



Diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías Speech to Text y generación de códigos QR para el control de acceso en una Urbanización

Albán Alvear, Mónica Mishell y Villacrés Veloz, César Augusto

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

Ing. Aguilar Salazar, Darwin Leónidas

09 de enero del 2023

10/1/23, 6:48

Tesis Final

Informe de originalidad

NOMBRE DEL CURSO

Tesis

NOMBRE DEL ALUMNO

MONICA MISHHELL ALBAN ALVEAR

NOMBRE DEL ARCHIVO

Tesis_Albán_Villacrés_Final

SE HA CREADO EL INFORME

9 ene 2023



firmado electrónicamente por:
DARWIN LEONIDAS
AGUILAR SALAZAR

ING. DARWIN AGUILAR SALAZAR
DOCENTE DEEE - TUTOR

Resumen

Fragmentos marcados	10	0,7 %
Fragmentos citados o entrecuillados	9	0,4 %

Coincidencias de la Web

espe.edu.ec	4	0,4 %
movistar.es	7	0,2 %
docplayer.es	3	0,2 %
ipn.mx	2	0,1 %
passeidireto.com	1	0,1 %
doccity.com	1	0,1 %
slideplayer.es	1	0 %



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Certificación

Certifico que el trabajo de titulación, "Diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías Speech to text y generación de códigos QR para el control de acceso en una Urbanización" fue realizado por los señores **Albán Alvear, Mónica Mishell y Villacrés Veloz, César Augusto**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 09 de enero del 2023

Firma:



Ing. Aguilar Salazar, Darwin Leónidas

C. C 1103036826



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Responsabilidad de Autoría

Nosotros, **Albán Alvear Mónica Mishell**, con cédula de ciudadanía 1718439126 y **Villacrés Veloz César Augusto**, con cédula de ciudadanía 1721712758 declaramos que el contenido, ideas y criterios del trabajo de titulación: **Diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías Speech to text y generación de códigos QR para el control de acceso en una Urbanización** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 09 de enero del 2023

Albán Alvear Mónica Mishell

C.C.: 1718439126

Villacrés Veloz César Augusto

C.C.: 1721712758



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Autorización de Publicación

Nosotros **Albán Alvear Mónica Mishell**, con cédula de ciudadanía 1718439126 y **Villacrés Veloz César Augusto**, con cédula de ciudadanía 1721712758, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Título: Diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías Speech to text y generación de códigos QR para el control de acceso en una Urbanización** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 09 de enero del 2022

Albán Alvear Mónica Mishell

C.C.: 1718439126

Villacrés Veloz César Augusto

C.C.: 1721712758

Dedicatorias

Dedico este trabajo de titulación a mi hija Sarah Alejandra por ser el motivo de culminación de mis estudios, de crecer día a día profesionalmente y personalmente, por enseñarme que todo esfuerzo vale la pena y que la superación diaria es fundamental. A José por confiar en mí y en mi esfuerzo, por darme aliento y motivarme en los momentos más difíciles y duros, por brindarme cariño y enseñarme que todo es posible a base de esfuerzo y constancia.

A mi hermana Ma. Alejandra por transmitirme felicidad y por llenarme de momentos alegres en los momentos más oscuros de mi vida, por enseñarme a confiar en mí misma. Gracias hermanita por dedicarme tu tiempo, por ser mi mejor amiga y por creer en mis capacidades.

A mi mamá por dedicarme y ofrecerme todo su tiempo, por inculcarme valores y explicarme en base a su experiencia como reaccionar frente a las adversidades que se presentan en el camino de la vida. Muchas gracias mamá por darme amor, cariño y sobre todo por depositar en mí esperanza y fe para lograr terminar mi carrera.

A mis abuelitos Hugo y Hortencia por estar pendiente de mí y motivarme día a día a superarme, gracias por sus valiosos consejos y experiencia que en base a amor me han formado en una persona llena de valores.

A mi familia en general por brindarme apoyo y confianza para obtener mi título universitario.

A mis amigos, José Luis y Santiago que me acompañaron durante toda la carrera, ustedes me compartieron muchas cosas, entre ellos su valioso conocimiento pero sobre todo me enseñaron que lo fundamental es ser buena persona y tener presente los valores. Gracias por su amistad sincera.

A mi compañero de tesis César por escucharme y comprenderme, por compartir conmigo sus conocimientos para poder finalizar con este trabajo de titulación y por confiar en mis capacidades, dedicación y perseverancia.

Mónica Mishell Albán Alvear

Dedico este proyecto de titulación a mi padre César Villacrés y mi madre Magdalena Veloz que me guiaron por buen camino educándome de la mejor manera para convertirme en un excelente profesional, continuamente buscan que haga las cosas bien, me guían, me corrigen y apoyan incondicionalmente, los adoro con mi alma.

A mis hermanos Héctor García y Alicia Villacrés que a lo largo de mi vida compartimos muchos momentos maravillosos y sé que puedo contar con ellos en diversas situaciones tanto buenas como malas.

A mi novia Estefanía Sandoval que me hizo vivir experiencias inolvidables, me inspira a ser un mejor hombre día a día y su soporte sentimental que me ha dado constantemente en el tiempo que hemos estado juntos.

A mis amigos Santiago Altamirano, Gissela Vega y Richard Requelme que siempre me aconsejaron y trataron que yo tomé las mejores decisiones para mi vida.

Finalmente, a mi querida amiga y pareja de tesis Mishell Albán que me ha dado la confianza para realizar este proyecto de tesis con ella.

César Augusto Villacrés Veloz

Agradecimiento

Agradecemos a Dios por llenarnos de sabiduría, inteligencia y paciencia en el transcurso de nuestra carrera, así como también en nuestro trabajo de titulación.

A nuestros padres por darnos la oportunidad de ser profesionales y por inculcarnos que el esfuerzo y la constancia son muy importantes para alcanzar metas y objetivos. Agradecemos su amor y apoyo incondicional durante toda nuestra carrera y sobre todo por sus consejos, vivencias y experiencia que nos han servido para desarrollarnos tanto personal como profesionalmente.

Expresamos nuestro agradecimiento a hermanos y familiares por extendernos su afecto y confianza día tras día, por llenar nuestros días de alegrías y enseñanzas.

Agradecemos de una manera muy especial a nuestro Director de tesis, Ing. Darwin Aguilar por compartir sus conocimientos y experiencia en el desarrollo del proyecto, así como también por sus consejos y tiempo. De igual forma a nuestro Colaborador del proyecto al Ing. Raúl Haro por ser una persona atenta y constituirse en una guía importante al brindarnos su criterio de manera oportuna.

Reconocemos a la Universidad de las Fuerzas Armadas – ESPE que en conjunto con los profesores contribuyeron en nuestra formación profesional y personal, agradecemos a cada uno de los profesores que semestre a semestre nos llenaron de sólidos conocimientos, así como también supieron forjaron nuestro carácter.

Mónica Mishell Albán Alvear
César Augusto Villacrés Veloz

Índice de contenidos

Informe de Originalidad	2
Certificación	3
Responsabilidad de Autoría	4
Autorización de Publicación	5
Dedicatorias	6
Agradecimiento	8
Resumen	19
Abstract	20
Capítulo I	21
Definición del proyecto	21
Introducción	21
Antecedentes	22
Justificación e Importancia	24
Alcance	26
Objetivos	27
<i>Objetivo general</i>	27
<i>Objetivos específicos</i>	27
Capítulo II	28
Fundamento teórico	28
Introducción	28
Voz sobre IP (VoIP)	28

	10
Telefonía IP	29
<i>Características</i>	30
<i>Ventajas</i>	31
<i>Desventajas</i>	31
Breve historia de la telefonía IP	32
Telefonía IP en el Ecuador	33
Sistema de Telefonía basado en Protocolo de Internet.....	33
Central telefónica PBX IP	34
FreePBX.....	34
Asterisk.....	35
<i>Issabel</i>	35
<i>Fail2ban</i>	35
Códecs	36
<i>Características</i>	36
<i>Códecs de audio</i>	37
<i>Códecs de video</i>	40
<i>Códecs para VoIP</i>	41
Valores de medición.....	44
<i>Escala MOS</i>	44
<i>Modelo E</i>	45
Virtualización	46

Máquina virtual	46
Entorno de Desarrollo Integrado (IDE)	47
<i>Características:</i>	47
<i>Ventajas:</i>	47
Flutter	49
Android Studio.....	49
<i>Lenguaje de programación: Java</i>	50
<i>Lenguaje de programación: Kotlin</i>	50
Plataforma Xcode.....	51
<i>Lenguaje de programación: Objective-C</i>	52
Base de datos: MySQL.....	52
Base de datos: SQLite	53
Base de datos: Firebase	53
Softphone	55
<i>Códigos QR (Quick Response)</i>	55
<i>Reconocimiento de voz</i>	56
Capítulo III.....	56
Análisis, Diseño e Implementación del Sistema	56
Introducción.....	56
Etapas de desarrollo del sistema.....	57
Servicios requeridos de la Urbanización	58

	12
Análisis de la propuesta de diseño e implementación	60
<i>Sistema de control de acceso</i>	60
Análisis sobre la implementación del sistema de control de acceso según los servicios requeridos de la urbanización.....	62
<i>Elección de servidor de telefonía IP</i>	62
<i>Elección de código libre para la programación del softphone</i>	66
Diseño de software	67
<i>Creación del nombre de la app</i>	67
<i>Android Studio</i>	68
<i>Firebase</i>	99
Configuraciones	109
<i>Digital Ocean e Issabel</i>	109
Pruebas de desarrollo del sistema	118
Capítulo IV	121
Análisis de resultados.....	121
Introducción.....	121
Cálculo de la muestra	121
Análisis de la implementación.....	124
Análisis del desempeño y funcionamiento del sistema de control de Acceso considerando QoS.....	132
Análisis de paquetes de la central telefónica	153

Análisis de los servicios ofrecidos por el softphone implementado utilizando métodos objetivos y subjetivos MOS, PSQM y PESQ	155
Análisis de costos.....	160
Conclusiones.....	162
Recomendaciones	165
Trabajos futuros	166
Bibliografía	168

Índice de tablas

Tabla 1 <i>Códecs de audio</i>	37
Tabla 2 <i>Características de códecs para VoIP</i>	42
Tabla 3 <i>Estándares de codificación</i>	43
Tabla 4 <i>Escala MOS</i>	44
Tabla 5 <i>Niveles de confianza NC</i>	123
Tabla 6 <i>Parámetros de Calidad de Servicio</i>	145
Tabla 7 <i>Reportes CDR Issabel PBX</i>	146
Tabla 8 <i>Costos del proyecto de tesis</i>	160
Tabla 9 <i>Materiales usados para el desarrollo del proyecto de titulación</i>	161

Índice de figuras

Figura 1 <i>Componentes de los IDEs</i>	48
Figura 2 <i>Ventajas de Kotlin</i>	51
Figura 3 <i>Estructura del código QR</i>	56
Figura 4 <i>Esquema de herramientas utilizadas para el desarrollo de la aplicación.</i>	59
Figura 5 <i>Creación de droplet en Digital Ocean</i>	63
Figura 6 <i>Paquete básico usado</i>	64
Figura 7 <i>Selección de volumen adicional</i>	65
Figura 8 <i>Vinculación del dominio para el registro del softphone</i>	66
Figura 9 <i>Diagrama de bloques de la estructura de la app</i>	68
Figura 10 <i>Interfaz de autenticación</i>	69
Figura 11 <i>Código para el registro de datos en la central telefónica</i>	70
Figura 12 <i>Interfaz de conexión con la central telefónica</i>	71
Figura 13 <i>Clases Abstractas de los estados de la llamada</i>	72
Figura 14 <i>Función showCall</i>	73
Figura 15 <i>Interfaz del IncomingCallFragment (llamada entrante)</i>	74
Figura 16 <i>Interfaz del VoiceCallFragment (Llamada de voz)</i>	75
Figura 17 <i>Interfaz del VideoCallFragment</i>	76
Figura 18 <i>Interfaz del Call Activity</i>	77
Figura 19 <i>Interfaz del DialpadFragment</i>	78
Figura 20 <i>Interfaz del CallOutgoingFragment</i>	79
Figura 21 <i>Interfaz del Chat Activity</i>	80
Figura 22 <i>Código del RecyclerView</i>	81
Figura 23 <i>Código del ExtensionAdapter</i>	82
Figura 24 <i>RecyclerView del Message Activity</i>	82
Figura 25 <i>Interfaz del Message Activity</i>	83

Figura 26 Código del MessageAdapter.....	84
Figura 27 Código de la Interfa NotificationAPI.....	85
Figura 28 Función sendNotification.....	86
Figura 29 Código de generación del token para cada dispositivo.....	86
Figura 30 Código para envío de notificaciones.....	87
Figura 31 Interfaz del menú de códigos QR.....	88
Figura 32 Interfaz de la generación de códigos QR temporales	89
Figura 33 Generación de código QR temporal.....	90
Figura 34 Código de la función codeCreated.....	91
Figura 35 Interfaz de la generación de código QR permanente.....	92
Figura 36 Interfaz del Lector de códigos QR.....	93
Figura 37 Función processQr	94
Figura 38 Función getData	95
Figura 39 Datos de la lectura del código QR temporal.....	96
Figura 40 Interfaz de speech to text.....	97
Figura 41 Proveedores de acceso.....	99
Figura 42 Código del botón de Google.....	100
Figura 43 Líneas de comando usadas para subir datos a Realtime Database.....	101
Figura 44 Datos extraídos del Realtime Database	101
Figura 45 Datos almacenados en Realtime Database.....	102
Figura 46 Consola de Firestore Database	103
Figura 47 Código empleado para subir datos en Firestore Database.....	104
Figura 48 Código usado para la extracción de datos de Firestore Database	105
Figura 49 Imágenes almacenadas en Storage	106
Figura 50 Código para subir imágenes al Storage.....	106
Figura 51 Venta de Cloud Messaging	107

Figura 52 <i>Código de la Clase Constants</i>	108
Figura 53 <i>Interfaz de Issabel para la habilitación de la cámara</i>	109
Figura 54 <i>Selección de la opción para la habilitación de la cámara</i>	110
Figura 55 <i>Activación de los códecs para videollamada</i>	110
Figura 56 <i>Configuración de la cola de llamadas</i>	111
Figura 57 <i>Creación de extensiones</i>	112
Figura 58 <i>Creación de la clave para cada condómino</i>	112
Figura 59 <i>Creación total de extensiones</i>	113
Figura 60 <i>Instalación de Fail2ban</i>	114
Figura 61 <i>Ficheros de Fail2ban</i>	114
Figura 62 <i>Fail2ban activo</i>	115
Figura 63 <i>Configuración de los parámetros: bantime, findtime y maxretry</i>	116
Figura 64 <i>IPs baneadas</i>	117
Figura 65 <i>IPs baneadas luego de dos intentos</i>	117
Figura 66 <i>Evidencias de la primera fase de prueba</i>	125
Figura 67 <i>Inducción al personal de seguridad</i>	126
Figura 68 <i>Colocación del código QR</i>	127
Figura 69 <i>Colocación de stickers en los parabrisas de lo vehículo de los condóminos del Conjunto Habitacional Benalcázar Oriental</i>	128
Figura 70 <i>Llamadas, videollamadas y mensajes de prueba</i>	129
Figura 71 <i>Escaneo y lectura de códigos QR permanentes por parte del personal de seguridad</i>	130
Figura 72 <i>Proceso de ingreso de visitantes</i>	131
Figura 73 <i>Calidad del audio de la llamada</i>	133
Figura 74 <i>Calidad de la videollamada</i>	134
Figura 75 <i>Comunicación entre guardias y condóminos</i>	135

Figura 76 <i>Instalación de la app en los dispositivos móviles Android</i>	136
Figura 77 <i>Creación de códigos QR</i>	137
Figura 78 <i>Manejo de la app</i>	138
Figura 79 <i>Control de acceso del lugar de residencia</i>	139
Figura 80 <i>Conectividad de la app</i>	140
Figura 81 <i>Generación de notificaciones personales y generales</i>	141
Figura 82 <i>Nivel de complejidad de la lectura de los códigos QR</i>	142
Figura 83 <i>Lectura de códigos QR</i>	143
Figura 84 <i>Mapa de calor MOS y latencia de VoIP Spear</i>	144
Figura 85 <i>Captura de videollamada con Packet Capture</i>	154
Figura 86 <i>Contenido espectral de la llamada 1 mediante la app</i>	156
Figura 87 <i>Contenido espectral de la llamada 2 mediante la app</i>	157
Figura 88 <i>Contenido espectral de la llamada 3 mediante la app</i>	158

Resumen

En marzo del 2021, Cellere y Nuñez (2021) desarrollaron el proyecto de titulación “Diseño e implementación de un sistema de acceso y alarma comunitaria basada en PBX VOIP virtuales para áreas residenciales”, como parte de los trabajos futuros, se planteó la propuesta de que se realice un análisis y desarrollo de un softphone para dispositivos móviles que cuente con las características tradicionales (audio, video, seguridad) y también permita generar notificaciones push para los dispositivos de los usuarios.

