura 10. Creación de un bot con BotFather .....	69
Figura 11. Características ISO 9126 y aspectos que atiende cada una .....	71
Figura 12. Arquitectura de la Plataforma .....	78
Figura 13. Diagrama de Caso de Uso de Alto Nivel del Sistema .....	86
Figura 14. Caso de Uso "Administrar Tickets".....	87
Figura 15. Caso de Uso "Administrar Usuarios".....	88
Figura 16. Caso de Uso "Administrar Técnicos".....	89
Figura 17. Caso de Uso "Administrar Canales".....	90
Figura 18. Caso de Uso "Administrar Fallas".....	91
Figura 19. Caso de Uso "Gestionar Técnicos a Tickets".....	92
Figura 20. Caso de Uso "Administrar Clientes".....	93
Figura 21. Caso de Uso "Administrar Permisos".....	94
Figura 22. Caso de Uso "Administrar Canales Gmail".....	95
Figura 23. Caso de Uso "Administrar Canales Telegram".....	96
Figura 24. Caso de Uso "Administrar Canales Central Telefónica".....	97
Figura 25. Caso de Uso "Administrar Calendarios Google".....	98
Figura 26. Caso de Uso "Administrar Empresas".....	99
Figura 27. Modelo Entidad Relación de la Base de Datos .....	101
Figura 28. Modelo Relacional de la Base de Datos del Sistema .....	102
Figura 29. Modelo Conceptual en Diagrama de Clases para el Sistema .....	103
Figura 30. Comunicación entre el sistema y telegram .....	106

Figura 31. Comunicación entre el sistema y Gmail .....	107
Figura 32. Comunicación entre el sistema y la central telefónica en Asterisk .....	108
Figura 33. Modelo de Navegación para el Sprint 1.....	109
Figura 34. Modelo de Navegación para el Sprint 1 (apertura de ticket vía Telegram).....	109
Figura 35. Modelo de Navegación para el Sprint 1 (apertura de ticket vía Correo Electrónico) .....	110
Figura 36. Modelo de Navegación para el Sprint 1 (apertura de ticket vía Central Telefónica e IVR).....	110
Figura 37. Modelo de Presentación de Tickets .....	111
Figura 38. Modelo de Presentación de Usuarios .....	112
Figura 39. Modelo de Presentación de Técnicos .....	112
Figura 40. Modelo de Presentación de Canales .....	113
Figura 41. Pantalla de listado de tickets registrados .....	114
Figura 42. Pantalla del formulario de registro de tickets .....	114
Figura 43. Pantalla de apertura y seguimiento de tickets vía telegram.....	115
Figura 44. Envío de mensaje al correo de la empresa .....	116
Figura 45. Selección y respuesta al correo con la falla .....	117
Figura 46. Pantalla de apertura exitosa de tickets vía Gmail .....	117
Figura 47. Registro de apertura de tickets vía Central Telefónica .....	118
Figura 48. Registro de seguimiento de tickets vía Central Telefónica .....	119
Figura 49. Registro de comunicación con un operador vía Central Telefónica .....	120
Figura 50. Pantalla de listado de usuarios registrados .....	121
Figura 51. Pantalla del formulario de registro de usuarios .....	122
Figura 52. Pantalla de listado de técnicos registrados .....	122
Figura 53. Pantalla del formulario de registro de técnicos .....	123
Figura 54. Pantalla de listado de canales registrados .....	123
Figura 55. Pantalla del formulario de registro de canales .....	125
Figura 56. Modelo de Navegación para el Sprint 2.....	126
Figura 57. Modelo de Presentación de Fallas .....	126
Figura 58. Modelo de Presentación de Agendar Técnicos a Tickets .....	127
Figura 59. Modelo de Presentación de Clientes .....	127
Figura 60. Modelo de Presentación de Permisos .....	128

Figura 61. Pantalla de listado de fallas registradas .....	128
Figura 62. Pantalla del formulario de registro de fallas .....	129
Figura 63. Pantalla de listado de tickets con la opción de agendar técnicos .....	129
Figura 64. Pantalla de listado de clientes registrados .....	130
Figura 65. Pantalla del formulario de registro de clientes .....	131
Figura 66. Pantalla de listado de permisos registrados .....	131
Figura 67. Pantalla del formulario de registro de permisos .....	134
Figura 68. Modelo de Navegación para el Sprint 3.....	135
Figura 69. Modelo de Presentación de Gmail .....	135
Figura 70. Modelo de Presentación de Telegram .....	136
Figura 71. Modelo de Presentación de Centrales Telefónicas .....	137
Figura 72. Modelo de Presentación de Calendarios Google .....	137
Figura 73. Modelo de Presentación de Empresas .....	138
Figura 74. Pantalla de listado de cuentas Gmail registradas .....	138
Figura 75. Pantalla del formulario de registro de cuentas Gmail .....	139
Figura 76. Pantalla de listado de bots Telegram registrados .....	139
Figura 77. Pantalla del formulario de registro de Bots Telegram .....	140
Figura 78. Pantalla de listado de Centrales Telefónicas registradas .....	140
Figura 79. Pantalla del formulario de registro de Centrales Telefónicas .....	141
Figura 80. Pantalla de listado de Calendarios Google registrados .....	141
Figura 81. Pantalla del formulario de registro de Calendarios .....	142
Figura 82. Pantalla de listado de Empresas registradas .....	142
Figura 83. Pantalla del formulario de registro de Empresas .....	146
Figura 84. Cálculo en línea de la desviación estándar .....	146
Figura 85. Cálculo de la t Student [Aplicación Statistics Calculator].....	148
Figura 86. Grafica del resultado de t Student .....	149
Figura 87. Login del sistema .....	160
Figura 88. Configuración del fichero zone .....	162
Figura 89. Configuración del fichero interfaces .....	162
Figura 90. Configuración del fichero policy .....	163
Figura 91. Configuración del fichero masq .....	163

## INDICE DE TABLAS

Tabla 1. Precios tentativos para la implementación de la plataforma.....	27
Tabla 2. Precios tentativos de abono mensual por empresa.....	28
Tabla 3. Directorios importantes de Asterisk.....	64
Tabla 4. Identificación de Roles Scrum.....	77
Tabla 5. Descripción del Caso de Uso "Administrar Tickets".....	87
Tabla 6. Descripción del Caso de Uso "Administrar Usuarios".....	88
Tabla 7. Descripción del Caso de Uso "Administrar Técnicos".....	89
Tabla 8. Descripción del Caso de Uso "Administrar Canales".....	90
Tabla 9. Descripción del Caso de Uso "Administrar Fallas".....	91
Tabla 10. Descripción del Caso de Uso " Gestionar Técnicos a Tickets ".....	92
Tabla 11. Descripción del Caso de Uso "Administrar Clientes".....	93
Tabla 12. Descripción del Caso de Uso "Administrar Permisos".....	94
Tabla 13. Descripción del Caso de Uso "Administrar Canales Gmail ".....	95
Tabla 14. Descripción del Caso de Uso "Administrar Canales Telegram".....	96
Tabla 15. Descripción del Caso de Uso "Administrar Canales Central Telefónica".....	97
Tabla 16. Descripción del Caso de Uso "Administrar Calendarios Google ".....	98
Tabla 17. Descripción del Caso de Uso "Administrar Empresas ".....	99
Tabla 18. Pila del producto.....	100
Tabla 19. Pila del Sprint 1.....	104
Tabla 20. Pila del Sprint 2.....	124
Tabla 21. Pila del Sprint 3.....	132
Tabla 22. Valores de ajuste.....	143
Tabla 23. Factores de ponderación total.....	144
Tabla 24. Lista de entradas de usuario.....	151
Tabla 25. Lista de salidas de usuario.....	151
Tabla 26. Lista de peticiones de usuario.....	152
Tabla 27. Lista de todos los archivos lógicos.....	153
Tabla 28. Factores de ponderación .....	153
Tabla 29. Valores de ajuste de complejidad.....	154

Tabla 30. Preguntas para obtener el grado de usabilidad.....	157
Tabla 31. Factores de portabilidad y sus porcentajes.....	159
Tabla 32. Resultados de la calidad .....	160

# CAPITULO I

## MARCO INTRODUCTORIO

### 1.1.Introducción

En la actualidad la atención adecuada y seguimiento de los clientes de una empresa que brinda servicios en telecomunicaciones, es indispensable para la conservación y permanencia de los mismos, esto se torna aún mucho más complicado ahora que la humanidad atraviesa la pandemia del COVID-19 desde el 1 de diciembre de 2019, esto obliga a que las empresas mejoren sus servicios de atención, soporte y seguimiento a los reclamos de sus clientes de manera remota, eficiente y eficaz.

Muchas empresas no están preparadas para brindar soporte remoto, pues el problema radica en el tiempo y costo que se requiere para implementar una solución que realice el control y seguimiento de sus clientes sobre problemas que surjan de la obtención de sus servicios.

Del mismo modo muchas empresas no cuentan con una solución propia de telecomunicaciones y central telefónica puesto que los costos de implementación son muy altos en la actualidad para una empresa pequeña o que recién inicia, solo grandes entidades tienen estos servicios con infraestructura propia.

Por otro lado, muchas empresas aún realizan el soporte y seguimiento a sus clientes de manera manual y de poco beneficio, por ejemplo, mediante, hojas de rutas lo cual trae inconvenientes en el tema de logística, llamadas telefónicas que generan costos adicionales y en muchos casos obligando al agente o técnico a asistir a la fuente laboral.

Hoy en día existen plataformas a nivel internacional que brindan estos servicios en telecomunicación y soporte para empresas en la nube con un costo mensual, entre estas podemos mencionar algunas como: GotoConnect, AirCall, CouldTalk, Five9, ChaseData, Anura, etc. Los precios para contratar ese tipo de servicios también llegan a ser un inconveniente, y una opción que muchas empresas descartan de inicio.

Por tal motivo, el propósito de esta tesis aplicada es la de desarrollar una plataforma en la nube que brinde una solución que proporcione distintos canales de atención para la gestión, control y seguimiento de reclamos de los clientes para cualquier empresa que brinde servicios en telecomunicaciones, servicios relacionados o algún tipo de servicio que tenga gran cantidad de abonados o clientes en nivel departamental o nacional, ayudando en incrementar la satisfacción de los mismos, logrando la permanencia de los clientes y aumentando la rentabilidad de la empresa.

Christian Grönroos (1994) indica en su obra y aborda los problemas económicos de la relación con el cliente, más específicamente el vínculo entre la calidad del servicio y la rentabilidad desde una perspectiva de gestión y marketing relacional. En esta perspectiva, la tarea del marketing no es solo establecer relaciones con los clientes, sino también mantenerlas y mejorarlas para mejorar la rentabilidad del cliente. En la literatura sobre calidad del servicio se asume que una mayor calidad conduce a la satisfacción del cliente, lo que conduce a la lealtad del cliente y esto impulsa la rentabilidad del cliente. El marco destaca factores que, además de la calidad del servicio y la satisfacción del cliente, influyen en los vínculos entre la calidad del servicio y la rentabilidad. También analiza aspectos de la mejora de la rentabilidad de las relaciones, cómo mejorar los ingresos de las relaciones a través de mayores grados de

concentración del patrocinio y reducir el costo de las relaciones al cambiar la configuración del episodio de las relaciones con los clientes.

La metodología de investigación que se utilizará de acuerdo al objeto de estudio y tipo de datos será cuantitativa aplicada con un enfoque inductivo, en cuanto a la metodología de diseño se utilizará Scrum para el diseño del sistema de tickets el cual se integrará con una central de telefonía IP en la nube.

## **1.2.Estado del Arte**

Las plataformas a nivel internacional que brindan este tipo de servicios o aplicaciones similares utilizan tecnologías en su implementación como:

- Telefonía IP VoIP, Gateway proxy SIP (OpenSips, kamailio)
- Asterisk (PBX)
- CRM y Helpdesk
- Framework (PHP, Java, .NET, etc.)
- Bases de Datos (MySQL, PostgreSQL, etc.)
- PWA (Progressive Web Applications) (Push Notifications)
- Firewall de Seguridad (Shorewall / Iptables / Fail2ban)
- SO (Windows Server - Linux Server)
- Bootstrap, HTML5, CSS5
- Ajax, JQuery, JavaScript
- Arquitecturas SOAP, REST
- Web Services, Rest-Services

- Bots (Telegram, WhatsApp, Facebook, etc.)

Algunas empresas que brindan estas soluciones son:

**GoToconnect** brinda servicios de comunicaciones unificadas de nivel empresarial y soluciones de VoIP para empresas e instituciones a través de la nube. Se utilizan comunicaciones Jive y todos los planes estándar de Jive incluyen tono de marcación local, correo de voz y larga distancia con todas las funciones avanzadas de PBX. Incluye varias funciones principales de VoIP, como PBX alojado, informes/análisis de llamadas, modo día-noche, conferencia de 3 vías, música en espera, intercomunicador, marcación por nombre, llamada en espera, desvío de llamadas, fax virtual, saludos personalizados, acceso remoto, fax a correo electrónico y muchos más. Cualquier cambio en el sistema Jive se puede hacer en línea en tiempo real, lo que ayuda a las empresas a ahorrar en costos de soporte y mantenimiento.

**Aircall** es un sistema de teléfono basado en la nube elegido por las marcas modernas. Mediante una integración perfecta con las herramientas más populares de CRM y Helpdesk, contribuye a que los equipos de ventas y asistencia se comuniquen de manera clara y eficiente. Los administradores pueden agregar números instantáneamente de más de 100 países, ampliar y reducir sus equipos según la estacionalidad y obtener información detallada a través de las analíticas en tiempo real. Debido a la capacidad de acceder por medio de una aplicación móvil y de escritorio, más de 3000 compañías en todo el mundo confían en Aircall.

Esta plataforma te permite realizar y atender llamadas en cualquier lugar, obtener números de teléfono instantáneamente en más de 100 países y manejar las llamadas sobre la marcha con las aplicaciones móviles y de escritorio de Aircall. Encamina llamadas de manera automática y eficiente de acuerdo con la selección de IVR, habilidades del agente, zona horaria y otros detalles, incluido un panel intuitivo. Realiza un seguimiento del rendimiento y recibe

analíticas avanzadas sobre la productividad de los agentes y del equipo. Supervisa la actividad del equipo en tiempo real en la transmisión en vivo y los datos de referencia cruzada con un CRM y un servicio de asistencia existentes para lograr una mejor comprensión de los procesos.

**CloudTalk** que es un sistema de telefonía empresarial, inteligente e intuitivo, para empresas de todos los tamaños que hacen negocios por internet y que buscan mejorar la productividad del equipo y las experiencias de los clientes.

Accesible a través de aplicaciones de navegador y de escritorio, CloudTalk permite que los equipos de contacto y centro de atención telefónica hagan y reciban llamadas en cualquier lugar, en cualquier dispositivo, con números internacionales de más de 80 países de todo el mundo. El software procesa las llamadas entrantes con un menú de respuesta de voz interactiva (IVR) y enrutamiento de llamadas y puede utilizarse para interconectar departamentos, sucursales y empleados, automatizar el llamado a clientes potenciales para los equipos de ventas y mejorar la experiencia del cliente.

CloudTalk se integra con soluciones populares de CRM, servicio de ayuda y comercio electrónico (incluidas Zendesk, Live Agent, Zoho CRM y Magento) con integraciones predefinidas de un solo clic y puede conectarse a cualquier otra herramienta a través de una API abierta.

**Five9** es una solución de centro de contacto y centro de atención telefónica impulsada por inteligencia artificial. La solución basada en la nube proporciona herramientas omnicanal, que incluyen administración de llamadas entrantes y salientes, marcador predictivo, marcador progresivo, chat, atención al cliente social, ACD (distribución automática de llamadas, por sus siglas en inglés) y más.

Five9 es una solución de centro de contacto en la nube que lleva el poder de la IA práctica a miles de centros de atención telefónica de todo el mundo. Desde 2001, Five9 ayuda a los clientes a crear excelentes experiencias omnicanal para sus clientes. Triplica el tiempo de conversación de los agentes con los marcadores inteligentes que omiten automáticamente las llamadas sin respuesta y las señales ocupadas para garantizar que la próxima oportunidad esté disponible cuando el siguiente agente esté listo. Utiliza el enrutamiento inteligente para derivar las llamadas a los agentes correctos y obtener los datos de intención del cliente para predecir cuáles son los mejores pasos que deben dar los agentes. Ofrece opciones intuitivas de autoservicio con la función IVR (respuesta de voz interactiva, por sus siglas en inglés) de Five9.

Five9 lleva la IA práctica a los centros de contacto para aumentar la eficacia y la eficiencia de los agentes y crear excelentes experiencias para los clientes, siempre y en todos los canales: teléfono, chat, móvil, correo electrónico y redes sociales, entre otros.

### **Plataformas a Nivel Nacional**

En el contexto actual en Bolivia solo las empresas en el área de telecomunicaciones como Entel, NuevaTel, Viva, Tigo, AXS Bolivia, Cotel u otras empresas de servicios de gran alcance contarían con una plataforma o sistema con similares características.

Por citar el manejo de los clientes de la empresa Entel que realiza la gestión de reclamos mediante su CALL CENTER que trabaja con un menú telefónico para cada servicio que ofrecen, pues en caso de encontrarse una línea ocupada o colapsada no existe otro canal o medio por el cual realizar un reclamo, solo queda asistir a sus oficinas presenciales.

### **1.3.Marco Teórico**

Es importante que consideremos el camino más adecuado para lograr alcanzar los objetivos planteados, recopilando toda la información para el sustento del estudio que se realiza, estos elementos son la base para la descripción y explicación del problema a solucionar.

#### **Definición de Internet**

“International Network of Computer” o “Red Internacional de Computadoras”, mejor conocido como Internet; a su vez esta red mundial interconecta otras redes de mayor magnitud por lo que se puede referir a ella como la red de redes.

El internet ofrece muchas otras formas de comunicación, por ejemplo, el correo electrónico, Chat, etc. Además, proporciona muchas formas valiosas de intercambiar información, correo electrónico, investigaciones, documentos, etc.

#### **Navegadores**

Un navegador web (en inglés, web browser) es un software, aplicación o programa que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser vistos.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Además, permite visitar páginas web y hacer actividades en ella, es decir, enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades.

Los navegadores más conocidos en la actualidad son Chrome, Safari, Firefox, Edge, Explorer, etc. Casi la mayoría de estos navegadores tienen funciones multiplataforma.

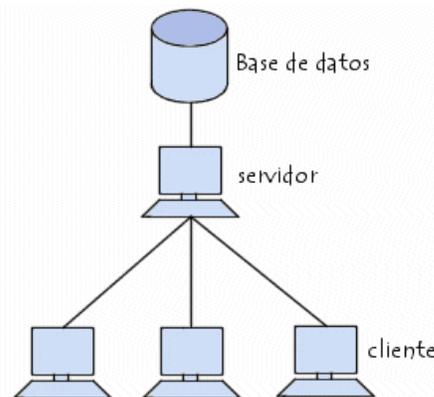
#### **Servidor Web**

Un servidor web es un equipo informático que tiene un programa ejecutándose en un sitio web para responder a la solicitud de los navegadores. Los servidores Web pueden ejecutar otros programas residentes tomando información que se provee junto a las solicitudes, esto permite el acceso a las bases de datos.

El software que hoy en día se utiliza es: Apache, Nginx, LiteSpeed, Microsoft IIS, Lighttpd, Caddy, Tomcat, Cherokee, etc.

### **Servidor de Base de Datos**

Un servidor de base de datos es un programa que provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor. También puede hacer referencia a aquellas computadoras (servidores) dedicadas a ejecutar esos programas, prestando el servicio.



*Figura 1.* Esquema de trabajo de un Servidor.

### **Servidor Apache**

Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

Les permite a los propietarios de sitios web servir contenido en la web, de ahí el nombre de «servidor web». Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Cuando alguien quiere visitar un sitio web, ingresa un nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual. Aunque llamamos a Apache un servidor web, no es un servidor físico, sino un software que se ejecuta en un servidor. Su trabajo es establecer una conexión entre un servidor y los navegadores de los visitantes del sitio web (Firefox, Google Chrome, Safari, etc.) mientras envían archivos entre ellos (estructura cliente-servidor). Apache es un software multiplataforma, por lo cual funciona tanto en servidores Unix como en Windows. Cuando un visitante quiere cargar una página de tu sitio web, su navegador le envía una solicitud a tu servidor y Apache le devuelve una respuesta con todos los archivos solicitados (texto, imágenes, etc.) El servidor y el cliente se comunican a través del protocolo HTTP y Apache es responsable de garantizar una comunicación fluida y segura entre las dos máquinas.

Apache es altamente personalizable, ya que tiene una estructura basada en módulos. Los módulos les permiten a los administradores del servidor activar y desactivar funcionalidades adicionales. Apache tiene módulos de seguridad, almacenamiento en caché, reescritura de URL, autenticación de contraseña y más.

### **Sistema Gestor de Base de Datos**

Los sistemas gestores de bases de datos son un tipo de software muy específico dedicado a servir de interfaz entre las bases de datos y las aplicaciones que los utilizan asegurando su integridad, confidencialidad y seguridad. Se componen de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y un lenguaje de consulta.

Permite definir las bases de datos especificando los tipos, estructuras y restricciones de datos, Además de manipular la base de datos realizando consultas, actualizaciones y generando informes.

En esta tesis aplicada se utilizará el Sistema Gestor de Base de Datos MySQL.

## **Modelo UML**

Unified Modeling Language o UML es un lenguaje orientado a objetos mediante el cual se puede modelar, construir y además documentar los elementos que forman un sistema de software.

UML fue creado por Grady Booch, Ivar Jacobson y Jim Rumbaugh, creadores de los métodos: Booch, OMT y OOSE, respectivamente.

Este lenguaje posibilita el intercambio de los modelos entre las distintas herramientas CASE Orientados a Objetos.

## **Modelado Orientado a Objetos**

El modelo orientado a objetos utiliza el paradigma de la orientación a objetos para el desarrollo de software. Este enfoque realiza la construcción de modelos de un sistema por medio de la identificación y la especificación de un conjunto de objetos relacionados, que colaboran entre sí de acuerdo a los requerimientos establecidos para el sistema de objetos.

La definición del modelado orientado a objetos puede claramente dividir el enfoque en tres dimensiones:

- La dimensión estructural.
- La dimensión dinámica.
- La dimensión funcional.

Este tipo de modelado implica la realización de las siguientes actividades:

1. Identificar las clases, modelos y objetos. (objetos y atributos).
2. Asociar estáticamente los objetos. (relaciones dependientes del dominio del problema).
3. Especificación del comportamiento de los objetos. (Definir cómo se comportarán los objetos).
4. Definir la jerarquía de herencia de las clases. (Definir la jerarquía de clases, para que el sistema quede lo más abstracto posible).

Características de los modelos Orientados a Objetos.

- EL modelado Orientado a Objetos está basado en el paradigma orientado a objetos.
- Trata el almacenamiento de objetos (persistencia de los objetos).
- Define un lenguaje para la definición y manipulación de objetos.
- Incluye mecanismos para optimizar el acceso (Indexación y Clustering), el control de la concurrencia, seguridad y gestión de usuarios, facilidad de consulta y recuperación ante fallos.
- Debido a que es un esquema orientado a objetos incluye: Encapsulamiento, herencia, polimorfismo, etc.

### **Lenguajes Orientados a Objetos**

Para considerar un lenguaje de programación como orientado a objetos se deben cumplir varias condiciones. Debe soportar objetos, que deben tener un tipo o clase asociado y que estas clases pueden heredar atributos de una clase superior o superclase.

Actualmente existen una gran variedad de lenguajes de programación orientados a objetos entre los que podemos destacar: C++, Objective C (es el lenguaje principal de

programación para Mac OS X, iOS), Java, Ruby, Visual Basic, Visual C Sharp, Perl, Smalltalk, PHP o Python.

C++ junto con Java, son los dos lenguajes de programación orientada a objetos más utilizados actualmente (según datos proporcionados por el índice de programación Tiobe). Por lo tanto, cuando se quiere acceder a un curso de programación orientada a objetos, siempre hay que tener ambos lenguajes en cuenta, sobre todo para ampliar las opciones de encontrar ofertas de empleo en el mercado laboral.

Python, PHP y Ruby son tipos de lenguajes de programación orientada a objeto muy populares y enfocados a la programación y desarrollo web (este tipo de programación, así como las aplicaciones para móviles son ejemplos de la programación orientada a objetos).

Existen otros lenguajes de programación más específicos como ADA, centrado en la seguridad y utilizado para aplicaciones de defensa. líneas aéreas, etc.

### **Desarrollo (Front End / Back End)**

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Un programador Frontend debe saber de códigos (HTML, CSS y JavaScript) para poder usar algunos frameworks o librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios. React, Redux, Angular, Bootstrap, Foundation, LESS, Sass, Stylus y PostCSS son algunos de ellos.

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja

dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

Algunos de los lenguajes de programación de Backend son Python, PHP, Ruby, C# y Java, y así como en Frontend, cada uno de los anteriores tiene diferentes frameworks que te permiten trabajar mejor según el proyecto que estás desarrollando. Se tiene a Django, Laravel, Ruby On Rails y ASP.Net, estos tienen una gran comunidad que los respalda.

## **Bootstrap**

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

## **Frameworks en el Desarrollo Web**

Un framework es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas.

Los framework hacen que el desarrollador no esté continuamente “reinventado la rueda” y se centre en el problema que quiere resolver y no en la implementación de funcionalidades que normalmente son de uso común y que ya están resueltas por otros.

Ventajas:

- El código está organizado desde el primer momento.
- Proporcionan soluciones a la mayoría de problemáticas comunes del desarrollo de aplicaciones web.

- Arquitectura MVC (Model - View - Controller).
- No tienes que preocuparte por la seguridad, el framework te ayuda.
- Respaldo y protección de la comunidad.
- Perfectos para fomentar el trabajo en equipo.

Frameworks utilizados en la actualidad:

- Symfony 4
- Laravel
- Codeigniter
- CakePHP

## **Desarrollo de Software**

El Proceso para el desarrollo de software, también denominado ciclo de vida del desarrollo de software es una estructura aplicada al desarrollo de un producto de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso. Algunos autores consideran un modelo de ciclo de vida un término más general que un determinado proceso para el desarrollo de software.

Fases del proceso de desarrollo de software

- Análisis de requisitos. Extraer los requisitos de un producto de software es la primera etapa para crearlo.
- Diseño y arquitectura.
- Programación.
- Pruebas.
- Documentación.

- Mantenimiento.
- Modelo en cascada o convencional.
- Modelo evolutivo.

#### Enfoques de desarrollo de software

- Modelo en cascada: Framework lineal.
- Prototipado: Framework iterativo.
- Incremental: Combinación de framework lineal e iterativo.
- Espiral: Combinación de framework lineal e iterativo.
- RAD: Rapid Application Development, framework iterativo.

Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

#### **Metodología de Desarrollo Ágil Scrum**

Es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental. Se orienta a la auto-organización de equipos y colaboración entre sus miembros para entregar un incremento del producto funcional con un grado de prioridad cada cierto periodo de tiempo y bajo unos roles que tienen que asumir las personas que conforman el equipo desarrollo.

#### **Aplicación Web Progresiva (PWA)**

Una aplicación web progresiva (PWA por sus siglas en inglés) es un tipo de software de aplicación que se entrega a través de la web, creado utilizando tecnologías web comunes como HTML, CSS y JavaScript. Está destinado a funcionar en cualquier plataforma que use un navegador compatible con los estándares. La funcionalidad incluye trabajar sin conexión,

notificaciones push y acceso al hardware del dispositivo, lo que permite crear experiencias de usuario similares a las aplicaciones nativas en dispositivos móviles y de escritorio. Dado que una aplicación web progresiva es un tipo de página web o sitio web conocido como aplicación web, no hay ningún requisito para que los desarrolladores o usuarios instalen las aplicaciones web a través de sistemas de distribución digital como Apple App Store o Google Play.

Los PWA no requieren agrupación o distribución por separado. La publicación de una aplicación web progresiva es como sería para cualquier otra página web. Los PWA funcionan en cualquier navegador, pero las características "similares a las aplicaciones", como ser independientes de la conectividad, instalarse en la pantalla de inicio y enviar mensajes, dependen de la compatibilidad del mismo. A partir de abril de 2018, esas características son compatibles en diversos grados con los navegadores Microsoft Edge, Google Chrome, Mozilla Firefox y Apple Safari, pero es posible que más navegadores admitan las funciones necesarias en el futuro. Varios negocios destacan mejoras significativas en una amplia variedad de indicadores clave de rendimiento después de la implementación de PWA, como un mayor tiempo dedicado a la página, conversiones o ingresos.

En esta tesis se implementará las notificaciones push en la plataforma.

### **Servidor SIP Proxy para Telefonía IP**

Un servidor SIP es el principal componente de una central IP o PBX IP, y principalmente se encarga de la administración de todas las llamadas SIP en la red. Un servidor SIP es también conocido como SIP Proxy o un Registrador.



*Figura 2.* Esquema de trabajo de un Servidor SIP Proxy.

A pesar de que el Servidor SIP puede ser considerado como la parte más importante del sistema telefónico PBX IP basado en SIP, sólo se ocupa de administrar las sesiones; más específicamente, un Servidor SIP puede:

- Establecer una sesión entre dos (o más) endpoints (un audio conferencia puede involucrar más endpoints).
- Negociar los parámetros media y especificaciones para la sesión para cada endpoint utilizando el protocolo SDP.
- Ajusta los parámetros media y especificaciones de una sesión DURANTE la sesión (poner una llamada en espera).
- Sustituir un endpoint con un nuevo endpoint (transferencia de llamada).
- Terminar una sesión.

El servidor SIP de hecho no transmite o recibe ninguna media, esto se hace a través del media server usando el protocolo RTP. Dentro del contexto de un ecosistema PBX IP, casi siempre es verdad que el servidor SIP y su acompañante, el servidor media, residen en una misma máquina. Sin embargo, que los servidores SIP de muy alto volumen (tales como un

Proveedor VoIP, por ejemplo), pueden separar el servidor media en una máquina diferente para administrar mejor la carga de trabajo, y posiblemente poder, además, distribuir la carga en múltiples servidores media.

Actualmente existen dos softwares SIP: Kamailio y OpenSIPS.

### **Sistemas de Telefonía VoIP Basados en OpenSIPS**

OpenSIPS es un maduro servidor SIP de código abierto (Open Source), que por su versatilidad puede jugar el papel de cualquiera de los elementos que conforman una red de servicios de voz sobre IP basada en SIP.

OpenSIPS puede utilizarse como:

- Servidor SIP de Registro
- Enrutador o Servidor Proxy de SIP
- Servidor SIP de Redireccionamiento
- Agente de Presencia SIP
- Servidor de IM (Instant Messaging) SIP
- Gateway SIP-SMS y SMS-SIP
- Gateway SIP-XMPP y XMPP-SIP para Presencia e IM
- Distribuidor de Cargas SIP
- Unidad para NAT Traversal de SIP
- Servidor de Aplicaciones SIP

OpenSIPS es uno de los servidores SIP más rápidos que existen en el mercado, con un rendimiento y robustez que lo convierten en una elección para aplicaciones a nivel de proveedores de servicio.

Nuestras implementaciones de OpenSIPS en conjunto con otras aplicaciones, también de código abierto, permiten contar con soluciones VoIP de alta disponibilidad. Esto se logra con diseños en los que las cargas se distribuyen en nodos y en cada uno de ellos se cuenta con servidores maestros respaldados por servidores de backup en línea (hot backup). La distribución de las cargas en nodos permite desarrollar redes de servicios altamente escalables.

Estos son algunos de los escenarios en los que se puede utilizar OpenSIPS:

- Proveedores de Telefonía por Internet (ITSP)
- Proveedores de Servicios de SIP Trunk
- Servicios Corporativos Basados en SIP
- Enrutamiento SIP con LCR y Múltiples Gateways
- Distribución de Cargas en Redes SIP

### **Asterisk PBX**

Es un programa de software libre (bajo licencia GPL) que proporciona funcionalidades de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí dentro de una misma organización e incluso acceder a comunicaciones fuera de la misma a la PSTN o conectando a un proveedor de VoIP o bien a una RDSI tanto básicos como primarios.

Mark Spencer, de Digium, inicialmente creó Asterisk y actualmente es su principal desarrollador, junto con otros programadores que han contribuido a corregir errores y añadir novedades y funcionalidades. Originalmente desarrollado para el sistema operativo GNU/Linux, Asterisk actualmente también se distribuye en versiones para los sistemas operativos BSD, Mac OS X, Solaris y Microsoft Windows, aunque la plataforma nativa (GNU/Linux) es la que cuenta con mejor soporte de todas.

Asterisk incluye muchas características que anteriormente sólo estaban disponibles en costosos sistemas propietarios PBX, como buzón de voz, conferencias, IVR, distribución automática de llamadas, y otras muchas. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado en GNU/Linux.

Para conectar teléfonos estándares analógicos son necesarias tarjetas electrónicas telefónicas FXO y/o FXS fabricadas por Digium u otros proveedores. La primera consiste en cualquier dispositivo que actúe como un teléfono tradicional, conectado a una línea telefónica convencional, o sea es capaz de dar tono de marcado, hacer ring, descolgar y colgar. El segundo es un generador de línea telefónica interna, provee la alimentación, y las señales eléctricas (tensiones) necesarias para generar el tono de llamada y para establecer la comunicación del audio. En resumen FXO se conecta a la línea telefónica de un proveedor y FXS genera una línea telefónica interna.

Quizá lo más interesante de Asterisk es que reconoce muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un registrador y como gateway entre ambos.

Uno de los puntos fuertes del software Asterisk es que permite la unificación de tecnologías: VoIP, GSM y PSTN.

Asterisk se empieza a adoptar en algunos entornos corporativos como una gran solución de bajo coste junto con SER (Sip Express Router) aunque este último desde el 2007 que no recibe actualizaciones o nuevas versiones. Hoy en día Asterisk promete mucho más desarrollo y soporte que la mayoría de plataformas abiertas de telefonía.

## **Google APIs**

Son desarrolladas por Google, las cuales permiten la comunicación e integración de sus servicios con otros servicios. Ejemplos incluyen las APIs de Búsqueda, Gmail, Calendario, Traductor o Maps. Las aplicaciones de terceros pueden usar esas APIs para extender la funcionalidad de sus servicios.

Las APIs proveen funcionalidades como análisis, aprendizaje automático (machine learning), o acceso a los datos de usuario (donde estén establecidos los permisos de lectura).

Se puede acceder a los productos de Google Cloud desde cualquier código. Las API de Cloud ofrecen una funcionalidad similar a la del SDK y la consola de Cloud, además de permitirte automatizar los flujos de trabajo utilizando el lenguaje que uno prefiera. Se puede usar estas API de Cloud con llamadas REST o biblioteca de cliente en los lenguajes de programación más extendidos.

## **Telegram - Bots**

Telegram sube de nivel respecto a otras aplicaciones permitiendo utilizar automatizaciones creadas por otros que ayudan con determinadas tareas.

Los Bots de Telegram son una serie de aplicaciones de terceros que se ejecutan dentro de la aplicación de mensajería. No necesitas instalarlos ni hacer nada diferente para poder utilizarlos, ya que se integran de manera que se utilizan como si fueran una persona real con la que interactúas. Son multiplataforma, y los puedes usar en la versión de Telegram del sistema operativo que quieras.

La interfaz de estas aplicaciones es puramente textual, y los controlarás enviándoles mensajes que contengan determinados comandos o instrucciones que ellos entiendan para

reaccionar a ellos. Los comandos, las instrucciones o las palabras que les tienes que escribir para utilizarlos dependerá de cada bot.

Estas aplicaciones están totalmente automatizadas. No hay nadie detrás de ellas, sino que están preparadas para dar determinados tipos de respuestas dependiendo del comando que les escribas tú. Puedes encontrar varios tipos de bots, desde los más sencillos que simplemente muestran un contenido básico hasta otros algo más complejos y enfocados a alguna tarea concreta con la que puedan ayudarte.

Los Bots llegaron a Telegram en junio del 2015, y la aplicación de mensajería ofrece las herramientas necesarias para que cualquiera pueda crear un bot con el que automatizar tareas en la app. Al ser bots conversacionales, por lo general no requieren que interactúen pulsando o haciendo clic en ningún sitio, aunque muchos también incluyen botones en sus respuestas con los que enviar comandos o peticiones sin tener que escribirlas.

### **Gestión de la Seguridad - Firewall Linux (Shorewall/Iptables)**

Shorewall es una herramienta de firewall de código abierto para Linux que se basa en el sistema Netfilter ( iptables / ipchains ) integrado en el kernel de Linux , lo que facilita la administración de esquemas de configuración más complejos al proporcionar un mayor nivel de abstracción para describir reglas utilizando archivos de texto.

Características funcionales de Shorewall:

- Permite el filtrado de paquetes stateful gracias a las capacidades de monitoreo de conexiones de Netfilter.
- Puede utilizarse en múltiples aplicaciones de router (enrutador), firewall (corta-fuegos) o gateway (pasarela).
- Completamente configurable mediante el uso de archivos de configuración.

- Número de interfaces de red ilimitadas.
- Permite dividir las redes en zonas, permitiendo extenso control sobre las conexiones establecidas entre ellas.
- Múltiples interfaces por zona y múltiples zonas por interfaz.
- Soporta zonas anidadas y solapadas.

Todas estas características convierten a Shorewall en un firewall extremadamente flexible y configurable.

#### **1.4.Planteamiento del Problema**

Las empresas que prestan servicios en telecomunicaciones carecen de sistemas propios o contratados para la atención de quejas y reclamos de sus clientes así como el respectivo seguimiento mediante el área técnica, esto hace que la solución de cada problema sea muy lenta, y ocasiona que el cliente evalúe la calidad y su permanencia con el servicio que se compra o contrata, si se mejorara la atención esto elevaría el nivel de satisfacción del cliente que es indispensable para el crecimiento empresarial.

En muchos casos el soporte y seguimiento del mismo aún se lo realiza manualmente mediante la elaboración de hojas de ruta, llamadas telefónicas, mediante la atención de un solo agente por empresa que no satisface la demanda de reclamos y esto provoca que los servicios demandados por la empresa sean cancelados en el peor de los casos.

Por otro lado, los precios altos que implican implementar una solución propia de este tipo no son accesibles para todos, ya que implica una inversión alta inicialmente en hardware, software y recursos humanos capacitados en el área de informática y telecomunicaciones.