En el presente proyecto de titulación se planteó que dicho softphone admita generar y leer códigos QR, en el caso de los residentes tendrán pegado el código QR a manera de sticker en el parabrisa de sus vehículos los cuales serán leídos mediante la cámara del dispositivo del personal de seguridad. Para las visitas, se dispusieron de librerías que faciliten la generación de códigos QR por parte de los residentes (mediante la aplicación del softphone implementada) para poder enviar a los visitantes, posterior a esto, los visitantes mostraron el código QR al llegar a la urbanización y adicionalmente, mediante librerías “Speech to Text” se confirmó la siguiente información: número de casa a la que se dirige y apellido de la familia, es decir se realizó doble validación.

Para el desarrollo del mencionado proyecto se realizó la implementación de un servidor de Telefonía IP en la nube, que, a más de admitir el uso de comunicaciones de voz basado en IP, permitió incorporar el softphone y a su vez desarrollarlo e implementarlo con algoritmos de código abierto para luego modificarlo para la conversión de voz a texto mediante librerías Speech to Text. Para el registro de la información se generó una base de datos mediante una app con el fin de almacenar la información recopilada.

Palabras clave: softphone, códigos QR, Speech To Text, telefonía IP, base de datos.

Abstract

In March 2021, Cellere and Nuñez (2021) developed the degree project "Design and implementation of a community access and alarm system based on virtual PBX VOIP for residential areas", as part of the future work, the proposal of to carry out an analysis and development of a softphone for mobile devices that has traditional features (audio, video, security) and also allows generating push notifications for user devices.

In this titling project, it was proposed that said softphone admits generating and reading QR codes, in the case of residents they will have the QR code pasted as a sticker on the windshield of their vehicles, which will be read through the camera of the staff device. of security. For the visits, there were bookstores that facilitate the generation of QR codes by the residents (through the implemented softphone application) to be able to send them to the visitors, after this, the visitors showed the QR code when they arrived at the urbanization and additionally, through Speech to Text libraries, the following information was confirmed: number of the house to which it was addressed and surname of the family, that is, double validation was carried out.

For the development of the aforementioned project, an IP Telephony server was implemented in the cloud, which, in addition to supporting the use of voice communications based on IP, allowed the softphone to be incorporated and, in turn, to develop and implement it with code algorithms. Open and then modify it for Speech To Text conversion using Speech to Text libraries. For the registration of the information, a database was generated through an app in order to store the information collected.

Keywords: softphone, QR codes, Speech To Text, IP telephony, database.

Capítulo I

Definición del proyecto

Introducción

La Fiscalía General del Estado [FGE] (2021) ha realizado un estudio sobre las estadísticas de robos que se han dado en el Ecuador, entre ellas, robos a personas, domicilios, carros, motos, establecimientos económicos, bienes, accesorios, autopartes de vehículos. Haciendo un énfasis en el robo a domicilios, el horario que presenta mayores robos, es en la noche. De acuerdo a datos provinciales, Pichincha es la provincia que presenta el mayor porcentaje de robo a domicilios, es decir el 21% hasta diciembre del 2021.

El incremento de inseguridad en diferentes zonas del país ha obligado a que las personas busquen una manera de salvaguardar y proteger su vida así como también sus bienes materiales. Las personas que residen en un conjunto habitacional o urbanización pretenden que al salir de sus hogares todas sus pertenencias queden bajo el cuidado del personal de seguridad, sin embargo, se han dado algunos casos en donde los delincuentes han sustraído bienes en diversos horarios. Conviene enfatizar que los controles de acceso que manejan los conjuntos residenciales y urbanizaciones son mediante personal de seguridad quienes deben hacer un registro manual.

Cabe destacar que tener un sistema de control de acceso no es un privilegio o un lujo sino más bien una necesidad que los moradores están obligados a tener, el desarrollo del presente trabajo de titulación “Diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías *Speech to Text* y generación de códigos QR para el control de acceso en una Urbanización” tuvo por objetivo brindar seguridad a los moradores y que su residencia sea mucho más controlada puesto que por medio de la colocación de códigos QR en el parabrisa de los automóviles el personal de seguridad pudo escanear y de esta forma saber

que dicho auto pertenece a la urbanización, el uso librerías Speech To Text ayudaron a verificar el acceso de visitas, cabe señalar que se dispuso de una base de datos para almacenar la información proporcionada y a su vez que pueda servir de respaldo en caso de que algún morador necesite algún dato. Todo este proceso tuvo el objetivo de aumentar la seguridad del Conjunto y agilizar el acceso de moradores y visitas en donde prevalezca la seguridad por medio de tecnología, como lo son smartphones, códigos QR, servidor de telefonía IP, librerías Speech To Text, notificaciones push, etc.

El avance y progreso de la tecnología en cuanto a sistemas de seguridad ha contribuido a que lugares residenciales, instituciones públicas y privadas, negocios dispongan de equipos que ayuden proteger y mantener seguras sus propiedades. Hay que tener en cuenta que la inseguridad no tiene un horario por esta razón es importante un monitoreo en tiempo real. (Aviles & Cobeña, 2015)

Antecedentes

A continuación, se indican algunos trabajos de titulación que se relacionan con sistemas de control de acceso ya sea para áreas residenciales o de trabajo, en dichos trabajos se han utilizado códigos QR y aplicaciones para monitorear controles de acceso, sistemas biométricos que identifiquen el iris de los trabajadores, este sistema es usado ya que los rasgos biométricos son únicos de cada persona y por otro lado una aplicación web para el envío de sms (Short Message Service) entre el personal de seguridad y los habitantes.

La tesis de Mingolla (2019) trata de un sistema de control de accesos en donde los habitantes del *country* tenían que generar el código QR para luego enviar a sus invitados mediante email, los habitantes eran responsables de proporcionar la siguiente información: datos personales, día, hora, cantidad de acompañantes, toda esta información fue almacenada en una base de datos en formato de lista diaria. Al llegar a la guardianía, los invitados presentan dicho código QR el cual es escaneado, luego la aplicación indica la información

proporcionada, así como también quien originó el código. En caso de que el invitado desee salir del country, se repite el proceso de escaneado facilitando el registro de la hora de salida.

De acuerdo a Aguilar y Chulde (2020) realizaron un sistema de control de acceso enfocado en áreas industriales para reconocer y autenticar a los trabajadores en tiempo real a través de la biometría del iris, en el caso de querer acceder a áreas de seguridad dichos trabajadores tendrán que poseer la capacitación suficiente, así como un registro personal en la base de datos. Cabe recalcar que los lectores de iris tienen la función de admitir el acceso únicamente a los trabajadores capacitados para dicha área, de lo contrario se rechazará el ingreso. En dicho trabajo de titulación existieron dos métodos para reportar el acceso a áreas indebidas de los trabajadores hacia el jefe del personal, se trata del envío de memorandos o notificaciones, el primer método fue enviar las notificaciones al celular y el segundo la visualización en sus computadoras.

Según la tesis de Velasco (2015) se desarrolló una aplicación web para gestión de acceso a urbanizaciones privadas en la ciudad de Guayaquil que sirvió para el registro, actualización, eliminación y emisión de reportes para propietarios y visitas. Dicha aplicación fue instalada en la PC del personal de seguridad puesto que por este medio se logró enviar mensajes al correo o sms a los habitantes para informar el ingreso de alguna visita. El objetivo de esta tesis es evitar la delincuencia dentro de la urbanización privada y que los habitantes sientan más seguridad en sus viviendas.

Una vez mencionados los anteriores trabajos de titulación, se puede expresar que la implementación de los sistemas de control de acceso se pueden instalar tanto en áreas de trabajo como en residenciales y que no solo los códigos QR son un medio sino también los sistemas biométricos o la generación de aplicaciones web, por consiguiente, se puede resaltar que el presente proyecto de titulación es considerado más completo y eficiente teniendo en cuenta su objetivo el cual es desarrollar un servidor de telefonía IP y un softphone

personalizado de acuerdo a las necesidades de la urbanización y de los moradores, contar con notificaciones push privadas y generales, generar códigos QR únicos para los vehículos de los moradores y visitas, ya que cada habitante fue el único responsable de la emisión de dichos códigos. Se tuvo doble validación para las visitas mediante librerías Speech To Text, dicho proceso fue realizado por parte del personal de seguridad. Lo que se pretende con la implementación de este proyecto de titulación es brindar seguridad y tranquilidad a los moradores de la urbanización.

Justificación e Importancia

En la Tesis de Cellere y Nuñez (2021) se ha presentado un inconveniente en el ingreso a las Urbanizaciones Residenciales, el cual es el tiempo que tarda el personal de seguridad para registrar el ingreso de cada vehículo de los moradores y visitantes, quienes generalmente deben ser interrogados a fin de poder generar un registro manual con la información proporcionada que se almacenó “manualmente” acorde a la respuesta del conductor de cada vehículo.

Lo que se pretendió es no detener a los vehículos visitantes para preguntarles: ¿A dónde se dirige?, ¿A qué familia va a visitar?, ¿Cuál es el motivo?, y en algunos casos incluso hacer esperar a los visitantes para establecer una llamada y confirmar que la visita ha sido aceptada para permitir el ingreso a la Urbanización.

Por tal motivo se consideró que el desarrollo del presente proyecto es viable con el fin de que permita el ingreso de moradores y visitantes de una manera más segura. Con este proyecto se deseó que el tiempo de acceso a la urbanización sea eficiente y que el tiempo en la garita con los guardias sea mínimo, una consecuencia de esto es que los carros accedan rápido sin hacer tráfico, adicionalmente contar con una base de datos “registro” que permita conocer de primera mano y en cualquier momento las personas que accedieron a la Urbanización.

Por otro lado, el sistema que se desarrolló fue personalizado en cuanto a las necesidades de la urbanización, al hablar de seguridad, se tuvo una base de datos con información de: hora de ingreso, hora de salida, lugar de destino y personas que ingresaron a la urbanización. Esta información sirvió ya que al darse un acto delincuencia se podrá respaldar o realizar la verificación de la información e imágenes disponibles en la base.

La implementación del servidor de telefonía fue evaluada en las viviendas de la urbanización. Esta evaluación se realizó directamente con los habitantes, en consecuencia, se pudieron obtener los resultados y a su vez el usuario tuvo conocimiento del tipo de servidor fue implementado y el uso eficiente de la aplicación, igualmente se brindó una inducción en donde se explicó el manejo del sistema, las ventajas, niveles de estabilidad y escalabilidad, al hablar de estabilidad se engloba lo siguiente: mediciones para garantizar niveles de calidad en cuanto a las llamadas, métricas de mediciones de los niveles de satisfacción de los usuarios, tomas de mediciones de cobertura, ancho de banda, latencia, jitter y configuraciones de QoS. Con respecto a la escalabilidad el sistema va a poder crecer en número de usuarios sin ningún problema en el caso de que se necesiten más usuarios que requieran el servicio o una actualización del softphone o de la central telefónica IP.

Las pruebas de funcionamiento y métricas obtenidas en este análisis fueron contrastadas con recomendaciones como la ITU G.114 en la cual se mencionaron tiempos máximos para retardos o jitter, rangos que permitieron ofrecer una solución confiable para los residentes y usuarios, en general, pudieron hacer uso de este servicio que se propuso como desarrollo.

Esta implementación permitió a la Urbanización residencial contar con sistemas actuales de control de acceso basados en tecnología, que a su vez dispusieron de bases de datos y registros de acceso de forma tal que se pudo implementar junto con “otras” políticas y

disminuir las brechas de inseguridad mismas que actualmente se han incrementado de forma alarmante en la ciudad de Quito.

Alcance

Este proyecto de titulación tuvo el objetivo de realizar una investigación que aporte con datos relevantes mediante el estado del arte de artículos investigativos, tesis, entre otras fuentes de importancia bibliográfica que detallen sistemas de control de acceso en urbanizaciones, así como también las características y funcionamiento de los softphones de código libre existentes en el mercado además de la modificación de su código, en el caso que así sea posible.

Se buscó información sobre los tipos de servidores de telefonía IP, las plataformas en las que pueden operar, los programas más convenientes y fiables a utilizar y su respectiva configuración para integrarlo al funcionamiento general del sistema de control de acceso.

La implementación del servidor de telefonía IP permitió que el softphone realice: llamadas de voz, videollamadas, envío y recepción de mensajes de texto, por otro lado, el servidor alojado en la nube, ayudó a registrar la información de los residentes o visitantes conectados, llamadas realizadas, protocolos utilizados, acceso a nuevos usuarios, entre otras.

De igual modo se propuso diseñar y desarrollar un sistema de control de acceso mediante la modificación del código abierto de un softphone, además que el servidor de telefonía IP dió soporte a la aplicación del softphone para generar códigos QR y usar librerías Speech To Text con el propósito de facilitar el registro de la información en la base de datos. Dicha base se encargó de almacenar todos los registros del acceso y salida de los moradores y visitantes de la urbanización, incluyendo: llamadas, mensajes de texto, horas de ingreso y salida, número de casa a la que se dirige, nombre de la familia, toma de foto facial a las visitas, es por esta razón que se sincronizó en conjunto el softphone con dicho servidor. Por otra parte,

existieron notificaciones push destinadas a todos los habitantes de la urbanización en donde se indicó algún mensaje de alerta o comunicado general, en el caso de tener una visita se informó mediante dicha notificación únicamente al domicilio correspondiente.

Se consideró de suma importancia la base de datos puesto que facilitó y agilitó el trabajo del personal de seguridad, así como también sirvió de respaldo en caso existir alguna anomalía o caso de inseguridad dentro de la urbanización.

Objetivos

Objetivo general

Realizar el diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías Speech to Text y generación de códigos QR para el control de acceso en una Urbanización.

Objetivos específicos

- Implementar un servidor de Telefonía IP virtual alojado en la nube que permita gestionar llamadas, videollamadas y mensajes de texto entre la garita de los guardias y los habitantes en una Urbanización Residencial.
- Diseñar y desarrollar un softphone para un sistema de control de acceso que permita la generación y lectura de códigos QR y conversiones de voz a texto mediante librerías Speech To Text.
- Evaluar y realizar mediciones en sitio sobre el sistema implementado a fin de determinar el óptimo funcionamiento del sistema de control de acceso y el registro de la base de datos.
- Realizar el análisis de la calidad de servicio que ofrece el servidor y el softphone implementado utilizando métodos objetivos y subjetivos MOS y *PSQM (Perceptual*

Speech Quality Measurement), *PESQ (Perceptual Evaluation of Speech Quality)* respectivamente.

Capítulo II

Fundamento teórico

Introducción

En este capítulo se pretende dar a conocer información de conceptos básicos para el desarrollo del proyecto de titulación tales como: VoIP, telefonía IP, así como sus ventajas, desventajas, características, historia, diferencias, además se presentará información sobre la central telefónica PBX IP.

Se hablará sobre máquinas virtuales en conjunto con sus sistemas operativos y lenguajes de programación, también de entornos de desarrollo integrados (IDE) y bases de datos con el propósito de hacer una comparación entre dichas temáticas.

Finalmente se expondrán definiciones sobre softphones, generación de códigos QR y reconocimiento de voz.

Voz sobre IP (VoIP)

Conjunto de normas, dispositivos y protocolos que posibilitan a las señales de voz viajar mediante internet por medio del protocolo IP (Protocolo de Internet), en otras palabras, la señal de voz se envía de forma digital y no de forma análoga. Al hablar de protocolos de voz sobre IP se hace referencia a protocolos que facilitan el envío de señales de voz sobre la red IP, se considera pertinente aclarar que la telefonía sobre IP es un servicio telefónico disponible al público, por otro lado, VoIP es un conjunto de recursos que conceden la comunicación de la voz sobre el protocolo IP (Quintana, 2016).

Telefonía IP

Consiste en un servicio de telecomunicaciones en donde los usuarios pueden establecer llamadas telefónicas mediante redes que usan el protocolo IP, por consiguiente, tener este tipo de comunicaciones implica cierto mecanismo que se encarga de que la voz sea digitalizada y empaquetada para luego poder enviar por la red y concluir con la comunicación en el destinatario final (Chávez, 2014).

La difusión del sistema VoIP se dio entre domicilios por medio de un software el cual requería de Internet para lograr una comunicación entre el PC de las máquinas, por otro lado, las llamadas no tenían un costo, no obstante, era lenta y con ecos. La tecnología ADSL la cual estuvo presente en domicilios y empresas, brindó un ancho de banda que permitió mantener una conversación telefónica PC a PC usando internet y proporcionando una buena calidad de sonido (Aguilar C. , 2015).

Desde el aparecimiento de la tecnología ADSL que permitió una ampliación del ancho de banda posibilitando la comunicación entre teléfonos y computadoras, hasta el actual apogeo de las redes de fibra óptica *FTTH (Fiber To The Home)*, el sistema VoIP tiene una gran acogida debido a que ofrece mejores servicios en cuanto a las llamadas simultáneas o las videollamadas que gracias al crecimiento exponencial del ancho de banda se volvieron servicios muy comunes en las aplicaciones de VoIP (Quintana, 2016).

A continuación se mencionan protocolos usados en redes de comunicación de acuerdo con Quintana (2016):

- IAX (Inter Asterisk eXchange): utilizado para conexiones VoIP entre servidores como por ejemplo Asterisk (servidor PBX) o clientes que manejan IAX2 como protocolo de conexión ya que la primera versión está en desuso, no obstante, facilita el control de un gran número de *streams* y de códecs, en definitiva, es

posible manejar cualquier tipo de dato. El objetivo principal de este protocolo es brindar prioridad a los paquetes de voz sobre una red IP, reducir el ancho de banda que se ocupa en la transmisión de voz y video en la red IP, por último, en un único puerto UDP se da la señalización y el flujo de datos.

- SIP (Session Initiation Protocol): usado para iniciar, modificar y finalizar sesiones interactivas como por ejemplo: voz, mensajería instantánea, video, realidad virtual y juegos en línea. Además, es compatible con comunicaciones en tiempo real, del mismo modo, los usuarios pueden conocer sobre su estado antes de que inicie la sesión.

Como menciona Berruecos et al. (2015) dicho protocolo pertenece a la capa TCP/IP y a su vez guarda relación con el protocolo *SDP (Session Description Protocol)*, así como también existen ciertos protocolos que le dan soporte, como por ejemplo: *RSTP (Real-Time Streaming Procol)* para el control de flujos y sesión, *RSVP (Resource Reservation Protocol)* y *DiffServ (Differentiated Services)* gestionan la calidad de servicio y la reserva de recursos y RTP/RTCP usados para transportar datos en tiempo real.

Características

Desde el punto de vista de Choez y Pérez (2016) se han mencionado características que han sido acogidas en las comunicaciones como se detallan a continuación.

- Basada en estándares internacionales con el propósito de englobar varios equipos, protocolos y códecs.
- Posee elementos de seguridad para que el usuario pueda autenticarse a través de un servidor.
- Al tener los mismos recursos de red se facilitan las conexiones al mismo tiempo.
- Posibilita el envío de información de voz por paquetes IP sobre redes: LAN, WAN con la intención de unir servicios de fax, mensajería, videoconferencias, etc.

- La implementación de un servicio de telefonía IP puede ser sin costo y sobre la misma red LAN y WAN, esto se debe a la misma infraestructura de red de datos.

Ventajas

Según Choez y Pérez (2016) existen aportes que ofrece la telefonía IP, como por ejemplo: agregar datos y voz en una misma red de comunicaciones con el fin de reducir hasta en un 50% el precio que representaría su construcción y mantenimiento, obtener varios números telefónicos sin pagar líneas a los respectivos proveedores mensualmente.

Jalca (2017) plantea que con VoIP es posible realizar una llamada independientemente del lugar en donde se encuentre el usuario siempre y cuando disponga de conexión a internet, esto resulta ser una gran ventaja debido a que si el usuario presenta viajes constantemente a través de su dispositivo móvil puede establecer comunicación mediante el acceso a su servicio de telefonía IP, para sintetizar, VoIP engloba lo siguiente: identificación, repetición, devolución, transferencia de llamadas, servicio de llamadas en espera, etc.

Desventajas

Se manifiesta que los problemas en cuanto al uso de VoIP son debido a limitaciones tecnológicas, como por ejemplo, al referirse de los hogares, las conexiones por modem dificultan el establecimiento de una óptima conversación por VoIP, en el caso de llamadas de emergencia (911) resulta complicado determinar el área geográfica al que pertenece el número telefónico, puesto que la telefonía IP se basa en direcciones IP para determinar un número telefónico, por otro lado, el no contar con estabilidad y calidad en la línea de datos dificulta las conversaciones telefónicas, en otras palabras, la calidad de una conexión VoIP se puede ver alterada por la alta latencia o pérdida de paquetes. En el caso de tener un softphone es de suma importancia contar con un equipo en donde su configuración sea personalizada y no existan programas que creen interferencia en el procesador de dicho equipo, así como también en la comunicación (Jalca, 2017)

Breve historia de la telefonía IP

Para resumir, en Israel en 1995 un grupo de jóvenes iniciaron con la tecnología VoIP a base de la realización de un fuerte trabajo, debido a que la comunicación en ese entonces era de computadora a computadora, luego Volcatec, Inc. lanzó *Internet Phone Software*, el cual fue un softphone cuyo funcionamiento se basaba en que dos PC debían disponer del mismo software y hardware, por otro lado, en ese tiempo las comunicaciones de banda ancha no estaban aptas, entonces dicho lanzamiento no fue un éxito como tal. En el año de 1998, en EEUU, se inició la fabricación de los primeros *ATA/Gateways* por parte de emprendedores, esto facilitó la comunicación de computadora a teléfono convencional y posterior a esto, de teléfono convencional a computadora por medio de adaptadores ATA en cada extremo. La realización de las llamadas se hizo posible a causa de la publicidad tanto al inicio como al final de las llamadas, de este modo, los usuarios no tenían que cubrir ningún gasto, esta fue una forma de probar la calidad de la comunicación. Luego en el año 2000, Mark Spencer un joven universitario crea Asterisk, la cual fue una central telefónica basada en Linux con código abierto (Caldera & Suazo, 2011).

En definitiva, al momento de hablar de las implementaciones que se han dado en Latinoamérica se puede tomar como ejemplo Colombia, como en la Universidad de San Buenaventura en el año 1999 se instaló una red en donde fue posible la transmisión de voz y datos, años más tarde, en 2003 se desarrolló Skype, el cual es un software que brinda a los usuarios la opción de realizar llamadas con una buena calidad y sin costo, siempre y cuando se disponga de conexión a internet (Aguilar C. , 2015).

Sintetizando, la telefonía IP va de la mano con el desarrollo del internet debido a que la transmisión de datos se basa en el protocolo IP, la tecnología IP se da posterior a la VoIP, de esta manera se lograron servicios de comunicación sobre redes IP por medio de teléfonos estándares o gateways. Cabe recalcar que la telefonía IP hace posible que los usuarios se

puedan comunicar por voz a través de toda red que disponga del protocolo IP. El funcionamiento de dicha telefonía, inicia cuando la emisión sonora se comprime a través de un códec de audio, posterior a esto por medio de paquetes IP llega al destinatario, finalmente un códec de audio descomprime la voz con el objetivo de retornar a su estado original (Chávez, 2014).

Telefonía IP en el Ecuador

La evolución de la telefonía fija corporativa se conecta a través del protocolo IP, mientras que en la telefonía fija la Corporación Nacional de Telecomunicaciones (CNT) brinda servicios de troncales telefónicas con el mismo protocolo. En el año 2005, en Ecuador se inician investigaciones sobre la telefonía IP con el propósito de abaratar costos de servicio de Internet, posterior a esto en el 2009 muchos ecuatorianos a nivel nacional optaron por dicho servicio, se puede destacar que la mayoría de los ecuatorianos usan la aplicación de WhatsApp para su comunicación diaria. A nivel empresarial resulta conveniente el uso de telefonía IP debido al bajo costo que representa, es decir que la comunicación entre sucursales de dicha empresa resultaría barata, esto también implica que los clientes puedan contactarse sin existir algún problema. Una vez que apareció el software Asterisk contribuyó con la adopción de telefonía IP en el país debido a su bajo costo y sus servicios, sin lugar a dudas, muchas empresas han dejado de usar servicios brindados por ciertos proveedores calificados (Estrada, et al., 2016).

Sistema de Telefonía basado en Protocolo de Internet

Es importante destacar que las telecomunicaciones hacen posible la transmisión de voz sobre IP, actualmente existen muchas aplicaciones que el usuario puede utilizar para su comunicación a un costo relativamente asequible. Por otro lado, existe un código específico para el cliente el cual fue establecido por proveedores del servicio de telecomunicaciones,

denominado UIT-T E.164 el cual está enfocado al traslado del código a cualquier zona (Santos, 2020).

Central telefónica PBX IP

Plataforma que hace posible la administración de llamadas telefónicas mediante un software libre a través de internet, dada la implementación de la central se puede tener el servicio de mensajería o video conferencias. Además, existe la posibilidad de administrar una gran cantidad de teléfonos, análogos, digitales, IP. De igual forma, las llamadas nacionales e internacionales son posibles gracias al enlazamiento entre troncales que dispongan de VoIP a un menor costo y garantizando una óptima comunicación. Este sistema de la central telefónica depende del software y hardware, del mismo modo, si posee licencia o es libre (Santos, 2020).

Desde la perspectiva de Santos (2020) existen algunas funcionalidades tales como: “configuración de correo de voz”, “servicio de grabación para llamadas”, “identificación y transferencia de llamadas”, “conectividad para troncales SIP”, “instalación de mensajería”, “sala de conferencias”, “sistema compatible para IP”, “teléfonos digitales”, “análogos” y SIP (p.33).

FreePBX

Es un software que funciona junto con Asterisk y algunos otros, tal es el caso de Linux, Dahdi, Postfix, etc con el propósito de, desarrollar, instalar, configurar y controlar un PBX VOIP, por otro lado, es una plataforma basada en código abierto que ayuda al usuario con el registro de archivos de configuración y planes de marcación de una manera rápida. Existen algunos servicios tales como: *operadora automática, distribución de llamadas, salas de conferencias, indicaciones de voz personalizadas, extensiones, compatibilidad con otros idiomas, usuarios remotos, estándares SIP*, etc. En lo que respecta a códecs se manejan los siguientes: *G.722, G.711, G.723, G.729A/B, T.38, AMR-NB, AMR-WB* y en cuanto a protocolos se tienen: *SIP V2, H.323, MGCP*, etc. (Ramírez, 2021).

Asterisk

Desde el punto de vista de Choez y Pérez (2016) se trata de un software que sirve de central telefónica para voz sobre IP el cual es de libre acceso y ha sido usado por empresas para la implementación de proyectos de telefonía, por consiguiente, Asterisk ofrece enlaces de líneas de telefonía convencional o troncales como SIP e IAX, extensiones analógicas, comunicación por conferencia entre algunas personas, correo de voz, respuestas de voz interactiva, integración de sistemas de telefonía, etc.

Issabel

Desde la perspectiva de Castro (2019), es necesario aclarar que Elastix fue una plataforma de software libre de telefonía usada por empresas, dicha plataforma tuvo una buena aceptación a tal punto que fue comprada por la empresa *3Cx Phone*, posterior a esto los creadores de Elastix iniciaron con Issabel. En efecto, Issabel es un software de código abierto en el cual su antecesor fue Elastix y con el pasar de los años ha podido ofrecer comunicaciones unificadas basadas en Asterisk, en otras palabras, incluye PBX IP. Su funcionamiento es mediante Linux con Asterisk, su distribución se basa en Centos, por otra parte, dicha plataforma utiliza protocolos SIP e IAX2.

Este software sirve de conmutador telefónico, posee módulos para llamadas, correo electrónico, video conferencias, chats, etc. (Lavariega, 2020).

Fail2ban

Al implementar una IP - PBX la cual está conectada a internet puede sufrir ataques de fuerza bruta, robo de información de los propios usuarios o alteraciones que conlleven a un mal funcionamiento del sistema, otro tipo de ataques, son de *DoS (Denial of Service)* o *password cracking* en SSH que resultan de gran interés para los atacantes debido a que al realizar un sin número de llamadas telefónicas, la empresa vulnerada es la responsable de pagar miles de dólares gastados en dichas llamadas.

Con el fin de tener un servidor seguro es necesario la instalación de Fail2ban, puesto que monitorea los intentos de registro y las direcciones IP de donde provienen los ataques. Esta herramienta puede ser configurada de acuerdo a las necesidades, por ejemplo, el número fallido de registros puede ser bloqueado por un tiempo definido de acuerdo a los administradores del servidor. Fail2ban se distribuye bajo licencia Linux, por esta razón es que se pueden realizar configuraciones de acuerdo a lo que establece dicha licencia (Peláez, 2016).

Códecs

Son algoritmos que por medio de técnicas de procesamiento convierten la voz analógica en información digital, esto quiere decir, que por medio del muestreo de la voz humana se empaquetan datos para viajar por medio de la red, esto implica que dependiendo del códec usado el ancho de banda puede ser mayor o menor (Villacís, 2011).

Un codificador-decodificador se encarga de comprimir la secuencia de datos para lograr cancelar el eco a fin de minimizar el ancho de banda, por otra parte, a la salida del códec se obtiene una secuencia de datos los cuales serán puesto en paquetes IP de la red para llegar a su destino (Guanga & León, 2012).