Por lo anteriormente expuesto se plantea la siguiente interrogante:

¿La implementación de una plataforma multicanal en línea optimizará la gestión y seguimiento de tickets en empresas de telecomunicaciones?

#### **1.4.1. Problema General**

La falta de una plataforma multicanal en línea para la gestión y seguimiento de tickets para empresas en telecomunicaciones.

#### **1.4.2. Problemas Específicos**

- Costos altos para implementar una plataforma multicanal propia.
- El hardware necesario no se fabrica en el país y su importación eleva los costos.
- Falta de software en el país que integre la gestión de reclamos y la atención multicanal en una misma solución.
- Falta de plataformas que brindan atención 24/7 para la apertura de reclamos.
- Clientes insatisfechos por la falta de canales de atención.
- Necesidad de realizar trabajo y soporte técnico en línea o de forma remota a causa de la pandemia de salud COVID-19.

### **1.5. Objetivos**

#### **1.5.1. Objetivo General**

Implementar una plataforma multicanal en línea para optimizar la gestión y seguimiento de tickets para empresas en telecomunicaciones.

#### **1.5.2. Objetivos Específicos**

- Reducir costos para las empresas al ofrecer una plataforma multicanal ya implementada en línea.
- Reducir costos en las empresas para adquirir hardware costoso o importado.

- Diseñar y aplicar varias tecnologías en una plataforma para ofrecer muchas más opciones y beneficios a las empresas en la gestión de sus clientes.
- Brindar una plataforma con disponibilidad 24/7 para la atención y apertura de reclamos.
- Implementar distintos medios o canales de atención para la comodidad de los clientes al momento de presentar un reclamo.
- Implementar una plataforma que permita realizar y colaborar con la gestión y soporte remoto o en línea a los clientes de las empresas en telecomunicaciones.

### **1.6.Hipótesis**

La implementación de una plataforma multicanal en línea permitirá optimizar la gestión y seguimiento de tickets para empresas en telecomunicaciones.

### **1.7.Justificación**

#### **1.7.1. Justificación Social**

La Plataforma mejorará y facilitará la atención de reclamos de los clientes que utilizan los servicios de una empresa en telecomunicaciones en un menor tiempo, de forma eficiente y oportuna. La satisfacción de los clientes es la clave para una mejor fidelización con la empresa, ya que se atiende las necesidades y expectativas de cada usuario.

Por otra parte, las empresas tendrán una mejor gestión de sus recursos humanos, tanto técnicos como operadores, al estar estos menos ocupados en la atención directa de los clientes, así como la opción de poder trabajar remotamente o en línea, fuera de la empresa en la atención, seguimiento y cierre de cada reclamo.

La implementación de esta plataforma agilizará los procesos de atención de reclamos de servicios, ya que cuenta con varios canales de atención (llamadas, IVRs, correo electrónico,

bots de Telegram) los cuales permitirán y garantizarán el alta del reclamo del cliente en el sistema y su posterior control y seguimiento. Por otro lado, el personal técnico no tendrá que estar necesariamente en la fuente de trabajo pudiendo estos estar trabajando remotamente desde sus hogares o directamente en la calle recibir las notificaciones push mediante el agendado en Google calendar o aviso a Telegram de las asistencias a los clientes que tienen que realizar el soporte de esta manera estos ser más productivos y eficientes en las empresas.

### **1.7.2. Justificación Técnica**

La plataforma ofrecerá la atención por distintos canales ya sea mediante llamadas telefónicas que interactúan con un IVR (Menú interactivo de voz), Bots de atención por Telegram o por Correo Electrónico, los cuales permitirán y garantizarán la apertura del reclamo por parte del cliente en cualquier horario las 24 horas.

La plataforma en su implementación estará integrada por varias tecnologías tanto del área informática como ser, un servidor web en linux, lenguaje de programación PHP, Java Script, Bootstrap, Codeigniter, JQuery, PWA (Progressive Web Applications), Base de Datos MySQL para la parte de Software, Firewall de Seguridad, y del área de telecomunicaciones como Opensips (Proxy SIP), Asterisk PBX (Central Telefónica), Protocolos de telefonía SIP, IAX, RTP, codec en la transferencia de voz y audio G729, G723, ULAW, ALAW, así como Softphones como puntos de acceso para la atención telefónica.

En la parte de Hardware se necesitará un servidor apropiado que pueda tener un procesador i7, i9 o Xeon de 10ma generación con memoria RAM mínima de 16GB, además de adaptadores o gateways telefónicos necesarios que permitan integrar la central telefónica actual de la empresa con la plataforma.

Dicho Hardware no es requerido para la empresa solo para la implementación de la plataforma.

### 1.7.3. Justificación Económica

Desde el punto de vista económico el servicio que ofrece la plataforma será accesible mediante un abono mensual de las empresas de telecomunicaciones a diferencia de querer adquirir toda la solución de manera propia, por otra parte, la implementación y desarrollo tanto en Hardware como en Software de toda la plataforma implica una inversión inicial alta por las características que ofrece.

A continuación, detallamos un esquema de precios tentativos para la implementación de toda la plataforma:

**Tabla 1**  
*Precios tentativos para la implementación de la plataforma*

Item	Descripción	Marca	Cant.	Precio/Unitario \$us	Total
1	Servidor Tipo Rack para plataforma WEB/SIP con raid 1 con hotswap	DELL PowerEdge T-140	1	1400	1400
2	Adaptadores Telefónicos	SPA 3102	1	70	70
3	Teléfonos IP (Opcional)	CISCO GRANDSTREAM	1 1	110 90	90
4	Instalación - Configuración Servidor Web/Firewall/Central Telefónica/Base de Datos	Asterisk Opensips Shorewall Mysql	1	1000	1000
5	Desarrollo e integración sistema de tickets multicanal	Programación con PHP, Telegram, Correo, PWA,	1	1000	1000
6	Dominio propio		1	20 / Anual	20
7	Servicio de Internet Corporativo con 5 IPs Públicos	30 MB/ Simétrico 1 a 1	1	500 / Mes	500
8	Licencia para Codecs de voz g729 para softphone EyeBeam / Zoiper	Digium	1	30	30
9	Router Mikrotik para control de ancho de banda/	RB 951	1	200	200

**Nota.** Elaboración Propia.

También se detalla el esquema de precios tentativos de abono mensual por contrato mínimo de 12 meses para uso de la plataforma:

Tabla 2  
*Precios tentativos de abono mensual por empresa*

Item	Descripción	Precio Bs	Impuestos Bs	Total Bs
1	Acceso a la plataforma multicanal con uso de adaptador telefónico	500	80	580
2	Acceso a la plataforma multicanal sin uso de adaptador telefónico	400	64	464

**Nota.** Elaboración Propia.

## 1.8.Límites y Alcances

### 1.8.1. Delimitación Temática

Delimitamos el alcance de la plataforma con los siguientes módulos:

Módulos del Sistema de Tickets

- Módulo de apertura de tickets de reclamos por los canales habilitados.
- Módulo de gestión de agentes y operadores.
- Módulo de asignación de los tickets a técnicos de la empresa para su seguimiento.
- Módulo para agendar las visitas técnicas en sitio.
- Módulo de notificaciones cuando se apertura un reclamo.
- Módulo de categorización de fallas de los servicios de la empresa.
- Módulo de sincronización de clientes.

Módulos de la Plataforma Multicanal

- Módulo de creación de IVR (Menú interactivo de voz).
- Módulo de creación de Bot con Telegram y creación de cuenta Gmail.

### **1.8.2. Delimitación Espacial**

El alcance será para empresas bolivianas de telecomunicaciones en función de su alcance territorial en 1 o varios departamentos, que cuenten con acceso a internet y con los requerimientos necesarios para poder utilizar la plataforma como ser:

- Computadora personal
- Acceso a Internet
- Dispositivo Móvil
- Navegador Web

### **1.8.3. Delimitación Temporal**

La plataforma analizará el comportamiento de los datos de reclamos para informar a la empresa:

- Los reclamos más comunes en un periodo de tiempo.
- Clientes con más reclamos.
- Tiempo de seguimiento desde la apertura hasta el cierre o solución de los reclamos.
- Evaluación de los técnicos por reclamos atendidos.

## **1.9. Metodología**

### **1.9.1. Metodología de Investigación**

La metodología de investigación que se utilizará de acuerdo al objeto de estudio y tipo de datos será cuantitativa aplicada con un enfoque inductivo.

### **1.9.2. Ingeniería**

Se utilizará la metodología de desarrollo ágil SCRUM para el sistema integrado a la plataforma, también el método de ingeniería de software UWE para el desarrollo de aplicaciones Web, el cual hará uso de la herramienta de modelado UML (Unified Modeling Language), para el modelado de datos, componentes y flujos de trabajo del sistema y la plataforma.

La solución de telefonía cuenta con la metodología de desarrollo orientada a objetos y también la estructurada en el manejo del ruteo de atención de las llamadas al IVR (menú interactivo de voz).

Las herramientas que se utilizarán para la implementación de la plataforma multicanal serán: Sistema Operativo Linux, Software de PBX Asterisk, Servidor Proxy OpenSIPs, Framework Codeigniter 3, Base de datos MySQL, JQuery, Ajax, HTML5, Bootstrap 4, Google APIs (Google calendar, Gmail), Bots de Telegram, WPA (Web Progressive Application - Push Notification).

### **1.10. Aporte**

El principal aporte de este trabajo radica en el hecho de que muchas empresas en telecomunicaciones en Bolivia tendrán acceso una plataforma multicanal económica y segura que ayude al tratamiento y gestión de reclamos de sus clientes, obtendrán muchos beneficios que ayuden al crecimiento empresarial, podrán evaluar y mejorar los procesos que influyen en la pronta solución de los reclamos de sus clientes y la utilidad que trae un buen manejo de los mismos. Tendrán la posibilidad de optar por el trabajo remoto o en línea y de esta manera mejorar la organización de los recursos humanos.

## **CAPITULO 2**

### **MARCO TEORICO**

#### **2.1. Introducción**

##### **2.1.1. La atención y soporte al cliente en Bolivia**

El Covid-19, además de generar un efecto devastador en las personas y los sistemas de salud, está afectando duramente la economía global, con graves consecuencias para el empleo y los servicios. Uno de los sectores más golpeados son las empresas que ofrecen distintos servicios en telecomunicaciones.

Inicialmente la atención y soporte a los clientes fue evolucionando desde obtener una ficha, llenar un formulario de reclamo todo en forma presencial, hasta la actualidad que muchas empresas ya tienen implementado un call center o por lo menos poseen un número telefónico para la asistencia a sus clientes, incluso muchas innovaciones en la actualidad destacan como por ejemplo la empresa Conecta (Santa Cruz de la Sierra) que ofrece servicios de call center y atención al cliente y que por esta pandemia implemento el teletrabajo para sus operadores asegurando así la asistencia y el contacto con sus clientes.

En países como Argentina, Colombia, Perú, México, Brasil y El Salvador, los gobiernos y los gremios de call center ya vienen trabajando integrando actuales y nuevas tecnologías en mejorar más las condiciones para este sector que es intensivo en la generación de fuentes de empleo. Para eso, están tomando medidas como el incremento de la velocidad y baja de costos de Internet, la utilización de plataformas de call center en la nube, integración con plataformas sociales (Telegram, WhatsApp, Facebook, etc.) y la modificación de normativas

laborales que permitan enmarcarse en las nuevas tecnologías y formas de trabajo con mayor flexibilidad. Esto es lo que necesitamos en Bolivia para ser competitivos a nivel mundial.

Otro aspecto a considerar son los recursos económicos tanto por parte de los clientes como por parte de las empresas, estos con mayor razón disminuyen la posibilidad de dar un mejor servicio a sus clientes, lo cual desencadena una serie de sucesos terminando en el cierre de la empresa por disminución de clientes que adquieren sus servicios.

Juan Pablo Ayala (Executive Director & CEO at Hispanic Synergy Florida Group) analista en desarrollo de negocios y un capacitador experto en Emprendedurismo, Servicio al cliente, y Orgullo laboral afirma. “La experiencia genera ventas, la experiencia genera retención, la experiencia genera diferenciación”.

## **2.2. Ingeniería de software**

La Ingeniería de Software es la rama de la ingeniería que estudia todo lo relacionado con la informática o sistemas de computación, con una orientación metódica, ordenada y cuantificable al incremento, ejecución y conservación del software.

La ingeniería de software es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de los programas informáticos.

La ingeniería de software, también, incorpora el análisis precedente de la situación, el bosquejo del proyecto, el desarrollo del software, el ensayo necesario para comprobar su funcionamiento correcto y poner en funcionamiento el sistema.

Se debe señalar, que el desarrollo del software va unido a lo que se conoce en el campo del software “ciclo de vida del software” que consiste en cuatro etapas que se conocen como: concepción, elaboración, construcción y transición.

### 2.2.1 Metodología ágil de desarrollo de software

Por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

Las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo piezas pequeñas de sistemas de software en funcionamiento para mejorar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos autoorganizados de desarrolladores de software y representantes empresariales realizan el trabajo multidisciplinario regularmente durante el ciclo de vida del desarrollo de software. La metodología ágil favorece un enfoque sencillo de la documentación de software, y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos.

Metodologías tradicionales	Metodologías ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

*Figura 3.* Comparación de Metodologías de Desarrollo de Software.

**Nota.** Tomado de Navarro, Fernández & Morales (2013)

## **2.2.2 SCRUM**

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum es un modelo de desarrollo ágil caracterizado por (Palacio 2014):

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada.

### **2.2.2.1. *Gestión de la evolución del Proyecto.***

Scrum maneja de forma empírica la evolución del proyecto con las siguientes tácticas (Palacio 2014):

- Revisión de las Iteraciones: Al finalizar cada sprint se revisa funcionalmente el resultado con todos los implicados en el proyecto. Es por tanto la duración del sprint, el periodo máximo para descubrir planteamientos erróneos, mejorables o malinterpretaciones en las funcionalidades del producto.
- Desarrollo incremental: El desarrollo incremental ofrece al final de cada iteración una parte de producto operativa, que se puede usar, inspeccionar y evaluar. Durante la construcción se depura el diseño y la arquitectura, y no se cierran en una primera fase del proyecto. Las distintas fases que el desarrollo en cascada realiza de forma secuencial, en Scrum se solapan y realizan de forma continua y simultánea.
- Autoorganización: Son muchos los factores impredecibles en un proyecto. La gestión predictiva (métodos tradicionales) asigna al rol de gestor del proyecto la responsabilidad de su gestión y resolución. En Scrum (metodología ágil) los equipos son autoorganizados, con un margen de maniobra suficiente para tomar las decisiones que consideren oportunas.
- Colaboración: Es un componente importante y necesario para que a través de la autoorganización se pueda gestionar con solvencia la labor que de otra forma realizaría en gestor de proyectos. Todos los miembros del equipo colaboran de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

#### **2.2.2.2. Roles.**

- Propietario del producto (Product Owner)

El propietario del producto es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto. Decide en última instancia cómo será el resultado final,

y el orden en el que se van construyendo los sucesivos incrementos (que se pone y que se quita de la pila del producto), y cuál es la prioridad de las funcionalidades. Conoce el plan del producto, sus posibilidades y plan de inversión, así como del retorno esperado a la inversión realizada, y se responsabiliza sobre fechas y funcionalidades de las diferentes versiones del mismo. Para ejercer este rol es necesario (Palacio 2014):

- Conocer perfectamente el entorno del negocio del cliente, las necesidades y el objetivo que se persigue con el sistema que se está construyendo.
- Tener la visión del producto, así como las necesidades concretas del proyecto, para poder priorizar eficientemente el trabajo.
- Disponer de atribuciones y conocimiento del plan del producto suficiente para tomar las decisiones necesarias durante el proyecto, incluidas para cubrir las expectativas previstas de retorno de la Inversión del proyecto.
- Recibir y analizar de forma continua retroinformación del entorno de negocio (evolución del mercado, competencia, alternativas) y del proyecto (sugerencias del equipo, alternativas técnicas, pruebas y evaluación de cada incremento).
- Scrum Master

El Scrum Master es el responsable del cumplimiento de las reglas del marco técnico Scrum, asegurándose de que se entiendan en la organización, y se trabaje conforme a ellas. Realiza su trabajo con un modelo de liderazgo servil, al servicio y en ayuda del equipo y del propietario del producto, proporciona (Palacio 2014):

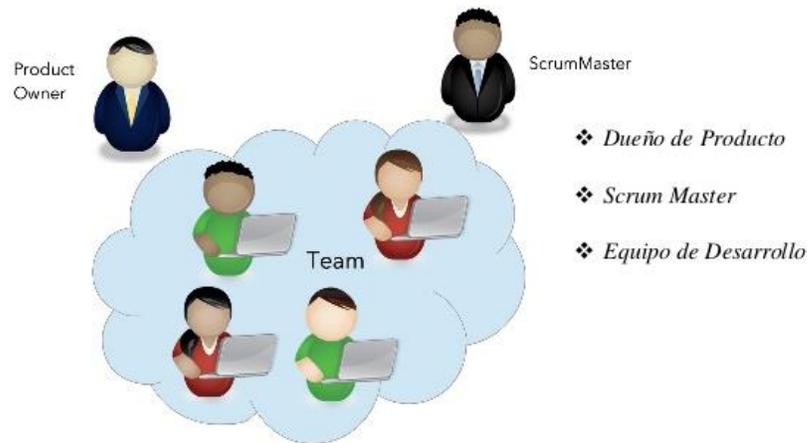
- Asesoría y formación al equipo para trabajar de forma autoorganizada y con responsabilidad de equipo.

- Revisión y validación de la pila del producto.
- Moderación de las reuniones.
- Resolución de impedimentos que en el sprint puedan entorpecer la ejecución de las tareas.
- Gestión de las dinámicas del grupo en el equipo.
- Configuración, diseño y mejora continua de las prácticas de Scrum en la organización, respeto de la organización y los implicados, con las pautas de tiempos y formas de Scrum.
- Equipo de Desarrollo

Lo forman el grupo de profesionales que realizan el incremento de cada sprint. Se recomienda que un equipo Scrum tenga entre 4 a 8 personas. No se trata de un grupo de trabajo formado por un arquitecto, diseñador o analista, programadores y testers. Es un equipo multifuncional, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida. Es posible que algunos miembros sean especialistas en áreas concretas, pero la responsabilidad es el incremento de cada sprint y recae sobre el equipo de desarrollo en conjunto. Un equipo scrum responde en su conjunto. Trabaja de forma cohesionada y autoorganizada. No hay un gestor para delimitar, asignar y coordinar las tareas. Son los propios miembros los que lo realizan. Entonces en el equipo (Palacio 2014):

- Todos conocen y comprenden la visión del propietario del producto.
- Aportan y colaboran con el propietario del producto en el desarrollo de la pila del producto.
- Comparten de forma conjunta el objetivo de cada sprint y la responsabilidad del logro.
- Todos los miembros participan en las decisiones.

- Se respetan las opiniones y aportes de todos.
- Todos conocen el modelo de trabajo con scrum.



*Figura 4.* Descripción de los Roles Scrum (Webipedia Scrum).

### **2.2.2.3. Artefactos.**

- **Product Backlog:** consiste en un documento que recoge el conjunto de requerimientos que se asocian al proyecto y de todo lo que podría necesitarse en el producto y es la única fuente de requerimientos para los cambios que se realizarán en el producto. El Product Owner es responsable del Product Backlog incluyendo su contenido, disponibilidad y jerarquización. Inicialmente, contiene los requerimientos de lo que se conoce y entiende al momento de iniciar el desarrollo. El Product Backlog evoluciona conforme el producto y su entorno evolucionan. El Product Backlog es dinámico; cambia constantemente de acuerdo a lo que el producto requiere para mantenerse adecuado, competitivo y útil. Mientras exista un producto, su Product Backlog también existe. El Product Backlog incluye todas las características, funciones, requerimientos, mejoras y correcciones que constituyen los cambios que deben introducirse en el producto en

futuras versiones. Los elementos del Product Backlog deben contener los siguientes atributos: descripción, orden, estimación y valor.

Conforme el producto se utiliza y su valor útil aumenta y recibe retroalimentación del mercado, el Product Backlog crece y se convierte en una lista más exhaustiva. Cambios en las necesidades del negocio, las condiciones del mercado o la tecnología tienen un impacto directo en el Product Backlog, por lo que es un artefacto dinámico.

- **Sprint Backlog:** un sprint debe entenderse como un subconjunto de requerimientos, extraídas del Product backlog, para ser ejecutadas durante un periodo entre 1 y 4 semanas de trabajo. El sprint backlog sería el documento que describa las tareas que son necesario realizar para abordar el dicho subconjunto de requerimientos.

El Sprint Backlog es el conjunto de elementos seleccionados del Product Backlog para el Sprint, además de un plan para la entrega del Incremento y el cumplimiento del objetivo del Sprint. El Sprint Backlog es una proyección realizada por el equipo de desarrollo sobre la funcionalidad que estará en el próximo incremento y el trabajo necesario para convertir esa funcionalidad en un incremento "Terminado".

El Sprint Backlog hace visible todo el trabajo que el equipo de desarrollo ha identificado como necesario para cumplir con el objetivo del Sprint.

Es un plan con suficiente detalle como para que los cambios en curso puedan ser entendidos durante el Scrum Diario. El equipo de desarrollo modifica el Sprint Backlog durante todo el ciclo y, conforme avanza en el plan y entiende mejor el trabajo necesario para lograr el objetivo, el Sprint Backlog emerge y se concreta. Conforme se identifican nuevas tareas, el equipo de desarrollo las añade al Sprint Backlog. Conforme se realizan o completan, el tiempo para completar el trabajo se actualiza. Cuando se identifican tareas

que ya no son necesarias, se eliminan. Sólo el equipo de desarrollo puede realizar cambios en el Sprint Backlog, el cual representa una imagen en tiempo real de lo que planea realizar durante el Sprint. Por lo tanto, este artefacto pertenece únicamente al equipo de desarrollo.

La ejecución del Sprint sería el periodo de entre 1 y 4 semanas (periodo definido previamente en base a las tareas recogidas en el sprint backlog) durante el cual el equipo de trabajo abordaría las tareas de desarrollo correspondientes. Una vez iniciada la ejecución de un sprint definido, este no podrá ser modificado, y en caso de ser necesario introducir cambios estos se harán una vez concluido el periodo a través de la definición de otro sprint backlog.

- **Burn down:** Es un documento que refleja el estado del proyecto, indicando el volumen de requerimientos que en ese momento se encuentran pendientes de ser abordados (en el product backlog), los requerimientos que en ese momento se están desarrollando (sprint backlog) y los requerimientos cuyo desarrollo ya se ha completado en su totalidad.

#### **2.2.2.4. Eventos.**

- **Sprint 0 – Planificación del Proyecto:** También es conocido como la preparación del proyecto, que corresponde a la definición de las historias de usuario para la pila del producto en base a los requerimientos y necesidades del cliente, y la determinación del tamaño y la duración de los sprints y cuántos de estos se realizarán, además conjuntamente también se realizan las actividades preparatorias de análisis, diseño, selección de las herramientas tecnológicas, identificación de políticas de calidad y seguridad para el sistema a desarrollar, y si es necesario la capacitación del equipo de

desarrollo. Deben estar presentes el Propietario del Producto, el Scrum Master, el equipo de desarrollo, los interesados y dura de una a dos semanas (Menzinsky, 2015).

- **Reunión de planificación de Sprint:** En esta reunión se toman como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint (Palacio, 2014). El responsable de que se realice esta reunión es el Scrum Master y debe asegurarse de que estén presentes el propietario del producto, el equipo de desarrollo y si es necesario los implicados o interesados del proyecto. La reunión de planificación de Sprint dura aproximadamente una jornada de trabajo completa, cuando se trata de planificar un sprint largo (de un mes de duración) o un tiempo proporcional para planificar un sprint más breve (Palacio, 2014).

El propietario del producto presenta la pila de producto, exponiendo los requisitos de mayor prioridad que se necesita y prevé que se podrán desarrollar en el siguiente sprint. Si algún elemento de la pila del producto ha tenido cambios significativos desde la anterior reunión, explica las causas que los han ocasionado (Palacio, 2014).

El equipo de desarrollo hace preguntas y pide aclaraciones necesarias, además pueden proponer sugerencias, modificaciones y soluciones alternativas, como respuesta el propietario del producto atiende las dudas y sugerencias del equipo asegurándose de que tengan un buen entendimiento de la idea y visión del producto. El equipo desglosa cada funcionalidad en tareas, y estiman el tiempo para cada una de ellas, autoasignándose en base a sus conocimientos, intereses y una distribución homogénea de trabajo, componiendo así las tareas que forman la pila de sprint (Palacio, 2014).

- **Sprint:** Se denomina sprint a cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa (incremento). La duración de cada sprint puede ser desde una hasta seis semanas, aunque se recomienda que no excedan de un mes. El equipo monitoriza la evolución de cada sprint en reuniones breves diarias (reuniones de pie – stand-up meeting) que duran entre 5 a 15 minutos donde se revisa en conjunto las tareas realizadas por cada miembro el día anterior, y previsto para el día en curso (Palacio, 2014).
- **Reunión Diaria:** Las reuniones diarias, no debe durar más de 15 minutos, deben estar presentes todo el equipo. En esta reunión cada miembro del equipo explica lo que ha logrado desde la anterior reunión diaria, lo que va a hacer en la jornada laboral actual y si está teniendo algún problema, o si prevé que puede encontrar algún impedimento, para que los demás miembros del equipo puedan colaborarle en solucionarlo (Palacio, 2014).
- **Revisión del sprint:** Se lo realiza al finalizar un sprint para comprobar el incremento. No debe durar más de 4 horas en caso de sprints largos. Para sprints de 1 o 3 semanas, de 1 a 2 horas es suficiente y deben estar presentes el equipo de desarrollo, el propietario del producto y el Scrum Master. El equipo expone el objetivo del sprint, las listas de funcionalidades deseadas, y las que se han desarrollado, haciendo una demostración de las partes construidas. Luego se hace una ronda de preguntas y sugerencias, donde se genera información valiosa respecto a la visión del producto. El propietario del producto comprobando el progreso del sistema, ve si se puede considerar si se ha logrado el objetivo y el Scrum Master, de acuerdo a las agendas del propietario del producto y el equipo, cierra la reunión para la preparación del siguiente sprint (Palacio, 2014).

- **Retrospectiva del sprint:** Se los realiza tras la revisión de cada sprint, y antes de la reunión de planificación del siguiente sprint, puede durar de 1 a 3 horas, según la duración del sprint terminado. En ella el equipo realiza un autoanálisis de su forma de trabajar, identificando fortalezas para consolidarlos y puntos débiles para tomar acciones de mejoras (Palacio, 2014).

En la siguiente figura se muestra un resumen gráfico de los eventos Scrum que ya fueron expuestos.



Figura 5. Descripción resumida de los Eventos Scrum.

**Nota.** Tomado de Palacio (2014)

## 2.3. Desarrollo Web basada en U.M.L. (U.W.E.)

### 2.3.1. Lenguaje de Modelado de datos U.M.L.

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que programación estructurada es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML solo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas y algunos son más necesarios dependiendo del tipo del proyecto.

### ***Estructurales***

Los diagramas estructurales muestran la estructura estática del sistema y sus partes en diferentes niveles de abstracción. Existen un total de siete tipos de diagramas de estructura:

- Diagrama de clases

Los diagramas de clase son, sin duda, el tipo de diagrama UML más utilizado. Es el bloque de construcción principal de cualquier solución orientada a objetos. Muestra las clases en un sistema, atributos y operaciones de cada clase y la relación entre cada clase. En la mayoría de las herramientas de modelado, una clase tiene tres partes, nombre en la parte superior, atributos en el centro y operaciones o métodos en la parte inferior. En sistemas grandes con muchas clases

relacionadas, las clases se agrupan para crear diagramas de clases. Las diferentes relaciones entre las clases se muestran por diferentes tipos de flechas.

- Diagrama de componentes

Un diagrama de componentes muestra la relación estructural de los componentes de un sistema de software. Estos se utilizan principalmente cuando se trabaja con sistemas complejos que tienen muchos componentes. Los componentes se comunican entre sí mediante interfaces. Las interfaces se enlazan mediante conectores.

- Diagrama de despliegue

Un diagrama de despliegue muestra el hardware de su sistema y el software de ese hardware. Los diagramas de implementación son útiles cuando la solución de software se despliega en varios equipos, cada uno con una configuración única.

- Diagrama de objetos

Los diagramas de objetos, a veces denominados diagramas de instancia, son muy similares a los diagramas de clases. Al igual que los diagramas de clases, también muestran la relación entre los objetos, pero usan ejemplos del mundo real. Se utilizan para mostrar cómo se verá un sistema en un momento dado. Debido a que hay datos disponibles en los objetos, a menudo se utilizan para explicar relaciones complejas entre objetos.

- Diagrama de perfiles

Diagrama de perfil es un nuevo tipo de diagrama introducido en UML 2. Este es un tipo de diagrama que se utiliza muy raramente en cualquier especificación.

- Diagrama de estructura compuesta

Los diagramas de estructura compuesta se utilizan para mostrar la estructura interna de una clase.

### *De comportamiento*

Muestran el comportamiento dinámico de los objetos en el sistema.

- Diagrama de actividades

Los diagramas de actividad representan los flujos de trabajo de forma gráfica. Pueden utilizarse para describir el flujo de trabajo empresarial o el flujo de trabajo operativo de cualquier componente de un sistema. A veces, los diagramas de actividad se utilizan como una alternativa a los diagramas de máquina del estado.

- Diagrama de casos de uso

Como el tipo de diagrama de diagramas UML más conocido, los diagramas de casos de uso ofrecen una visión general de los actores involucrados en un sistema, las diferentes funciones que necesitan esos actores y cómo interactúan estas diferentes funciones. Es un gran punto de partida para cualquier discusión del proyecto, ya que se pueden identificar fácilmente los principales actores involucrados y los principales procesos del sistema.

- Diagrama de máquina de estados

Los diagramas de máquina de estado son similares a los diagramas de actividad, aunque las anotaciones y el uso cambian un poco. En algún momento se conocen como diagramas de estados o diagramas de diagramas de estado también. Estos son muy útiles para describir el comportamiento de los objetos que actúan de manera diferente de acuerdo con el estado en que se encuentran en el momento.

- Diagrama de interacción

Los diagramas de interacción incluyen distintos tipos de diagramas:

- Diagrama de secuencia

Los diagramas de secuencia en UML muestran cómo los objetos interactúan entre sí y el orden en que se producen esas interacciones. Es importante tener en cuenta que muestran las interacciones para un escenario en particular. Los procesos se representan verticalmente y las interacciones se muestran como flechas. Los diagramas de secuencia de UML forman parte de un modelo UML y solo existen dentro de los proyectos de modelado UML.

- Diagrama de comunicación

El diagrama de comunicación se llamó diagrama de colaboración en UML 1. Es similar a los diagramas de secuencia, pero el foco está en los mensajes pasados entre objetos.

- Diagrama de tiempos

Los diagramas de sincronización son muy similares a los diagramas de secuencia. Representan el comportamiento de los objetos en un marco de tiempo dado. Si es solo un objeto, el diagrama es directo, pero si hay más de un objeto involucrado, también se pueden usar para mostrar interacciones de objetos durante ese período de tiempo.

- Diagrama global de interacciones

Los diagramas generales o globales de interacción son muy similares a los diagramas de actividad. Mientras que los diagramas de actividad muestran una secuencia de procesos, los diagramas de interacción muestran una secuencia de diagramas de interacción. En términos simples, pueden llamarse una colección de diagramas de interacción y el orden en que suceden.

Como se mencionó anteriormente, hay siete tipos de diagramas de interacción, por lo que cualquiera de ellos puede ser un nodo en un diagrama de vista general de interacción.

### **2.3.2. Ingeniería Web basada en U.M.L. (U.W.E.)**

U.W.E. es una metodología que permite especificar de mejor manera una aplicación web, para el proceso de creación de aplicaciones y describe que es lo que se debe utilizar. Este proceso iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el proceso unificado de modelado (Pérez, 2010).

La metodología U.W.E. define vistas especiales representadas gráficamente por diagrama en UML. Los principales aspectos en los que se fundamenta U.W.E. son los siguientes:

- Notación estándar: el uso de la metodología U.M.L. para todos los modelos.
- Métodos definidos: pasos definidos para la construcción de cada modelo.
- Especificación de restricciones: recomendables de manera escrita, para que la exactitud en cada modelo aumente.

#### **2.3.2.1. Fases de Desarrollo.**

Las actividades base de modelado de U.W.E. son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como son el modelo de usuario, modelo de adaptación y modelo de tareas para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. De esta manera se obtiene una colección de modelos y diagramas que describen una aplicación web de manera integral (Pérez, 2010).

- **Análisis de requisitos:** plasma los requerimientos funcionales de la aplicación web, mediante modelos de casos de uso.

Según Novilos, Coronel & Campaña (2010), el análisis de requisitos se expresa a través de la especificación de los casos de uso del sistema. Un caso de uso en UML es una unidad coherente de la funcionalidad proporcionada para la aplicación que obra recíprocamente con unos o más actores de la aplicación. Describe una parte del comportamiento de la aplicación sin revelar la estructura interna. De esta manera, los requisitos para una aplicación web se pueden especificar con un modelo de casos de uso.

- **Diseño conceptual:** se define mediante un modelo de dominio, considerando los requisitos plasmados en los casos de uso, el diagrama de clases representará los conceptos con un gran porcentaje de detalle.

El modelado conceptual para aplicaciones web dentro de UWE no difiere del modelado conceptual de las aplicaciones normales. Se utiliza un diagrama de clases para representar gráficamente un modelo conceptual como visión estática que demuestre una colección de los elementos estáticos del dominio (Novilos, Coronel, & Campaña, 2010).

La construcción del modelo conceptual se lleva a cabo de acuerdo con los casos de uso que se definieron en la especificación de los requerimientos de usuario.

- **Diseño navegacional:** Comprende la construcción del modelo de navegación en dos pasos:
  - Modelo de espacio de navegación: su objetivo es especificar qué objetos pueden ser visitados a través de la aplicación.
  - Modelo de estructura de navegación: amplía el modelo con un modelo de estructuras de acceso necesarias para la navegación como los índices, consultas y visitas guiadas.

- **Diseño de presentación:** permite la especificación lógica de la aplicación web. Basada sobre este modelo lógico, una presentación física puede ser construida. Representa las interfaces de usuario por medio de vistas estándares de interacción U.M.L. Dentro de este modelo se distinguen dos diferentes vistas:
  - **Estructura de vista:** Muestra la estructura del espacio de presentación.
  - **Interfaz de usuario (UI por sus siglas en inglés de User Interface):** Se refleja los detalles acerca de los elementos de la interfaz de usuario dentro de las páginas.

#### 2.4. Patrón de diseño

Según Cáceres. J. En el desarrollo software el patrón de diseño describe las clases y objetos que se comunicarán entre sí de manera que puedan resolver un problema general de diseño en un contexto particular. En un contexto informático un patrón de diseño es similar a conceptos como biblioteca de clases, frameworks, técnicas y/herramientas de refactorización o programación extrema.

La clasificación de los patrones de diseño en este contexto atiende a dos características como son:

- Atendiendo a su propósito
  - Creacionales: Resuelven problemas relativos a la creación de objetos
  - Estructurales: Resuelven problemas relativos a la composición de objetos
  - de Comportamiento: Resuelven problemas relativos a la interacción entre objetos.
- Atendiendo a su ámbito
  - Clases: Relaciones estáticas entre clases
  - Objetos: Relaciones dinámicas entre objetos

### **2.4.1. Modelo vista controlador MVC**

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores. (Fernández & Díaz, 2012).

#### ***Definición de partes***

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo.

Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. La figura 6 ilustra los elementos del patrón y la interrelación entre estos.

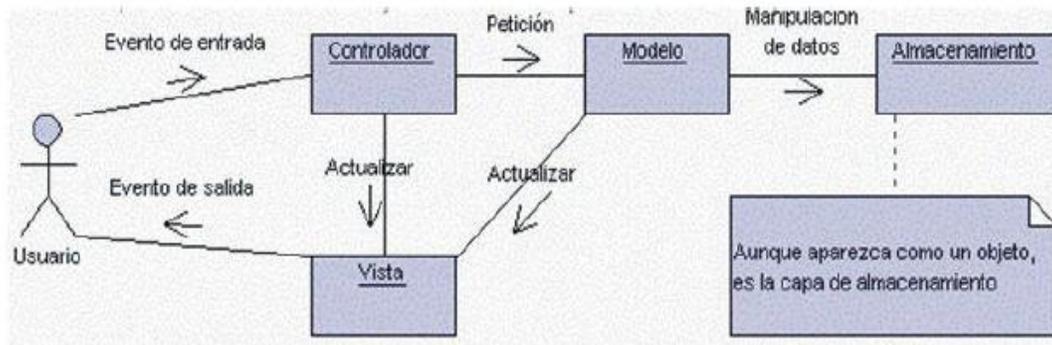


Figura 6. Interrelación entre los elementos del patrón MCV.

**Nota.** Tomado de Fernández & Díaz, 2012.

## 2.5. Tecnologías

### 2.5.1. Tecnologías Web

#### 2.5.1.1. PHP.

Es un lenguaje de programación diseñado específicamente para la Web, ya que todo el código es interpretado y ejecutado en un servidor Web. Además, que se puede incluir etiquetas HTML con código PHP embebido en un mismo archivo, permitiendo que las páginas web sean dinámicas.

Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. Algunas de sus características más sobresalientes son (Welling & Thomson, 2005):

- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que el servidor es el que se encarga de ejecutar el código y enviar su resultado en HTML u otro formato Json al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- Permite aplicar técnicas de programación orientada a objetos.

### ***2.5.1.2. Codeigniter 3 Framework.***

CodeIgniter es un entorno de desarrollo de aplicaciones para gente que construye sitios web usando PHP. El objetivo es habilitar el desarrollo de proyectos de forma mucho más rápida de lo que podría si escribiese código desde cero, a través de proveer un rico conjunto de librerías para tareas comúnmente necesarias, tanto como una simple interface y estructura lógica para acceder a estas librerías. CodeIgniter permite concentrarse creativamente en un proyecto, minimizando el volumen de código necesario para una tarea determinada.