Características

Desde el punto de vista de Guanga y León (2012) se establece lo siguiente: “frecuencia de muestreo: número de muestras tomadas en un segundo”, “tamaño de la trama: número de paquetes enviados por segundo”, tasa de comprensión nativa: valor de comprensión de la señal al salir de la codificación”, “tamaño del frame y pay load: se trata de datos digitalizados de cada duración, en cambio el pay load, es la cantidad de datos enviados por paquetes”. Al momento de elegir un códec resulta importante tener presente los siguientes parámetros: “Bit rate, indica la cantidad de información enviada en un segundo”, “Sampling rate, muestra la frecuencia de muestreo de la señal vocal”, “Frame size, indica cada cuantos milisegundos se

envía un paquete con la información sonora”, “MOS, señala la calidad del códec en un rango del 1 al 5”.

Códecs de audio

Según Castorena (2012) son usados para comprimir señales de audio con una mínima cantidad de datos de manera que no se ocupe demasiado espacio y se logre una buena calidad, para luego descomprimir y reproducir. Existen algunos parámetros como se indican a continuación:

- Número de canales: se refiere al número de señales de audio simultáneos que forman el flujo de datos, al hablar del número de canales se debe tener en cuenta lo siguiente: si se tiene un canal, dos canales y seis canales se denominan mono, estéreo y multicanal respectivamente.
- Pérdida de información: sucede cuando algún códec suprime frecuencias de la señal original.
- Número de bits por muestra: señala la precisión con la que se reproduce la señal original y su rango.
- Tipo de compresión: se diferencian compresiones con y sin pérdidas.
- Tasa de bits: indica el número de bits que son necesarios por unidad de tiempo.

En la Tabla 1 que se indica a continuación se detallan los códecs de audio.

Tabla 1

Códecs de audio

Nombre	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones	MOS
--------	-------------	--------------------	------------------------	--------------------	---------------	-----

G.711	PCM (<i>Pulse code modulation</i>)	64	8	Muestreada	Posee dos versiones u-law y a-law.	4.1
G.721	ADPCM (<i>Adaptive differential pulse code modulation</i>)	32	8	Muestreada	Obsoleta	-
G.722	7 kHz audio-coding within 64 Kbit/s	64	16	Muestreada	Divide los 16 kHz en dos bandas usando ADPCM	-
G.722.1	Codificación a 24 y 32 kbit/s con baja pérdida de paquetes	24/32	16	20	-	-
G.723	Extensión de la norma G.721 a 24 y 40 kbit/s para aplicaciones en circuitos digitales	24/40	8	Muestreada	Obsoleta por G.726 y diferente a G.723.1	-
vG.723.1	Dual rate 5.3 y 6.3 kbit/s	5.6/6.3	8	30	Para el codificador de high rate se emplea MP-MLQ	

*(Multipulse
Maximum
Likelihood
Quantization)* y
para el de low-
rate ACELP
*(Algebraic-
Code-Excited
Linear-
Prediction)*

Nombre	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones	MOS
G.726	40, 32, 24, 16 kbits/s ADPCM	16/24/32/ 40	8	Muestreada	ADPCM, reemplaza a G.721 y G.723	3.85
G.727	5-,4-, 3- y 2- bit/sample ADPCM	var.	-	Muestreada	ADPCM relacionada con G.726	-
G.728	Coding of speech at 16 kbit/s using low- delay code excited linear prediction	16	8	2.5	CELP	3.61
G.729	Coding of speech at 8 kbit/s using CS-ACELP <i>(Conjugate-</i>	8	8	10	Bajo retardo (15 ms)	3.92

	<i>Structure</i> <i>Algebraic-Code-Excited Linear-Prediction)</i>					
GSM 06.10	RPE-LTP (Regular Pulse Excitation Long-term Predictor)	13	8	22.5	Usado por la tecnología celular GSM	-

Nota. En la tabla se pueden notar las diferencias existentes entre los códecs bajo los parámetros de: modulación, bit rate, sampling rate, frame size y MOS. Tomado de *Análisis y evaluación de códecs de audio para transmisión de VoIP en infraestructuras de redes sobre Linux* (p.30), R, Guanga y Y, León, 2012.

Códecs de video

H.261. Teniendo en cuenta a Guerrero (2016), fue el primer estándar de comprensión de video, de esta manera, se puede, decir que los códecs posteriores se basan en este.

H.263. En un inicio fue usado para videoconferencia y video por internet, actualmente permite comprimir video en formato Flash el cual es usado en YouTube, Google Video, etc. Mediante este códec se llegó a la estandarización de la capacidad de comprensión de video de escaneo progresivo.

MPEG-4. También conocido como H.264, en cuanto a calidad es superior al H.263 puesto que está orientado a objetos, por otro lado, es compatible con múltiples estándares, dado que, parte de videos de baja resolución hasta DVD.

XVID, 3IVX. Ofrecen una calidad de imagen similar al DVD, en cuanto al factor de comprensión es muy alto gracias al uso del bitrate semejante del VCD o SVCD.

WMV (WINDOWS MEDIA VIDEO). Son la familia de códecs de video de Microsoft, además, este códec tiene la capacidad de almacenar videos de cualquier calidad y bitrate.

VP8. Según Padilla y Zuta (2020), es un códec usado en la codificación y decodificación de videos de alta definición, el cual está en contraste con el H.264 códec.

Códecs para VoIP

Desde la posición de Guanga y León (2012), trata de algoritmos encargados en convertir la voz tanto de la PBX o del teléfono a datos para poder transmitir por medio de una red IP, es necesario recalcar, que la elección del códec juega un papel muy importante debido a que una mala elección puede afectar en la calidad de la voz, puesto que los algoritmos trabajan con la compresión y el ancho de banda. En el caso de un enlace LAN que tenga un ancho de banda mínimo y se elija el códec G.711 el cual dispone de un gran ancho de banda lo que se esperaría es que la voz tenga una baja calidad, es por esta razón, que para este caso resultaría mejor elegir el códec G.729 puesto que al presentar un ancho de banda menor resultaría óptimo para dicho enlace. Se considera importante señalar las características de los códecs para VoIP, es por esta razón, que se ha realizado la Tabla 2 como se indica a continuación.

Tabla 2*Características de códecs para VoIP*

Códec	Características
G.711	Conocido como a-law/ μ -law, sin compresión.
G.723.1	Gran compresión, uso intenso de CPU.
G.726	Buena compresión con poco uso de CPU.
G.729	Excelente relación ancho de banda-calidad, requiere licencia.
GSM	Usado en redes GSM.
LPC-10	Mínimo ancho de banda, voz robótica.
iLBC	Robusto ante pérdida de paquetes.
Speex	Gran flexibilidad, gran uso de CPU.

Nota. La presente tabla detalla las características de los códecs más usados en VoIP. Tomado de *Análisis y evaluación de códecs de audio para transmisión de VoIP en infraestructuras de redes sobre Linux* (p.33), R, Guanga y Y, León, 2012.

Una vez detallados algunos códecs tanto en las Tabla 1 y Tabla 2, se considera importante describir los más utilizados en VoIP, así como:

A-Law y μ -law. De acuerdo al estándar G.711 ambos son considerados como formatos de compresión de datos de voz, usan modulación PCM para que los datos puedan ser muestreados a intervalos regulares. Dicho estándar posee una frecuencia de 8kHz, es decir 8.000 muestras por segundo, en el caso de A-law cada muestra tiene 13 bits y en el de μ -law 14 bits dando lugar a una alta calidad al presentar pequeños errores debido a la cuantización de la señal. El uso de A-law se da en la mayoría de lugares del mundo mientras que μ -law en América del Norte y Japón (Jalca, 2017).

G.723. Citando a Guanga y León (2012), se recomienda usarlo para la transmisión de VoIP en conexiones WAN de bajo ancho de banda, se caracteriza por tener un MOS de 3.9, adicionalmente existen dos versiones: 5,3 kbps y 6,4 kbps y un bajo ancho de banda.

G.723.1. Opera en bajas tasas de muestreo tales como: 5.3 kbps y 6.3 kbps, es de alta calidad y comprensión elevada, su ancho de banda es mínimo. Al usarse con el protocolo H.323 se debe pagar licencias para emplearlo comercialmente.

G.726. Reemplazó al G.721 por disponer de tres velocidades, tales como: 16, 24 y 32 kbps, cabe señalar que, es usado por Asterisk al soportar ADPCM-32.

G.729. Brinda una alta comprensión al presentar un bajo ancho de banda, su MOS es de 4 dando lugar a una buena calidad de voz. Es compatible con el códec G.729a, el cual ofrece una alta calidad de sonido puesto que usa 8 kbps de ancho de banda, lo que implica un significativo procesamiento del CPU.

Gsm. Choez y Pérez (2016) sostienen que este códec es utilizado en los canales de líneas móviles, de este modo, el flujo de datos en una conexión *Full-Rate* es de 13 kbit/s, es decir en este códec lo primordial es la cantidad y no la calidad, por otro lado, al hablar de ancho de banda es mejor opción que el códec *A-law* debido a que consume cinco veces menos; sin embargo, cuando se tiene una mala conexión y por ende una mala calidad el nivel del tiempo de respuesta hace que la voz sea irreconocible.

De acuerdo con la Unión Internacional de Telecomunicaciones [ITU] se han establecido estándares de codificación los cuales se indican en la Tabla 3.

Tabla 3

Estándares de codificación

Códec	Ancho de banda	Periodo Muestreo	Tamaño Frame	Frames Paquetes	Ethernet BW
G.711	Kbps	20ms	160	1	95,2Kbps
G.729 ^a	5,3Kbps	30ms	20	1	26,1Kbps
G.723 ^a	6,4Kbps	30ms	24	1	27,2Kbps
G.726	32Kbps	20ms	80	1	63,2Kbps

Códec	Ancho de banda	Periodo Muestreo	Tamaño Frame	Frames Paquetes	Ethernet BW
G.728	16Kbps	2,5ms	5	4	78,47Kbps
G.722	6,6Kbps	20ms	17	1	28,5Kbps

Nota. La tabla indica los estándares de codificación cubiertos por la ITU. Esta tabla ha sido tomada de *Análisis y evaluación de códecs de audio para transmisión de VoIP en infraestructuras de redes mp/s sobre Linux* (p.28), R, Guanga y Y, León, 2012.

Valores de medición

Escala MOS

Como expresa Ayala y Lomas (2010), resulta ser una recomendación de la ITU-T P800, que en base a toma de muestras subjetivas a través de técnicas ACR (*Absolute Category Rating*) se tiene una escala de calidad de voz como se puede evidenciar en la Tabla 4.

Tabla 4

Escala MOS

Calificación MOS	Calidad	Esfuerzo
5	Excelente	No hace falta esfuerzo alguno.
4	Buena	Es necesario prestar atención, pero no requiere un esfuerzo apreciable.
3	Aceptable	Esfuerzo moderado.
2	Pobre	Gran esfuerzo.
1	Mala	No es posible entender la conversación.

Nota. La tabla muestra las ponderaciones existentes en la escala MOS, las cuales van de forma descendente del 5 al 1. Tomado de *Análisis del desempeño de los codec's de telefonía para voz sobre IP (VoIP)* (p.90), A, Ayala y J, Lomas ,2010.

Para poder determinar la calidad en las redes telefónicas se realiza la prueba MOS en donde las personas involucradas o a su vez que hayan experimentado dicha calidad de audio como por ejemplo en una llamada telefónica, puedan expresar su punto de vista. Los valores pueden oscilar del 1 al 5, en donde valores menores a 3.5 indican una mala calidad y los valores mayores a 4.5 una buena calidad. Por medio de esta prueba se puede analizar valores de latencia, jitter, paquetes perdidos, etc (Guamán & Rubio, 2022).

Modelo E

Es una recomendación de la ITU-T G.107 que se basa en un modelo matemático más objetivo de medición de la calidad de voz, tomando en cuenta ciertos parámetros como por ejemplo: jitter, retardo y pérdida de paquetes. Para dicho modelo se han establecido dos fórmulas (1) y (2) que se indican a continuación.

$$R = R_0 - I_s - I_d - I_e + A \quad (1)$$

Donde:

R_0 : representa la relación a ruido.

I_s : representa la degradación por conversión.

I_d : representa el retardo.

I_e : representa la degradación producida por los elementos de la red, depende del códec y las pérdidas.

A : representa el margen de seguridad.

Finalmente, una vez obtenidos los resultados en el modelo E se puede convertir a escala MOS y de esta manera es posible medir la calidad de la voz.

Virtualización

Hace algunos años a nivel empresarial, se destinaba un presupuesto alto para la renovación de equipos, sin embargo, con el pasar del tiempo se volvían obsoletos y esto reflejaba una pérdida para la empresa, puesto que la tecnología avanza cada día. Hoy en día, la virtualización ayuda a reducir gastos relacionados a sistemas informáticos. La virtualización es una tecnología de software que mediante la ejecución de varias máquinas virtuales en la misma máquina física permite compartir sus recursos. Esto quiere decir que, el servidor virtual posee recursos que se acogen al servidor físico, por ejemplo: placa madre, procesador, memoria, etc. (Castro, 2019).

Una vez mencionadas las diferentes centrales telefónicas, se eligió trabajar con Issabel PBX por brindar una mayor facilidad y fiabilidad en el servicio, puesto que, con Asterisk y FreePBX se tuvieron problemas tales como: registro de softphones, actualización del dialplan, activación constante de module load chan_sip cada que se reiniciaba el servidor de Asterisk. Issabel PBX ofrece una interfaz gráfica en donde la configuración resulta intuitiva, en cuanto al uso y realizar las modificaciones son más fáciles.

Máquina virtual

Desde el punto de vista de Palmieri (2015), es un sistema de virtualización que depende tanto de hardware como de dispositivos físicos, consta de CPUs virtuales, tarjetas de red, discos, etc., que en la configuración el usuario puede determinar la capacidad que desea.

Dicha máquina se ejecuta sobre un equipo físico que en muchos de los casos es denominado *host*, el mismo que comparte su hardware de una manera separada. Al hablar de la funcionalidad se puede decir que algunos sistemas operativos se ejecutan simultáneamente, así como también los servidores (Soria, 2013).

Entorno de Desarrollo Integrado (IDE)

Desde la perspectiva de Ponce (2016), se refiere a un programa el cual abarca todo un sistema encargado de incrementar la productividad y brindar componentes necesarios para la realización de interfaces de usuario. Además, es posible editar, compilar, depurar y construir una interfaz gráfica a través de distintos lenguajes de programación, por ejemplo: C++, Delphi, Visual Basic, etc. El objetivo de los IDE se basa en tener una misma estructura de servicios con el fin de minorar tiempos de ajustes para lograr una productividad de desarrollo. A continuación, se indican algunos IDEs usados en diferentes lenguajes de programación, como lo son: Xcode, NetBeans, Eclipse, Codenvy.

De esta manera resulta conveniente detallar características y ventajas de conformidad con Ponce (2016), a fin de brindar un conocimiento claro sobre el entorno de desarrollo integrado.

Características:

- Manejo de algunos lenguajes de programación, así como, diferentes idiomas.
- Posee extensiones y componentes.
- Facilita la importación y exportación de proyectos.
- Depurador.