#### **La estructura de CodeIgniter**

El diseño orientado al rendimiento de este framework de desarrollo web se revela en su parca arquitectura, pues se basa en el patrón Modelo-Vista-Controlador (MVC). El principio fundamental que sustenta a la arquitectura de desarrollo MVC es la estricta separación entre el

código y la presentación, gracias a una estructura modular de software y a la externalización del código PHP. Esta separación se realiza en estos tres grupos: el modelo (model), la vista (view) y el controlador (controller), que se explica a continuación:

- El modelo representa la estructura de datos de una aplicación web desarrollada con CodeIgniter. Para ello, en el código fuente se definen las denominadas clases (“model classes”), que contienen funciones especiales con las cuales se puede recibir, insertar o actualizar la información de la base de datos.
- La vista es aquello que se le presenta al usuario final. Por lo general, se trata de un documento HTML en el cual se ha insertado contenido de forma dinámica con PHP, convirtiéndose en una especie de plantilla. CodeIgniter también permite definir fragmentos de una página web como la cabecera y el pie de página o páginas RSS como vista. Normalmente las aplicaciones web utilizan varias vistas, que toman su contenido desde el mismo modelo, de tal forma que es posible presentar diversas características del programa en vistas diferentes.
- El controlador media entre el modelo, la vista y cualquier otro recurso necesario para procesar una petición HTTP o generar una página web de forma dinámica. Este componente recibe las peticiones entrantes, valida la entrada, selecciona la vista deseada y le entrega el contenido que el modelo ha cargado desde una base de datos.

En la siguiente figura se muestra el flujo de datos en el sistema.

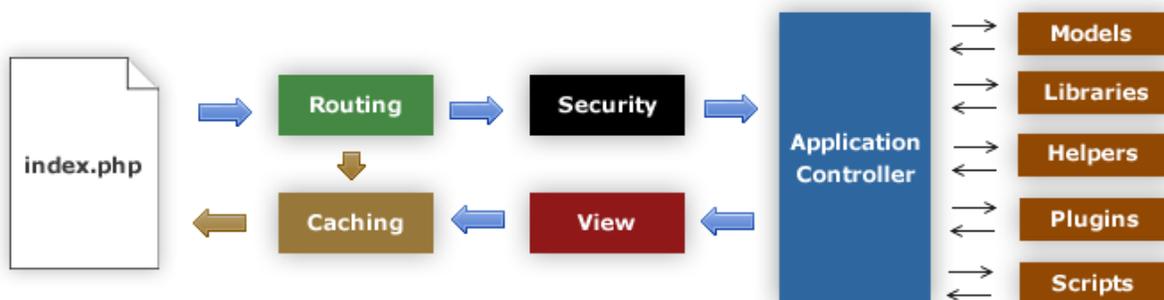


Figura 7. Diagrama del flujo de una aplicación en CodeIgniter.

**Nota.** Tomado de CODEIGNITER sitio oficial

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para ejecutar CodeIgniter.
2. El enrutador examina la solicitud HTTP para determinar qué se debe hacer con ella.
3. Si existe un archivo de caché, se envía directamente al navegador, sin pasar por la ejecución normal del sistema.
4. Seguridad. Antes de que se cargue el controlador de la aplicación, la solicitud HTTP y los datos enviados por el usuario se filtran por motivos de seguridad.
5. El controlador carga el modelo, las bibliotecas centrales, los ayudantes y cualquier otro recurso necesario para procesar la solicitud específica.
6. La Vista finalizada se procesa y luego se envía al navegador web para que se vea. Si el almacenamiento en caché está habilitado, la vista se almacena en caché primero para que se pueda atender en solicitudes posteriores.

### 2.5.1.3. JavaScript.

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Por lo general se emplea del lado del cliente (lo que se conoce como client-side), aunque también hay una forma de este lenguaje del lado del servidor (server-side).

Con javascript se puede crear efectos visuales en las páginas, como movimientos dinámicos, cambio de colores, etc. como también controlar algunas acciones de los usuarios, puede ser validar los datos de los formularios, manipular los elementos de una página web, entre otros.

#### **2.5.1.4. *Jquery.***

jQuery es una biblioteca multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

jQuery incluye las siguientes características:

- Selección de elementos DOM utilizando el motor de selección de código abierto de múltiples navegadores Sizzle, un spin-off del proyecto jQuery.

- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Objetos diferidos y de promesa para controlar el procesamiento asíncrono
- Soporta extensiones (JSON).
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Métodos de compatibilidad que están disponibles de forma nativa con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.20
- Soporte para múltiples navegadores

#### **2.5.1.5. Google APIs.**

API es la sigla de Application Programming Interface o, en español, Interfaz de Programación de Aplicaciones. Es básicamente un conjunto de aplicaciones que permiten la construcción de una interfaz inteligente, configurando un medio por el cual dos sistemas se comunican.

### **Autenticación y autorización**

El uso de algunas API de Google requiere autenticación y autorización, utilizando el protocolo OAuth 2.0 protocol. OAuth 2.0 is a simple protocol. Para empezar, es necesario obtener las credenciales provistas por la Consola de Desarrolladores; entonces, el desarrollador puede solicitar un token de acceso desde el servidor de autorización de Google, y la aplicación cliente puede usar ese token para acceder a los servicios provistos por la API de Google.

### **Características**

- El Registro de usuarios es hecho comúnmente a través del servicio de inicio de sesión de Google+, que permite a servicios de terceros iniciar sesión de forma segura a través de la cuenta de Google del usuario usando la API de Google+ (generalmente mediante un botón en las páginas).
- Drive apps consiste en varias aplicaciones web (generalmente de terceros) que pueden interactuar con la API de Drive, lo que permite integrar las aplicaciones de Chrome Store para trabajar íntegramente en la nube. Entre los usos más comunes incluyen herramientas de edición colaborativa (como Google Docs y Hojas de cálculo), edición multimedia, diagramado, y sincronización de archivos.
- Custom Search (Búsqueda Personalizada) permite a los desarrolladores web usar un cuadro con un motor de búsqueda personalizado usando la API provista. Como su nombre lo indica, permite personalizar el motor en base al acotado de resultados al sitio web donde se integra, la apariencia, y cómo monetizar utilizando AdSense.
- Las aplicaciones App Engine son aplicaciones web que corren en Google App Engine, un servicio basado en la nube que permite a los desarrolladores correr servicios web en los datacenters de Google. Esas aplicaciones pueden verse beneficiadas con la API para manipular servicios como TaskQueue, BigQuery o DataStore.

- Los Gadgets son mini aplicaciones escritas en HTML, JavaScript, Flash y Silverlight, las que se pueden incrustar dentro de páginas web u otras aplicaciones.

#### **2.5.1.6. Aplicación Web Progresiva (PWA).**

Las aplicaciones web progresivas están diseñadas para funcionar en cualquier navegador que cumpla con los estándares web. Al igual que con otras soluciones multiplataforma, el objetivo es ayudar a los desarrolladores a crear aplicaciones multiplataforma más fácilmente de lo que lo harían con las aplicaciones nativas. Según los desarrolladores de Google, las características de una PWA son (Jason Rzutkiewicz & Jeremy Lockhorn):

- Progresiva: funciona para todos los usuarios, independientemente de la elección del navegador, utilizando principios de mejora progresiva.
- Responsiva: se adapta a cualquier factor de forma: computadora de escritorio, dispositivo móvil, tableta o formato que aún no han surgido
- Más rápido después de la carga inicial: una vez finalizada la carga inicial, no es necesario volver a descargar el mismo contenido y elementos de página cada vez. Los sitios web ordinarios a menudo utilizan la memoria caché del navegador para evitar volver a descargar los mismos datos de forma redundante, mientras que, para las PWA, el navegador asigna su propio espacio de almacenamiento independiente.
- Conectividad independiente: los service worker permiten usos fuera de línea o en redes de baja calidad.
- Como una aplicación: se siente como una aplicación para el usuario con interacciones y navegación de estilo de aplicación.

- Actualizado: siempre actualizado debido al proceso de actualización del trabajador de servicio.
- Seguro: se sirve a través de HTTPS para evitar espiar y garantizar que el contenido no haya sido manipulado.
- Detectable: identificable como una "aplicación" por manifest.json y el registro del service worker, y detectable por los motores de búsqueda.
- Reconectable: capacidad de utilizar notificaciones push para mantener la experiencia de usuario.
- Instalable: proporciona iconos de la pantalla de inicio sin el uso de una tienda de aplicaciones.
- Vinculable: se puede compartir fácilmente a través de una URL y no requiere una instalación compleja.

## **2.5.2. Software, Protocolos y aplicaciones de comunicación.**

### **2.5.2.1. OpenSIPS.**

OpenSIPS es un servidor / proxy SIP de código abierto para voz, video, mensajería instantánea, presencia y cualquier otra extensión SIP.

OpenSIPS es un servidor SIP de señalización multifuncional y multipropósito utilizado por operadores, telecomunicaciones o ITSP para soluciones como plataformas residenciales de clase 4/5, servicios troncales / mayoristas, soluciones PBX empresariales / virtuales, controladores de borde de sesión, servidores de aplicaciones, carga de front-end Equilibradores, plataformas IMS, centros de llamadas y muchos otros.

OpenSIPS se recomienda para cualquier tipo de escenario / servicio SIP por: el alto rendimiento : decenas de miles de CPS, millones de llamadas simultáneas, la flexibilidad del enrutamiento y la integración : secuencia de comandos de enrutamiento para implementar lógicas de enrutamiento personalizadas, varias API de interfaz, la construcción de aplicaciones efectiva : más de 120 módulos para proporcionar funciones, para el manejo de SIP, para las operaciones de backend, para la integración, para las lógicas de enrutamiento.

### **Características de OpenSIPS:**

- Servidor de registro SIP (RFC3261) robusto y eficaz, servidor de ubicación, servidor proxy y servidor de redireccionamiento.
- Procesamiento de proxy SIP sin estado y transaccional con estado completo.
- Soporte para capas de transporte UDP / TCP / TLS / SCTP.
- IPv4 e IPv6.
- Soporte para SRV y NAPTR DNS.
- Conmutación por error de DNS SRV.
- Listas negras de IP.
- Compatibilidad con múltiples hogares (mhomed) y multidominio.
- Lenguaje de secuencias de comandos flexible y potente para la lógica de enrutamiento.
- Soporte de variables en el script: variables del script, pseudovariables (acceso a los mensajes SIP), AVP (valores persistentes por transacciones SIP).
- Autenticación, autorización y contabilidad (AAA) a través de base de datos (MySQL, Postgress, archivos de texto).
- SNMP: interfaz para el protocolo simple de administración de redes.

- Soporte transversal de NAT para tráfico SIP y RTP.
- Múltiples backends de bases de datos: MySQL, PostgreSQL, Oracle, Berkeley, archivos planos y otros tipos de bases de datos que tienen controladores unixodbc.

### Arquitectura de OpenSIPS

La arquitectura del servidor OpenSIPS se basa en:

- Nucleo pequeño y funcionalidades provistas por núcleos.
- Lenguaje de Scripting para la configuración.
- Información de gestión: Datagramas.

En la figura 8 podemos ver la arquitectura de OpenSIPS.

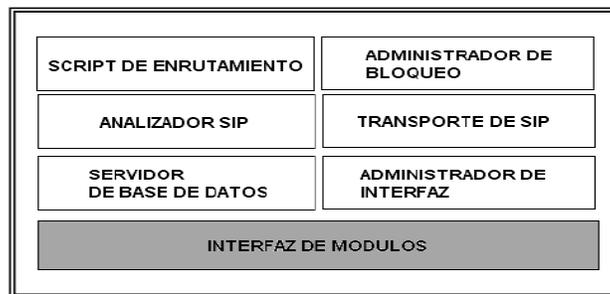


Figura 8. Contenido de la arquitectura del núcleo de OpenSIPS.

**Nota.** Tomado de OpenSIPS sitio web oficial

#### 2.5.2.2. Asterisk PBX.

Asterisk es un software de central telefónica con capacidad para voz sobre IP que es distribuido bajo licencia libre.

#### Características más relevantes

- Contestación Automática de Llamadas

- Transferencia de Llamadas
- Opción de No Molestar
- Parqueo de Llamadas
- Contestación de una llamada a una extensión remota
- Monitoreo y Grabación de Llamadas
- Voicemail
- Conferencias
- Reportación de Llamadas
- Colas de atención
- Llamada en espera
- Identificador de Llamante
- Bloqueo por llamante identificado
- Recepción de Fax
- Listado Interactivo del directorio de extensiones
- Interactive Voice Response (IVR)
- Música en espera
- Manejo de comportamiento por tiempo (Time Conditions)
- Follow me

### **Directorios de Asterisk**

Asterisk organiza sus archivos en algunos directorios. Entre los más importantes tenemos a los siguientes.

Tabla 3  
*Directorios importantes de Asterisk*

<b>Directorio</b>	<b>Descripción</b>
/etc/asterisk/	Aquí residen los archivos de configuración de asterisk
/usr/lib/asterisk/modules/	Este directorio contiene los módulos de Asterisk
/usr/sbin/	Aquí reside el binario de Asterisk
/var/log/asterisk/	Contiene los logs de Asterisk
/var/lib/asterisk/agi-bin/	Directorio donde residen los scripts AGI
/var/lib/asterisk/mohmp3 /var/lib/asterisk/sounds	Carpeta que contiene archivos para música en espera Sonidos que Asterisk utiliza como prompts de voz
/var/spool/asterisk/	Directorio donde Asterisk guarda archivos que genera producto de su funcionamiento como voicemails y grabaciones de llamadas
/var/run/	Archivos con información de PIDs
/var/log/asterisk/	Aquí residen los archivos de log de Asterisk como el /var/log/asterisk/full o el log de texto de CDRs

**Nota.** Tomado de Comunicaciones\_Unificadas\_con\_Elastix\_Volumen\_1.

A continuación, una figura un poco más ordenada donde se pueden observar los componentes de Elastix y su relación entre sí.

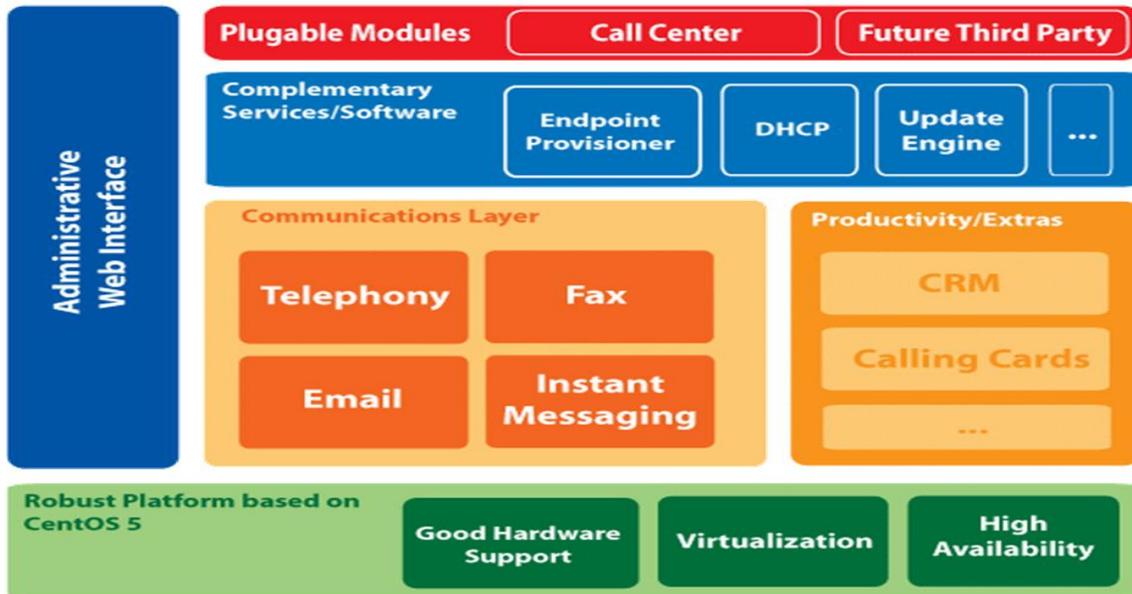


Figura 9. Esquema general de los componentes de Elastix.

**Nota.** Tomado de Comunicaciones\_Unificadas\_con\_Elastix\_Volumen\_1.

### 2.5.2.3. Protocolo SIP.

SIP es un protocolo de control de la capa de aplicación, para crear modificar y finalizar sesiones con uno o más participantes. Estas sesiones incluyen llamadas telefónicas por internet, distribución multimedia y conferencias multimedia.

SIP hace uso de elementos llamados servidores proxy para ayudar a enrutar las solicitudes a la ubicación actual del usuario, autenticar y autorizar los servicios a usuarios, proveedor de aplicaciones en las políticas de enrutamiento en las llamadas, y proporcionar características a los usuarios. SIP también proporcionan una función de registro que permite a los usuarios subir sus ubicaciones actuales para el uso del servidor proxy, SIP se ejecuta en la parte superior de varios protocolos de transporte.

### Descripción de la funcionalidad de SIP

SIP es un protocolo de control de la capa de aplicación que puede establecer, modificar y finalizar sesiones multimedia (conferencias), como llamadas de telefonía por Internet. SIP también puede invitar a los participantes a sesiones ya existentes, tales como conferencias de multidifusión. Medios de comunicación pueden ser añadidos (y retirados de) una sesión existente. SIP soporta de forma transparente la asignación de nombre y servicios de redirección, que apoya la movilidad personal (los usuarios pueden mantener un solo identificador visible externamente, independientemente de su ubicación de red).

SIP utiliza cinco facetas en la creación y la terminación de sesiones en comunicaciones multimedia:

- Ubicación del usuario: determinación de la terminal para ser utilizado para la comunicación;
- Disponibilidad del usuario: determinación de la voluntad de la terminal para participar en las comunicaciones;
- Capacidades del usuario: determinación de los parámetros de los medios de comunicación para ser utilizado;
- Configuración de la sesión: "Ringing", el establecimiento de parámetros de la sesión en ambas partes, llamada y llamando;
- La administración de sesiones: incluida la transferencia y la terminación de sesiones, la modificación de parámetros de la sesión, e invocando los servicios.

SIP funciona tanto con IPv4 como con IPv6.

#### **2.5.2.4. *Telegram Bots.***

Los bots de Telegram son una serie de aplicaciones de terceros que se ejecutan dentro de la aplicación de mensajería, para la programación de los bots, Telegram nos ofrece una interfaz, llamada Telegram Bot API. Esta es una interfaz basada en HTTPS creada para desarrolladores interesados en construir bots para Telegram. El API que nos ayudara a interactuar con los bots está diseñado para ocultar toda la sección del protocolo de cifrado de MTProto.

BotFather es el “padre de todos los bots”, y proporciona tokens únicos para cada uno. Con BotFather podremos crear nuevos bots o cambiar la configuración de los que ya existen, es decir, llevar a cabo su administración y gestión.

Los bots tienen características diferentes respecto al resto de usuarios, entre los cuales cabe destacar:

- No pueden iniciar conversaciones.
- Sus nombres tienen que acabar en ‘bot’.
- No tienen estado de conexión ni marcas de visto.
- Mensajes antiguos del bot pueden ser eliminados al poco tiempo.

Los bots procesan ordenes enviadas mediante un formato que se compone de ‘/comando [opcional1] [argumentoOpcional]’. Estos no podrán superar los 32 caracteres y pueden contener letras, numero o guiones bajos.

El nombre del bot se mostrará en los datos de contacto y en otros lugares, y dicho nombre será un nombre corto, que se utilizará en menciones y enlaces telegram.me.

Para crear un nuevo bot se utiliza el comando `‘/newbot’`, con el cual BotFather pedirá un nombre para dicho bot y él generará un Token de autorización, cadena que se requiere para autorizar el bot y enviar solicitudes a la API.

Comando que admite BotFather:

- `/token` – Genera un token de autorización.
- `/revoke` – Revoca el token de acceso a bot.
- `/setname` – Cambia el nombre del bot.
- `/setdescription` – Cambia la descripción del bot.
- `/setabouttext` – Cambia la información acerca del bot.
- `/setuserpic` – Cambia la foto del perfil del bot.
- `/setcommands` – Cambia la lista de comandos del bot.
- `/setjoingroups` – Agregar el bot a grupos.
- `/setprivacy` – Mensajes que puede ver el bot en un grupo.
- `/deletebot` – Elimina el bot.
- `/cancel` – Cancela la operación actual.

A continuación, en la figura se muestra la creación de un bot.

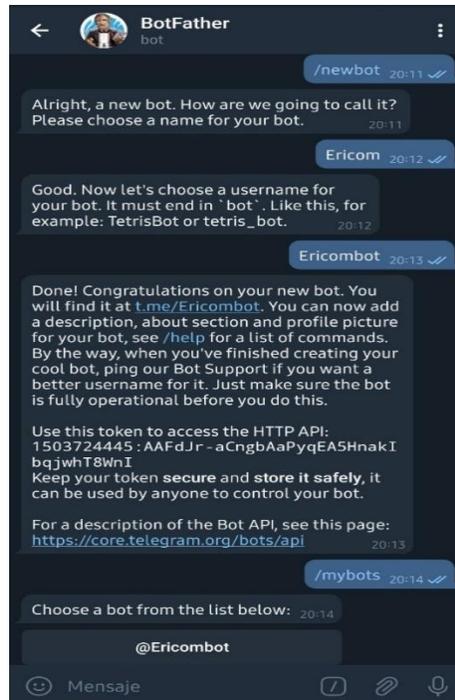


Figura 10. Creación de un bot con BotFather

**Nota.** Elaboración propia

## 2.6. Gestor de Base de datos

### 2.6.1. MySQL

MySQL es un sistema para la administración de base de datos relacionales rápido, sólido, fiable y fácil de usar. Utiliza SQL el lenguaje estandarizado para la consulta de base de datos más utilizado en todo el mundo. Es un servidor multiusuario y de subprocesamiento múltiple. Está disponible de manera gratuita, bajo una licencia de código abierto, o por un precio reducido en forma de licencia comercial (Welling & Thomson, 2005).

MySQL es considerada como la base de datos de código abierto más popular del mundo. Además, es usado por muchos sitios web grandes y populares como Wikipedia, Twitter, YouTube, etc.

## **2.7. Calidad de software**

En la industria del software se pueden evidenciar necesidades de satisfacción del cliente de productos o servicios de software, de reducción de recursos invertidos en proyectos de software y de la efectiva asignación de recursos humanos. Si hablamos de la calidad del software, tenemos varias definiciones.

Según Pressman (2007). La calidad de software es el cumplimiento de los requisitos de la funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados y las características implícitas que se esperan de todo el software de desarrollo profesionalmente.

La definición de la calidad del software según la IEEE, Std. 610-1990, es “el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

### **2.7.1. Norma ISO 9126**

ISO 9126 era un estándar internacional para la evaluación de la calidad del software. Fue reemplazado en 2005 por el conjunto de normas SQuaRE, ISO 25000:2014, la cual desarrolla los mismos conceptos.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.

- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

En la siguiente figura se muestra las características y la pregunta central que atiende cada una de estas.

<i>Características</i>	<i>Pregunta central</i>
<i>Funcionalidad</i>	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué . . . ?
<i>Confiabilidad</i>	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
<i>Usabilidad</i>	¿El software es fácil de usar y de aprender?
<i>Eficiencia</i>	¿Es rápido y minimalista en cuanto al uso de recursos?
<i>Mantenibilidad</i>	¿Es fácil de modificar y verificar?
<i>Portatilidad</i>	¿Es fácil de transferir de un ambiente a otro?

*Figura 11.* Características ISO 9126 y aspectos que atiende cada una.

**Nota.** Tomado de Abud, 2016

### **2.7.1.1 Funcionalidad.**

Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

- Adecuación - Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas especificadas.

- Exactitud - Atributos del software relacionados con la disposición de resultados o efectos correctos o acordados.
- Interoperabilidad - Atributos del software que se relacionan con su habilidad para la interacción con sistemas especificados.
- Seguridad - Atributos del software relacionados con su habilidad para prevenir acceso no autorizado ya sea accidental o deliberado, a programas y datos.
- Cumplimiento funcional.

#### **2.7.1.2. Confiabilidad.**

Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

- Madurez - Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.
- Recuperabilidad - Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- Tolerancia a fallos - Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.
- Cumplimiento de Fiabilidad - La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.

#### **2.7.1.3. Usabilidad.**

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- Aprendizaje- Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Comprensión - Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Operatividad - Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.
- Atractividad

#### **2.7.1.4. Eficiencia.**

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

- Comportamiento en el tiempo - Atributos del software que se relacionan con los tiempos de respuesta y procesamiento y en las tasas de rendimientos en desempeñar su función.
- Comportamiento de recursos - Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

#### **2.7.1.5. Mantenibilidad.**

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

- Estabilidad - Atributos del software relacionados con el riesgo de efectos inesperados por modificaciones.

- Facilidad de análisis - Atributos del software relacionados con el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallos, o identificaciones de partes a modificar.
- Facilidad de cambio - Atributos del software relacionados con el esfuerzo necesario para la modificación, corrección de falla, o cambio de ambiente.
- Facilidad de pruebas - Atributos del software relacionados con el esfuerzo necesario para validar el software modificado.

#### **2.7.1.6. Portabilidad.**

Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra.

- Capacidad de instalación - Atributos del software relacionados con el esfuerzo necesario para instalar el software en un ambiente especificado.
- Capacidad de reemplazamiento - Atributos del software relacionados con la oportunidad y esfuerzo de usar el software en lugar de otro software especificado en el ambiente de dicho software especificado.

## **2.8. Seguridad**

### **2.8.1. Firewall Linux (Shorewall)**

Shorewall es un lenguaje de alto nivel de propósito específico para manipular la infraestructura de control de paquetes del núcleo Linux, Netfilter.

Más específicamente, Shorewall es un script en lenguaje BASH (o en Perl en caso que deseemos utilizar shorewall-perl) que interpreta una serie de archivos de configuración a partir

de los cuales hace sucesivas llamadas a iptables para definir el conjunto de reglas necesarias representadas por la configuración.

Además de iptables, Shorewall también utiliza otras herramientas para controlar otros módulos de red núcleo Linux como modprobe (para cargar los módulos de Netfilter), iproute (para la definición de reglas de ruteo) y tc (para el control de tráfico de paquetes).

### **2.8.2. Características funcionales de Shorewall**

Permite el filtrado de paquetes stateful gracias a las capacidades de monitoreo de conexiones de Netfilter.

- Puede utilizarse en múltiples aplicaciones de router (enrutador), firewall (corta-fuegos) o gateway (pasarela).
- Completamente configurable mediante el uso de archivos de configuración.
- Número de interfaces de red ilimitadas.
- Permite dividir las redes en zonas, permitiendo extenso control sobre las conexiones establecidas entre ellas.
- Múltiples interfaces por zona y múltiples zonas por interfaz.
- Soporta zonas anidadas y solapadas.

### **2.8.3. ¿Cuándo conviene usar Shorewall?**

A juzgar por la sección anterior, debería ser bastante claro que Shorewall es un firewall extremadamente flexible y configurable. Hasta ahora no he encontrado ninguna situación en donde Shorewall no pudiera adecuarse de manera fácil y rápida.

A pesar de esto, creo que Shorewall puede ser demasiado cuando uno solo quiere asegurar una estación de trabajo conectada a Internet. En estos casos yo generalmente

recomiendo FirewallBuilder, FireStarter o herramientas similares (o las herramientas provistas por la distro como system-config-firewalld de Fedora).

Shorewall realmente se destaca cuando queremos conectar dos zonas (una red hogareña contra la Internet) o más, siendo cada vez más conveniente a medida que la situación se torna cada vez más sofisticada, por ejemplo, conectar múltiples ISP con control de tráfico, DMZ y zonas internas distinguidas.

Ahora bien, habiendo tantas herramientas de gestión de firewalls disponibles, ¿Cuándo conviene construir un firewall utilizando iptables desde shell scripts?

La respuesta, yo creo, es bastante sencilla: Cuando queremos aprender el funcionamiento interno de Netfilter. Una vez que uno domina esta técnica es cuando realmente puede apreciar una herramienta como Shorewall.

## CAPITULO 3

### MARCO APLICATIVO

#### 3.1. Introducción

Uno de los aspectos más importantes en esta etapa es tener la base sólida del problema para luego asumir la fase de construcción de la plataforma con las herramientas elegidas para tal objetivo.

En este capítulo se describe la planificación y el desarrollo de la plataforma multicanal en línea para la gestión y seguimiento de tickets para empresas en telecomunicaciones, desarrollando un sistema aplicando la metodología Scrum y complementando con la metodología de ingeniería Web U.W.E. basada en U.M.L. (Lenguaje Unificado de Modelado).

#### 3.2. Identificación de roles Scrum

En la siguiente tabla se tiene una lista de las personas implicadas e identificadas con sus roles Scrum (Ver acápite 2.2.2.2) para el desarrollo de proyecto:

Tabla 4  
*Identificación de Roles Scrum*

ROL		NOMBRE
Propietario del Producto		Erick Aguilar Lozano
Scrum Master		Erick Aguilar Lozano
Equipo Scrum	Analista Diseñador Desarrollador Testeador	Erick Aguilar Lozano

**Nota.** Elaboración propia.

### 3.3. Análisis de Requerimientos

El análisis de requerimientos permite definir las funciones y el rendimiento del software y determina las restricciones que debe cumplir el mismo, además determina la interfaz que tendrá el software con otros elementos del sistema, cada uno de los requerimientos que se puedan identificar en el proceso de análisis ayudará a la realización del diseño del sistema.

En la Figura 12 se muestra la arquitectura de la plataforma.

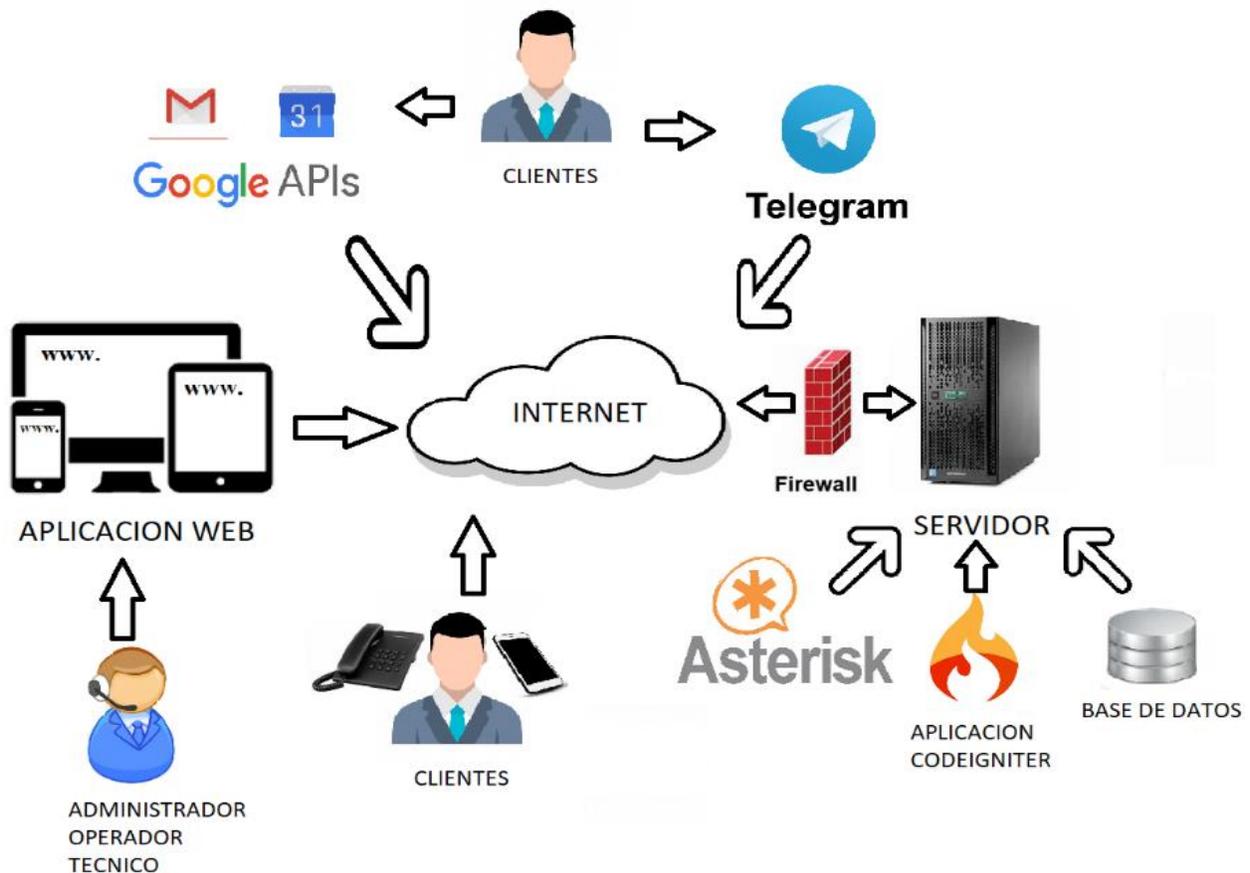


Figura 12. Arquitectura de la Plataforma.

**Nota.** Elaboración propia.

Como consecuencia del análisis de la situación actual y luego de recolectar toda la información definimos los requisitos del sistema.

- Requisitos globales
- Requisitos elementales
- Requisitos funcionales
- Requisitos de navegación

### **3.3.1. Requisitos Globales**

El fin principal de esta tesis es desarrollar una plataforma multicanal en línea que gestione y realice el seguimiento de Tickets a empresas en telecomunicaciones, debido a esto se tiene que contar con un dominio en internet el cual permitirá brindar el servicio a empresas y acortar la distancia entre ellas y sus clientes.

### **3.3.2. Requisitos Elementales**

Existen 5 tipos fuentes que generan y reciben información ambiente actual entorno a la plataforma.

- Administrador
- Empresas
- Operadores
- Técnicos
- Clientes

De acuerdo al análisis realizado logramos establecer las necesidades de los usuarios mencionados a continuación detallamos dichas necesidades.

#### ***Administrador***

La información que requiere este administrador servirá para poder gestionar y administrar los canales de atención, para cada empresa, así también realizar toda la gestión y

administración de los usuarios de empresas y sus respectivos sub usuarios como operadores, técnicos y clientes. Previo a listar los requerimientos definiremos lo que significa un canal de atención en el contexto de la plataforma.

- **Canal (Multicanal)**

Se trata de todas las herramientas de gestión y relación con los usuarios que permiten interactuar mediante contactos e información entrantes o salientes por alguno de los canales de comunicación que establece la empresa con sus clientes. Estas herramientas son: correo electrónico, telegram, llamadas telefónicas.

Los requerimientos son:

- Información sobre la configuración e instalación de cada canal de comunicación.
- Administración de Canales.
- Información y administración de Usuarios.
- Información y administración de Tickets.
- Información y administración de Clientes.
- Información y administración de Fallas.
- Información y administración de Empresas.
- Información y administración de Usuarios.
- Información y administración de Técnicos.
- Información y administración de Permisos de usuario.

### ***Empresas***

La información que requiere este administrador es de vital importancia para la toma de decisiones, para obtener reportes de altas, bajas y continuidad de los servicios que brinda su empresa. Los requerimientos son:

- Información de cantidad de Tickets de reclamos por servicios prestados.
- Información de seguimiento y soporte a los Tickets.
- Información de técnicos disponibles y ocupados.
- Información de Operadores.
- Información de las Fallas de la empresa.
- Información de los Canales activos en la empresa.
- Información actualizada sobre las quejas y reclamos de los clientes.
- Información y administración de Usuarios.
- Información y administración de Permisos.

### ***Operadores***

Los requerimientos son:

- Administración e información de cada Ticket.
- Asignación de Técnicos a Tickets.
- Administración de Técnicos.

### ***Técnicos***

Los requerimientos son:

- Administración e información de Tickets.
- Información sobre el seguimiento y solución de los Tickets.

### ***Clientes***

Los requerimientos son:

- Información sobre el estado actual del Ticket.
- Varios canales para aperturar un Ticket.

### **3.3.3. Requisitos Funcionales**

De acuerdo a la estructura organizacional de cualquier departamento que brinda soporte técnico y atención al cliente de una empresa en telecomunicaciones, se identificaron los roles de usuario para la plataforma con los que se controla el acceso a ciertas partes del sistema y ciertas acciones.

- Administrador
- Empresas
- Operadores
- Técnicos
- Clientes

El ingreso a la plataforma es a través de un módulo de identificación, cada usuario tiene acceso a ciertas áreas del sistema, las cuales están diseñadas para cada uno de ellos de acuerdo a sus requerimientos, sin embargo cabe mencionar que de variar las necesidades o requerimientos de dichos usuarios dependiendo de la estructura organizacional de cada empresa en telecomunicaciones existirá un módulo de permisos el cual solo podrá ser administrado por el usuario Administrador o Empresa pudiendo este otorgar acceso a otras funcionalidades del sistema a los usuarios.

- **Administrador**

Este usuario tiene acceso a todas las funcionalidades de la plataforma, haciendo énfasis en la administración del módulo de permisos anteriormente descrito.

- **Empresas**

Este usuario tiene acceso a todas las funcionalidades del sistema relacionadas con su empresa.

- **Operadores**

Este usuario puede realizar la gestión de los tickets, asignar técnicos a tickets, verificar toda la información concerniente a la apertura, seguimiento y solución del ticket.

- **Técnicos**

Este usuario puede realizar la gestión de tickets, verificar la apertura, seguimiento y solución que se da a los tickets.

- **Clientes**

Son usuarios los cuales tienen 4 formas de aperturar un ticket:

- **Mediante un Bot de Telegram.** – El cliente podrá aperturar un ticket escribiendo al Bot asociado a la empresa, ingresando su CI por primera vez para ser identificado y posteriormente aperturarlo.
- **Mediante correo electrónico.** – El cliente podrá aperturar un ticket desde su cuenta de correo electrónico personal, enviando previamente un mensaje con la palabra TICKET al correo asociado a la empresa, para ser identificado y posteriormente aperturar un ticket.
- **Mediante llamada telefónica e interacción con un IVR (menú interactivo de voz).**
  - El cliente podrá aperturar un ticket llamando al número asociado a la empresa, este

será atendido por una contestadora de voz (IVR) que le proporcionará un menú de las opciones disponibles.

- **Mediante llamada telefónica con un operador derivado desde el IVR.** – La última opción del IVR será ser derivado a un operador de la empresa, el cual realizará la apertura del ticket de manera convencional.