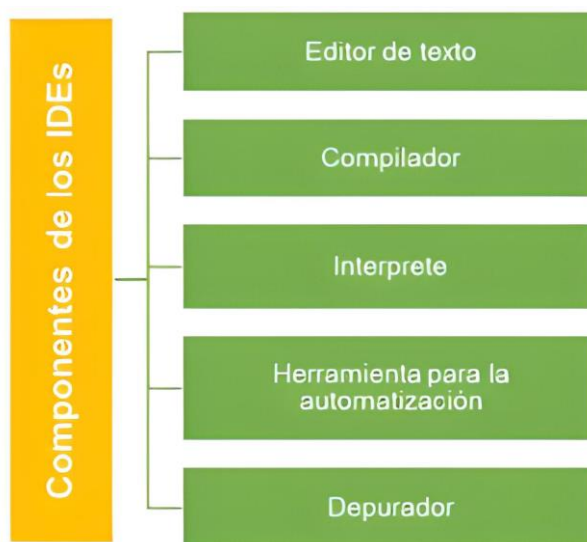
Ventajas:

- El código fuente se encuentra acoplado con el lenguaje de programación.
- El desarrollo de la programación resulta más fácil debido a que el usuario tiene opciones de auto completado de código fuente.
- En el caso de tener errores al compilar el programa, se pueden solucionar acudiendo a una lista de métodos en donde existen posibles soluciones, sin embargo, también existen alertas de mensajes de error de sintaxis.

En la Figura 1 se muestran algunos componentes que forman parte del IDE, tales como: editor de texto, compilador, interprete, herramienta para la automatización, depurador.

Figura 1

Componentes de los IDEs



Nota. En la presente figura se pueden visualizar los componentes que forman de los IDEs.

Obtenido de (Ponce, 2016).

En concordancia con Ponce (2016), se considera pertinente detallar cada uno de los componentes como se puede visualizar a continuación.

- Editor de texto: es una herramienta de escritura que a su vez sirve para la gestión de código o para la escritura de notas rápidas. En otras palabras, dicho editor ayuda a la lectura del código fuente posibilitando la visualización de palabras reservadas, variables y sentencias.
- Compilador: software encargado de compilar un programa con su respectivo lenguaje de programación para luego entregar como resultado un lenguaje de máquina.

- Interprete: cumplen la función contraria al compilador, es decir los procesos se realizan instrucción por instrucción.
- Herramienta para la automatización: se pueden construir aplicaciones al disponer de componentes los cuales poseen elementos que hacen posible la automatización de dichas aplicaciones.
- Depurador: herramienta que ayuda con la depuración del código fuente a través del análisis de futuras amenazas en el software, es decir, dicha depuración colabora con el proceso de prueba-eliminación de errores existentes.

Flutter

Es un *framework* desarrollado por Google a finales del año 2018 para aplicaciones móviles multiplataforma, el lenguaje de programación fue desarrollado de la misma forma por Google y se denomina *Dart*, es de código abierto y permite a los programadores realizar aplicaciones para Android y para iOS. Existen *widgets* para la creación de botones, entradas de texto, animaciones y controles deslizantes, los cuales pueden ser personalizados según la necesidad del programador y la temática que trate la aplicación con el propósito de que sea atractiva y novedosa (Torres, 2021).

Android Studio

Como lo hace notar Quezada (2020), es un IDE que contribuye con el desarrollo y compilación de apps además de facilitar la programación mediante sus distintas herramientas, por consiguiente, por tener características complejas depende de los requisitos de computadora. Cabe señalar que, es un IDE oficial para el desarrollo de software de la plataforma Android. Posee un sistema sólido haciendo posible las compilaciones y configuraciones, es preciso señalar, que dicho IDE facilita la integración de *Firebase*.

Lenguaje de programación: Java

Disponible desde el año de 1995, este tipo de lenguaje es orientado a objetos para el desarrollo de plataformas, su función es compilar el código proporcionado por el programador para luego ejecutar en un lenguaje que sea comprendido por la máquina. Java es semejante a C y C++ lo cual facilita la programación, también permite la multi-verificación de código para evitar errores y garantizar un correcto funcionamiento, al ser independiente de su sistema operativo se puede ejecutar en Windows, Mac, Linux, etc.; al poseer una programación de alto rendimiento se debe a la configuración dinámica en la máquina virtual (Montalvo, 2018).

Utiliza la máquina virtual de Java para establecer una comunicación entre la aplicación y el sistema operativo, sin embargo, es independiente a dicho sistema, por otra parte, se caracteriza por su portabilidad y escalabilidad, debido a que puede ser usado en dispositivos inteligentes, móviles, aplicaciones web, etc. (Andrade y Manosalvas, 2014).

Lenguaje de programación: Kotlin

Es un lenguaje oficial orientado al desarrollo de aplicaciones móviles por la compañía JetBrains, lo que se pretende es que en un futuro se pueda sustituir el uso de java, con el fin de tener aplicaciones que superen la velocidad en comparación de las que antiguamente fueron programadas en Java, además Kotlin posibilita el desarrollo de aplicaciones de escritorio multiplataforma (Enríquez, 2018).

Como expresa Narváez et al. (2019), es un lenguaje que depende de la máquina virtual de java, por lo tanto, interactúan módulos programados en java y otros en Kotlin, haciendo posible la comunicación entre ellos.

Mediante dicho lenguaje de programación es posible reducir líneas de código, así como también mantener una programación orientada a objetos o funcional. En la Figura 2 se indican algunas ventajas que ofrece Kotlin.

Figura 2

Ventajas de Kotlin



Nota. La figura presenta alguna de las ventajas que ofrece el lenguaje de programación Kotlin.

Tomado de (Narváez, et al., 2019).

Plataforma Xcode

Es un IDE que se encuentra de forma gratuita en conjunto con el sistema operativo MAC y actúa como plataforma oficial de Apple, trabaja en conjunto con *Interface Builder*. Xcode facilita la programación en distintos lenguajes tal como: C, C++, Java, AppleScript, etc. Dicha plataforma distribuye el proceso de construcción del código fuente entre varios ordenadores, además es posible realizar aplicaciones para iPhone mediante mecanismos que ofrece la misma plataforma como la añadidura de efectos y plantillas de acceso de acuerdo a los requerimientos. El programador dispone de algunas ventajas al trabajar en esta plataforma, así como: archivos y paquetes son generados de forma automática y a su vez colocados en carpetas de acuerdo al código, si se trata de la creación del aplicativo, resulta sencillo la modificación de parámetros, versión del software, orientación, en pocas palabras, el tipo de

programación depende de la versión utilizada, por lo que resulta mejor emplear versiones actuales (Villalba, 2013).

Lenguaje de programación: Objective-C

Diseñado para programación orientada a objetos por medio del estándar ANSI del lenguaje C, es decir soporta su misma sintaxis, de igual manera, es posible definir ficheros de cabeceras, de código fuente, clases, métodos o estructuras con el fin de impulsar la extensión dinámica de clases (Vergara, 2018).

Si bien es cierto, la opción de Flutter parece muy llamativa al momento de diseñar una aplicación multiplataforma a partir de un mismo lenguaje de programación para Android, sin embargo, se pueden presentar problemas de compatibilidad a largo plazo puesto que cada plataforma tiene actualizaciones constantes, lo que generará un retraso en cuanto a mejorar y depurar la aplicación, por otra parte, Android Studio ofrece las facilidades de un IDE nativo siendo confiable con actualizaciones constantes.

Android Studio utiliza tanto Java como Kotlin para programar aplicaciones, en este caso, resulta mejor utilizar Kotlin puesto a que es la recomendación de Google debido a que todas las aplicaciones están migrando a este nuevo lenguaje de programación. Actualmente existe diversa documentación que facilita el diseño, por otro lado, dicho lenguaje es más eficiente al ofrecer facilidades que antes con Java no se tenía, como por ejemplo la reducción de código en muchos comandos y la eliminación del “;” al final de cada sentencia.

Base de datos: MySQL

Trata de un sistema de gestión de bases de datos que es considerada la más popular a nivel mundial en lo que respecta a desarrollo web, de hecho, para usar dicha base no se requiere disponer de un gran conocimiento, así como tampoco requiere la inversión de una gran cantidad de dinero en capacitaciones, lo que se requiere es un manejo básico de base de

datos SQL, por consiguiente, existen capas de seguridad en donde la información almacenada en el sistema es protegida (Quillay, 2018).

Es una base de datos con código abierto que se puede usar en conjunto con PHP y Apache, de manera que, al visualizar los datos resulte fácil y sencillo comprender debido al formato usado, gracias a su diseño es posible realizar procesamiento de consultas complejas (Jimenez, 2017).

Base de datos: SQLite

Es un motor de base de datos SQL que maneja una gran cantidad de datos puesto que trabaja directamente con aplicaciones, funciona en sistemas operativos de 32 y 64 bits. El software de esta base de datos es de fácil integración tanto en dispositivos como televisores o teléfonos móviles, en cuanto a la instalación únicamente se requiere la descarga de bibliotecas SQLite, resulta una base de datos eficiente al cargar solo los datos que necesita evitando leer todo el archivo y mantener en la memoria, es decir, el contenido se actualiza de manera continua (Tomalá, 2020).

Base de datos: Firebase

Como lo hace notar Sani (2020), una base de datos es una colección de datos que siguen un orden, dichos datos pueden ser símbolos. Firebase tiene su *backend* (base de datos y servidor) en la nube, mientras que, la programación se encuentra en el *frontend* (capa de programación), finalmente Firebase ejecuta el backend. Adicionalmente, facilita la creación de aplicaciones de una forma rápida, así como también resultan muy provechosas sus notificaciones para atraer nuevos usuarios.

Desde la posición de Pérez y Torres (2020), Firebase Inc. desarrolló dicha aplicación web con el fin de que programadores realicen aplicaciones, luego de tres años fue adquirida

por Google. En cuanto al almacenamiento de la información se realiza en formato de notación de objetos. A continuación, se detallan los servicios:

- *Firebase Analytics*: da información de la aplicación, así como también, hace posible que los usuarios usen el servicio.
- *FirebaseCloud Messaging*: trata de un servicio de pago para mensajes y notificaciones para Android o iOS.
- *Firebase Auth*: se puede administrar por correo electrónico y contraseña el inicio de sesión para autenticar a los usuarios.
- *Real-Time Database*: permite acceder a la DataBase para generar un *API (Application Programming Interfaces)* el cual se encarga de proporcionar servicios para que sea usado por su propia aplicación con el fin de unir datos de usuario y almacenar en la sección correspondiente.
- *Firebase Test Lab for Android*: facilita la toma de capturas de pantalla.
- *Firebase Storage*: ayuda con la transferencia de archivos sin involucrar la calidad de las aplicaciones.
- *Firebase Crash Reportinf*: detalla los errores creados dentro de la aplicación de forma tal que se puedan corregir a tiempo para evitar el mal funcionamiento en la plataforma.
- *Firebase Notifications*: hace posible el envío de notificaciones hacia aplicaciones móviles.

Entre los gestores de bases de datos consultados la mejor opción es la de Firebase, dado que cuenta con una sincronización directa con Android Studio, en cambio con MySQL y SQLite no se tiene dicha facilidad, además que obliga a tener conocimiento sobre el lenguaje SQL, por esta razón Firebase es más eficiente para la aplicación que se realizó.

Softphone

En definitiva, es una aplicación multimedia que usa VoIP para que el usuario tenga la posibilidad de realizar llamadas desde su PC hacia teléfonos convencionales, celulares u otra computadora por medio de internet, es decir, hace que se transforme la computadora en un teléfono sin tener que pagar ningún costo (López, 2013).

En términos generales es un software usado para llamar a otros teléfonos o propiamente a softphones e incluso recibir llamadas por medio de Volp, dado que está basado en el estándar SIP/H.323. Asimismo, existen algunos softphones basados en SIP como por ejemplo: sipXphone, Express Talk, Zoiper, Calls, Vippie, StarTele (Huallpa, 2017).

Códigos QR (Quick Response)

Según Ávila (2016), fue inventado por Denso Wave, dicho código consiste en barras dimensionales que codifican una gran cantidad de información tanto en dirección vertical como horizontal, la diferencia con un código tradicional radica en el número de datos debido a que la información se encuentra únicamente en dimensión horizontal. A través de la cámara de un teléfono inteligente se escanea la imagen del código QR y luego se procesa con el lector QR con el objetivo de acceder a los datos, dicho lector consiste en un programa que el usuario debe instalar en su dispositivo previamente. Estos códigos pueden ser encontrados en transportes públicos, letreros, tarjetas de visita, ventas de productos, enciclopedias, educación, etc. Actualmente se acostumbra ubicar a dichos códigos en anuncios para que las personas al escanear puedan leer información, ir a páginas web, videos, mapas, direcciones URL, etc. En la Figura 3 se indica que dicho código consta de un cuadrado con tres cuadrados más pequeños ubicados cada uno en tres esquinas independientes, en cuanto a la codificación se trata de un cubo de datos bidimensional, es decir la estructura es en alto y ancho.

Figura 3

Estructura del código QR



Nota. En la figura se observa la distribución de la información del código QR, claves, patrones y posicionamientos. Tomado de (Ávila, 2016).

Reconocimiento de voz

Una de las librerías usadas es la `SpeechRecognition`, su funcionamiento se basa en escuchar la voz emitida por el usuario para luego interpretar el mensaje, posterior a esto, se obtiene una cadena de texto para reconocer los comandos existentes (Santander, et al., 2020).

Capítulo III

Análisis, Diseño e Implementación del Sistema

Introducción

Este capítulo tiene el objetivo de indicar las etapas que se siguieron para diseñar e implementar el servidor de telefonía IP y el softphone, para esto se realizaron algunas investigaciones con el fin de garantizar que los softwares usados sean los más óptimos y eficientes para el desarrollo de la programación. Por consiguiente, en la etapa de la implementación se instaló el softphone en los dispositivos móviles de los habitantes

residenciales y personal de seguridad, conviene señalar que dicha instalación se realizó en smartphones cuyo sistema operativo fue Android.

Etapas de desarrollo del sistema

En el presente proyecto de titulación se pretende diseñar e implementar un servidor de Telefonía IP y un softphone basado en librerías Speech To Text y generación de códigos QR para el control de acceso en una Urbanización, por lo que se refiere, que se necesitarán smartphones para hacer posible la lectura de códigos QR por medio de su respectiva cámara.

Mediante el desarrollo del servidor de telefonía IP el cual está alojado en la nube se registró información de los residentes y visitantes de manera que la base de datos pueda almacenar información como por ejemplo: nombre, número de casa, hora de ingreso y salida. Además, dicho servidor permitió incorporar el softphone, el mismo que admitió realizar llamadas, videollamadas, envió y recepción de mensajes, notificaciones push, etc. Se debe tener en cuenta que, la generación de códigos QR proviene del softphone, sin embargo, se tienen dos casos puntuales, el de habitantes y visitas. En cuanto al primero, todos los vehículos pertenecientes a la urbanización tendrán pegado el código QR en su respectivo parabrisa del vehículo, para el segundo caso, cada habitante tendrá que generar el código QR y enviar a su visita de tal modo que, al llegar a la guardianía el personal de seguridad pueda escanear, así como también confirmar la información proporcionada por las visitas a través de librerías que puedan convertir el audio a texto, toda esta información será almacenada en la base de datos. De esta manera, el personal de seguridad podrá escanear cada código QR generado y simultáneamente llevar un registro en la base de datos brindando seguridad a los moradores de una forma eficiente.

La implementación de este proyecto de titulación se realizó en el “Conjunto Benalcazar Oriental”, el cual se encuentra situado en Antonio de sierra N15-16 y Verde Cruz, dicho conjunto se encuentra compuesto por 107 casas y 107 vehículos. Cabe mencionar que algunos

habitantes poseen más de un vehículo es por esta razón que se tiene el número de vehículos mencionado anteriormente. Se contó con 30 familias para llevar a cabo las pruebas piloto con el propósito de obtener resultados y evaluar el desempeño de la app, central IP y base de datos.