#### **3.3.4. Requisitos de Navegación**

Una vez definidos los usuarios del sistema, consideremos los siguientes puntos para la navegación.

- La plataforma maneja un sistema abierto, es decir que considera adicionar otro usuario o actor no definido anteriormente para adecuarse a las necesidades de cualquier empresa en telecomunicaciones.
- Existen módulos para cada tipo de usuario o actor.
- Está definido un módulo de permisos únicamente para el usuario administrador y empresa.
- Está definido un módulo exclusivo para los usuarios.
- Existen módulos comunes para la administración de los tickets dependiendo del tipo de usuario.
- Existe un módulo para que la empresa pueda migrar la lista de sus clientes.
- Existe un módulo para la administración de las fallas comunes de la empresa solamente para el administrador de la empresa.
- Existe un módulo para habilitar y deshabilitar los canales de atención.

### **3.3.5. Modelado de casos de uso**

Continuando el proceso de desarrollo de la ingeniería web de la metodología UWE, se propone los siguientes diagramas de casos de uso.

En la Figura 13 se puede ver el diagrama de Caso de Uso de Alto Nivel del sistema a desarrollar:

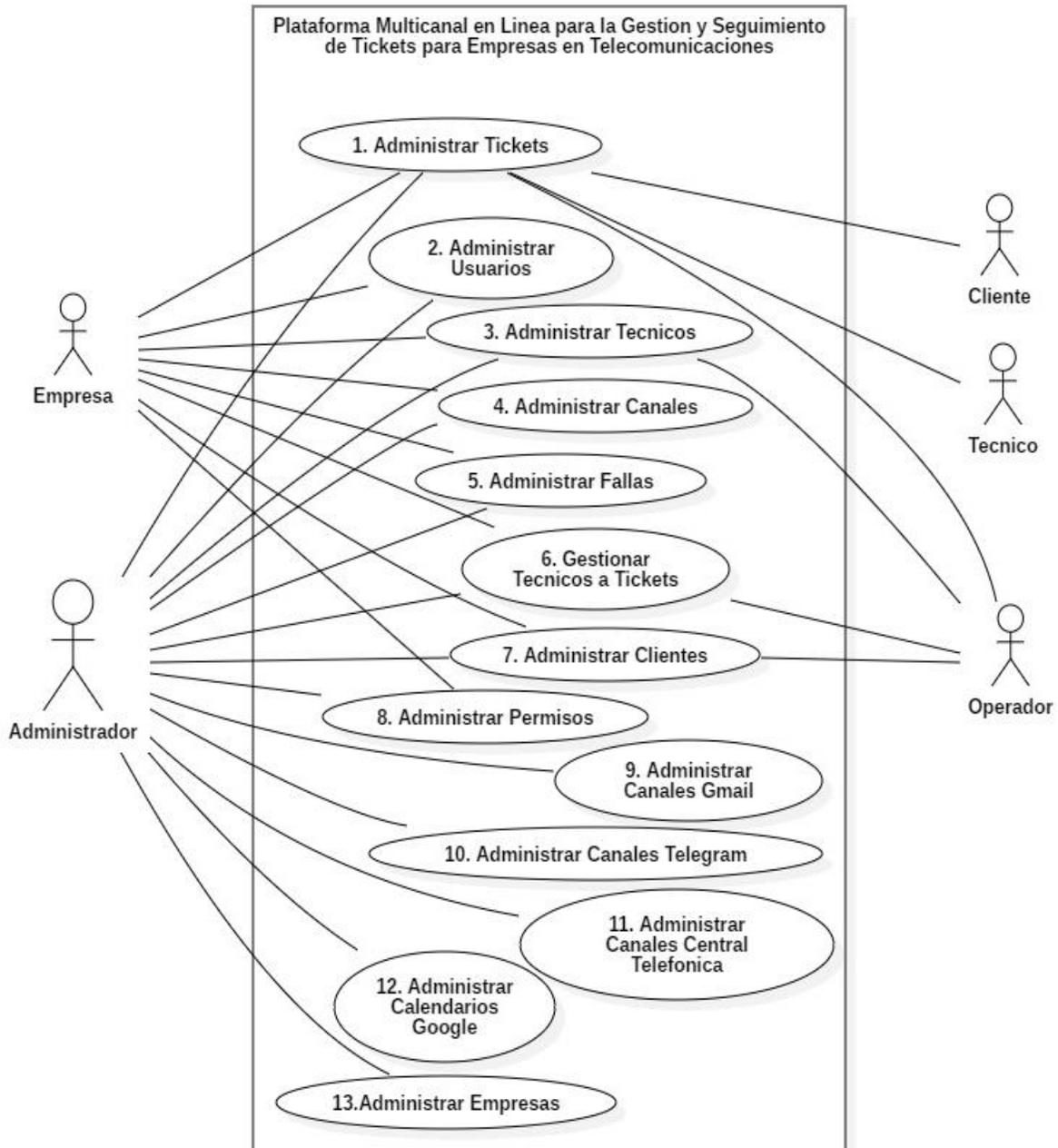


Figura 13. Diagrama de Caso de Uso de Alto Nivel del Sistema.

Se muestra el caso de uso para la Administración de Tickets en la Figura 14 y en la Tabla 5 sus especificaciones:

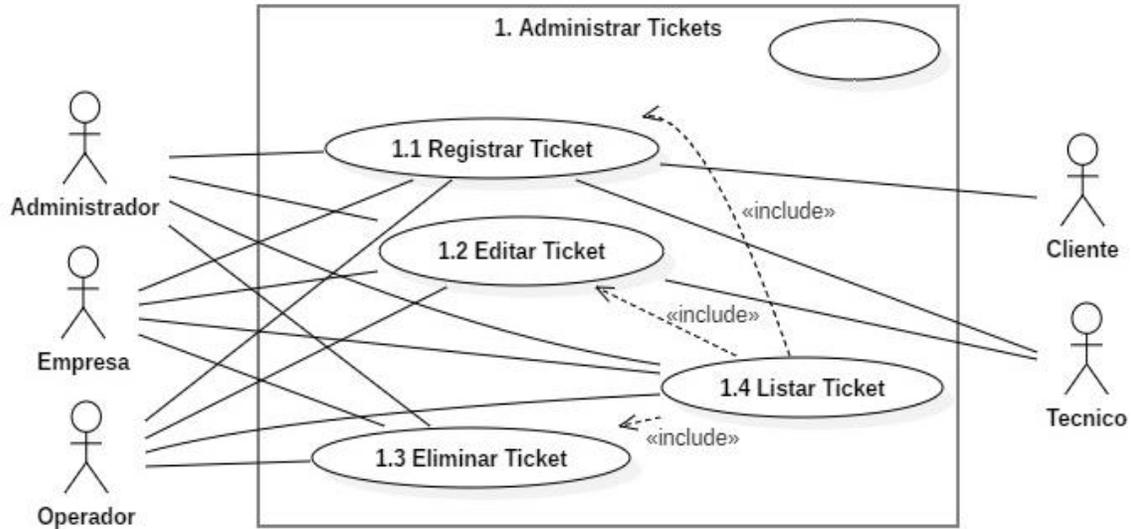


Figura 14. Caso de Uso "Administrar Tickets".

Tabla 5

Descripción del Caso de Uso "Administrar Tickets"

<b>Caso de Uso:</b> Administrar Tickets.
<b>Actores:</b> Administrador, Empresa, Operador, Técnico, Cliente.
<b>Descripción:</b> Permite registrar, editar o eliminar tickets.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador, Empresa, Operador, Técnico, en el caso del actor Cliente este debe pertenecer a una empresa.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador, Empresa u Operador.
2. El actor ingresa a la opción de tickets.
3. El sistema despliega los tickets registrados, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
1A. El actor ingresa al sistema como Técnico.
2A. El actor ingresa a la opción de tickets.
3A. El sistema despliega los tickets registrados, junto con las opciones de registrar o editar.
4A. El actor acciona una de las opciones.
1B. El actor escoge un canal para la apertura de un ticket y se identifica.
2B. El actor recibe la lista de fallas comunes de la empresa.
3B. El actor responde con el problema que refiere.
4B. El sistema apertura el ticket.
5B. El sistema responde ticket aperturado con éxito.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.

**Nota.** Elaboración Propia.

A continuación, se muestra en la Figura 15 el diagrama de caso de uso para la Administración de Usuarios junto con sus especificaciones en la Tabla 6:

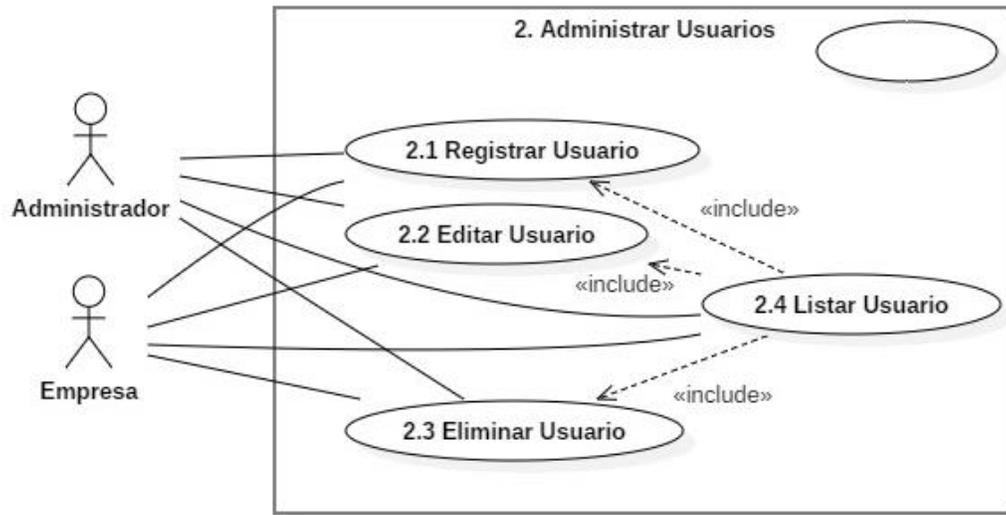


Figura 15. Caso de Uso "Administrar Usuarios".

Tabla 6

Descripción del Caso de Uso "Administrar Usuarios"

<b>Caso de Uso:</b> Administrar Usuarios.
<b>Actores:</b> Administrador, Empresa.
<b>Descripción:</b> Permite registrar, editar o eliminar usuarios.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador o Empresa.
<b>Flujo Normal:</b> 1. El actor ingresa al sistema como Administrador o Empresa. 2. El actor ingresa a la opción de usuarios. 3. El sistema despliega los usuarios registrados, junto con las opciones de registrar, editar o eliminar. 4. El actor acciona una de las opciones.
<b>Postcondiciones:</b> El sistema ejecuta la opción elegida y muestra los cambios realizados.

**Nota.** Elaboración Propia.

En la Figura 16 se puede apreciar el diagrama de caso de uso para la Administración de Técnicos y en la Tabla 7 están sus especificaciones:

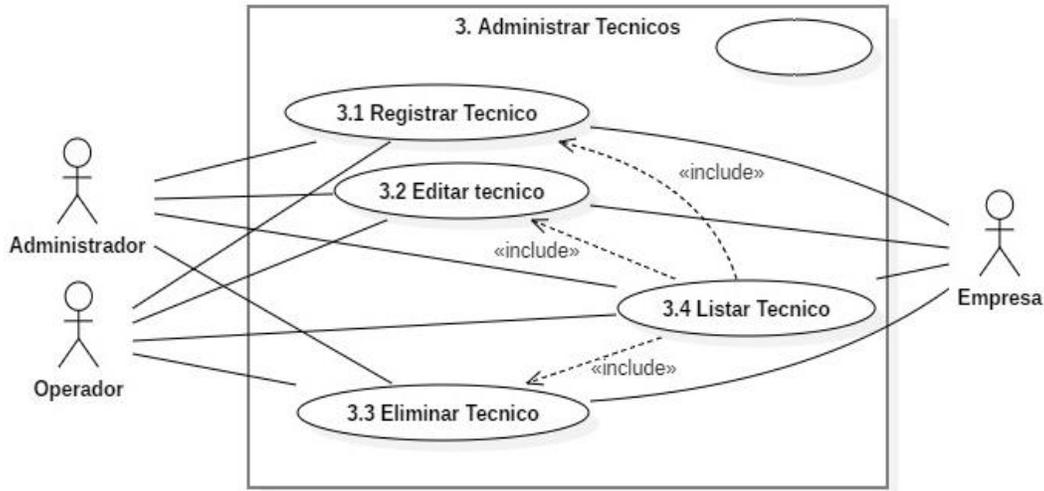


Figura 16. Caso de Uso "Administrar Técnicos".

Tabla 7

Descripción del Caso de Uso "Administrar Técnicos"

<b>Caso de Uso:</b> Administrar Técnicos.
<b>Actores:</b> Administrador, Empresa, Operador.
<b>Descripción:</b> Permite registrar, editar o eliminar técnicos.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador, Empresa u Operador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador, Empresa u Operador.
2. El actor ingresa a la opción de técnicos.
3. El sistema despliega los técnicos registrados, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 17 se muestra el diagrama de caso de uso para la Administración de Canales junto a sus especificaciones en la Tabla 8:

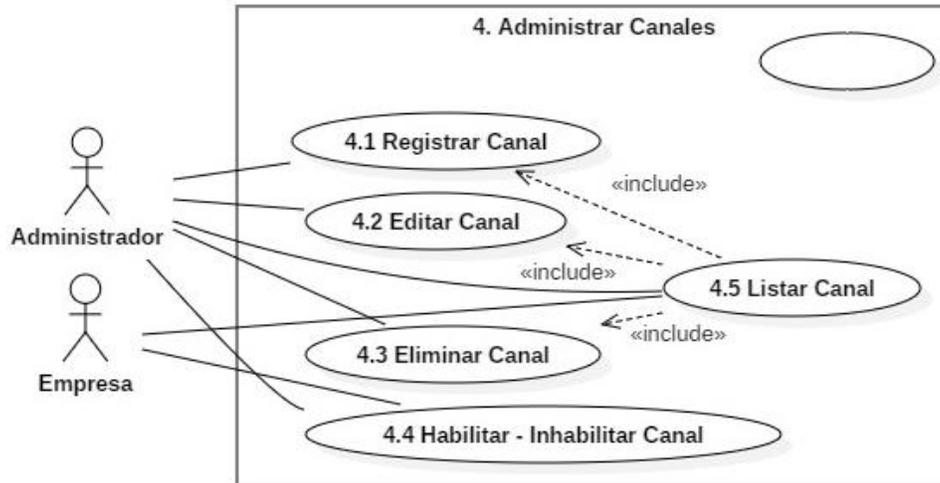


Figura 17. Caso de Uso "Administrar Canales".

Tabla 8

Descripción del Caso de Uso "Administrar Canales"

<b>Caso de Uso:</b> Administrar Canales.
<b>Actores:</b> Administrador, Empresa.
<b>Descripción:</b> Permite registrar, editar, eliminar, habilitar o inhabilitar canales.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador o Empresa.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador.
2. El actor ingresa a la opción de canales.
3. El sistema despliega los canales registrados, junto con la opción de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
1A. El actor ingresa al sistema como Empresa.
2A. El actor ingresa a la opción de administración de canales.
3A. El sistema despliega los canales registrados, junto con la opción de habilitar o inhabilitar el canal.
4A. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 18 se muestra el diagrama de caso de uso para la Administración de Fallas y en la Tabla 9 sus especificaciones:

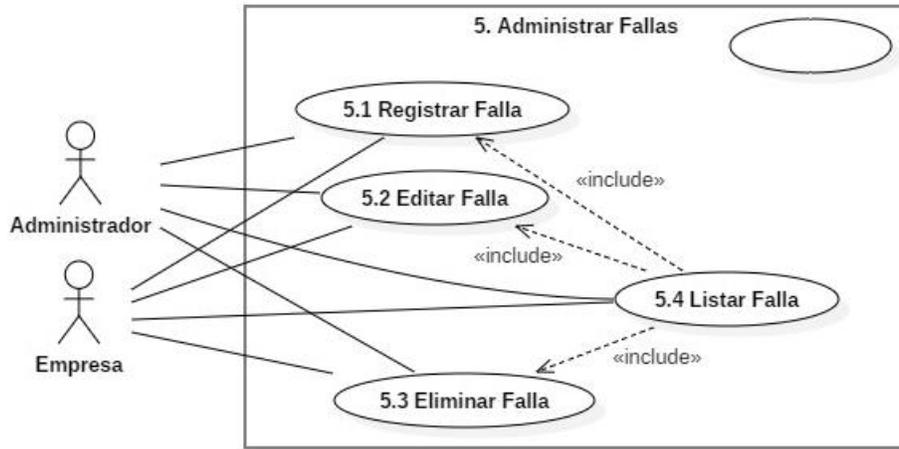


Figura 18. Caso de Uso "Administrar Fallas".

Tabla 9

Descripción del Caso de Uso "Administrar Fallas"

<b>Caso de Uso:</b> Administrar Fallas.
<b>Actores:</b> Administrador, Empresa.
<b>Descripción:</b> Permite registrar, editar o eliminar fallas.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador o Empresa.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador o Empresa.
2. El actor ingresa a la opción de fallas.
3. El sistema despliega las fallas registradas, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 19 se muestra el diagrama de caso de uso para la Gestión de Técnicos a Tickets junto a sus especificaciones en la Tabla 10:

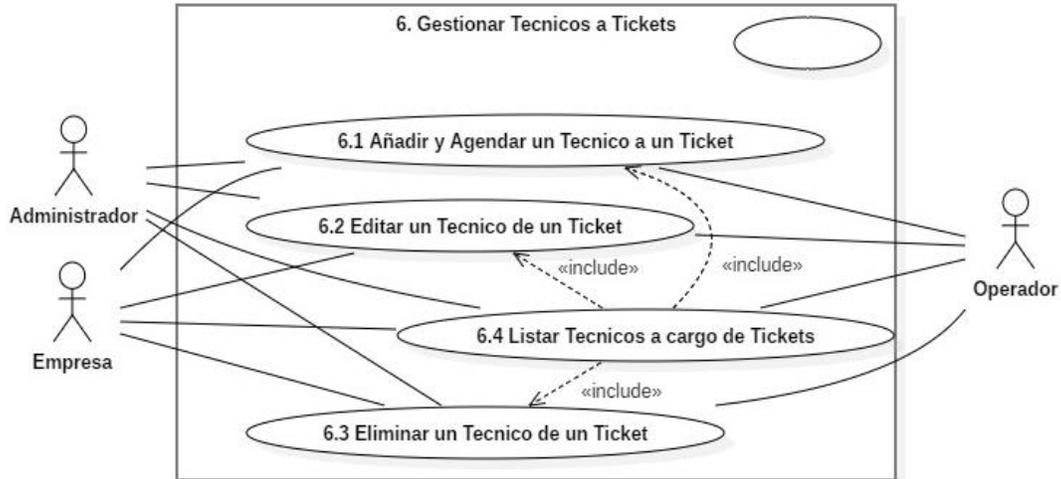


Figura 19. Caso de Uso "Gestionar Técnicos a Tickets".

Tabla 10

Descripción del Caso de Uso " Gestionar Técnicos a Tickets "

<b>Caso de Uso:</b> Gestionar Técnicos a Tickets.
<b>Actores:</b> Administrador, Empresa, Operador.
<b>Descripción:</b> Permite agendar, editar o eliminar técnicos a tickets.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador, Empresa u Operador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador, Empresa u Operador.
2. El actor ingresa a la opción de tickets.
3. El sistema despliega los tickets registrados, junto con las opciones de agendar, editar o eliminar técnicos a cada ticket.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 20 se muestra el diagrama de caso de uso para la Administración de Clientes junto a sus especificaciones en la Tabla 11:

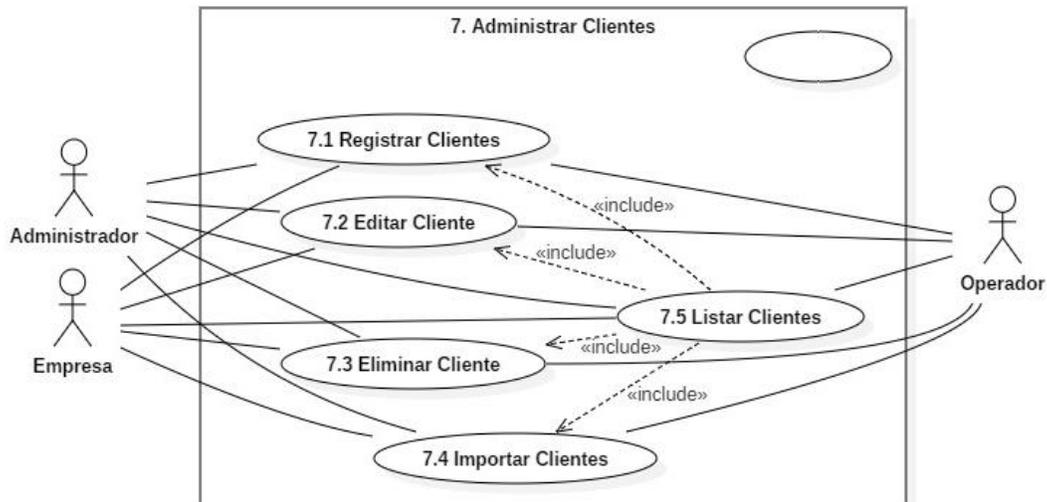


Figura 20. Caso de Uso "Administrar Clientes".

Tabla 11

Descripción del Caso de Uso "Administrar Clientes"

<b>Caso de Uso:</b> Administrar Clientes.
<b>Actores:</b> Administrador, Empresa, Operador.
<b>Descripción:</b> Permite registrar, editar, eliminar o importar clientes.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador, Empresa u Operador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador, Empresa u Operador.
2. El actor ingresa a la opción de clientes.
3. El sistema despliega los clientes registrados, junto con las opciones de registrar, editar, eliminar o importar una lista de clientes desde un Excel.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 21 se muestra el diagrama de caso de uso para la Administración de Permisos junto a sus especificaciones en la Tabla 12:

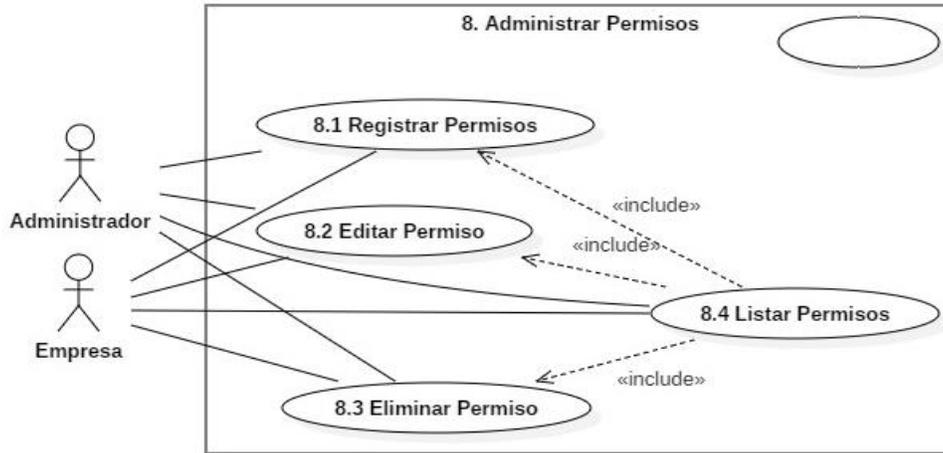


Figura 21. Caso de Uso "Administrar Permisos".

Tabla 12

Descripción del Caso de Uso "Administrar Permisos"

<b>Caso de Uso:</b> Administrar Permisos.
<b>Actores:</b> Administrador, Empresa.
<b>Descripción:</b> Permite registrar, editar, eliminar permisos.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador o Empresa.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador o Empresa.
2. El actor ingresa a la opción de permisos.
3. El sistema despliega los permisos registrados, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 22 se muestra el diagrama de caso de uso para la Administración de Canales Gmail junto a sus especificaciones en la Tabla 13:

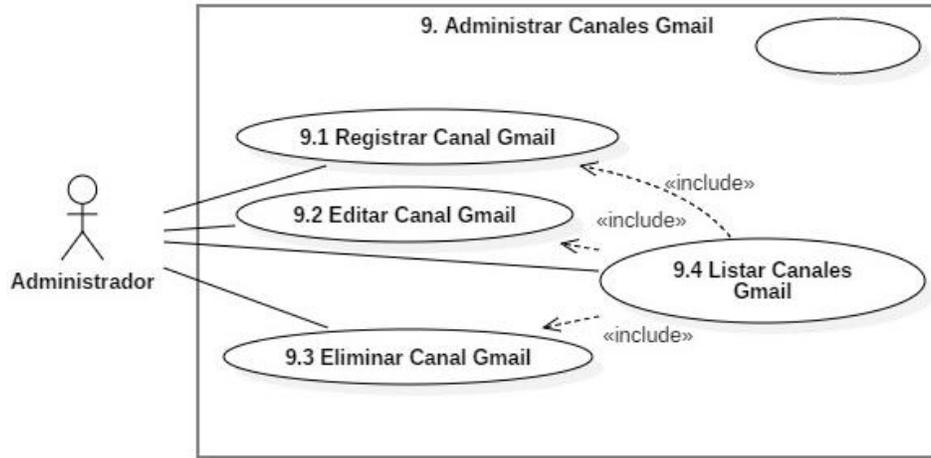


Figura 22. Caso de Uso "Administrar Canales Gmail".

Tabla 13

Descripción del Caso de Uso "Administrar Canales Gmail "

<b>Caso de Uso:</b> Administrar Canales Gmail.
<b>Actores:</b> Administrador.
<b>Descripción:</b> Permite registrar, editar o eliminar canales de Gmail.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador.
2. El actor ingresa a la opción de canales - Gmail.
3. El sistema despliega los correos de Gmail registrados por empresa, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.

**Nota.** Elaboración Propia.

En la Figura 23 se muestra el diagrama de caso de uso para la Administración de Canales Telegram junto a sus especificaciones en la Tabla 14:

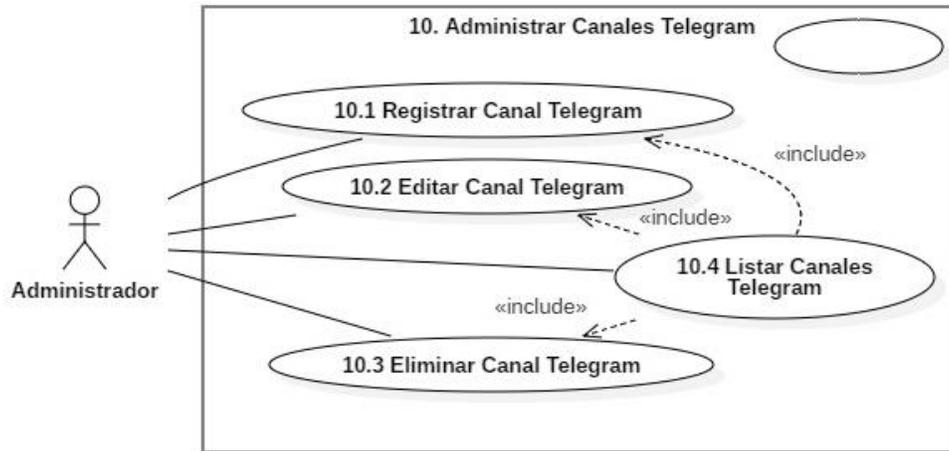


Figura 23. Caso de Uso "Administrar Canales Telegram".

Tabla 14

Descripción del Caso de Uso "Administrar Canales Telegram"

<b>Caso de Uso:</b> Administrar Canales Telegram.
<b>Actores:</b> Administrador.
<b>Descripción:</b> Permite registrar, editar o eliminar canales de telegram.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador.
2. El actor ingresa a la opción de canales - telegram.
3. El sistema despliega los canales de telegram registrados por empresa, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 24 se muestra el diagrama de caso de uso para la Administración de Canales Central telefónica junto a sus especificaciones en la Tabla 15:

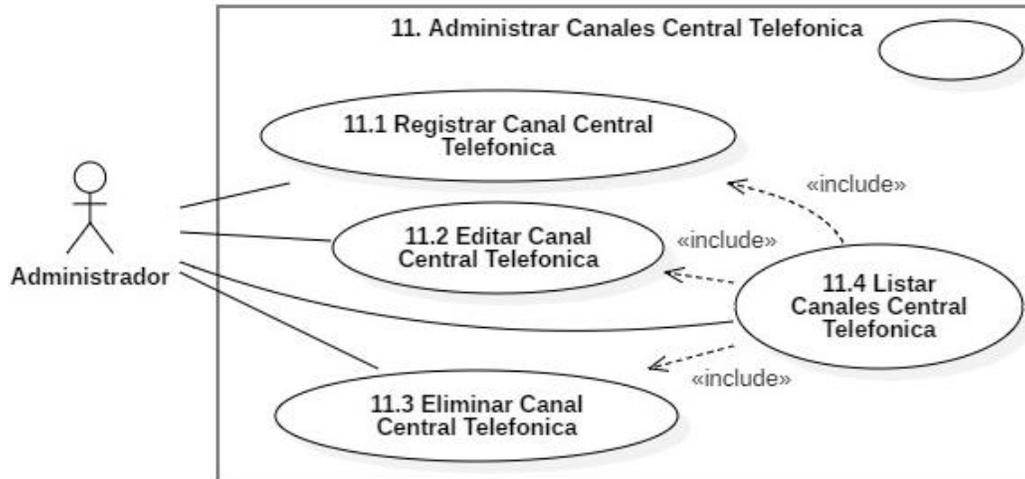


Figura 24. Caso de Uso "Administrar Canales Central Telefónica".

Tabla 15

Descripción del Caso de Uso "Administrar Canales Central Telefónica"

<b>Caso de Uso:</b> Administrar Canales Central Telefónica.
<b>Actores:</b> Administrador.
<b>Descripción:</b> Permite registrar, editar o eliminar canales en la central telefónica.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador.
2. El actor ingresa a la opción de canales – central telefónica.
3. El sistema despliega los canales de la central telefónica registrados por empresa, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

En la Figura 25 se muestra el diagrama de caso de uso para la Administración de Calendarios Google junto a sus especificaciones en la Tabla 16:

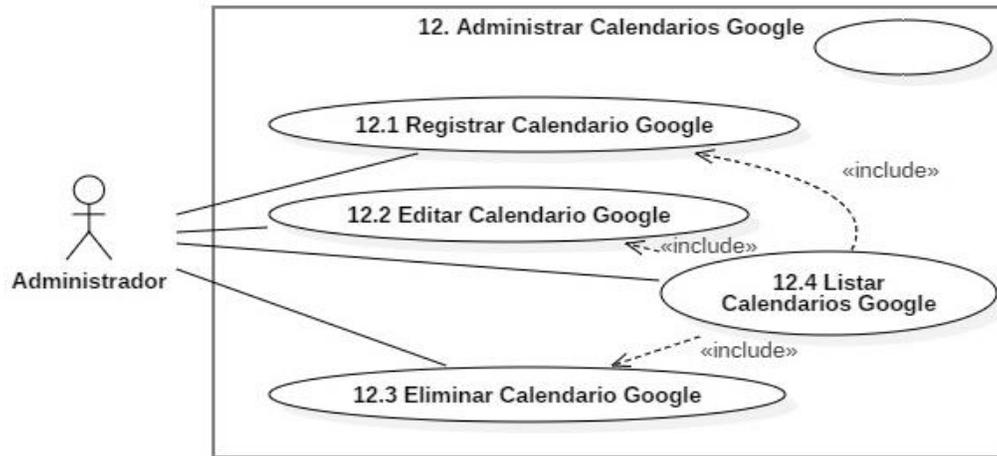


Figura 25. Caso de Uso "Administrar Calendarios Google".

Tabla 16

Descripción del Caso de Uso "Administrar Calendarios Google "

<b>Caso de Uso:</b> Administrar Calendarios Google.
<b>Actores:</b> Administrador.
<b>Descripción:</b> Permite registrar, editar o eliminar cuentas de Google calendar.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador.
2. El actor ingresa a la opción de canales – Calendarios Google.
3. El sistema despliega las cuentas de Google Calendar registrados por empresa, junto con las opciones de registrar, editar o eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.

**Nota.** Elaboración Propia.

En la Figura 26 se muestra el diagrama de caso de uso para la Administración de Empresas junto a sus especificaciones en la Tabla 17:

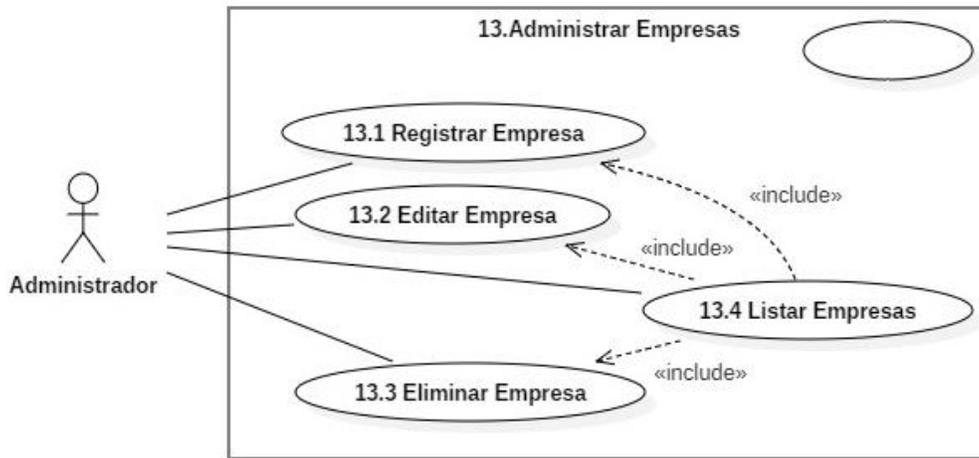


Figura 26. Caso de Uso "Administrar Empresas".

Tabla 17

Descripción del Caso de Uso "Administrar Empresas "

<b>Caso de Uso:</b> Administrar Empresas.
<b>Actores:</b> Administrador.
<b>Descripción:</b> Permite registrar, editar o eliminar empresas.
<b>Precondiciones:</b> El actor debe ser un usuario registrado y tener el rol de Administrador.
<b>Flujo Normal:</b>
1. El actor ingresa al sistema como Administrador.
2. El actor ingresa a la opción de empresas.
3. El sistema despliega las empresas registradas, junto con las opciones de registrar, editar, eliminar.
4. El actor acciona una de las opciones.
<b>Postcondiciones:</b>
El sistema ejecuta la opción elegida y muestra los cambios realizados.
<b>Nota.</b> Elaboración Propia.

### 3.4. Sprint 0 – Preparación del proyecto

En base al análisis plasmado en los modelos de casos de uso y al análisis de requerimientos, a continuación, se muestra en la Tabla 18 una lista de actividades priorizadas, siendo estas actividades los elementos de la pila del producto tomando las estimaciones de demora en horas de trabajo.

Tabla 18  
*Pila del producto*

<b>Id</b>	<b>Actividad</b>	<b>Estimación [Horas]</b>	<b>Prioridad</b>
1	Administrar tickets	120	Alta
2	Administrar usuarios	40	Alta
3	Administrar técnicos	24	Alta
4	Administrar canales	72	Alta
5	Administrar fallas	48	Alta
6	Gestionar técnicos a tickets	40	Media
7	Administrar clientes	72	Alta
8	Administrar permisos	48	Alta
9	Administrar canales Gmail	72	Alta
10	Administrar canales telegram	48	Alta
11	Administrar canales central telefónica	72	Alta
12	Administrar calendarios Google	72	Media
13	Administrar empresas	24	Alta

**Nota.** Elaboración Propia.

En base a la pila del producto, se propone los modelos entidad relación y modelo relacional para generar la estructura de la base de datos.

### 3.4.1. Modelo Entidad-Relación

A continuación, se muestra el modelo Entidad Relación de la base de datos del sistema.

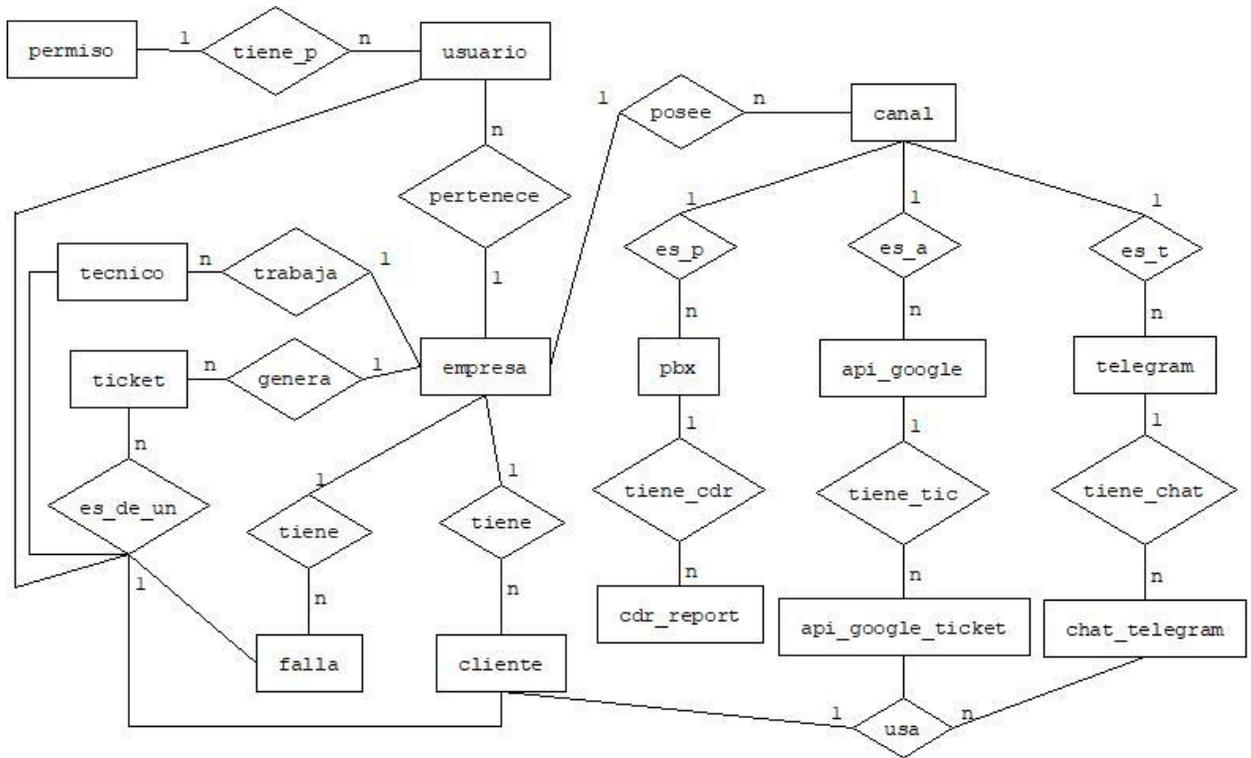


Figura 27. Modelo Entidad Relación de la Base de Datos.

### 3.4.2. Modelo Relacional

En la Figura 28 se puede ver el modelo relacional de la base de datos del Sistema.

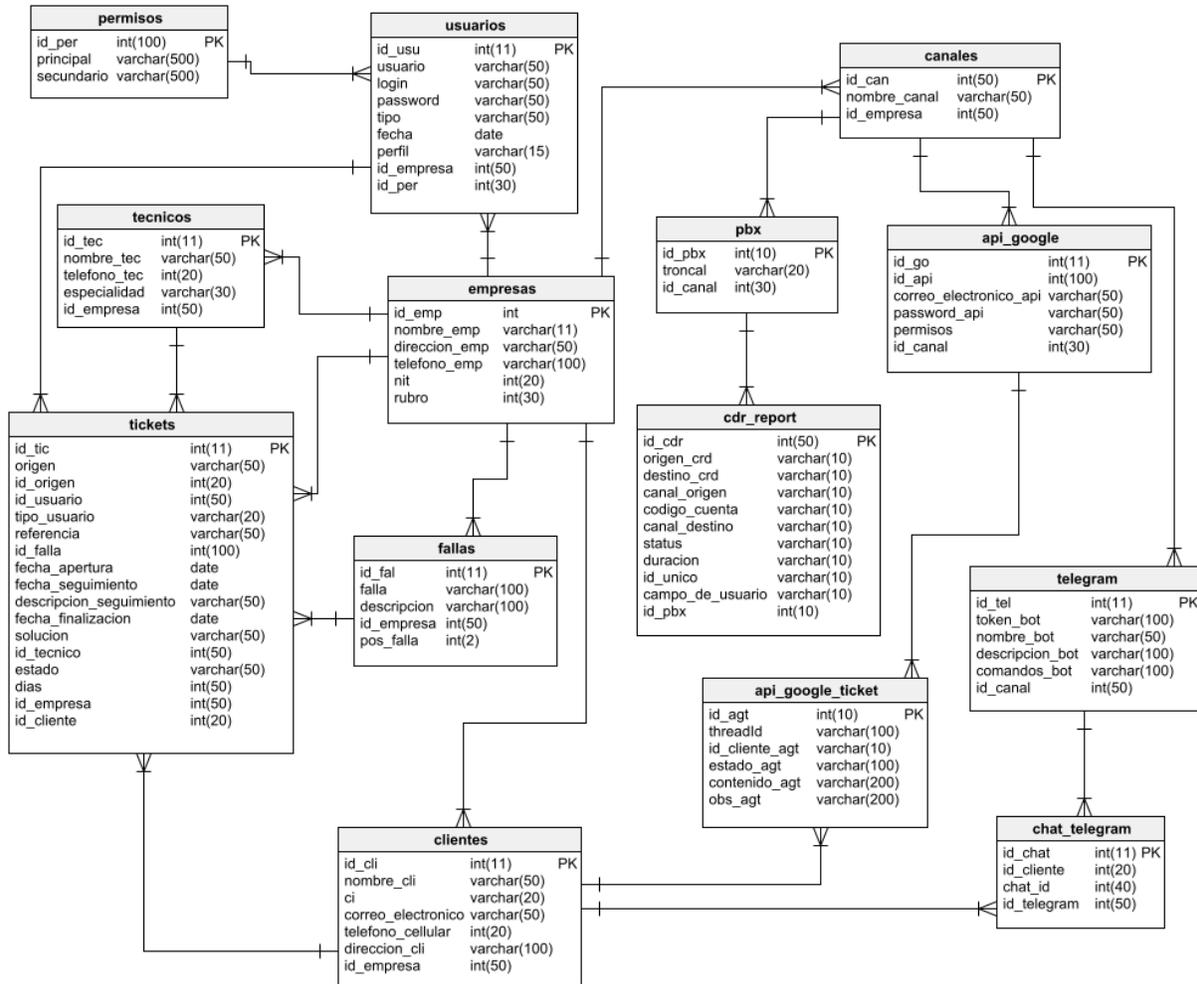


Figura 28. Modelo Relacional de la Base de Datos del Sistema.

### 3.4.3. Modelo Conceptual

En la Figura 29 se puede observar el modelo conceptual del sistema.

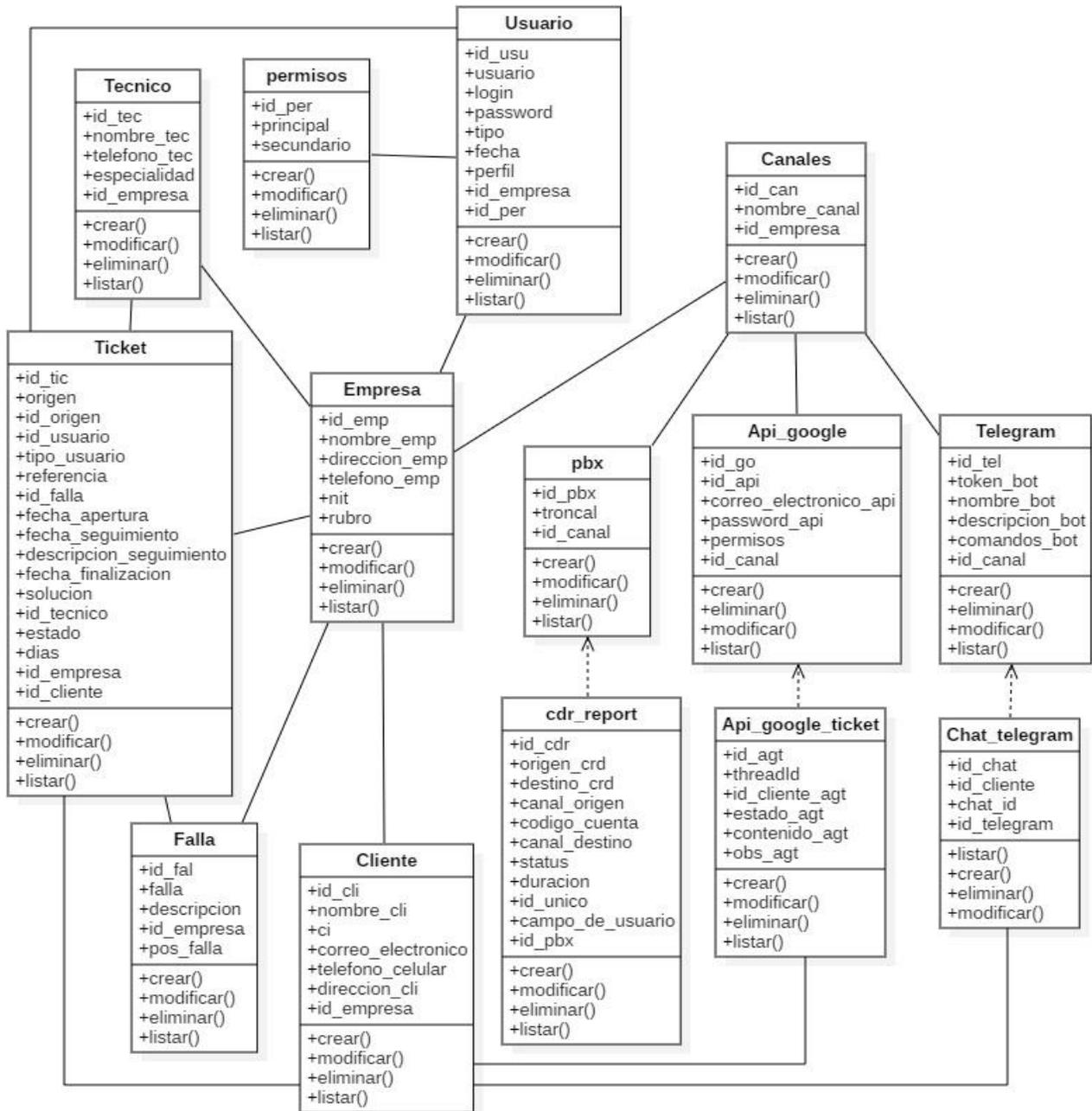


Figura 29. Modelo Conceptual en Diagrama de Clases para el Sistema.

### 3.5. Desarrollo de los Sprints

En esta fase, se implementa el sistema en 3 sprints de aproximadamente 30 días de duración cada uno. De cada sprint se obtiene los modelos de navegación, presentación y una entrega funcional del sistema.

#### 3.5.1. Sprint 1

En la Tabla 19 se muestra las tareas a desarrolladas en el Sprint 1:

Tabla 19  
*Pila del Sprint 1*

<b>Sprint 1</b>		<b>Inicio</b>	01/02/2021		
		<b>Finalización</b>	28/02/2021		
<b>Id</b>	<b>Descripción del elemento</b>	<b>Tareas</b>			
		<b>Id Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación [Horas]</b>	<b>Prioridad</b>
1	Administrar tickets	1.1	Crear operaciones CRUD para los tickets (registrar, editar, eliminar, actualizar)	15	Alta
		1.2	Implementar integración con Telegram para aperturar Tickets	35	Alta
		1.3	Implementar integración con Google Gmail para aperturar Tickets	35	Alta
		1.4	Implementar integración con Central Telefónica para aperturar Tickets	35	Alta
2	Administrar usuarios	2.1	Crear formulario para registrar usuarios	10	Alta
		2.2	Crear funcionalidad para listar usuarios	10	Alta
		2.3	Crear formulario para actualizar usuarios	10	Media
		2.4	Crear funcionalidad para eliminar usuarios	10	Alta
3	Administrar técnicos	3.1	Crear formulario para registrar técnicos	6	Alta
		3.2	Crear funcionalidad para listar técnicos	6	Alta

		3.3	Crear formulario para actualizar técnicos	6	Alta
		3.4	Crear funcionalidad para eliminar técnicos	6	Alta
4	Administrar canales	4.1	Crear formulario para registrar canales	15	Alta
		4.2	Crear funcionalidad para listar canales	15	Alta
		4.3	Crear formulario para actualizar canales	15	Media
		4.4	Crear funcionalidad para eliminar canales	12	Alta
		4.5	Crear funcionalidad para habilitar y deshabilitar canales	15	Alta

**Nota.** Elaboración Propia.

### **3.5.1.1. Integración entre el sistema y los canales de atención**

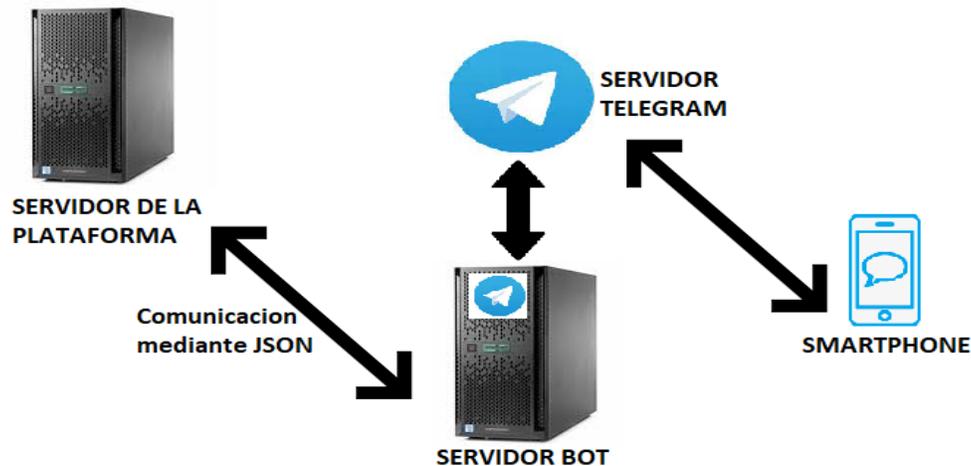
A continuación, se explica cómo es que el sistema se interconecta con los 3 canales de atención para la apertura de tickets: Telegram, Gmail y la Central Telefónica (PBX).

De manera general el sistema posee la estructura de una API (Application Programming Interfaces), es decir que posee funciones y procedimientos que le permiten interactuar con los canales de atención mediante formatos de texto JSON (JavaScript Object Notation).

#### **Telegram**

La aplicación posee la opción de creación de Bot Api, cada Bot previamente registrado vía @botfather, que es la única cuenta para la creación y revocación de ese tipo de cuentas, tiene una serie de parámetros y actividades, entre las cuales se encuentra la clave secreta o token (Para integración con servicios ajenos a Telegram como páginas web o servicios en la nube).

Una vez creado un Bot Api, el sistema ya puede recibir y enviar datos mediante formatos de texto JSON, bajo el protocolo de seguridad MTproto y el tipo de Autenticación JWT (Json Web Token), lo cual permite un nivel alto de seguridad en el intercambio de información.

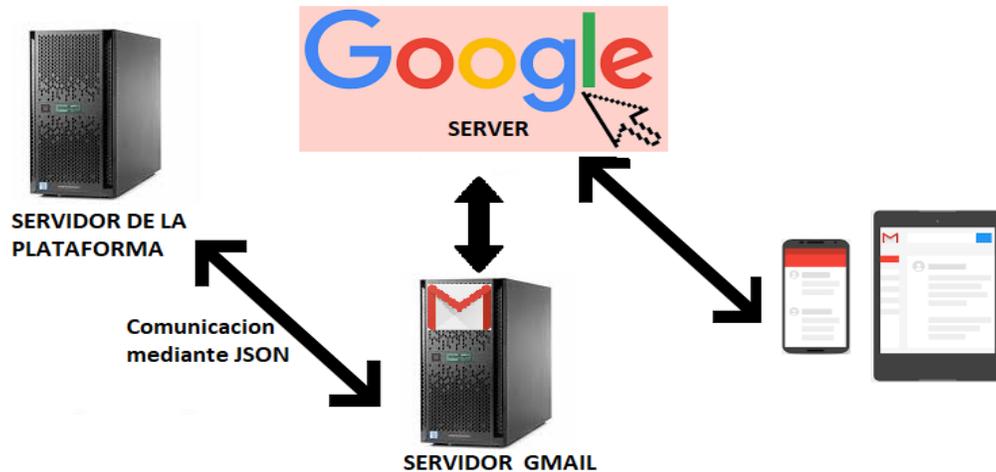


*Figura 30.* Comunicación entre el sistema y telegram.

## **Gmail**

Google Cloud Platform (Nube de Google) es una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado. Es utilizada para crear ciertos tipos de soluciones a través de la tecnología almacenada en la nube y permite por ejemplo destacar la rapidez y la escalabilidad de su infraestructura en las aplicaciones del buscador.

Dicha plataforma permite habilitar el uso de cualquier Api de Google, previamente creada una instancia o un proyecto, para este caso se habilita y utiliza los servicios de la Api de Gmail (correo electrónico), una vez creado el perfil de desarrollador, Google designa una llave de seguridad única por Api (Token) manejando el tipo de autenticación OAuth 2.0, de esta forma ya se puede interactuar con la Api de Gmail mediante formatos de texto JSON de manera segura.



*Figura 31.* Comunicación entre el sistema y Gmail.

### **Central Telefónica**

Para la conexión con Asterisk se utilizará la interfaz AGI (Asterisk Gateway Interface), o Interfaz Pasarela de Asterisk, es una interfaz del sistema Asterisk, que permite la comunicación con sistemas terceros, que se realiza de forma directa, dado que ejecuta un script a voluntad, tan pronto la aplicación AGI sea invocada. La interfaz AGI, fue diseñada originalmente, para servir de pasarela entre los distintos lenguajes de programación y Asterisk específicamente, los canales de comunicación especialmente.

El funcionamiento de AGI es relativamente sencillo:

- Primero hay que establecer una pasarela, entre el canal en curso, y un script ejecutable a voluntad.
- Cuando la conexión se establece, lanza una serie de variables al script ejecutado, que son accesibles desde el mismo. La mayoría de estas variables tienen algo que ver con el canal

- El script puede ejecutar cualquier algoritmo, y entre sus secuencias, puede ejecutar comandos específicos de AGI para influenciar en el flujo de llamada en curso. Este script puede perdurar en ejecución incluso después de finalizar la llamada.

De esta manera es como Asterisk enviará y recibirá información del sistema para identificar al cliente, listar las fallas comunes de la empresa y aperturar tickets.

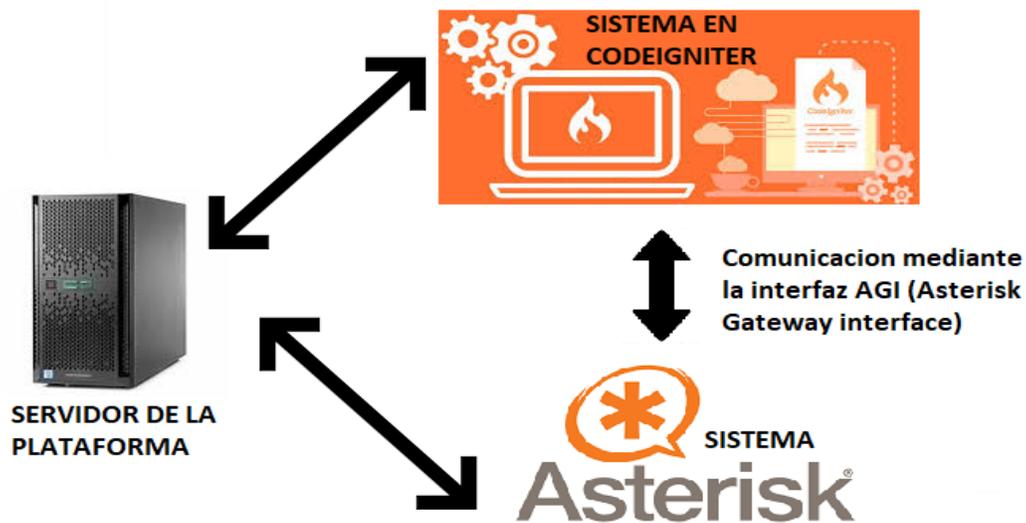


Figura 32. Comunicación entre el sistema y la central telefónica en Asterisk.

### 3.5.1.2. Modelo de Navegación.

La Figura 33 muestra el Modelo de Navegación en base a la pila del Sprint 1.

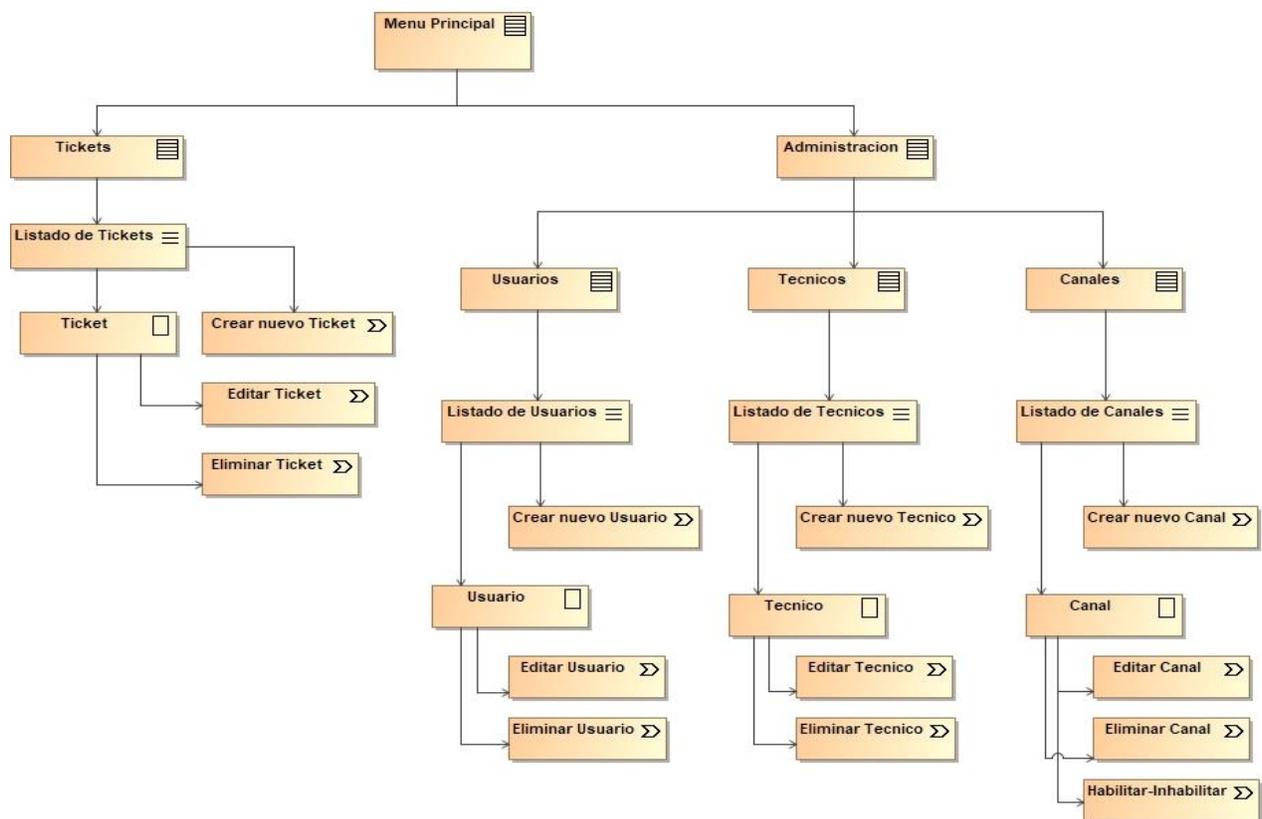


Figura 33. Modelo de Navegación para el Sprint 1.

La Figura 34 muestra el Modelo de Navegación de apertura de un ticket vía Telegram en base a la pila del Sprint 1.

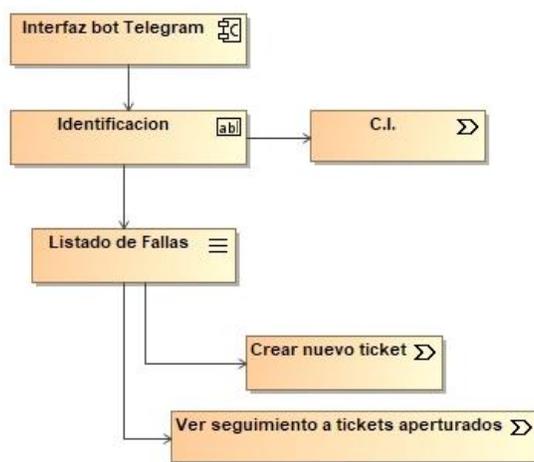


Figura 34. Modelo de Navegación para el Sprint 1 (apertura de ticket vía Telegram).

La Figura 35 muestra el Modelo de Navegación de apertura de un ticket vía Correo Electrónico en base a la pila del Sprint 1.

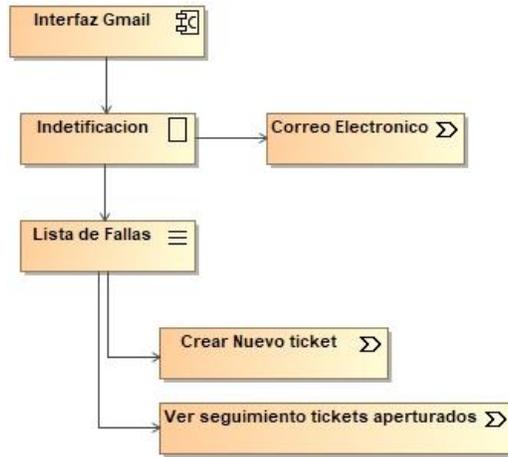


Figura 35. Modelo de Navegación para el Sprint 1 (apertura de ticket vía Correo Electrónico).

La Figura 36 muestra el Modelo de Navegación de apertura de un ticket vía Central Telefónica e IVR en base a la pila del Sprint 1.

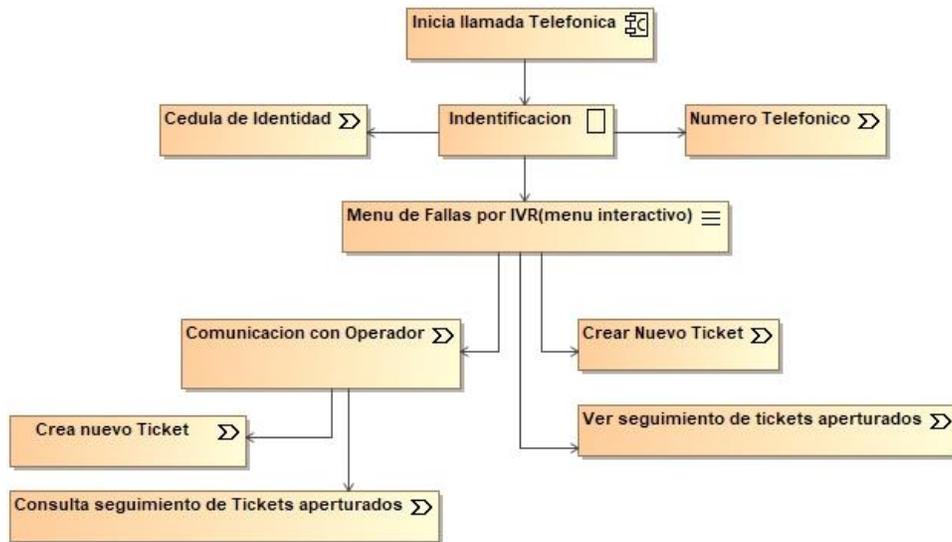


Figura 36. Modelo de Navegación para el Sprint 1 (apertura de ticket vía Central Telefónica e IVR).

### 3.5.1.3. Modelo de Presentación.

En la Figura 37 se muestra el modelo de presentación para la pantalla principal de Tickets.



Figura 37. Modelo de Presentación de Tickets.

En la Figura 38 se ve el modelo de presentación para la pantalla principal Usuarios.



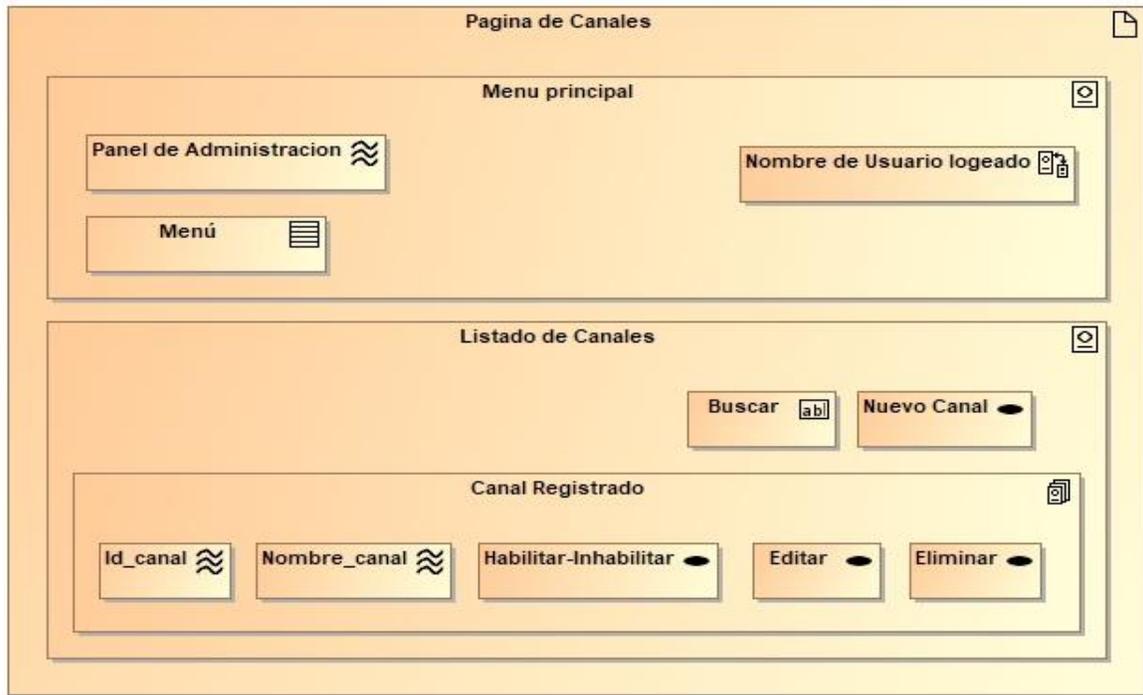
Figura 38. Modelo de Presentación de Usuarios.

En la Figura 39 se muestra el modelo de presentación para la pantalla principal de Técnicos.



Figura 39. Modelo de Presentación de Técnicos.

En la Figura 40 se ve el modelo de presentación para la pantalla principal Canales.



*Figura 40.* Modelo de Presentación de Canales.

A continuación, se tiene las capturas de pantalla de las tareas implementadas del Sprint 1 en funcionamiento.

En la Figura 41 se puede ver la pantalla de tickets con las opciones de crear, editar y eliminar.

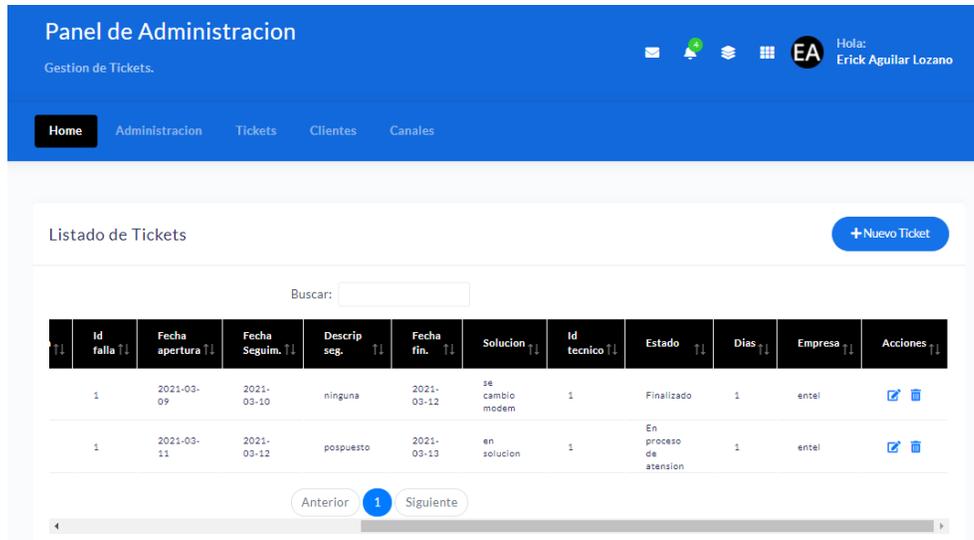


Figura 41. Pantalla de listado de tickets registrados.

En la Figura 42 se muestra una captura de pantalla con el formulario de registro de tickets.

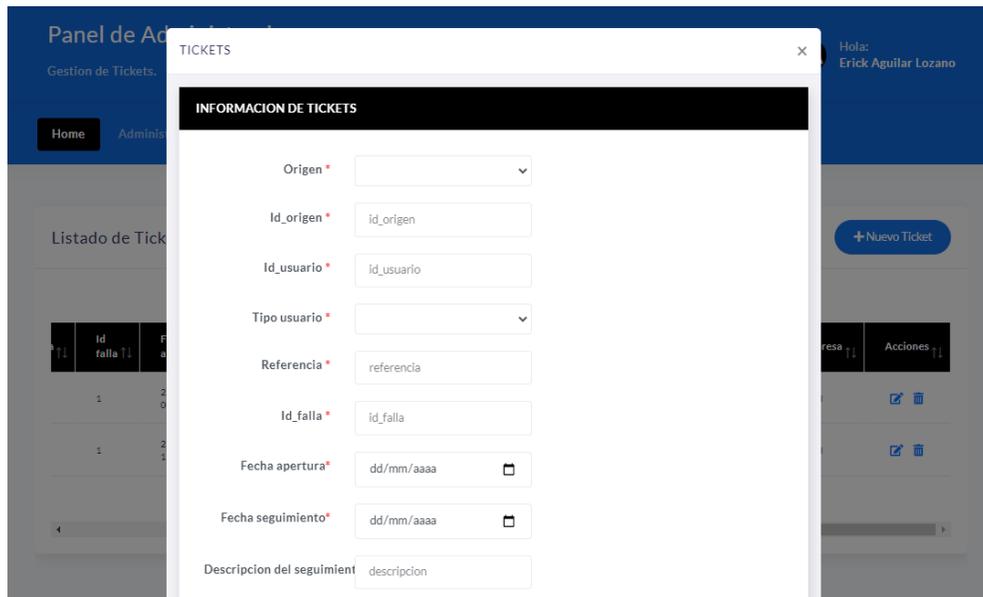


Figura 42. Pantalla del formulario de registro de tickets.

En la Figura 43 se puede ver la apertura y consulta de seguimiento de tickets mediante Telegram.

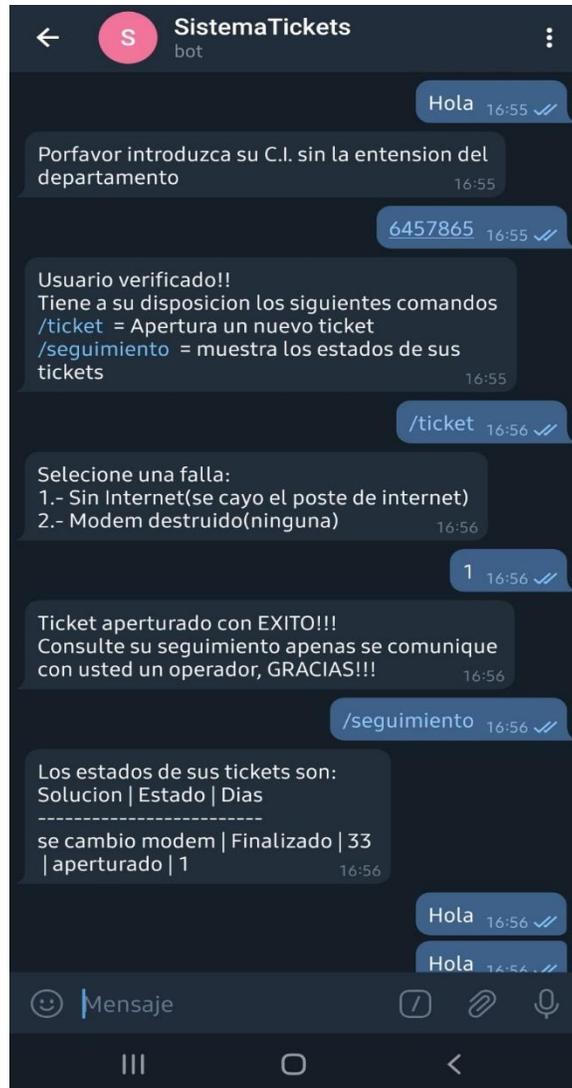


Figura 43. Pantalla de apertura y seguimiento de tickets vía telegram.

En las siguientes Figuras se puede ver la apertura de tickets mediante correo electrónico Gmail. Primeramente, se envía un mensaje al correo de la empresa.

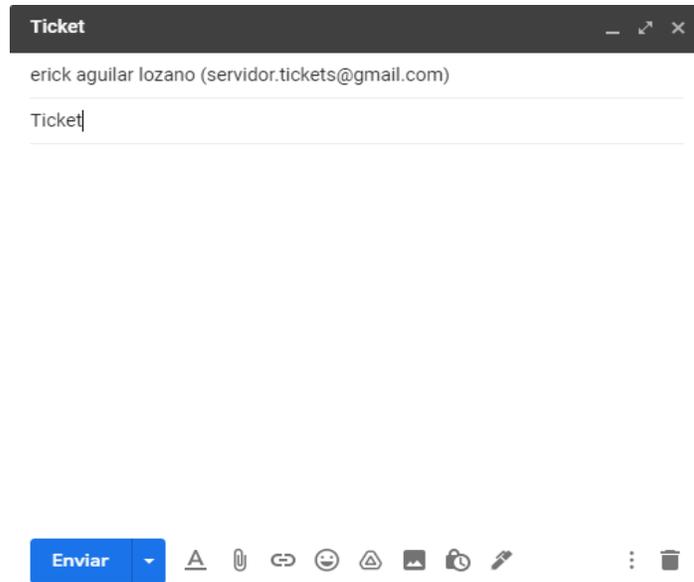


Figura 44. Envió de mensaje al correo de la empresa.

Se reciben las fallas y se responde al mensaje con la falla y una descripción.



erick aguilar lozano <erickaguilarlozano@gmail.com>  
para SERVIDOR-TICKETS ▾

13 abr 2021 19:45

1  
nada que decir



Figura 45. Selección y respuesta al correo con la falla.

Finalmente se recibe un mensaje con la confirmación de apertura del ticket como se muestra en la Figura 46.