Servicios requeridos de la Urbanización

Inicialmente se expuso el proyecto a la directiva del conjunto de forma tal que, conozcan el funcionamiento y los requerimientos que demanda el sistema de control de acceso, tales como, conexión a internet y smartphones, se debe tener en cuenta que el softphone diseñado va trabajar en Android, dicho sistema operativo es el más común en el mercado. De igual forma se comentaron los beneficios que obtendrán con dicho sistema de control, por otro lado, las personas que conforman la directiva manifestaron algunos inconvenientes existentes los cuales fueron solventados dentro del desarrollo de este proyecto. A continuación se presentan los servicios a ofrecer al conjunto Benalcazar Oriental.

- Instalación y manejo del softphone de fácil interacción con el usuario, que en este caso serían el personal de seguridad y los habitantes.
- Base de datos encargada del almacenamiento de información proporcionada para habitantes y visitantes.
- Contacto eficiente entre el personal de seguridad y los habitantes gracias a los servicios que ofrece el softphone, esto quiere decir, mensajería, notificaciones push (generales y privadas), videollamadas, llamadas, etc.
- Disminución de cola de espera o tráfico generado al acceder a la urbanización, teniendo en cuenta que no existen registros manuales.
- Los habitantes serán quienes generen códigos QR mediante el softphone, de esta manera podrán enviar a sus visitas con el propósito de que su acceso sea sencillo, consecuentemente, se convertirán en los responsables directos. Por

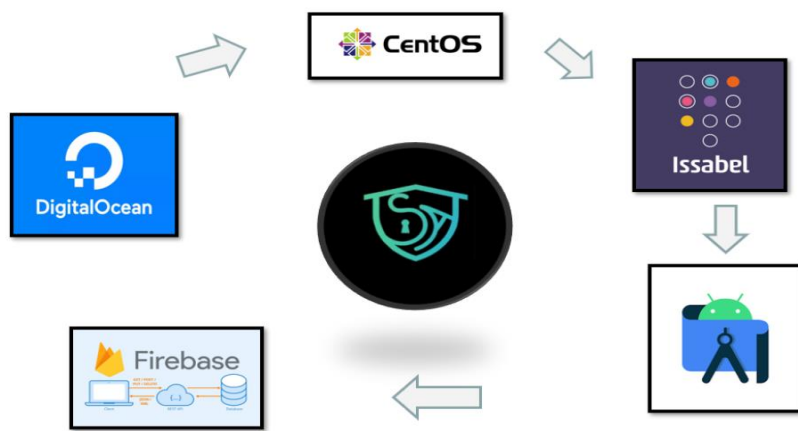
otro lado, mediante librerías que ayuden con la conversión de audio a texto, la información proporcionada por parte de las visitas también será guardada en la base de datos, es decir, habrá doble confirmación de acceso.

- Cada automóvil dispondrá de un único código QR el cual será colocado en el parabrisa, independientemente si una familia dispone de más de un automóvil, los códigos QR serán distintos, pues, se debe garantizar el acceso de los vehículos y esto se logra cada que se escanea dicho código QR.
- El personal de seguridad tendrá conocimiento de que tipo de persona entrará al conjunto, puesto que se especificarán datos por parte de los condóminos como por ejemplo: nombres y apellidos, cédula, N° de placa del medio de transporte, nombre de la Cooperativa / Empresa, etc.

Resulta conveniente resaltar que el diseño y construcción de dicho softphone fue personalizado de acuerdo a las necesidades de la urbanización Benalcázar Oriental utilizando como software Digital Ocean, CentOS, Issabel PBX, Android Studio, Firebase Database para que funcione diseñado llamado SoftAccess como se puede observar en la Figura 4.

Figura 4

Esquema de herramientas utilizadas para el desarrollo de la aplicación.



Análisis de la propuesta de diseño e implementación

En la mayoría de casos el acceso a urbanizaciones es realizado de forma manual por el personal de seguridad que se encuentra en la garita, lo cual ha generado que dicho acceso sea lento y la información permanezca escrita en hojas, las mismas que se pueden perder o manipular, en definitiva, esta información puede ser alterada, esto quiere decir que, si algún habitante requiere de información sería imposible dar datos concretos.

Es preciso señalar que la configuración del servidor de telefonía IP y el diseño del softphone engloba parámetros como códecs, ancho de banda, compatibilidad con los smartphones, actualizaciones, sincronizaciones con el servidor, etc., sin embargo, es de suma importancia tener acceso a internet para que el softphone pueda ser ejecutado en los smartphones. Existen dos escenarios puntuales, los habitantes y el personal de seguridad, para el primero, se podría afirmar que cada familia dispone de un servicio de internet, lo cual indica que el softphone podrá funcionar sin ningún problema, en cambio, para el segundo, se debe garantizar que exista una conexión fija de internet para que el personal de seguridad pueda utilizar dicho softphone y de esta forma brindar seguridad permanente a la urbanización. De igual forma, lo que se pretende es mantener una conexión óptima entre ambas partes para que la comunicación sea oportuna y el nivel de seguridad sea alto.

Sistema de control de acceso

Este sistema funcionará en conjunto con el servidor de telefonía IP alojado en la nube y el softphone, en donde el servidor ayudará a la recopilación de información, mientras que la app del softphone permitirá que el personal de guardia y los condóminos puedan comunicarse mediante los servicios ofrecidos. Para iniciar con el cumplimiento del control de acceso se deberá instalar la app en los smartphones y registrar la información necesaria de los moradores en la base de datos, igualmente, se deberán colocar los stickers de los códigos QR en los parabrisas de los automóviles, específicamente en el lado izquierdo inferior, entonces, con la

llegada de un condómino a la puerta principal en donde se encuentra el guardia, el procedimiento es escanear el código QR colocado en el parabrisa, posterior a esto, el guardia verificará y registrará la información en la base de datos, en el caso de ser correcta, podrá entrar, caso contrario, la puerta no se abrirá.

Para el caso de los visitantes, Uber, taxis, repartidores de comida, víveres o delicatessen, etc., previamente los moradores deberán generar un código QR para su correspondiente acceso, el cual deberá ser enviado para que puedan presentar al llegar a la urbanización, sin embargo, es necesario especificar el contenido que tendrá cada código para los siguientes casos:

- Visitantes: al momento de generar el código QR, el morador deberá escoger la opción de personas en general.
- Uber o taxis: al momento de generar el código QR, el morador deberá escoger la opción de servicio de transporte solicitado.
- Empresas de repartición de comida, víveres o delicatessen a domicilio como Glovo, Uber Eats, Rappi, Tipti, PedidosYa, etc: al momento de generar el código QR, el morador deberá escoger la opción de servicio de entrega a domicilio solicitado.

En síntesis, el guardia tendrá conocimiento de que tipo de persona o servicio de entrega está accediendo a la urbanización, dado que el registro permanecerá en la base de datos.

De manera general, los códigos proporcionados por los condóminos hacia visitantes, servicios de transporte y repartición de comida anteriormente mencionados, podrán ser usados una única vez, por lo tanto, se debe tener en cuenta que todos los registros quedarán almacenados en la base de datos. En el caso de escanear más de dos veces el mismo código QR aparecerá un mensaje de código expirado, de esta manera, el guardia negará el acceso. Al

tratar de los vehículos de quienes residen en la urbanización no tendrán límite de expiración por su condición de moradores.

No se debe pasar por alto la posibilidad de que por algún motivo no se genere el código QR por parte de los moradores, no obstante, al estar en la guardianía se podrá establecer una videollamada entre el morador y quien desea entrar siendo de intermediario el personal de guardia y a su vez podrá constatar, verificar y permitir el acceso de una forma segura gracias al servicio ofrecido por el softphone.

Análisis sobre la implementación del sistema de control de acceso según los servicios requeridos de la urbanización

Elección de servidor de telefonía IP

El servidor de telefonía IP desempeña un rol muy importante en el desarrollo de este proyecto de titulación, de forma tal que, posibilitó la comunicación por medio de internet, en particular entre el guardia y los moradores, de la misma forma, facilitó la realización de llamadas, mensajería, videollamada, etc., de igual modo, el registro de información.

Inicialmente se realizaron las configuraciones pertinentes mediante Asterisk con la intención de disponer dicho servidor, no obstante, se presentaron algunos problemas en la programación y configuración lo que imposibilitó el registro de los números y por ende la realización de llamadas. Es así como se optó por usar la plataforma de Issabel puesto que, la creación de extensiones, el acceso a llamadas y la adición de un dominio resultaron fáciles y sencillos, por último, se pudo crear el certificado *let's encrypt* en la interfaz de la central.

Mediante Issabel se lograron establecer servicios como: llamadas, videollamadas y mensajería, lo cual fue configurado en dicha plataforma, inicialmente se crearon extensiones las mismas que representan los números de casa de las 30 personas que fueron parte de las

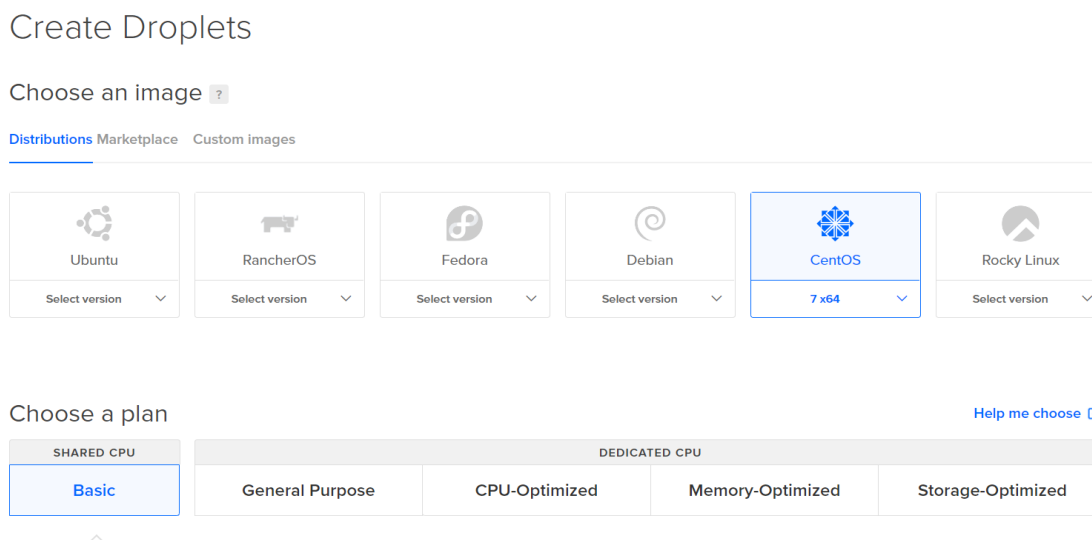
pruebas piloto, sin embargo, cada uno de estos servicios mencionados anteriormente requirieron una configuración diferente.

Digital Ocean. Dispone de una red de servidores los cuales están en la nube, así como también de un CPU virtual dedicado, sistemas operacionales basados en Linux tales como Ubuntu, CentOS, Debian, Fedora, etc., firewall, backup y escalabilidad. Para este último conviene enfatizar que, de acuerdo a la demanda obtenida Digital Ocean facilitará el almacenamiento requerido con una calidad óptima garantizando a los usuarios rapidez.

En cuanto a la creación de un droplet en Digital Ocean fue sencillo, puesto que, primero se crea una cuenta, se añade un método de pago, se selecciona la opción crear nuevo droplet, de la misma forma, se elige el sistema operativo, el cual fue CentOS 7 como se puede evidenciar en la Figura 5.

Figura 5

Creación de droplet en Digital Ocean



Nota. En la figura se evidencia la creación del droplet así como también que el sistema operativo elegido es CentOS.

Centos es ser un servidor robusto, confiable y dado que ha sido usado anteriormente en el desarrollo de proyectos orientados a telefonía se decidió emplear de la misma manera. En la Figura 6 se puede observar que el paquete elegido es el básico, es decir, de cinco dólares mensuales, lo cual hace posible trabajar con una capacidad de almacenamiento de 25 GB y una memoria RAM de 1 GB, por otra parte, si se diera el caso de necesitar mayor capacidad se puede cambiar al paquete requerido.

Figura 6

Paquete básico usado

The screenshot shows the 'Resize Droplet' interface. It includes a sidebar with navigation options like Graphs, Access, Power, Volumes, Resize, Networking, Backups, Snapshots, Kernel, History, Destroy, Tags, and Recovery. The main content area has a title 'Resize Droplet' and a note: 'This Droplet is on a Basic plan. You must turn off your Droplet to resize it.' There are two radio button options: 'CPU and RAM only' (selected) and 'Disk, CPU and RAM'. Below this, it says 'Currently using: Basic / 1 GB / 1 vCPU' with a 'Help me choose' link. A table of plans is displayed, with the 'Basic' plan highlighted.

Type	CPU Type	vCPUs	Memory	SSD	Transfer	Price	
<input type="radio"/>	Basic	Shared CPU	1 vCPU	512 MB	10 GB	0.5 TB	\$4/mo \$0.006/hr
<input checked="" type="radio"/>	Basic	Shared CPU	1 vCPU	1 GB	25 GB	1 TB	\$5/mo \$0.007/hr
<input type="radio"/>	Basic - Premium AMD	Shared CPU	1 vCPU	1 GB	25 GB	1 TB	\$6/mo \$0.009/hr
<input type="radio"/>	Basic - Premium Intel	Shared CPU	1 vCPU	1 GB	25 GB	1 TB	\$6/mo \$0.009/hr

Nota. En esta figura se pueden observar los paquetes ofertados por Digital Ocean, así como también ciertos parámetros tales como: CPU Type, vCPUs, Memory, SSD, Transfer, Price, por tanto, para el desarrollo de este proyecto de titulación se escogió el básico de cinco dólares.

En la Figura 7 se seleccionó un volumen adicional de 10GB de almacenamiento en el caso de que la memoria principal se llene, este costo es de un dólar adicional al mes.

Figura 7

Selección de volumen adicional

The screenshot shows the Digital Ocean control panel for a server named 'IssabelSoft'. The server configuration includes 1 GB Memory, 25 GB Disk, and 10 GB of additional storage in the NYC1 region, running CentOS 7 x64. The 'Volumes' section is active, displaying a table with the following data:

Name	Size	Created
volume-nyc1-01 NYC1 / 10 GB	10 GB	21 days ago

Nota. Esta figura indica que en la página de Digital Ocean se puede seleccionar un volumen extra en el caso de que la memoria principal se llene.

Para la configuración se usó el sistema operativo CentOS 7, así como también se instaló Issabel PBX mediante comandos obtenidos desde la página oficial de Issabel.org, luego se copiaron dichos comandos en un archivo, al que posteriormente se le dieron permisos de ejecución con el comando `chmod 777`, permitiendo instalar sin problemas.

Una de las funcionalidades que ofrece Digital Ocean es agregar dominios, lo cual fue de gran utilidad para que la central telefónica IP disponga de un *hostname* con el objetivo de facilitar el registro del softphone, puesto que, sin esta configuración se tendría que conectar al servidor mediante una dirección ip, de esta manera, al usuario se le facilita la conexión por medio de la dirección de la aplicación de SoftAccess en lugar de usar una dirección pública como se indica en la Figura 8.

Figura 8

Vinculación del dominio para el registro del softphone

DNS records

Type	Hostname	Value	TTL (seconds)
A	www.softacces	directs to 	3600 More ▾
A	softaccess.	directs to 	3600 More ▾
NS	softaccess	directs to ns1.digitalocean.com.	1800 More ▾
NS	softaccess	directs to ns2.digitalocean.com.	1800 More ▾
NS	softaccess.	directs to ns3.digitalocean.com.	1800 More ▾

Nota. En la figura se puede observar la vinculación del dominio a Digital Ocean, el cual es denominado softsccess.cf

Elección de código libre para la programación del softphone

Luego de haber investigado sobre algunos softphones conviene señalar que poseen características y funcionalidades similares, pese a esto, lo que se busca para el desarrollo del softphone de este proyecto de titulación es disponer del *Open Source* de forma tal que, se logre personalizar de acuerdo a los requerimientos sugeridos por parte de los usuarios de la urbanización.