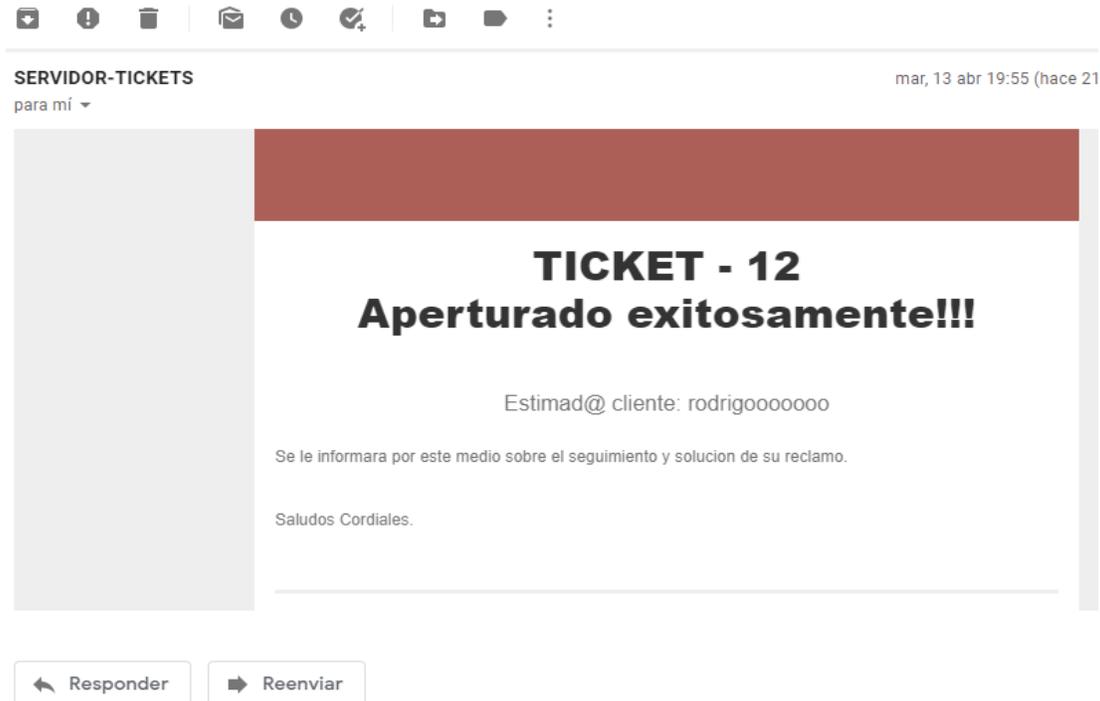


Figura 46. Pantalla de apertura exitosa de tickets vía Gmail.

En la Figura 47 se puede ver la apertura de tickets mediante la central telefónica, al no poder capturar una llamada telefónica se mostrará el proceso de apertura mediante consola de Asterisk.

```

== Using SIP RTP TOS bits 184
== Using SIP RTP CoS mark 5
> 0x7f8698006710 -- Strict RTP learning after remote address set to: 200.87.137.254:21130
-- Executing [s@from-trunkvozip:1] Answer("SIP/troncal-50150781-00000000", "") in new stack
> 0x7f8698006710 -- Strict RTP learning after remote address set to: 200.87.137.254:21130
-- Executing [s@from-trunkvozip:2] AGI("SIP/troncal-50150781-00000000", "erick2.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/erick2.php
erick2.php: Inicia script de IVR PBX
erick2.php: Informacion 1: SIP/troncal-50150781-00000000
erick2.php: Informacion 2: 72536177
erick2.php: Informacion 3: s
erick2.php: Informacion 4: 72536177
erick2.php: Informacion 5: unknown
erick2.php: Informacion 6: from-trunkvozip
erick2.php: Informacion 7:
erick2.php: nombre cliente:
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/ingrese_carnet.slin' (language 'es')
> 0x7f8698006710 -- Strict RTP switching to RTP target address 200.87.137.254:21130
> 0x7f8698006710 -- Strict RTP learning complete - Locking on source address
erick2.php: tecla:
erick2.php: nombre cliente obtenido por cedula:
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/ingrese_carnet.slin' (language 'es')
erick2.php: tecla: 233445
erick2.php: nombre cliente obtenido por cedula: gonzalo
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/inicio.slin' (language 'es')
erick2.php: tecla: 1
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/empresa_1/falla_1.slin' (language 'es')
erick2.php: Tecla del ticket:
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/empresa_1/falla_2.slin' (language 'es')
erick2.php: Tecla del ticket: 1
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/empresa_1/falla_3.slin' (language 'es')
erick2.php: Tecla del ticket:
erick2.php: Resultado de guardarrrrrrrrrrr: 1
erick2.php: Resultado de guardarrrrrrrrrrr: 1
-- AGI Script Executing Application: (Playback) Options: (/var/lib/asterisk/sounds/custom/gracias_ticket)
-- <SIP/troncal-50150781-00000000> Playing '/var/lib/asterisk/sounds/custom/gracias_ticket.slin' (language 'es')
-- AGI Script Executing Application: (SayNumber) Options: (14)
-- <SIP/troncal-50150781-00000000> Playing 'digits/14.gsm' (language 'es')
-- <SIP/troncal-50150781-00000000>AGI Script erick2.php completed, returning 0

```

Datos de quien realiza la llamada

Cliente Identificado mediante c.i. y atendido por el menú interactivo

El IVR dicta las fallas disponibles y el cliente elige una

Ticket aperturado con éxito

Figura 47. Registro de apertura de tickets vía Central Telefónica.

En la Figura 48 se puede observar la consulta del seguimiento de un ticket mediante la central telefónica.

```
ssabel*CLI>
= Using SIP RTP TOS bits 184
= Using SIP RTP CoS mark 5
> 0x7f8698005d20 -- Strict RTP learning after remote address set to: 200.87.137.254:21674
-- Executing [s@from-trunkvozip:1] Answer("SIP/troncal-50150781-00000001", "") in new stack
> 0x7f8698005d20 -- Strict RTP learning after remote address set to: 200.87.137.254:21674
-- Executing [s@from-trunkvozip:2] AGI("SIP/troncal-50150781-00000001", "erick2.php") in new stack
-- Launched AGI Script /var/lib/asterisk/agi-bin/erick2.php
erick2.php: Inicia script de IVR PBX
erick2.php: Informacion 1: SIP/troncal-50150781-00000001
erick2.php: Informacion 2: 72536177
erick2.php: Informacion 3: s
erick2.php: Informacion 4: 72536177
erick2.php: Informacion 5: unknown
erick2.php: Informacion 6: from-trunkvozip
erick2.php: Informacion 7:
erick2.php: nombre cliente:
-- <SIP/troncal-50150781-00000001> Playing '/var/lib/asterisk/sounds/custom/ingrese_carnet.slin' (language 'es')
> 0x7f8698005d20 -- Strict RTP switching to RTP target address 200.87.137.254:21674 as source
> 0x7f8698005d20 -- Strict RTP learning complete - switching on source address
erick2.php: tecla: 233445
erick2.php: nombre cliente obtenido por cedula: gonzalo
-- <SIP/troncal-50150781-00000001> Playing '/var/lib/asterisk/sounds/custom/inicio.slin' (language 'es')
erick2.php: tecla: 2
-- AGI Script Executing Application: (Playback) Options: (/var/lib/asterisk/sounds/custom/su_ticket_nro)
-- <SIP/troncal-50150781-00000001> Playing '/var/lib/asterisk/sounds/custom/su_ticket_nro.slin' (language 'es')
-- AGI Script Executing Application: (SayNumber) Options: (14)
-- <SIP/troncal-50150781-00000001> Playing 'digits/14.gsm' (language 'es')
-- AGI Script Executing Application: (Playback) Options: (/var/lib/asterisk/sounds/custom/Apertura)
-- <SIP/troncal-50150781-00000001> Playing '/var/lib/asterisk/sounds/custom/Apertura.slin' (language 'es')
-- <SIP/troncal-50150781-00000001> AGI Script erick2.php completed, returning 0
-- Executing [s@from-trunkvozip:3] Hangup("SIP/troncal-50150781-00000001", "16") in new stack
= Spawn extension (from-trunkvozip, s, 3) exited non-zero on 'SIP/troncal-50150781-00000001'
ssabel*CLI>
```

Datos de quien realiza la llamada

Cliente Identificado mediante c.i. y atendido por el menú interactivo

Cliente selecciona seguimiento

IVR dicta el estado de Tickets

Figura 48. Registro de seguimiento de tickets vía Central Telefónica.

En la Figura 49 se muestra la comunicación con un operador en caso de no poder aperturar tickets por los anteriores canales.

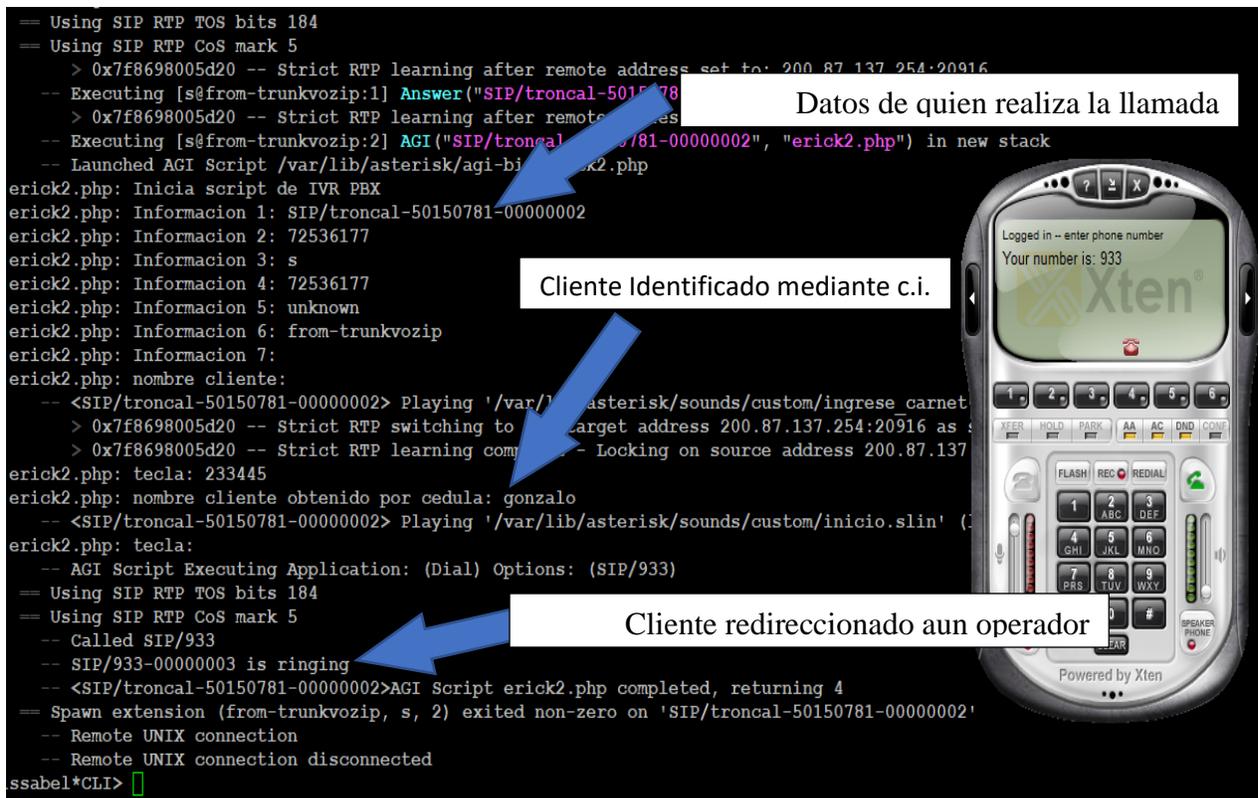
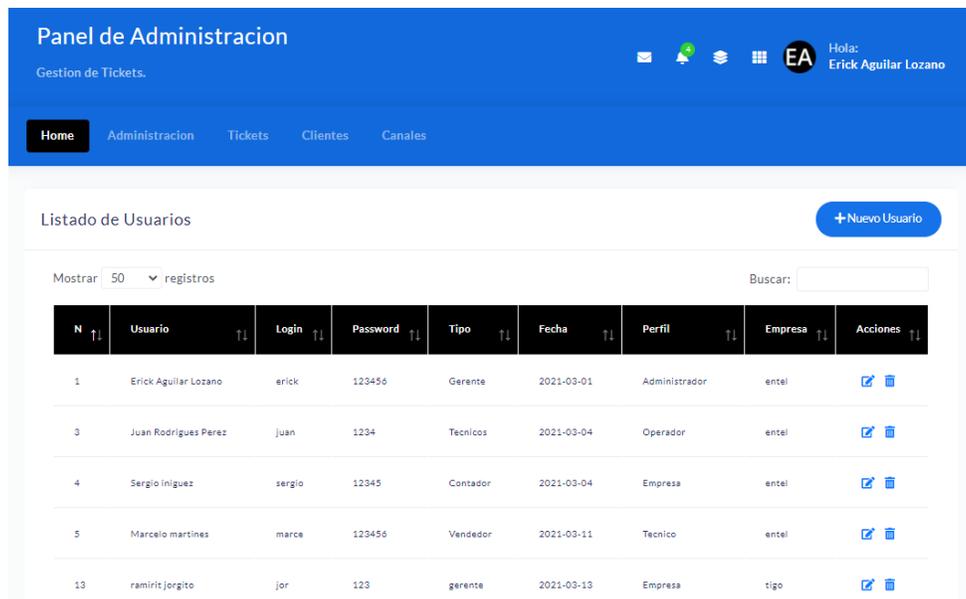


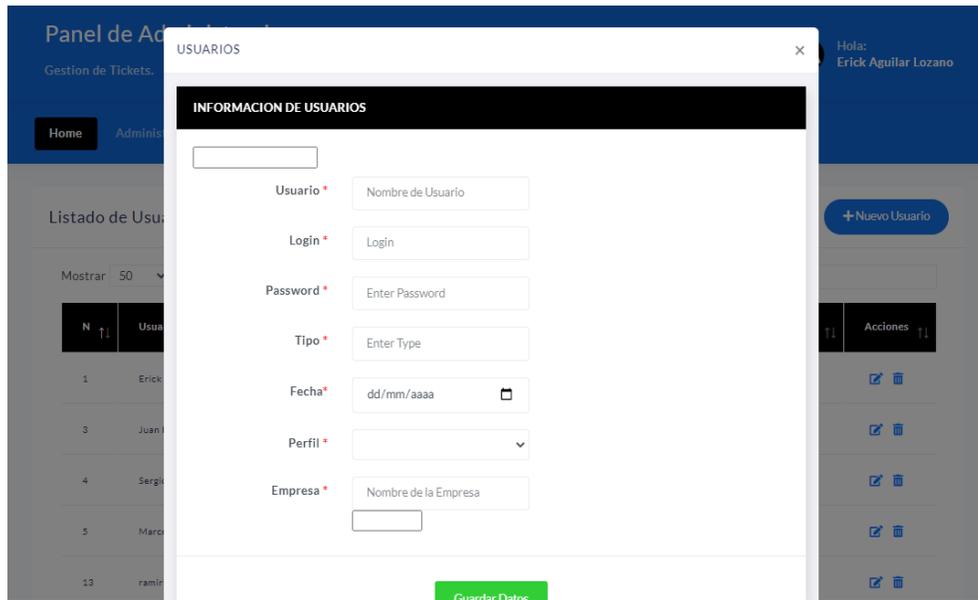
Figura 49. Registro de comunicación con un operador vía Central Telefónica.

En la Figura 50 se muestra la pantalla de usuarios con las opciones de crear, editar y eliminar.



*Figura 50.* Pantalla de listado de usuarios registrados.

En la Figura 51 se muestra una captura de pantalla con el formulario de registro de usuarios.



The image shows a web application interface with a modal window titled "USUARIOS" open over a "Listado de Usuarios" page. The modal window has a dark header "INFORMACION DE USUARIOS" and contains the following fields:

- Nombre de Usuario (input field)
- Login (input field)
- Enter Password (input field)
- Enter Type (input field)
- dd/mm/aaaa (date picker)
- Perfil (dropdown menu)
- Nombre de la Empresa (input field)

A green "Guardar Datos" button is located at the bottom of the modal. The background page shows a table of users with columns for "N" and "Usuario", and a sidebar with a "+Nuevo Usuario" button and "Acciones" (edit/delete icons).

*Figura 51.* Pantalla del formulario de registro de usuarios.

En la Figura 52 se muestra la pantalla de técnicos con las opciones de crear, editar y eliminar.

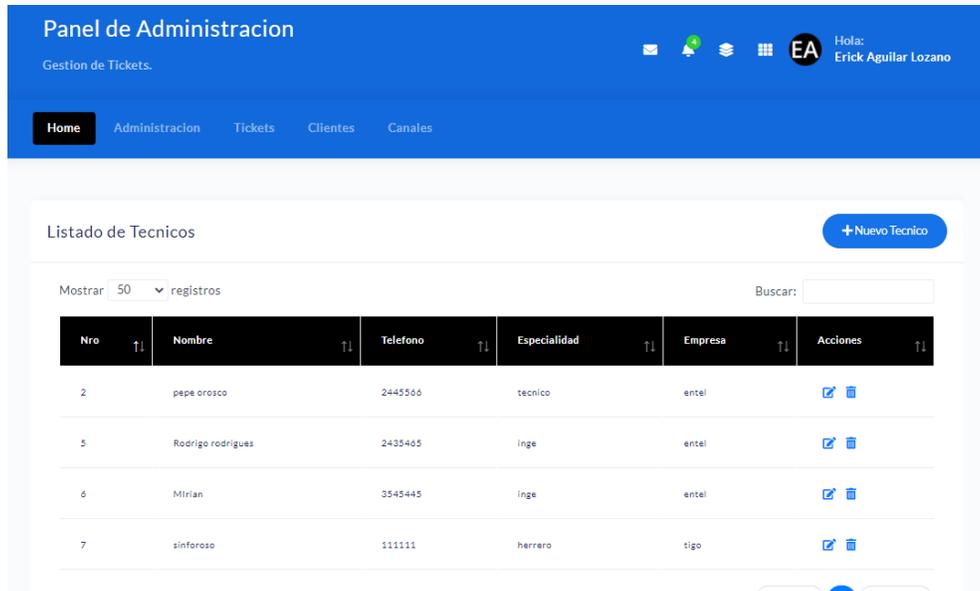


Figura 52. Pantalla de listado de técnicos registrados.

En la Figura 53 se muestra una captura de pantalla con el formulario de registro de técnicos.

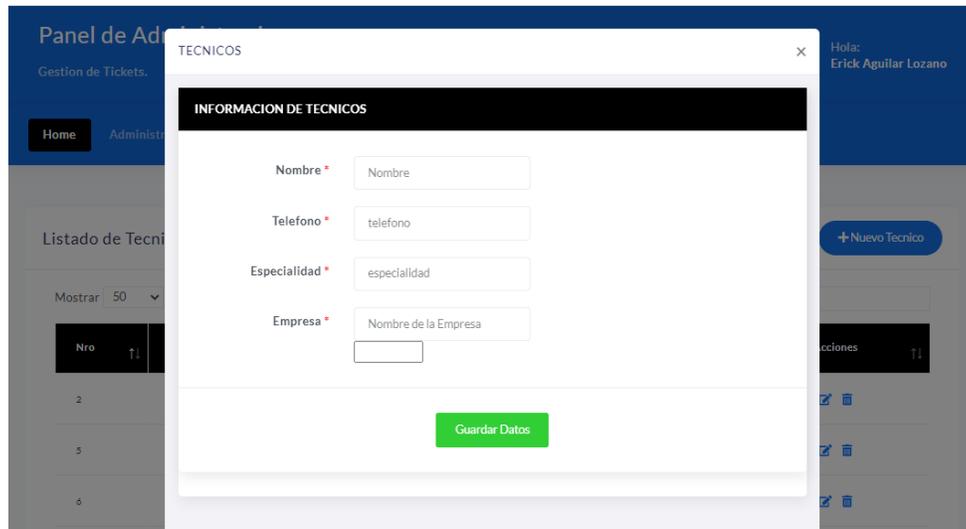


Figura 53. Pantalla del formulario de registro de técnicos.

En la Figura 54 se muestra la pantalla de canales con las opciones de crear, habilitar o inhabilitar, editar y eliminar.

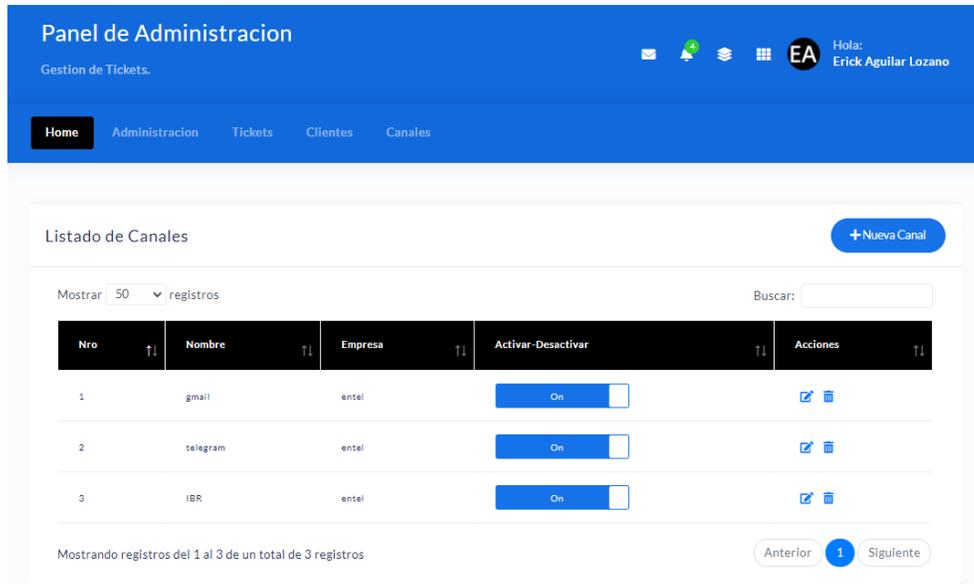


Figura 54. Pantalla de listado de canales registrados.

En la Figura 55 se muestra una captura de pantalla con el formulario de registro de canales.

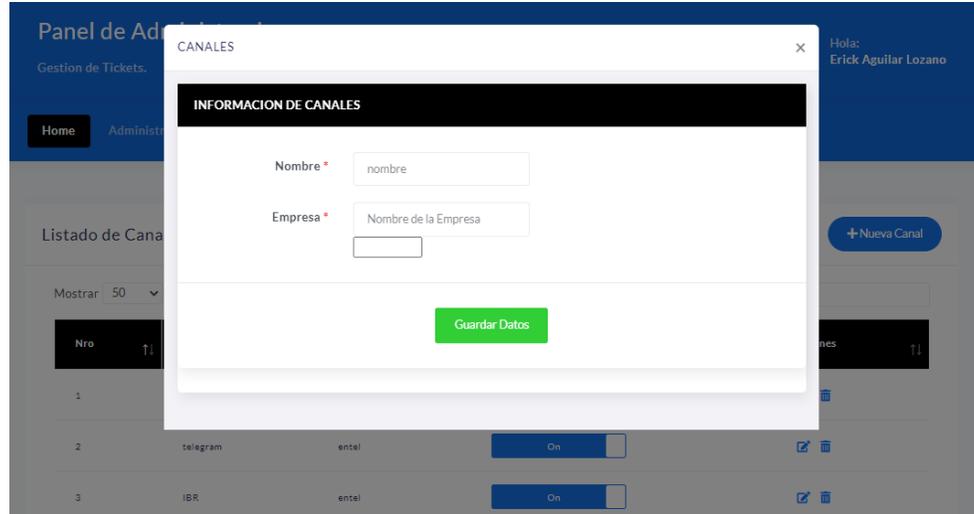


Figura 55. Pantalla del formulario de registro de canales.

### 3.5.2. Sprint 2

En la Tabla 20 se muestra las tareas desarrolladas en el Sprint 2.

Tabla 20  
Pila del Sprint 2

<b>Sprint 2</b>		<b>Inicio</b>	01/03/2021		
		<b>Finalización</b>	31/03/2021		
<b>Id</b>	<b>Descripción del elemento</b>	<b>Tareas</b>			
		<b>Id Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación [Horas]</b>	<b>Prioridad</b>
5	Administrar fallas	5.1	Crear formulario para registrar fallas	15	Alta
		5.2	Crear funcionalidad para listar fallas	10	Alta
		5.3	Crear formulario para actualizar fallas	13	Alta
		5.4	Crear funcionalidad para eliminar fallas	10	Alta
6	Gestionar técnicos a tickets	6.1	Crear funcionalidad para asignar y agendar técnicos a tickets, mediante Google Calendar.	40	Alta
7	Administrar clientes	7.1	Crear formulario para registrar clientes	15	Alta
		7.2	Crear funcionalidad para listar clientes	10	Alta
		7.3	Crear formulario para actualizar clientes	15	Media
		7.4	Crear funcionalidad para eliminar clientes	10	Alta
		7.5	Crear funcionalidad para importar clientes desde un archivo Excel	22	Alta
8	Administrar permisos	8.1	Crear funcionalidad para registrar, listar, actualizar y eliminar permisos.	48	Alta

**Nota.** Elaboración Propia.

### 3.5.2.1. Modelo de Navegación.

En la Figura 56 se ve el modelo de navegación en base a las tareas del Sprint 2.

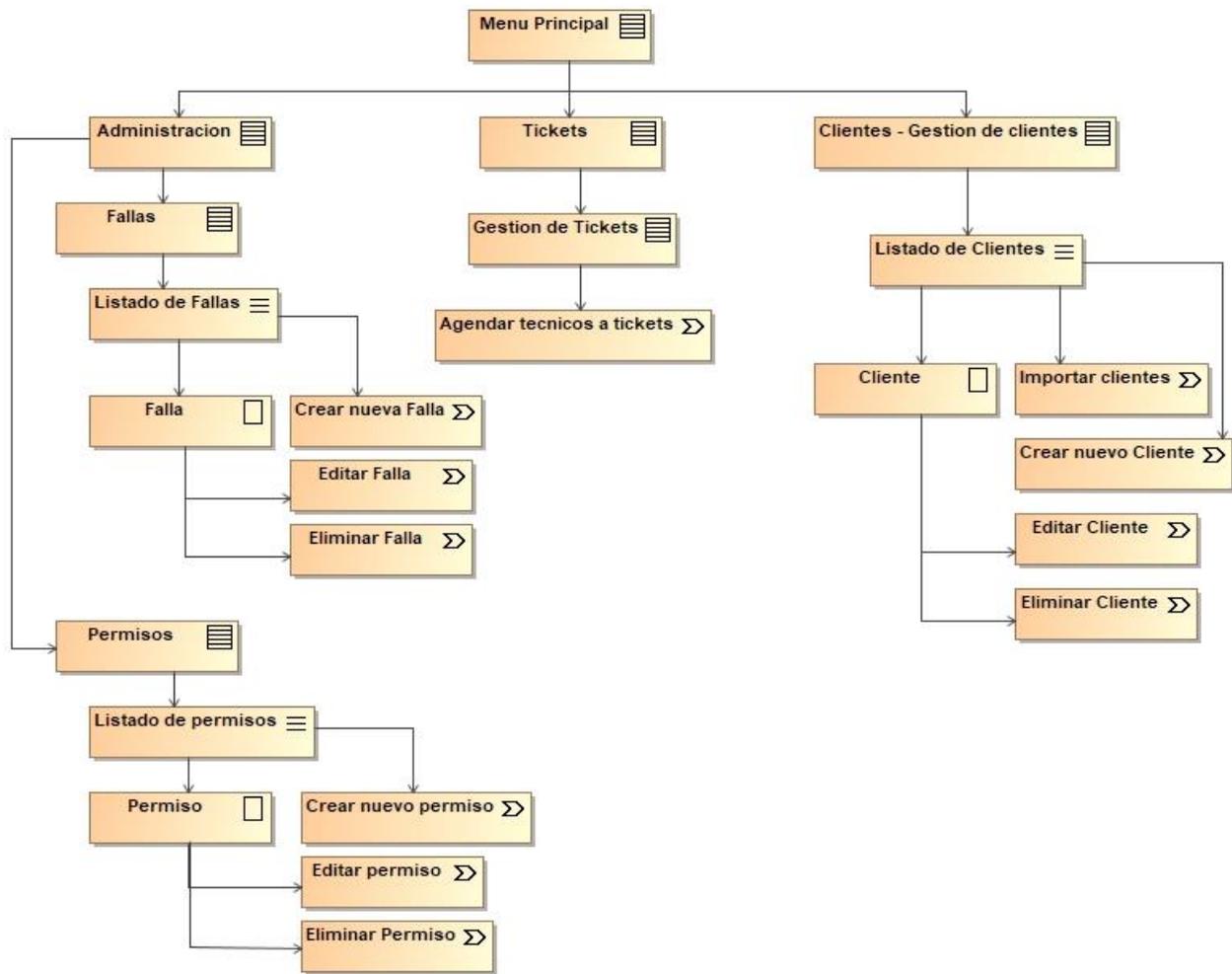


Figura 56. Modelo de Navegación para el Sprint 2.

### 3.5.2.2. Modelo de Presentación.

En la Figura 57 se muestra el modelo de presentación para la pantalla de fallas.

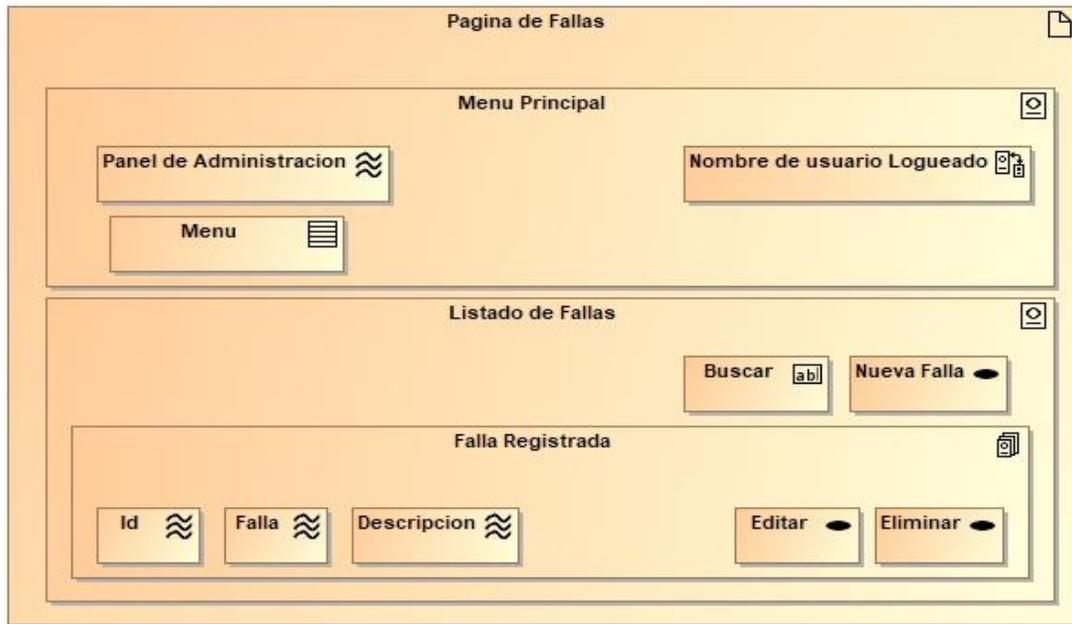


Figura 57. Modelo de Presentación de Fallas.

En la Figura 58 se muestra el modelo de presentación para la pantalla de agendar técnicos a tickets.

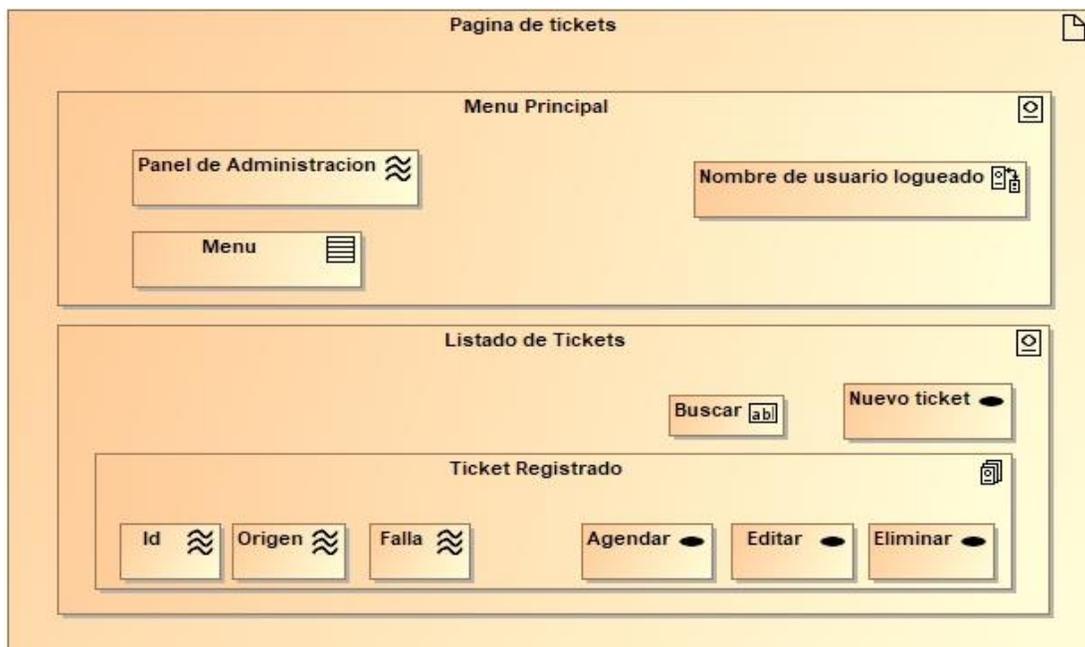


Figura 58. Modelo de Presentación de Agendar Técnicos a Tickets.

En la Figura 59 se muestra el modelo de presentación para la pantalla de clientes.

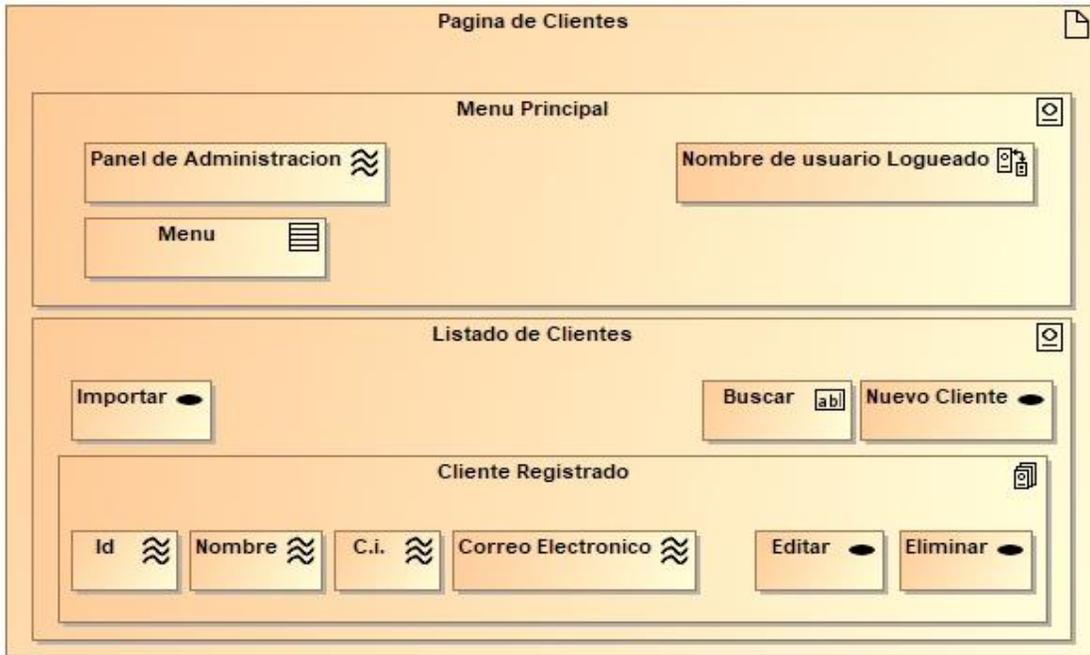


Figura 59. Modelo de Presentación de Clientes.

En la Figura 60 se muestra el modelo de presentación para la pantalla de permisos.

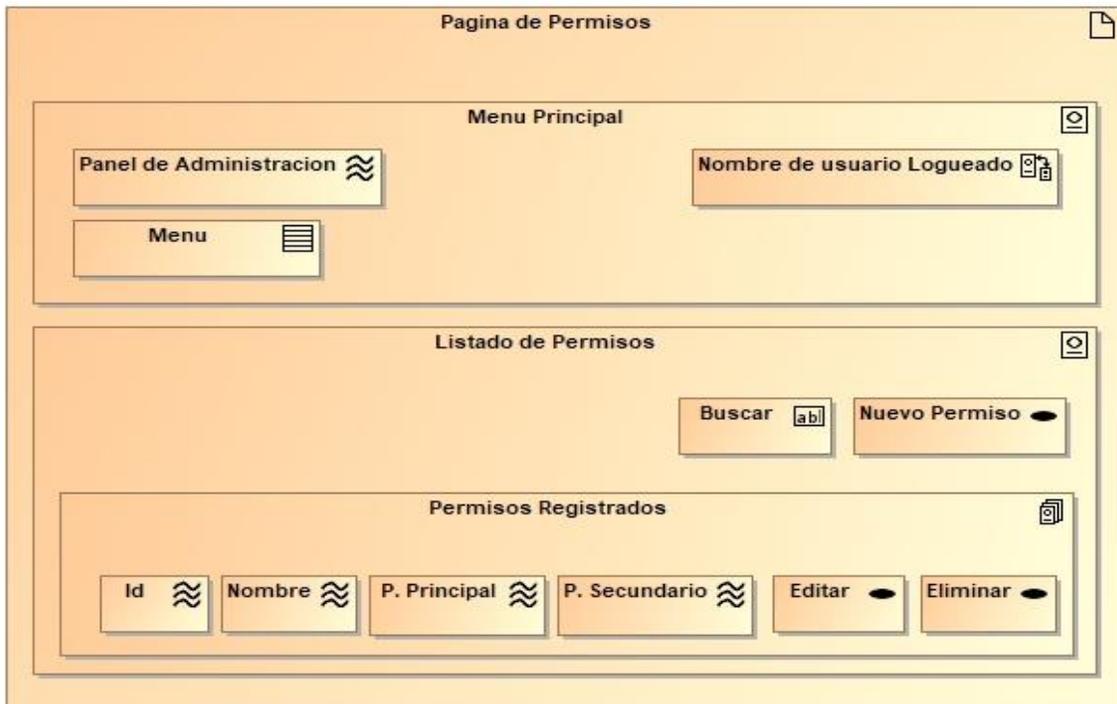


Figura 60. Modelo de Presentación de Permisos.

A continuación, se tiene las capturas de pantalla de las tareas implementadas en el Sprint 2. En la Figura 61 se muestra la pantalla de Fallas con las opciones de crear, editar y eliminar.

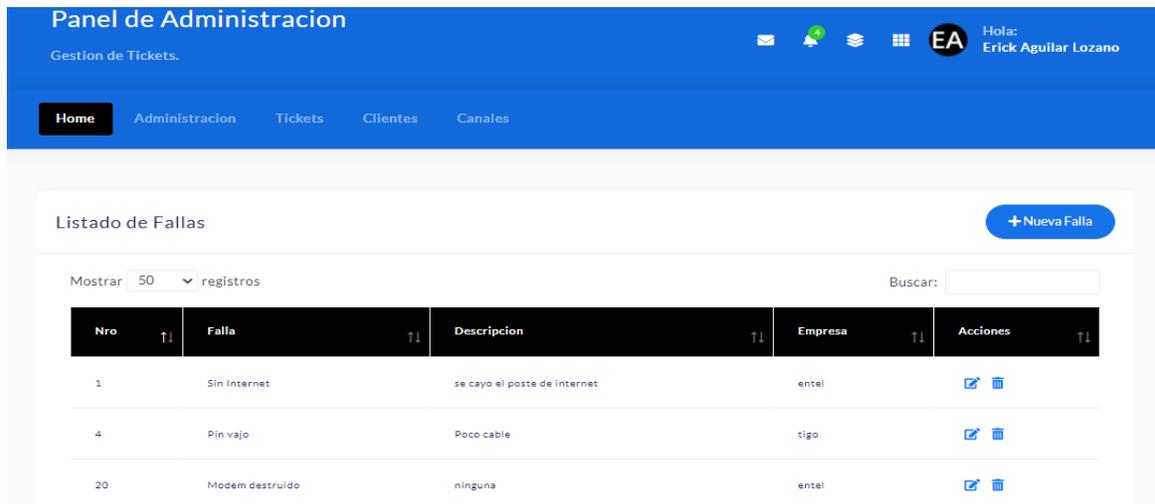


Figura 61. Pantalla de listado de fallas registradas.

En la Figura 62 se muestra una captura de pantalla con el formulario de registro de fallas.

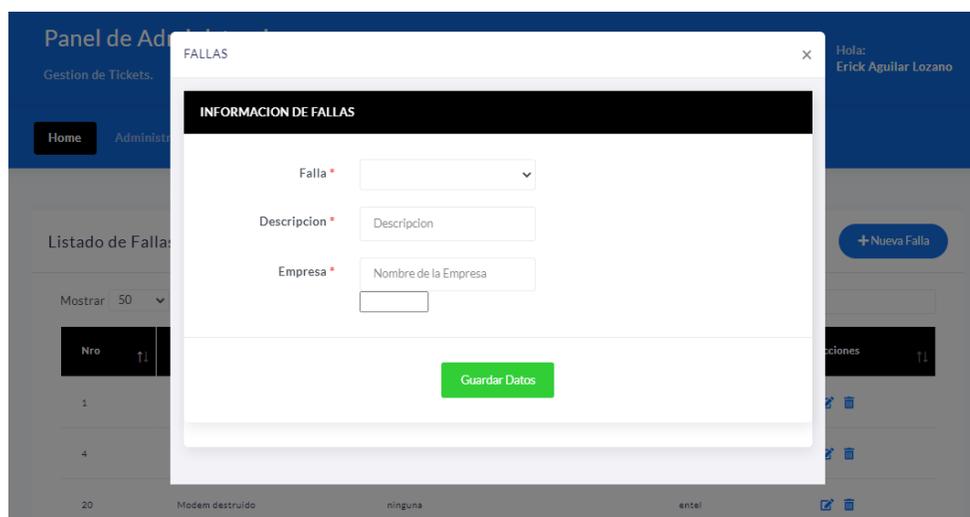


Figura 62. Pantalla del formulario de registro de fallas.

En la Figura 63 se muestra la pantalla de tickets con la opción de agendar técnicos.

**Panel de Administracion**  
Gestión de Tickets. Hola: Erick Aguilar Lozano

Home | Administracion | Tickets | Clientes | Canales

Listado de Tickets +Nuevo Ticket

Buscar:

Id	Fecha apertura	Fecha Seguim.	Descrip seg.	Fecha fin.	Solucion	Id tecnico	Estado	Días	Empresa	Acciones
1	2021-03-09	2021-03-10	ninguna	2021-03-12	se cambio modem	1	Finalizado	34	entel	Agendar
2	2021-03-11	2021-03-12	pospuesto	2021-03-13	en solucion	1	En proceso de atencion	32	entel	Agendar
3	2021-04-13	0000-00-00		0000-00-00		0	aperturado	0	entel	Agendar
4	2021-04-13	0000-00-00		0000-00-00		0	aperturado	0	tigo	Agendar
5	2021-04-	0000-		0000-						Agendar

Figura 63. Pantalla de listado de tickets con la opción de agendar técnicos.

En la Figura 64 se muestra la pantalla de clientes con las opciones de importar, crear, editar y eliminar.

**Panel de Administracion**  
Gestión de Tickets. Hola: Erick Aguilar Lozano

Home | Administracion | Tickets | Clientes | Canales

Listado de Clientes Importar +Nuevo Cliente

Mostrar 50 registros Buscar:

Nro	Nombre	C.I.	Correo Electronico	Telefono-Celular	Direccion	Empresa	Acciones
1	adal ramonez	6457865	adal@gmail.com	75643233	Miraflores	entel	Editar Eliminar
3	peluchin	4556577	pelu@gmail.com	232323	dfdfdf	entel	Editar Eliminar
5	rodrigoooooo	455657	erickaguilarlozano@gmail.com	232323	fdfdfdfdf	tigo	Editar Eliminar
13	jorge	567890	jorge@gmail.com	75676877	uyuni	tigo	Editar Eliminar

*Figura 64.* Pantalla de listado de clientes registrados.

En la Figura 65 se muestra una captura de pantalla con el formulario de registro de clientes.

The image shows a web application interface. At the top, there's a dark blue header with the text 'Panel de Administración' and 'Gestión de Tickets'. On the right, it says 'Hola: Erick Aguilar Lozano'. Below the header, there's a navigation bar with 'Home' and 'Administración'. The main content area is divided into two parts. On the left, there's a 'Listado de Clientes' section with a table showing columns for 'Nro', 'Nombre', and 'Acciones'. The table has several rows of data. On the right, there's a '+Nuevo Cliente' button. In the center, a modal window titled 'CLIENTES' is open, displaying a form for adding a new client. The form has a title 'INFORMACION DE CLIENTES' and several input fields: 'Nombre' (placeholder: nombre), 'C.I.' (placeholder: cedula), 'Correo electronico' (placeholder: Email), 'Telefono-Celular' (placeholder: telefono\_celular), 'Direccion' (placeholder: direccion), and 'Empresa' (placeholder: Nombre de la Empresa). A green 'Guardar Datos' button is located at the bottom of the form.

*Figura 65.* Pantalla del formulario de registro de clientes.

En la Figura 66 se muestra la pantalla de Permisos con las opciones de crear, editar y eliminar.

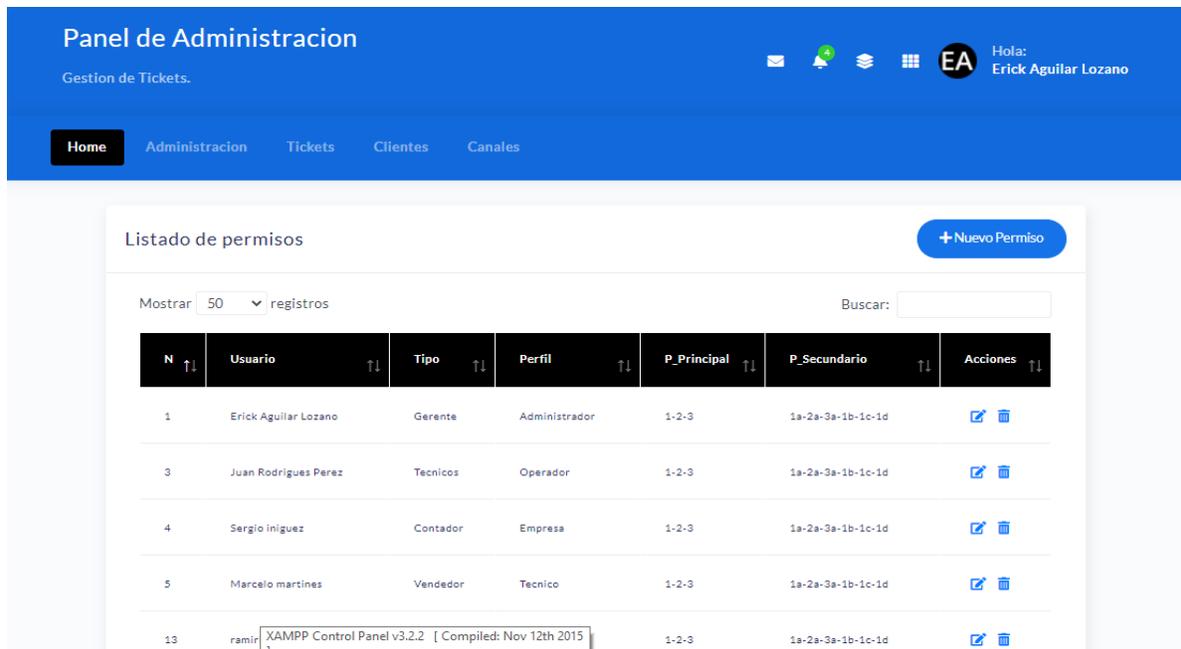


Figura 66. Pantalla de listado de permisos registrados.

En la Figura 67 se muestra una captura de pantalla con el formulario de registro de permisos.

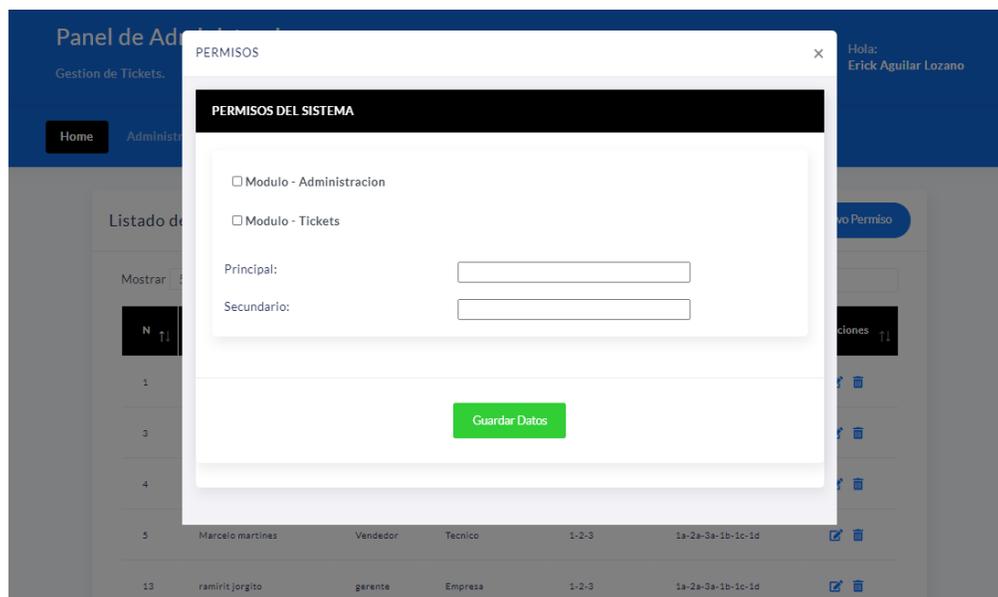


Figura 67. Pantalla del formulario de registro de permisos.

### 3.5.3. Sprint 3

En la Tabla 21 se muestra las tareas desarrolladas en el Sprint 3.

Tabla 21  
Pila del Sprint 3

<b>Sprint 3</b>		<b>Inicio</b>	01/04/2021		
		<b>Finalización</b>	30/04/2021		
<b>Id</b>	<b>Descripción del elemento</b>	<b>Tareas</b>			
		<b>Id Tarea</b>	<b>Descripción Tarea</b>	<b>Estimación [Horas]</b>	<b>Prioridad</b>
9	Administrar canales Gmail	9.1	Crear formulario para registrar canales Gmail	20	Alta
		9.2	Crear funcionalidad para listar canales Gmail	20	Alta
		9.3	Crear formulario para actualizar canales Gmail	20	Alta
		9.4	Crear funcionalidad para eliminar canales Gmail	12	Alta
10	Administrar canales telegram	10.1	Crear formulario para registrar canales de Telegram	18	Alta
		10.2	Crear funcionalidad para listar canales de Telegram	10	Alta
		10.3	Crear formulario para actualizar canales de Telegram	10	Alta
		10.4	Crear funcionalidad para eliminar canales de Telegram	10	Alta
11	Administrar canales central telefónica	11.1	Crear formulario para registrar canales de central telefónica	20	Alta
		11.2	Crear funcionalidad para listar canales de central telefónica	20	Alta
		11.3	Crear formulario para actualizar canales de central telefónica	20	Alta
		11.4	Crear funcionalidad para eliminar canales de central telefónica	12	Alta

12	Administrar calendarios Google	12.1	Crear formulario para registrar calendarios Google	20	Alta
		12.2	Crear funcionalidad para listar calendarios Google	20	Alta
		12.3	Crear formulario para actualizar calendarios Google	20	Media
		12.4	Crear funcionalidad para eliminar calendarios Google	12	Alta
13	Administrar empresas	13.1	Crear formulario para registrar empresas	7	Alta
		13.2	Crear funcionalidad para listar empresas	7	Alta
		13.3	Crear formulario para actualizar empresas	7	Media
		13.4	Crear funcionalidad para eliminar empresas	7	Alta

**Nota.** Elaboración Propia.

### 3.5.3.1. Modelo de Navegación.

En la Figura 68 se ve el modelo de navegación en base a las tareas del Sprint 3.

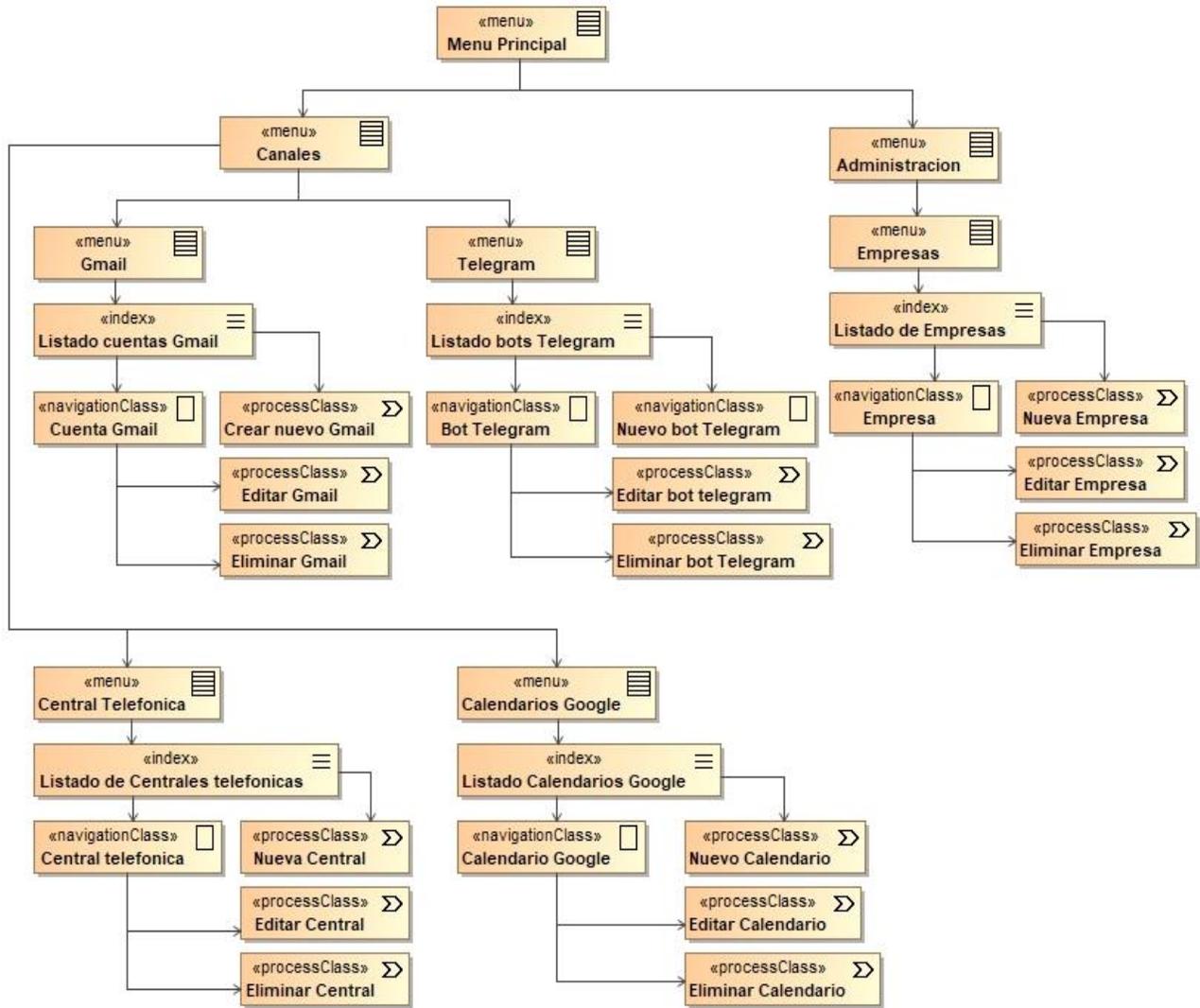


Figura 68. Modelo de Navegación para el Sprint 3.

### 3.5.3.2. Modelo de Presentación.

En la Figura 69 se muestra el modelo de presentación para la pantalla de cuentas de Gmail.



Figura 69. Modelo de Presentación de Gmail.

En la Figura 70 se muestra el modelo de presentación para la pantalla de Telegram.

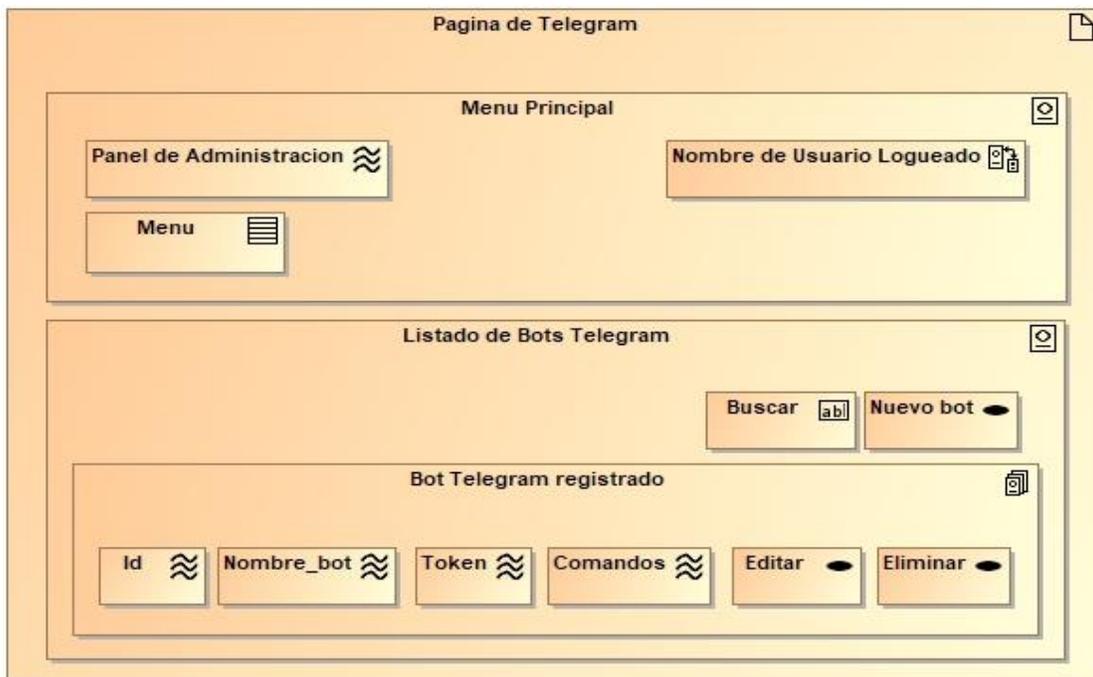


Figura 70. Modelo de Presentación de Telegram.

En la Figura 71 se muestra el modelo de presentación para la pantalla de Centrales Telefónicas.

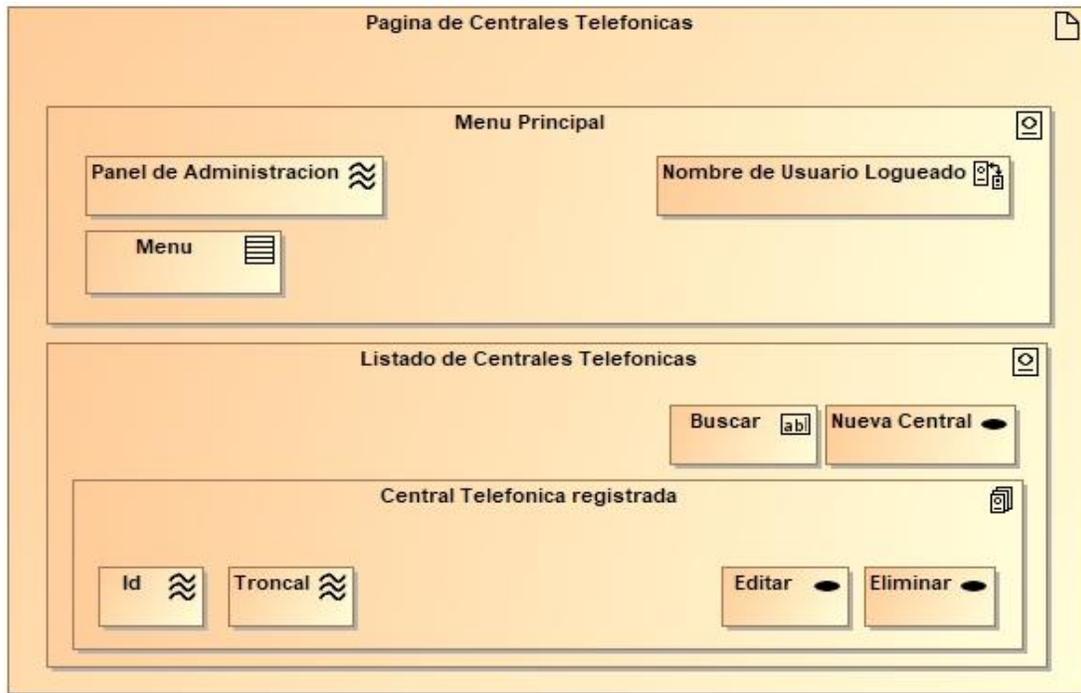


Figura 71. Modelo de Presentación de Centrales Telefónicas.

En la Figura 72 se muestra el modelo de presentación para la pantalla de Calendarios Google.

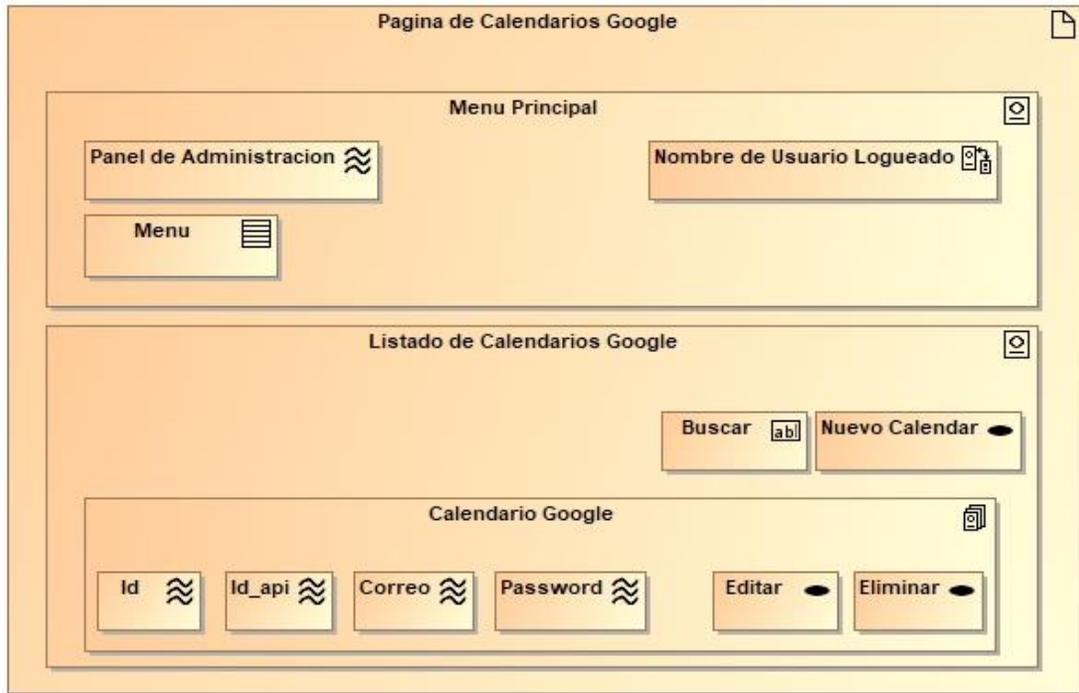


Figura 72. Modelo de Presentación de Calendarios Google.

En la Figura 73 se muestra el modelo de presentación para la pantalla de Empresas.

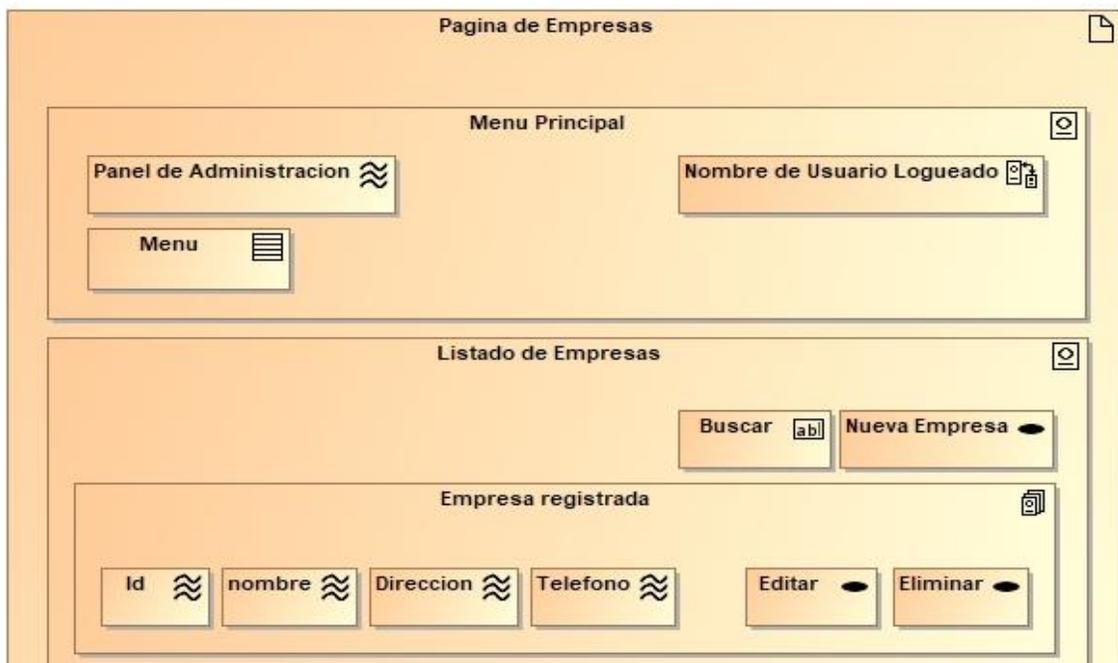


Figura 73. Modelo de Presentación de Empresas.

A continuación, se tiene las capturas de pantalla de las tareas implementadas en el Sprint 3. En la Figura 74 se muestra la pantalla de Gmail con las opciones de crear, editar y eliminar.

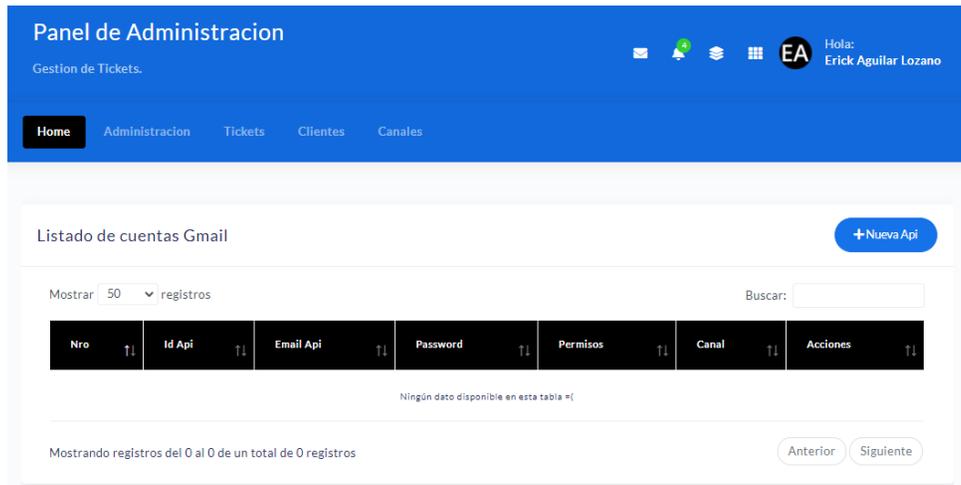


Figura 74. Pantalla de listado de cuentas Gmail registradas.

En la Figura 75 se muestra una captura de pantalla con el formulario de registro de cuentas Gmail.

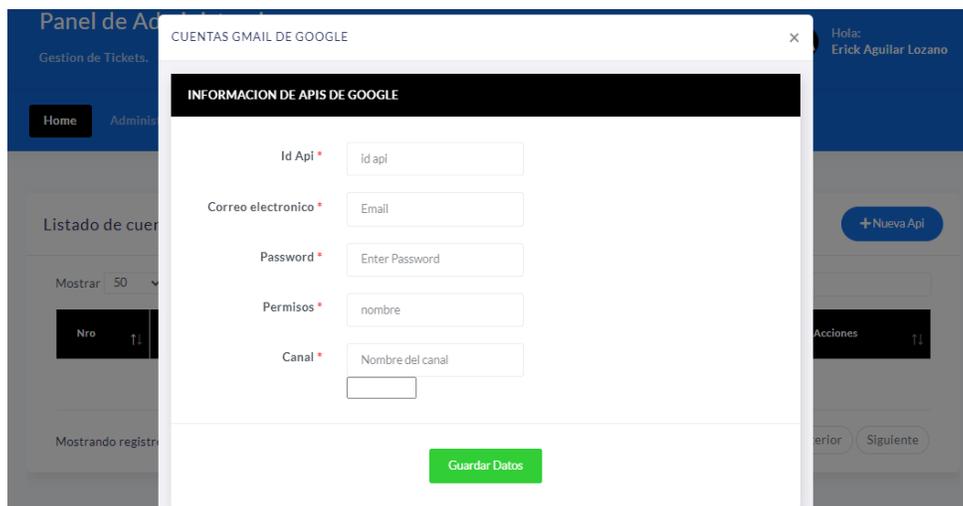


Figura 75. Pantalla del formulario de registro de cuentas Gmail.

En la Figura 76 se muestra la pantalla de Telegram con las opciones de crear, editar y eliminar.

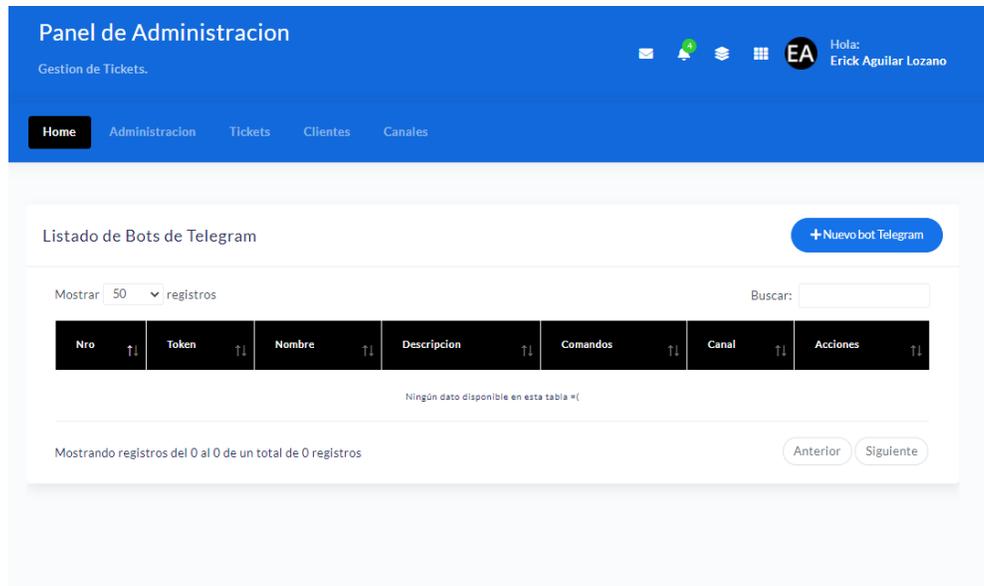


Figura 76. Pantalla de listado de bots Telegram registrados.

En la Figura 77 se muestra una captura de pantalla con el formulario de registro de bots Telegram.

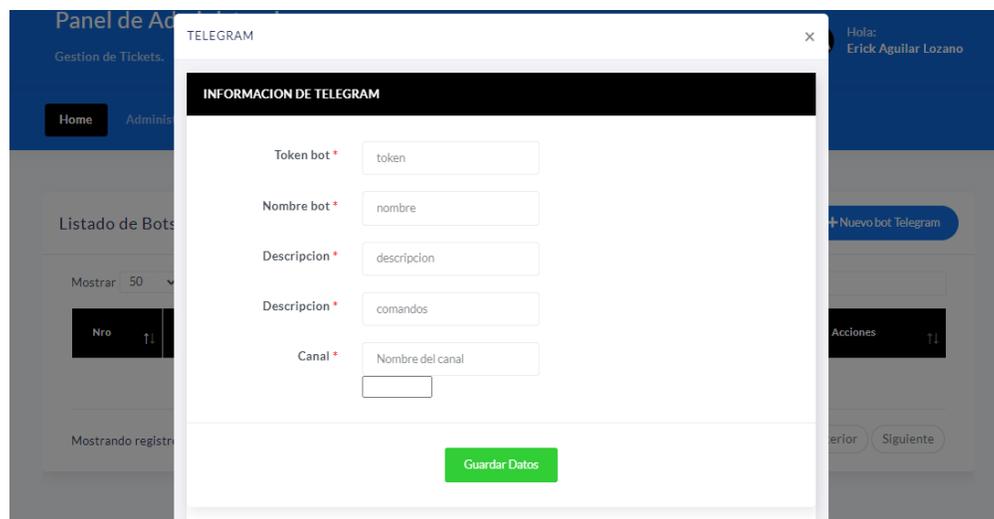


Figura 77. Pantalla del formulario de registro de Bots Telegram.

En la Figura 78 se muestra la pantalla de Central Telefónica con las opciones de crear, editar y eliminar.

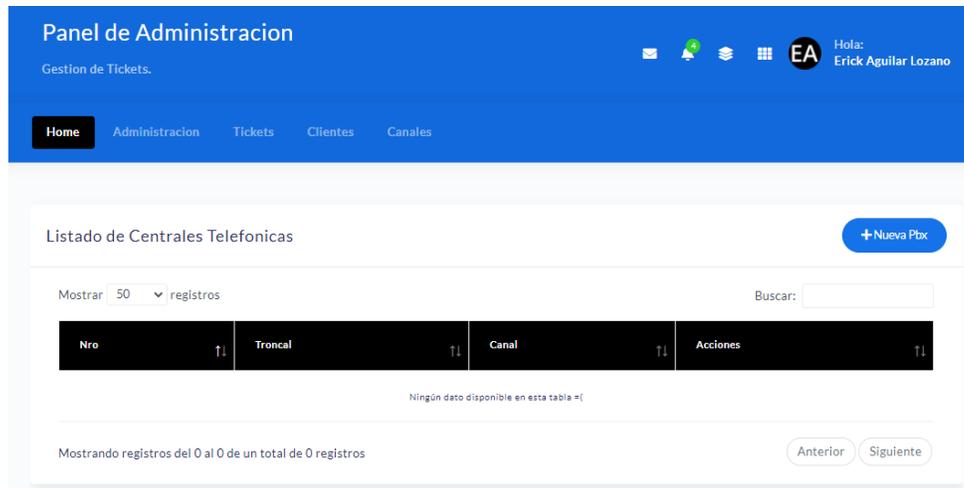


Figura 78. Pantalla de listado de Centrales Telefónicas registradas.

En la Figura 79 se muestra una captura de pantalla con el formulario de registro de Centrales Telefónicas.