Al referirse a open source o fuente abierta, se trata del código de programación que poseen los programas y se encuentran disponibles para el público en general, además se puede descargar con facilidad, de la misma forma, se debe reconocer que fueron creados por uno o varios programadores; en cambio el término de software libre, permite a millones de usuarios descargar y distribuir de forma gratuita, por consiguiente, no se pueden hacer modificaciones ni ediciones, mientras que en el primero, cada programador tiene la opción de modificar de acuerdo a su necesidad.

Es posible tener una comunicación libre con personas de todo el mundo por medio de internet, de igual forma, enviar mensajes de texto o voz, llamadas de audio y video. Los softphones han venido siendo usados para temas laborales además de personales, al hablar del costo de las llamadas, a través del uso de softphones resultan gratis sin importar el lugar en donde se encuentren las personas, tanto quien emite la llamada como quien recibe, el tiempo de establecimiento no resulta preocupante, puesto que el único requisito que se debe tener es una conexión a internet. Los softphones se basan en el protocolo SIP el cual trabaja con telefonía por internet.

Existen varios softphones así como Ekiga, Linnphone, 3CX, X-lite, Zoiper, entre otros, los tres primero que se mencionaron disponen de su fuente abierta mientras que los dos restantes únicamente disponen la opción de descargar.

Por lo tanto, se ha decidido trabajar con Linnphone por disponer de ejemplos de código orientados para cada funcionalidad generando una integración fácil entre el código y los respectivos segmentos tales como la conexión con la central IP, llamadas entrantes y salientes, chat, notificaciones push, entre otras, por otra parte, Linnphone está compuesto por una librería que se puede descargar desde Android Studio seleccionando la opción denominada como settings.gradle, de esta manera, se pueden utilizar todas las líneas de código necesarias para la programación del softphone.

Diseño de software

Creación del nombre de la app

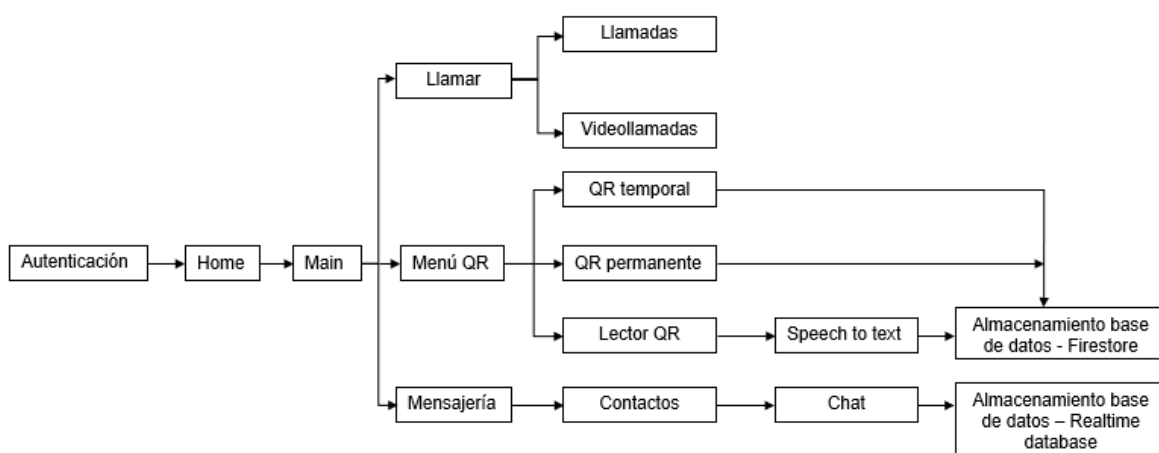
Dado que este trabajo de titulación está enfocado en el desarrollo de un softphone para áreas residenciales con la finalidad de brindar un control de acceso, su nombre SOFTACCESS, nace de la fusión de softphone y acceso.

Android Studio

Estructura de la app. La aplicación realizada cuenta con la estructura mostrada en la Figura 9 en donde se indican los servicios que fueron programados, así como también la conexión de la base de datos que trabaja conjuntamente con la “Autenticación”, de igual forma la central telefónica con el “Home”.

Figura 9

Diagrama de bloques de la estructura de la app



Nota. En esta figura se pueden observar todos los servicios que dispone la app desarrollada en Android Studio con su respectivo lenguaje de programación Kotlin, no obstante, se requirió de la base de datos Firebase para almacenar datos importantes de los condóminos.

Autenticación. Para el registro de los usuarios dentro de la aplicación se implementó el servicio de autenticación mediante correo electrónico, para esto, se han creado dos opciones, la primera es que el usuario ingrese cualquier correo y contraseña que disponga y la segunda es mediante la cuenta de Google, es decir el correo de Gmail configurado en el dispositivo móvil, en consecuencia, los datos serán almacenados en la base de datos de Firebase y se podrá acceder a todos los servicios implementados en la app. A continuación se presenta la Figura 10 como muestra de la autenticación.

Figura 10

Interfaz de autenticación



Nota. En la figura se evidencia la interfaz de autenticación mediante correo electrónico acompañado de los botones REGISTRAR y ACCEDER, los cuales facilitan el almacenamiento de datos en Firebase.

Llamadas y videollamadas. El servicio de llamadas y videollamadas se creó a partir de la librería denominada `org.linphone.sdk-android`, la cual permitió extraer las clases e instancias necesarias con el fin de activar el servicio de llamadas y videollamadas para la conexión con la central telefónica a través del núcleo de Linphone, dicha librería se actualiza constantemente para diferentes lenguajes de programación como swift, C++ y en nuestro caso kotlin. A partir de esta librería se realizó la siguiente configuración:

- Inicialmente se activó la conexión a la central telefónica para lo cual se creó una clase llamada *linphoneManager* para almacenar la extensión (*account*), la contraseña (*password*), el dominio (*ip*), el puerto (*port*) y el tipo de conexión (*type*) como se muestra en la Figura 11.

Figura 11

Código para el registro de datos en la central telefónica

```
binding.connect.setOnClickListener { it: View?
    val account = binding.username.text.toString()
    val password = binding.password2.text.toString()
    val ip = "softaccess.cf"
    val port = "5060"
    val type = TransportType.Udp
    if (account.isEmpty() || password.isEmpty() || ip.isEmpty() || port.isEmpty()) {
        return@setOnClickListener
    }else{
        linphoneManager.createProxyConfig(account, password, domain: "$ip:$port", type)
        prefes.saveNum(account)
        prefes.savePass(password)
        it.isEnabled = false
    }
}
```

Nota. La figura presenta los datos necesarios para la conexión de la app con la central telefónica con la ayuda de la clase Kotlin *linphoneManager*.

- Para el registro de cada usuario se tuvo que ingresar el número de extensión (número de casa) y la contraseña (creada por los desarrolladores) en cada uno de los campos correspondientes de la app, posterior a esto, mediante el botón

CONECTAR se generó la conexión con la central telefónica como se muestra en la Figura 12. Una vez realizado este proceso, ya no resulta necesario que cada usuario vuelva a digitar dichos datos de forma repetitiva debido a que quedarán guardados, es decir, se accederá a todos los servicios que contiene la app automáticamente. El botón DESCONECTAR eliminará el registro de la extensión y la contraseña, mientras que el botón CERRAR SESIÓN eliminará todos los datos de registro y enviará al usuario a la ventana anterior de autenticación con correo electrónico, finalmente el botón CONTINUAR permitirá acceder al *Call Activity* donde el usuario podrá utilizar todos los servicios.

Figura 12

Interfaz de conexión con la central telefónica



Nota. En la figura se puede observar que mediante esta pantalla es posible realizar la conexión con la central telefónica en donde el usuario debe escribir los datos: extensión y contraseña.

- Luego de la conexión exitosa con la central telefónica, el usuario accedió a la Actividad de las llamadas (Call Activity) en donde la clase abstracta `phoneCallback` permitió configurar los estados de una llamada que son:

`incomingCall`: llamada entrante.

`outgoingInit`: llamada saliente.

`callConnected`: llamada conectada.

`callReleased`: llamada completada.

`callEnd`: llamada finalizada.

En la Figura 13, dichas clases abstractas se implementaron en la configuración del Call Activity.

Figura 13

Clases Abstractas de los estados de la llamada

```
public abstract class PhoneCallback {  
    /** 来电状态 ... */  
    public void incomingCall(Call call) {  
    }  
  
    /** 呼叫初始化 */  
    public void outgoingInit(Call call) {  
    }  
  
    /** 电话接通 */  
    public void callConnected(Call call) {  
    }  
  
    /** 电话挂断 */  
    public void callEnd(Call call) {  
    }  
  
    /** 释放通话 */  
    public void callReleased(Call call) {  
    }  
  
    /** 连接失败 */  
    public void error(String string) {  
    }  
}
```

Nota. La figura muestra la clase abstracta `PhoneCallback` en donde se configura la lógica para cada una de los estados en una llamada.

- Se creó una función llamada `showCall` que contiene los siguientes parámetros:

Variable `type` de tipo entero: indica a qué ventana se va a transferir la interfaz de llamada.

Variable `call` de tipo `Call`: indica el estado de la llamada.

Variable `number` de tipo `string`: indica el número de extensión de la llamada entrante o saliente dependiendo si se realiza o recibe una llamada.

Variable `isvideo` de tipo `Boolean`: indica si se realiza una llamada con un `false` y una videollamada con un `true`.

El estado de estas variables cambia de acuerdo al estado de la llamada y con ayuda de la variable `type` la aplicación llamará a cada uno de los fragmentos para que se desplieguen como se muestra en la Figura 14.

Figura 14

Función showCall

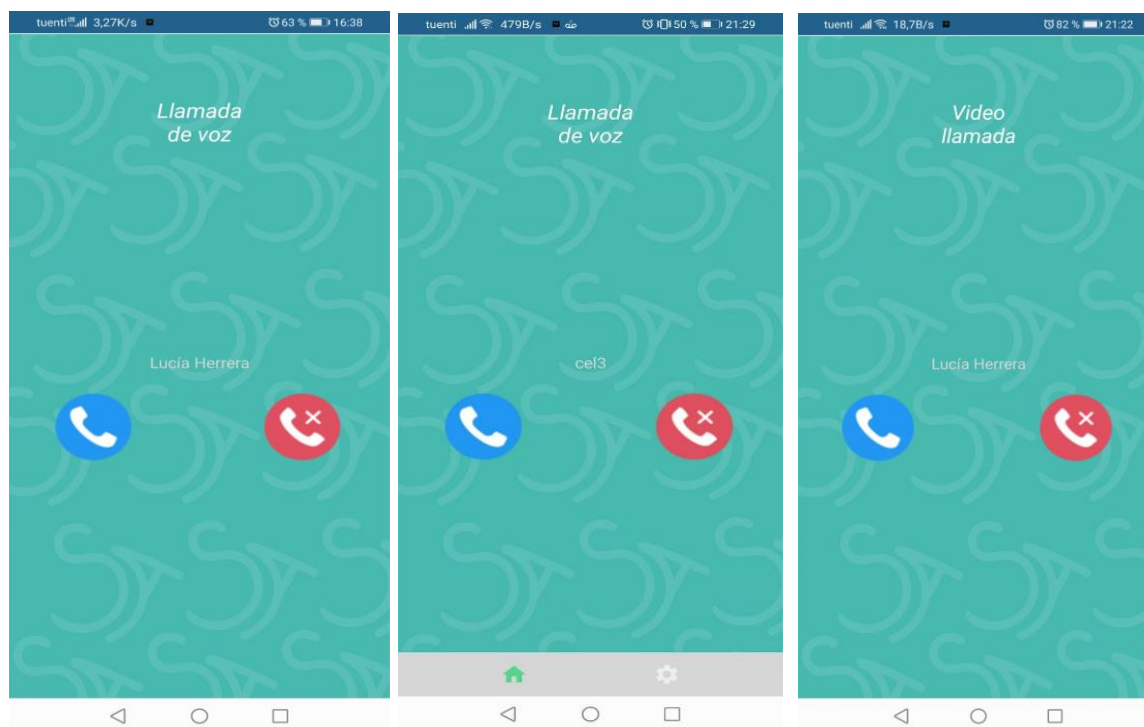
```
private fun showCall(type:Int, call:Call, number: String="", isVideo:Boolean=false) {
    when (type){
        0 -> {
            binding.frameLayout.visibility = View.INVISIBLE
            supportFragmentManager.beginTransaction()
                .remove(lastFragment!!)
                .commit()
        }
        1 -> {
            binding.frameLayout.visibility = View.VISIBLE
            lastFragment = CallIncomingFragment.newInstance(number, isVideo, call)
            supportFragmentManager.beginTransaction()
                .replace(frameLayout, lastFragment!!)
                .commit()
        }
        2 -> {
            binding.frameLayout.visibility = View.VISIBLE
            lastFragment = CallOutgoingFragment.newInstance(number, isVideo, call)
            supportFragmentManager.beginTransaction()
                .replace(frameLayout, lastFragment!!)
                .commit()
        }
    }
}
```

Nota. En esta figura se puede observar la estructura del funcionamiento de la función `showCall()` que dependiendo del estado de la llamada la aplicación enviará datos hacia los diferentes fragmentos para continuar con la llamada o videollamada.

- El fragmento *CallIncomingFragment* se despliega cuando una llamada de una extensión remota está ingresando, entonces, la interfaz de la aplicación cambia para que el usuario pueda contestar o rechazar la llamada como se puede observar en la Figura 15.

Figura 15

Interfaz del IncomingCallFragment (llamada entrante)



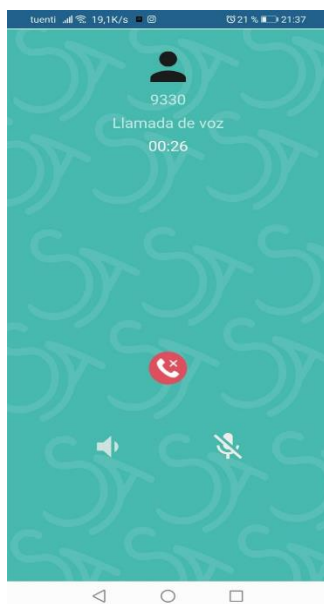
Nota. En la figura se visualiza la interfaz del *IncomingCallFragment*, donde el usuario puede contestar o rechazar la llamada, de igual forma, se muestra si es llamada o videollamada junto con el número de extensión del cual recibe la llamada.

- En el caso de escoger el botón colgar el *IncomingCallFragment* se ocultará y la llamada será finalizada, de esta manera, se volverá a la interfaz anterior del *CallActivity*. Caso contrario, si se presiona el botón contestar, la llamada pasará al estado de *CallConnected* y el fragmento desplegado será el

VoiceCallFragment o el *VideoCallFragment* en caso de que sea una videollamada. En la Figura 16 se puede visualizar lo mencionado.