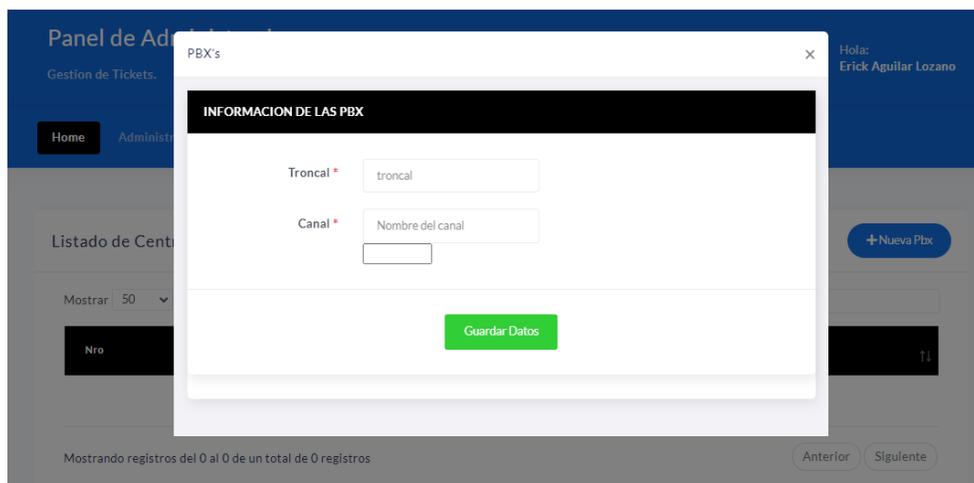
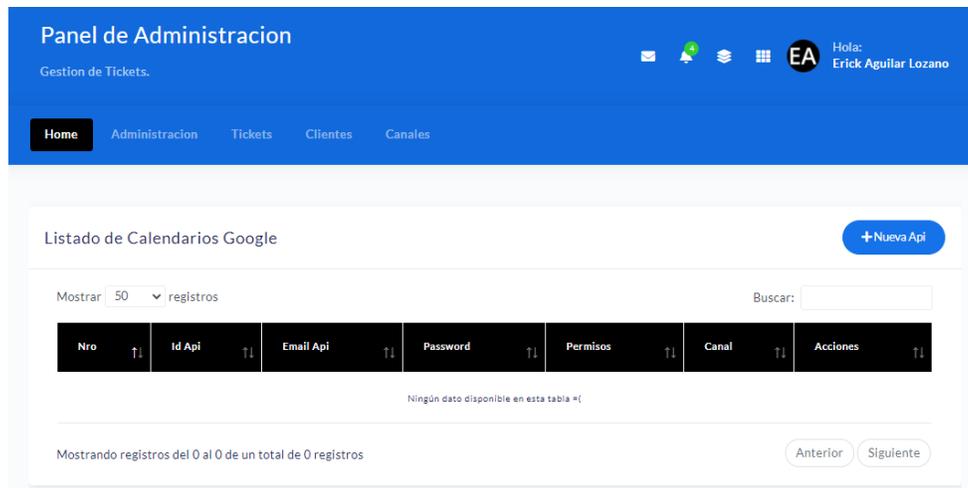


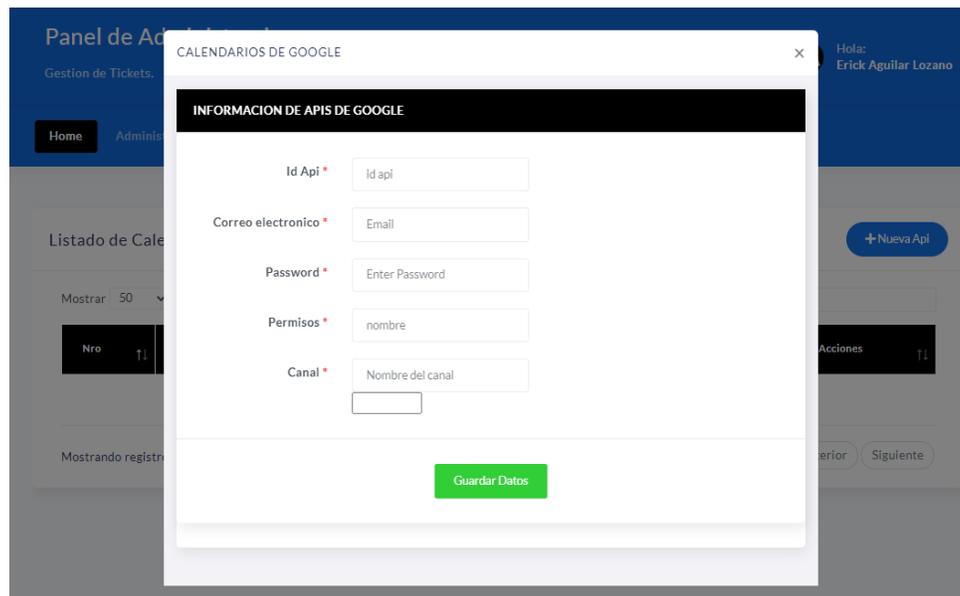
Figura 79. Pantalla del formulario de registro de Centrales Telefónicas.

En la Figura 80 se muestra la pantalla de Calendarios Google con las opciones de crear, editar y eliminar.



*Figura 80.* Pantalla de listado de Calendarios Google registrados.

En la Figura 81 se muestra una captura de pantalla con el formulario de registro de Calendarios.



*Figura 81.* Pantalla del formulario de registro de Calendarios.

En la Figura 82 se muestra la pantalla de Empresas con las opciones de crear, editar y eliminar.

Panel de Administración  
Gestión de Tickets.

Hola: Erick Aguilar Lozano

Home Administración Tickets Clientes Canales

Listado de Empresas + Nueva Empresa

Mostrar 50 registros Buscar:

Nro	Nombre	Dirección	Teléfono	N.I.T	Rubro	Acciones
1	entel	casimiro corales	2345678	232424	telecomunicaciones	
2	tigo	Av. buch	2365656	2323232	telecomunicaciones	
3	Cotas	Unduavi	2345678	2147483647	servicios publicos	

Mostrando registros del 1 al 3 de un total de 3 registros Anterior 1 Siguiente

Figura 82. Pantalla de listado de Empresas registradas.

En la Figura 83 se muestra una captura de pantalla con el formulario de registro de Empresas.

Panel de Administración  
Gestión de Tickets.

Hola: Erick Aguilar Lozano

Home Administración Tickets Clientes Canales

Listado de Empresas + Nueva Empresa

Mostrar 50 registros Buscar:

EMPRESAS

**INFORMACION DE EMPRESAS**

Nombre \*

Dirección \*

Teléfono \*

N.I.T. \*

Rubro \*

Guardar Datos

Mostrando registros del 1 al 3 de un total de 3 registros Anterior 1 Siguiente

Figura 83. Pantalla del formulario de registro de Empresas.

## CAPITULO 4

### DEMOSTRACION DE LA HIPOTESIS

#### 4.1. Introducción

En este capítulo se demostrará la hipótesis planteada aplicando un proceso de recolección de datos para su demostración. Para la obtención de datos estadísticos se optó por la técnica de recolección de datos por medio de una encuesta, planteados a los usuarios o entes de interés.

#### 4.2. Comprobación de hipótesis

Para la comprobación de hipótesis, es necesario establecer el entorno de prueba para la recolección de datos, con los que se realiza la prueba de hipótesis. Para ello se da los niveles de ponderación, los mismos serán utilizados para los cálculos de la muestra, esto se realiza según a los valores de ajuste establecidos como se observa a continuación en la tabla 22.

Tabla 22  
*Valores de ajuste*

<b>Descripción</b>	<b>Escala</b>	<b>Ponderación</b>
Excelente	5	10
Muy bien	4	8
Bien	3	5
Regular	2	2
Mal	1	0

**Nota.** Elaboración Propia.

La encuesta fue realizada a 25 personas entre ellas 2 gerentes de empresas de ventas de hardware en telecomunicaciones, 5 técnicos especializados en redes y las restantes 18 personas mayores de 35 años, considerando que poseen un criterio formado para la encuesta, por

lo tanto, la muestra tiene como  $N = 25$ , y los resultados obtenidos a partir de la recolección de datos se muestran a continuación en la tabla 23.

Tabla 23  
*Factores de ponderación total*

<b>Encuesta</b>	<b>Observado</b>	<b>Esperado</b>
1	84	85
2	98	85
3	78	85
4	80	85
5	86	85
6	83	85
7	86	85
8	88	85
9	90	85
10	83	85
11	87	85
12	93	85
13	86	85
14	90	85
15	90	85
16	90	85
17	87	85
18	86	85
19	88	85
20	87	85
21	83	85
22	93	85
23	86	85
24	90	85
25	93	85
<b>Total Promedio</b>	87.4	85

**Nota.** Elaboración Propia.

La prueba de hipótesis para medias usando la Distribución t Student, se usa cuando se cumplen las siguientes dos condiciones:

- Es posible calcular la media y la desviación estándar a partir de la muestra.
- El tamaño de la muestra es menor a 30.

### 4.3. Planteamiento de la hipótesis nula y alternativa

El procedimiento presenta 5 pasos esenciales que se muestran a continuación:

**Primer paso:** Se identifica la hipótesis nula y la hipótesis alternativa, bajo la siguiente notación: Hipótesis alternativa denotado por  $H_i$ , es la hipótesis planteada en el Capítulo 1 y la hipótesis nula denotado por  $H_0$ , es exactamente lo contrario de la hipótesis planteada.

$H_i$ :” La implementación de una plataforma multicanal en línea permitirá optimizar la gestión y seguimiento de tickets para empresas en telecomunicaciones”.

$H_0$ :” La implementación de una plataforma multicanal en línea no permitirá optimizar la gestión y seguimiento de tickets para empresas en telecomunicaciones”.

**Segundo paso:** Determinar el nivel de significancia, es el rango de aceptación de la hipótesis alternativa. Para ello se considera:

- 0.05 para proyectos de investigación.
- 0.01 para aseguramiento de calidad.

Por lo tanto, para nuestro caso es:  $\alpha = 0.05$  este valor utilizamos por conveniencia.

**Tercer paso:** Evidencia maestra, cálculo de la media y la desviación estándar a partir de la muestra.

Según los datos estadísticos obtenidos tenemos los siguientes datos:

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{X} = \frac{84 + 98 + 78 + 80 + 86 + 83 + 86 + 88 + 90 + 83 + 87 + 93 + 86 + 90 + 90 + 90 + 87 + 86 + 88 + 87 + 83 + 93 + 86 + 90 + 93}{25}$$

$$\bar{X} = \frac{2185}{25}$$

$$\bar{X} = 87.4$$

Donde  $\bar{X}$  es la media.

Ahora para la desviación estándar, tenemos:

$$S_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{X})^2}$$

$$S_x = \sqrt{\frac{1}{25-1} (464)}$$

$$S_x = \sqrt{19.33}$$

$$S_x = 4.39$$

Esto lo verificamos con una calculadora de Desviación estándar en línea, como se observa en la figura 84.

Tamaño de la muestra: 25  
Media aritmética ( $\bar{x}$ ): 87.4

**Desviación estándar (s): 4.3969686527576**

[Media aritmética](#) [Media geométrica](#) [Media armónica](#) [Mediana](#) [Moda](#) [Varianza](#)  
[Desviación media](#) [Rango](#) [Rango intercuartílico](#) [Cuartiles](#) [Todas medidas de dispersión](#)  
[Diagrama de caja](#)

Los datos son de una:  Población  Muestra

**Introduzca los datos separados por comas (sólo números):**

78, 80, 83, 83, 83, 84, 86, 86, 86, 86, 86, 87, 87, 87, 88, 88, 90, 90, 90, 90, 90, 93, 93, 93, 98

ENVÍA DATOS    RESET

Figura 84. Cálculo en línea de la desviación estándar.

[<http://www.alcula.com/es/calculadoras/estadistica/desviacion-estandar/>]

**Cuarto paso:** Se aplica la distribución t Student para calcular la probabilidad de error (P) por medio de la fórmula como se observa continuación.

$$t = \frac{\bar{X} - \mu}{S_x / \sqrt{n}}$$

Grados de libertad:  $gl = n - 1$

Media:  $\bar{X} = 87.4$

Valor a analizar:  $\mu = 85$

Valor a analizar:  $S_x = 4.39$

Tamaño de la muestra:  $n = 25$

Reemplazando en la fórmula de t Student tenemos:

$$t = \frac{\bar{X} - \mu}{S_x / \sqrt{n}} = \frac{87.4 - 85}{4.39 / \sqrt{25}} = 2.73$$

$$gl = n - 1 = 25 - 1 = 24$$

Esto lo podemos verificar con una aplicación que calcula la t Student como se muestra en la siguiente figura:

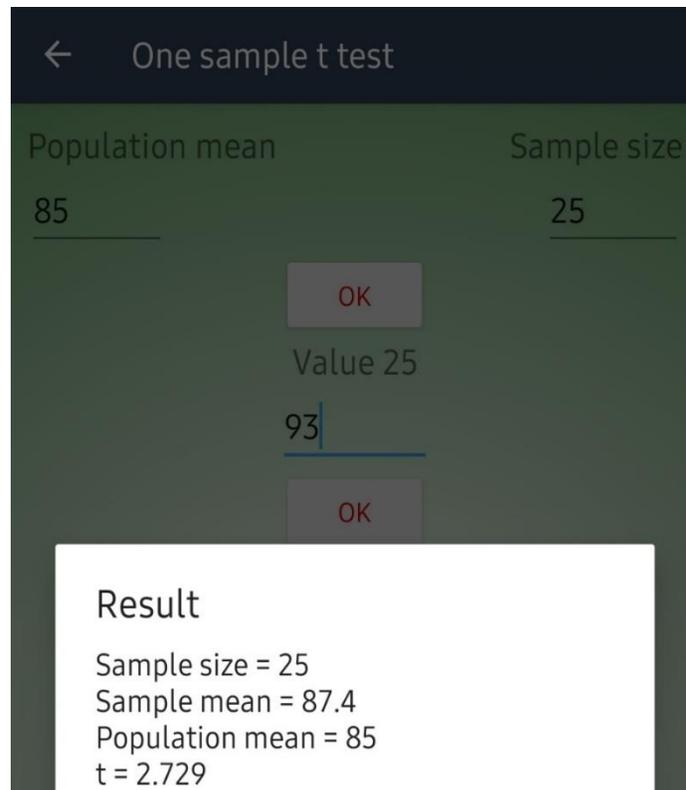


Figura 85. Cálculo de la t Student [Aplicación Statistics Calculator].

**Quinto paso:** En base a la evidencia disponible se acepta o se rechaza la hipótesis alternativa, bajo las siguientes condiciones:

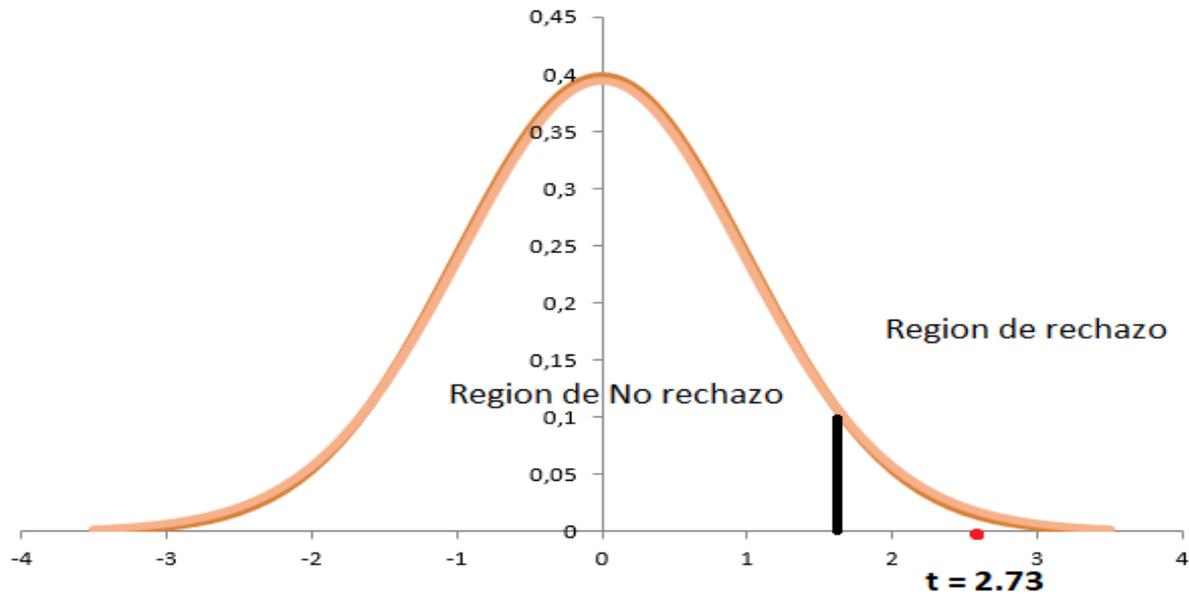
- Si la probabilidad de error (P) es mayor que el nivel de significancia, entonces se rechaza la hipótesis alternativa.
- Si la probabilidad de error (P) es menor que el nivel de significancia, entonces se acepta la hipótesis alternativa.

De acuerdo a los datos obtenidos con la fórmula de t Student procedemos a determinar si la hipótesis alternativa es válida o no. Viendo el resultado del cálculo se observa que:  $t = 2.73$ , se busca el valor más próximo en la tabla de distribución de t Student con el valor más próximo al resultado, tomando en cuenta la columna de grados de libertad.

$p = 0.04$

$\alpha = 0.05$

Siendo:  $p < \alpha$



*Figura 86.* Grafica del resultado de t Student.

#### 4.4. Interpretación de resultados

Como  $P < \alpha$ , es decir  $0.04 < 0.05$ , entonces se concluye que la hipótesis  $H_i$  es aceptable, por lo tanto, la implementación de una plataforma multicanal en línea permitirá optimizar la gestión y seguimiento de tickets para empresas en telecomunicaciones.

## **CAPITULO 5**

### **CALIDAD Y SEGURIDAD**

En este capítulo se determina la calidad de sistema web en base a los parámetros de medición de la norma ISO 9126 y las medidas de seguridad que debe tener el sistema.

#### **5.1. Calidad**

El objetivo es llegar a la mejor calidad necesaria de evaluación bajo la norma ISO 9126, para ello se realiza los cálculos referentes a la funcionalidad, confiabilidad, usabilidad, mantenibilidad y portabilidad del sistema.

##### **5.1.1. Funcionalidad**

Métrica para obtener una valoración mediante el cálculo del punto función en base a la evaluación de un conjunto de características y capacidades que debe cumplir el sistema:

- **Número de entradas del usuario**

Son medios por el cual se alimenta al sistema de información. En la Tabla 24 se lista las entradas de usuario que tiene el sistema.

Tabla 24  
*Lista de entradas de usuario*

N°	Entradas de Usuario
1	Ingreso al sistema
2	Registro de tickets
3	Registro de tickets vía telegram
4	Registro de tickets vía Gmail
5	Registro de tickets vía Central telefónica
6	Registro de usuarios
7	Registro de técnicos
8	Registro de canales telegram
9	Registro de canales Gmail
10	Registro de canales central telefónica
11	Registro de fallas
12	Registro de clientes
13	Registro de permisos
14	Registro de empresas
15	Registro de calendarios Google

**Nota.** Elaboración Propia.

- **Número de salidas del usuario**

Es información elaborada por el sistema para ser mostrada al usuario. En la Tabla 25 se muestra las salidas de usuario que tiene el sistema.

Tabla 25  
*Lista de salidas de usuario*

N°	Salidas de Usuario
1	Gestionar técnicos a tickets
2	Canal de apertura vía Telegram
3	Canal de apertura vía Gmail
4	Canal de apertura vía central telefónica
5	Permisos de acceso
6	Notificación Push de apertura de nuevo ticket

**Nota.** Elaboración Propia.

- **Número de peticiones de usuario**

Son entradas interactivas donde la respuesta es una salida interactiva es decir inmediata. En la Tabla 26 se puede observar todas las peticiones que puede hacer el usuario.

Tabla 26  
*Lista de peticiones de usuario*

N°	Peticiones de Usuario
1	Autenticación de usuario
2	Listar tickets
3	Listar usuarios
4	Listar técnicos
5	Listar canales telegram
6	Listar canales Gmail
7	Listar canales centrales telefónica
8	Listar fallas
9	Listar clientes
10	Listar permisos
11	Listar empresas
12	Listar calendarios Google
13	Editar tickets
14	Editar usuarios
15	Editar técnicos
16	Editar canales telegram
17	Editar canales Gmail
18	Editar canales centrales telefónica
19	Editar fallas
20	Editar clientes
21	Editar permisos
22	Editar empresas
23	Editar calendarios Google
24	Editar gestión técnicos a tickets

**Nota.** Elaboración Propia.

- **Número de archivos**

Son los archivos lógicos, los que tienen el control de la información de la base de datos. En la Tabla 27 se puede ver los archivos lógicos para cada tabla de la base de datos.

Tabla 27  
*Lista de todos los archivos lógicos*

N°	Archivos Logicos
1	Usuarios
2	Tickets
3	Telegram
4	Técnicos
5	Permisos
6	Pbx
7	Fallas
8	Empresas
9	Clientes
10	Chat telegram
11	Cdr report
12	Canales
13	Api Google ticket
14	Api Google

**Nota.** Elaboración Propia.

- **Número de interfaces externas**

Son las interfaces legibles por la máquina para pasar información de un sistema a otro.

En este caso solo son 2, intranet e internet.

- **Factor de ponderación**

Ahora con los anteriores datos obtenidos, se tiene que elegir un criterio de ponderación, para el caso se utilizará el factor medio. Como se puede ver en la Tabla 28.

Tabla 28  
*Factores de ponderación*

Parámetros de medición	Cuenta	Factores de ponderación			Valor Obtenido
		Simple	Media	Complejo	
Número de entradas de usuario	15			6	90
Número de salidas de usuario	6		5		30
Número de peticiones de usuario	24		5		120
Numero de archivos	14		5		70
Número de Interfaces externas	2		2		4
<b>Cuenta total</b>					<b>314</b>

**Nota.** Elaboración Propia.

- **Factor de ajuste de complejidad**

En la Tabla 29 se obtienen los valores de ajuste de complejidad en base a las respuestas de las siguientes preguntas evaluadas entre 0 y 5.

Tabla 29  
Valores de ajuste de complejidad

Factores de complejidad	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y de recuperación fiables?					x		4
¿Se requiere comunicación de datos?						x	5
¿Existen funciones de procesamiento distribuido?						x	5
¿Es crítico el rendimiento?				x			3
¿Se ejecutará el sistema con un entorno operativo existente y fuertemente utilizado?					x		4
¿Requiere el sistema entrada de datos interactiva?						x	5
Facilidad Operativa					x		4
¿Se actualiza los archivos maestros de forma interactiva?				x			3
¿Son complejas las entradas, las salidas, los archivos o las peticiones?					x		4
Procesamiento interno complejo						x	5
Diseño de código reutilizable					x		4
Facilidad de Instalación					x		4
¿Soporta múltiples instalaciones en diferentes sitios?					x		4
Facilidad de cambios					x		4
<b>Factor de ajuste de complejidad</b>							<b>58</b>

Nota. Elaboración Propia.

Ahora, para calcular el valor de punto función (PF), se usa la siguiente relación:

$$PF = CuentaTotal * (0.65 + 0.01 * \sum Fi)$$

Donde:

PF: Es la medida de funcionalidad

CuentaTotal: Es la suma de todas las entradas obtenidas en número de entradas, número de salidas, número de peticiones, número de archivos y número de interfaces externas.

$\Sigma F_i$ : Son los valores de ajuste de complejidad según las respuestas obtenidas en la Tabla 8.

Reemplazando:

$$PF = 314 * (0.65 + 0.01 * 58)$$

$$PF = 386.22$$

Para hallar el punto función ideal al 100% de los factores sería 70:

$$PF \text{ ideal} = 314 * (0.65 + 0.01 * 70)$$

$$PF \text{ ideal} = 423.9$$

Entonces la funcionalidad del sistema es:

$$Funcionalidad = \frac{PF}{PF \text{ ideal}} * 100$$

$$Funcionalidad = 91.1 \%$$

### 5.1.2. Confiabilidad

La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos usados, es decir, representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la privacidad de que un sistema presente fallas:

- Comportamiento con respecto al tiempo: Atributos de software relativos a los tiempos de respuesta y de procesamiento de los datos.

- Comportamiento con respecto a Recursos: Atributos software relativo a la cantidad de recursos usados y la duración de su uso en la realización de funciones.

La función a continuación muestra el nivel de confiabilidad del sistema:

$$F(t) = (\text{Funcionalidad}) * e^{(-\lambda t)}$$

Se ve el trabajo hasta que se observa un fallo en un instante t, la función es la siguiente:

Probabilidad de hallar una falla:  $P(T \leq t) = F(t)$

Probabilidad de no hallar una falla:  $P(T > t) = 1 - F(t)$

Donde:

Funcionalidad = 0,91

$\lambda = 0.14$  (1 error cada 7 ejecuciones)

Tomemos un tiempo t de 12 meses

Ahora hallando la confiabilidad del sistema:

$$F(12) = (0.91) * e^{(-0.14*12)}$$

$$F(12) = 0.169$$

Remplazando en la fórmula de no hallar una falla se tiene:

$$P(T > t) = 1 - F(t)$$

$$P(T > t) = 1 - 0.169$$

$$P(T > t) = 0.83$$

Con este resultado podemos decir que la probabilidad que el sistema no presente fallas es de 0,83 y que presente fallas es del 0,17.

$$\text{Confiabilidad} = 83 \%$$

Con este resultado se concluye que la aplicación tiene un grado de confiabilidad del 83% durante los próximos 12 meses.

### 5.1.3. Usabilidad

Se toma en cuenta la capacidad del sistema de ser comprendido, utilizado y atractivo para el usuario, en la Tabla 30 se muestra los resultados obtenido en base a las preguntas propuestas.

Tabla 30  
*Preguntas para obtener el grado de usabilidad*

N°	Pregunta	Valor Obtenido
1	¿Es entendible?	90
2	¿Las pantallas son agradables a la vista del usuario?	92
3	¿Es fácil de aprender?	90
4	¿Contiene la información necesaria?	90
5	¿Facilita su trabajo?	90
6	¿Los controles son fáciles de operar?	85
7	¿La navegabilidad es fluida?	90
<b>Promedio</b>		89.6

**Nota.** Elaboración Propia.

$$Usabilidad = 89.6 \%$$

### 5.1.4. Mantenibilidad

La Mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software.

Para calcular el índice hacen falta una serie de medidas anteriores:

Mt = número de módulos en la versión actual.

Fm = número de módulos en la versión actual que han sido modificados.

Fa = número de módulos en la versión actual que han sido añadidos.

Fe = número de módulos de la versión anterior que se han eliminado en la versión actual.

A partir de estas, el IMS se calcula de la siguiente forma:

$$IMS = \frac{[Mt - (Fa + Fm + Fe)]}{Mt}$$

Reemplazando:

$$IMS = \frac{[12 - (0 + 1 + 0)]}{12}$$

$$IMS = 0.91$$

La interpretación a este resultado establece un 91%, lo que indica que no requiere de mantenimiento inmediatamente.

$$\text{Mantenibilidad} = 91\%$$

### 5.1.5. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes sub características:

- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Capacidad para ser instalado.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Capacidad para ser reemplazado.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Para el cálculo de la portabilidad se tomó en cuenta la siguiente tabla que contiene las características que se mencionaron anteriormente.

Tabla 31  
*Factores de portabilidad y sus porcentajes*

<b>Factor de portabilidad</b>	<b>Valor %</b>
Puede ser transferido de un entorno a otro	95
Se puede adaptar a otros ambientes con facilidad (Instituciones similares)	100
Es fácil de instalar	95
Es capaz de reemplazar a una aplicación similar	90
<b>Total</b>	<b>95 %</b>

**Nota.** Elaboración Propia.

El resultado nos dice que nuestro sistema tiene la capacidad de ser transferido de una forma efectiva y eficiente a un 95%.

$$Portabilidad = 95\%$$

### **5.1.6. Resultados**

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente. La calidad está directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

Tabla 32  
*Resultados de la calidad*

<b>Características</b>	<b>Resultados</b>
Funcionalidad	91.1 %
Confiabilidad	83 %
Usabilidad	89.6 %
Mantenibilidad	91 %
Portabilidad	95 %
<b>Evaluación total</b>	<b>89.94 %</b>

**Nota.** Elaboración Propia.

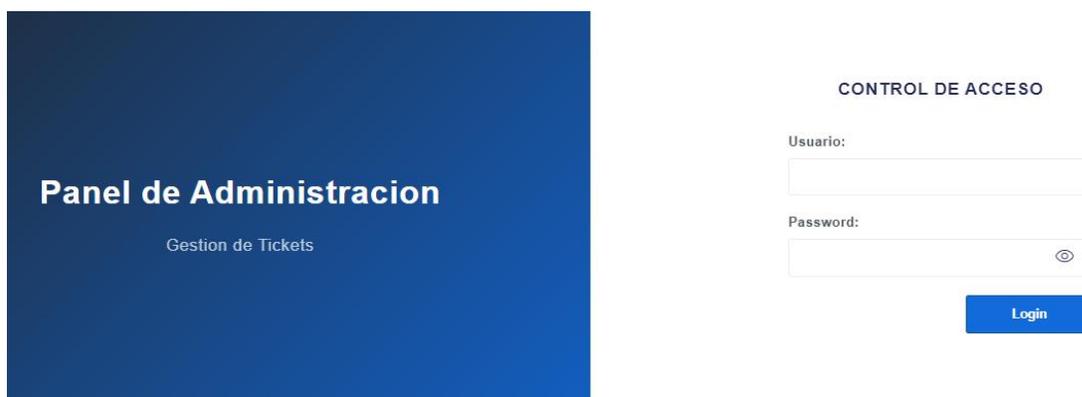
*Evaluacion de la calidad total = 90 %*

## 5.2. Seguridad

Los problemas de seguridad para sitios y sistemas web deben ser contemplados desde el momento del diseño lógico. A continuación, se detalla todas las medidas que se tomó al momento de implementar el sistema.

### 5.2.1. Autenticación de Usuarios

Los únicos que tienen acceso al sistema web son las empresas y los usuarios propios de cada una de ellas, estos están registrados en el sistema como usuarios y cada uno tiene un único nombre de usuario y contraseña que deben introducir en un formulario de Login.



*Figura 87.* Login del sistema.

### 5.2.2. Firewall Linux (Shorewall)

Shorewall es una robusta y extensible herramienta de alto nivel para la configuración de muros cortafuego. Shorewall solo necesita que le proporcionen algunos datos en algunos ficheros de texto simple y este creara las reglas de cortafuegos correspondientes a través de iptables.

Se configuro Shorewall de la siguiente manera:

#### *Shorewall.conf*

En Linux este archivo se encuentra en `/etc/shorewall/shorewall.conf`, del cual se modificó:

- `STARTUP_ENABLED` se utiliza para activar shorewall, de manera predefinida viene desactivado, solo basta cambiar NO por YES.
- `CLAMP MSS` se utiliza en conexiones PPP (PPTP o PPPoE) y sirve para limitar el MSS (Maximum Segment Size), se cambia su valor de NO por YES, shorewall calculara el MSS más apropiado para la conexión.

#### *Zones*

Su ubicación `/etc/shorewall/zones`.

Este fichero se utiliza para definir las zonas que se administran con shorewall y el tipo de zona (firewall, ipv4 o ipsec). La zona fw está presente en el fichero `/etc/shorewall.conf` como configuración predefinida. A continuación, se muestra la configuración de las zonas de internet (net), red local (loc) y zona desmilitarizada (dmz).

```
#####
#ZONE TYPE          OPTIONS          IN          OUT
#                   OPTIONS          OPTIONS
fw    firewall
net   ipv4
loc   ipv4
dmz   ipv4
#LAST LINE - ADD YOUR ENTRIES ABOVE THIS ONE - DO NOT REMOVE
```

*Figura 88.* Configuración del fichero zone.

### ***Interfaces***

Su ubicación /etc/shorewall/interfaces.

En este fichero se establecen cuáles serán las interfaces para las diferentes zonas. A continuación, se muestra la configuración realizada y en todas se solicita se calcule automáticamente la dirección de transmisión (Broadcast).

```
#ZONE INTERFACE BROADCAST OPTIONS
fw    firewall
net   eth0 detect
loc   eth1 detect
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT
REMOVE
```

*Figura 89.* Configuración del fichero interfaces.

### ***Policy***

Su ubicación /etc/shorewall/policy.

En este fichero se establece como se accederá desde una zona hacia otra y hacia la zona de internet.

```

#SOURCE          DEST          POLICY          LOG
      LIMIT: BURST
#
fw              all          ACCEPT
net            all          DROP
loc           net          ACCEPT
loc           fw          ACCEPT
all           all          REJECT
#LAST LINE -- DO NOT REMOVE

```

*Figura 90.* Configuración del fichero policy.

### *Masq*

Su ubicación /etc/shorewall/masq.

Se utiliza para definir a través de que interfaz o interfaces se habilitara enmascaramiento o NAT y para que interfaces o redes se aplicara. A continuación, se muestra la configuración realizada.

```

#INTERFACE      SOURCE          ADDRESS          PROTO PORT (S)
      IPSECMARK
eth0            eth1
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT
REMOVE

```

*Figura 91.* Configuración del fichero masq.

## CAPITULO 6

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1. Conclusiones

Luego de haber culminado con el desarrollo de la plataforma multicanal en línea, aplicando todas las metodologías de análisis y diseño de software en base a los problemas y objetivos planteados se puede afirmar que se han cubierto las metas propuestas.

Por lo que se concluye:

- Se logro implementar una plataforma multicanal en línea que permitirá a las empresas ofrecer a sus clientes la atención de sus reclamos mediante tickets desde cualquier parte del departamento, garantizando un acceso 24/7 a todos los usuarios.
- Se logro implementar distintos canales de atención como ser: Telegram, Gmail (correo electrónico), Central telefónica (atención por menú interactivo de voz u operadores) con el sistema para que los clientes tengan varias opciones al momento de realizar la apertura un reclamo de manera más cómoda y segura.
- El sistema implementado logra su propósito de administrar e integrar todos los canales de atención además de la gestión y seguimiento de los reclamos mediante tickets para los clientes de distintas empresas lo que permite una atención más eficiente y oportuna para los usuarios.

- La plataforma logra ofrecer un esquema de trabajo remoto, en línea a todos los operadores de la empresa lo que ayuda a agilizar los procesos de atención, seguimiento y solución de los reclamos.

## **6.2. Recomendaciones**

Las recomendaciones propuestas para mejorar la plataforma desarrollada son las siguientes:

- Integración de más canales de atención, por ejemplo, WhatsApp, etc.
- Integración de reportes como medida de seguimiento o gráficos estadísticos que muestren a cuantos tickets se dio solución en un periodo de tiempo.
- Expandir a otros ámbitos la solución propuesta, como la industria alimenticia, ministerios, fabricas, etc. Manteniendo los canales de atención y realizando cambios a la estructura del sistema.
- Implementar un módulo de llamadas telefónicas desde el sistema, que permita la comunicación inmediata entre el operador y los clientes.

## BIBLIOGRAFIA

Abud, M. A. (2016). *Calidad en la Industria del Software. Obtenido de La Norma ISO-9126.*

Aplicaciones Web Progresivas (s.f.).

Recuperado de: [https://developer.mozilla.org/es/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/es/docs/Web/Progressive_web_apps)

Bartó, A. (2005). *Introduccion a Shorewall*. Córdoba:

Recuperado de: <https://silو.tips/download/introduccion-a-shorewall>

Booch, G. y Rumbaugh, J. y Jacobson, I. (1999). *El lenguaje unificado de modelado*.

Madrid, España: Editorial Pearson Educación S.A.

Booch, G. (1994). *Análisis y Diseño Orientado a Objetos*. Estados Unidos: Díaz de Santos

Bots (s.f.). Recuperado de: <https://core.telegram.org/bots>

Cáceres, J. (2009): *Patrones de diseño: ejemplo de aplicación en los Generative Learning*

*Object*. Revista de Educación a Distancia, número monográfico X. Consultado en

<http://www.um.es/ead/red/M10>

Codeigniter Web oficial. (2014-2019). Recuperado de:

<https://www.codeigniter.com/userguide3/index.html>

Fernández, Y., & Díaz, Y. (2012). *Patrón Modelo-Vista-Controlador*. Recuperado de:

<https://www.diariodecolima.com/contenido/suscriptor/15-42-2-PB.pdf>

Grönroos, C. (1994). *Gestionar las relaciones con los clientes con ánimo de lucro:*

*La dinámica de la calidad de las relaciones*. Estados Unidos: Editorial MCB UP Ltd.

Goncalves, F. E. y Iancu, B. (2016). *Building Telephony Systems with OpenSIPS -*

*Second Edition.*: Editorial Packt Publishing.

Kendall, K. y Kendall, J. (2011). *Análisis y Diseño de Sistemas - Octava Edición*.

México: Editorial Luis Miguel Cruz Castillo.

- Kroiß, C. y Koch, N. (2008). *The UWE Metamodel and Profile – User Guide and Reference*.  
Recuperado de: <https://uwe.pst.ifi.lmu.de/download/UWE-Metamodel-Reference.pdf>
- Landivar, E. (2008-2009). *Comunicaciones Unificadas con Elastix Volumen 1*.  
Guayaquil, Ecuador: Editorial Edgar Land'var.
- Landivar, E. (2008-2009). *Comunicaciones Unificadas con Elastix Volumen 2*.  
Guayaquil, Ecuador: Editorial Edgar Land'var.
- Menzinsky, A. (23 de 03 de 2015). *¿Qué es el Sprint 0?*. Obtenido de Blog de un apóstol del  
Scrum Kanban: <http://scrum.menzinsky.com/2015/03/que-es-el-sprint-0-actualmentehay.html>
- Muñoz, A. (2010), *Elastix a Ritmo de Merengue - Versión 1.3*. xxxx: Editorial Alfio Muñoz.
- Navarro, A., Fernández, J. D., & Morales, J. (2013). *Revisión de Metodologías para el  
Desarrollo de Software*. Obtenido de  
[https://www.researchgate.net/publication/273302003\\_Revision\\_de\\_metodologias\\_agiles\\_ para\\_el\\_desarrollo\\_de\\_software](https://www.researchgate.net/publication/273302003_Revision_de_metodologias_agiles_para_el_desarrollo_de_software)
- Nolivos, G., Coronel, F., & Campaña, M. (2010). *Implementación de un Sistema Web para el  
Control de un Taller Técnico Automotriz en Plataforma PHP - MySQL utilizando  
UWE para la Empresa METROAUTOCERFRAN Cia. Ltda*. Ecuador: Escuela Politécnica  
del Ejército.
- OpenSips Web Oficial. (2021). Recuperado de: <https://www.opensips.org/>
- Palacio, J. (2014). *Gestión de Proyectos Scrum Manager*. Obtenido de:  
[http://www.scrummanager.net/files/sm\\_proyecto.pdf](http://www.scrummanager.net/files/sm_proyecto.pdf)
- Pérez, H. F. (2010). *Propuesta de Análisis y Diseño Basada en UML y UWE para la  
Migración de Arquitectura de Software Centralizada hacia Internet*. Guatemala:

Universidad de San Carlos de Guatemala. Facultad de Ingeniería. Escuela de Ingeniería en Ciencias y Sistemas.

Pressman, R. (2007). *Ingeniería de Software 6ta Edición*. Mac Graw Hill.

Rzutkiewicz, J. y Lockhorn, J. (2021). Recuperado de:

<https://www.webscale.com/blog/pwa-explained-expect-2021/>

Sharif, B. (2008). *Elastix Without Tears*. Australia: The Elastix.

Telegram Bot Api. (2013-2021). Recuperado de: <https://core.telegram.org/mtproto>

Welling, L., & Thomson, L. (2005). *Desarrollo Web con PHP y MySQL*. Anaya Multimedia.

# ANEXOS

## ANEXO A: ÁRBOL DE PROBLEMAS