Figura 16

Interfaz del VoiceCallFragment (Llamada de voz)

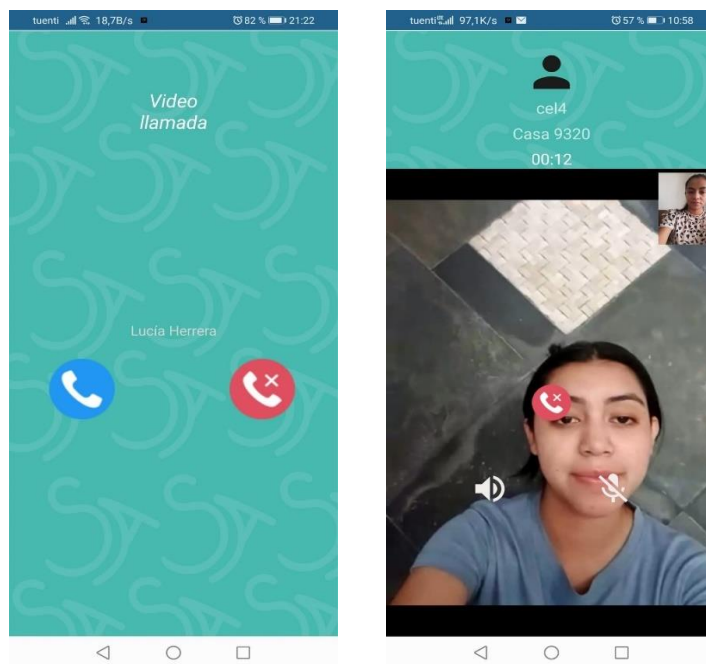


Nota. En esta figura se muestra la interfaz del *VoiceCallFragment* en donde el usuario puede observar algunos parámetros característicos de la llamada, como por ejemplo: la extensión, duración de la llamada, tipo de llamada (voz), de modo similar, los botones: colgar, altavoz y mute.

- Para el servicio de videollamada lo que difiere es la activación de la cámara de cada usuario como se puede ver en la Figura 17.

Figura 17

Interfaz del VideoCallFragment



Nota. Las figuras indican la interfaz del VideoCallFragment la cual está compuesta por los mismos botones que el VoiceCallFragment, la diferencia radica en que se puede visualizar el video del usuario remoto y del usuario local.

- En el caso de querer realizar una llamada se debe abrir el DialpadFragment presionando el botón llamar en el Call Activity como se muestra en la Figura 18.

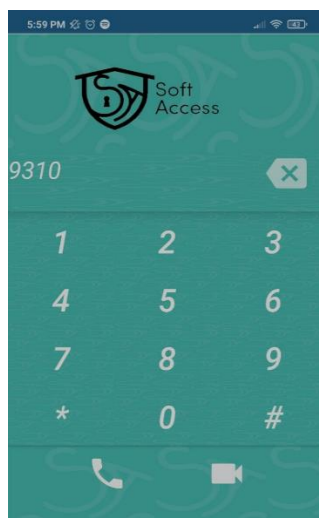
Figura 18

Interfaz del Call Activity



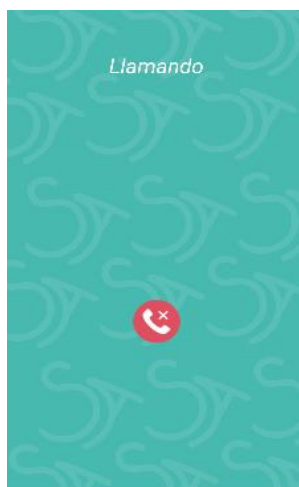
Nota. En esta figura se muestra la interfaz del Call Activity desde aquí se podrá acceder a cada uno de los servicios que brinda la aplicación “SoftAccess”.

- Al momento de escribir la extensión a la que se desea llamar, el usuario podrá elegir entre llamada con el botón de teléfono de la parte izquierda o videollamada con el botón de videocámara de la parte derecha, las cuales son mostradas en la Figura 19.

Figura 19*Interfaz del DialpadFragment*

Nota. La figura presenta la interfaz del DialpadFragment en donde se podrá realizar una llamada o videollamada a las extensiones que estén registradas dentro de la central telefónica.

- Automáticamente, después de presionar uno de los botones de llamada o videollamada se desplegará el *CallOutgoingFragment* donde aparecerá la información del tipo de llamada y la extensión a la que se está marcando como se muestra en la Figura 20.

Figura 20*Interfaz del CallOutgoingFragment*

Nota. La figura muestra la interfaz del CallOutgoingFragment en donde se brinda la opción de colgar la llamada mediante un botón.

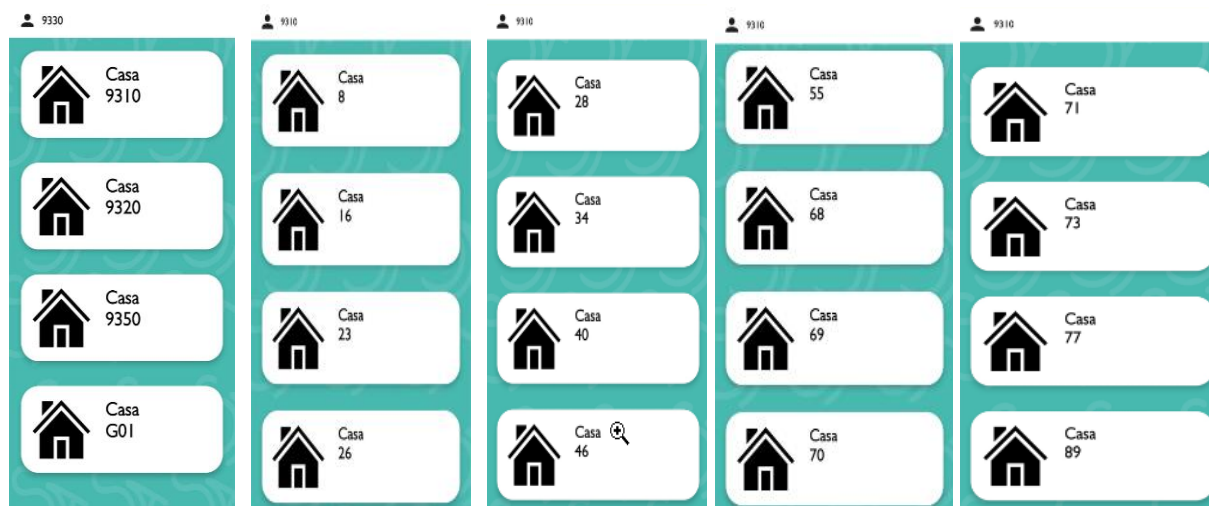
En consecuencia, si el usuario al que se está llamando contesta la llamada, la interfaz cambiará a la de VoiceCallFragment o a la de VideoCallFragment, dependiendo del caso.

Los usuarios podrán marcar a las extensiones que estén registradas en la central telefónica, caso contrario, la llamada no se concretará y se reproducirá un mensaje de voz por default indicando que no se puede acceder a dicha extensión.

Mensajería y notificaciones push. El servicio de mensajería se basó principalmente en la interfaz del *Chat Activity* donde se guardaron los contactos de cada una de las extensiones registradas en la aplicación. Esto quiere decir que, los usuarios que abran la interfaz del chat podrán visualizar todos los contactos registrados de los demás usuarios, puesto que se registraron con sus respectivas extensiones como se puede ver en la Figura 21. Cada contacto se encuentra dentro de un layout llamado *CardView* el cual tiene una imagen y dos campos de texto, con el fin de diferenciar cada extensión, cada *CardView* fue generado a través de un *RecyclerView* que es una lista de items que se despliegan en el *Chat Activity*.

Figura 21

Interfaz del Chat Activity



Nota. Esta figura muestra los ítems que permanecen dentro de la interfaz del *Chat Activity* correspondientes a los usuarios registrados en la aplicación.

Dentro de las extensiones se creó un chat grupal con el nombre de G01, únicamente los administradores podrán escribir temas importantes y urgentes con el propósito de informar a los demás usuarios registrados.

En cuanto al RecyclerView, trabajó con los datos de un adapter denominado *ExtensionAdapter*, el mismo que tomó el Contexto *this*, un ArrayList que es una lista de *strings* (conjunto de caracteres) con el nombre de *extList* y la variable *ext* que es la extensión del usuario como se puede observar en la Figura 22.

Figura 22

Código del RecyclerView

```
extList = ArrayList()
adapter = ExtensionAdapter( context: this, extList, ext) //Parámetros de envío (contexto, arraylist, extension)

extRecyclerView = findViewById(R.id.extRecyclerView)

extRecyclerView.layoutManager = LinearLayoutManager( context: this)

extRecyclerView.adapter = adapter //toda la vista del recyclerView viene de aqui llamando al ExtensionAdapter
```

Nota. En esta figura se observan los parámetros que necesita pasar el RecyclerView al *ExtensionAdapter* para crear la interfaz del Chat Activity.

La lista de extensiones *extList* fue llenada con información extraída del *Realtime Database* de *Firebase Database*. De forma similar, la configuración de la base de datos y los procesos de guardado y obtención de datos serán explicados más adelante, en la sección de *Firebase*.

Una vez extraídos los datos desde el *cloud* de *Firebase* serán procesados por el *ExtensionAdapter* el cual generará los *CardView* en conjunto con los datos para cada extensión como se observa en la Figura 23. Cuando se seleccione un contacto se generará un *intent* el cual hace que se inicie otra actividad llamada *Message Activity*, que mediante los *putExtra* se puedan incluir los datos de la extensión del usuario con el que se desea chatear y la extensión de quien envía el mensaje (emisor).

Figura 23

Código del ExtensionAdapter

```

override fun onBindViewHolder(holder: ExtensionViewHolder, position: Int) { //genera cada item de extensiones con sus respectivos datos
    val currentExt = extList[position] //toma las extensiones una por una
    holder.txtName.text = currentExt.extension

    holder.itemView.setOnClickListener{ //cuando se selecciona un item se lanza la Message Activity
        val intent = Intent(context, MessageActivity::class.java)
        intent.putExtra( name: "casa", currentExt.extension)
        intent.putExtra( name: "ext", ext)
        context.startActivity(intent)
    }
}

```

Nota. En la presente figura se pueden observar las líneas de código que permiten presionar el ítem correspondiente a un contacto de una extensión en particular para que pueda cambiar de interfaz hacia la Message Activity.

Al desplegar el Message Activity, aparecerán los ítems de los mensajes enviados y recibidos mediante un RecyclerView mediante un *MessageAdapter*, esto se puede evidenciar en la Figura 24. Por otra parte, se creó un *messageList* el cual es un arreglo de strings que se envió al MessageAdapter en donde se ordenan los mensajes enviados de cada usuario en dicha *messageList*, es importante resaltar, que los mensajes se guardan en la base de datos y se los extrae para que el MessageAdapter pueda ordenar en la *messageList*

Figura 24

RecyclerView del Message Activity

```

chatRecyclerView = binding.chatRecyclerView
messageBox = binding.messageBox
sendButton =binding.sentButton

messageList = ArrayList()
messageAdapter = MessageAdapter( context: this, messageList, ext)

chatRecyclerView.layoutManager = LinearLayoutManager( context: this)
chatRecyclerView.adapter = messageAdapter

```

Nota. En esta figura se indica el proceso de carga de mensajes dentro del RecyclerView del Message Activity con la ayuda del MessageAdapter.

En la Figura 25 se puede observar que dentro de la interfaz del Message Activity se tiene el *messageBox* el cual es usado para la escritura de mensajes, por otra parte, el *sendButton* sirve para enviar mensajes, de igual forma serán registrados en la base de datos conocida como realtime database, asimismo serán llamados cada vez que se escoja algún chat de una extensión.

Figura 25

Interfaz del Message Activity



Nota. Por medio de esta figura se observa la interfaz del Message Activity con un ejemplo de mensajes enviados y recibidos.

Dentro del MessageAdapter se crearon el *SentViewHolder* (parte derecha de la pantalla) y el *ReceiveViewHolder* (parte izquierda de la pantalla) con el propósito de dirigir al ítem correspondiente una vez que el usuario envíe o reciba un mensaje. Lo mencionado se puede notar en la Figura 26.

Figura 26

Código del MessageAdapter

```
override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {  
  
    val currentMessage = messageList[position]  
    if (holder.javaClass == SentViewHolder::class.java){  
        // hacer que se envíen los datos al sent view holder  
  
        val viewHolder = holder as SentViewHolder  
        holder.sendMessage.text = currentMessage.message  
  
    }else{  
        //hacer que se envíen los datos al receive view holder  
  
        val viewHolder = holder as ReceiveViewHolder  
        holder.receiveMessage.text = currentMessage.message  
  
    }  
}
```

Nota. Esta Figura presenta las líneas de programación y estructura usadas para la generación de mensajes de envío y recepción distribuidos al ViewHolder correspondiente.

Las notificaciones push sirvieron para que los usuarios tengan la facilidad de ver los mensajes recibidos, incluso cuando la aplicación no se encuentre abierta. Al momento de presionar dichas notificaciones se redirigirán al chat del cual fueron enviadas, de forma similar, esto ayudará a informar sobre cualquier novedad referente a la urbanización a los usuarios.

Para levantar este servicio fue necesario crear una clase con el nombre de *MyFirebaseMessagingService* que actuó como una clase kotlin de servicio, de igual forma una "Interface" de suma importancia denominada *NotificationAPI* la cual hizo posible la comunicación entre la base de datos y la aplicación por medio de la librería "retrofit2" facilitando el guardado y acceso de la información que en este caso se encuentra en Firebase, esto se realizó con el fin de no generar datos ni almacenarlos en cada dispositivo, al contrario, con retrofit se guardarán los datos en el servidor del Cloud Messaging Service usando internet.

Igualmente se utilizó la librería de corrutinas, puesto que, permiten efectuar procesos fuera del hilo principal de la aplicación, esto es necesario al momento de acceder a la información almacenada en Firebase, cabe recalcar que el intercambio de datos no se logra instantáneamente, sino que se tarda algunos milisegundos que en ciertos casos pueden llegar a segundos, esto depende de la red a la cual se esté conectando el dispositivo, aunque de manera general el retardo no supera los 100 ms. En caso de no realizar este proceso con corrutinas no se podría obtener la información de la base de datos y la programación de las notificaciones no podría ser completada.

En la Interface NotificationAPI fueron creados los campos de SERVER_KEY y CONTENT_TYPE que se encargan de identificar al servidor del cual estamos requiriendo el traspaso de información, en este caso sería la conexión con Firebase como se muestra en la Figura 27.

Figura 27

Código de la Interfa NotificationAPI

```
interface NotificationAPI {  
  
    @Headers("Authorization: key=${SERVER_KEY}", "Content-Type:${CONTENT_TYPE}")  
    @POST("fcm/send")  
    suspend fun postNotification(  
        @Body notification: PushNotification  
    ): Response<ResponseBody>  
}
```

Nota. En la Figura se pueden observar los parámetros de configuración para acceder al servicio de notificaciones.

Una vez creada la NotificationAPI se pudo configurar la función sendNotification como se puede notar en la Figura 28. De tal forma, se puede apreciar el uso de la CoroutineScope, puesto que, este proceso fue desarrollado fuera del hilo principal de la aplicación.

Figura 28

Función sendNotification

```
private fun sendNotification(notification: PushNotification) = CoroutineScope(Dispatchers.IO).launch { this: CoroutineScope
    try {
        val response = RetrofitInstance.api.postNotification(notification)
    } catch (e: Exception) {
        Toast.makeText(context: this@MessageActivity, e.toString(), Toast.LENGTH_LONG).show()
    }
}
```

Nota. En la figura se presenta la función `sendNotification` incluida dentro de un `try-catch` en el caso de que no se pueda acceder a la Instancia del Retrofit.

Un dato relevante dentro del servicio de notificaciones es poder detectar el *token* de cada dispositivo, puesto que, para poder enviar una notificación la base de datos necesita que cada dispositivo tenga un código distintivo único denominado token, por esta razón, que fue creado en la interfaz de conexión con la central telefónica, para que cada extensión se encuentre ligada al token de un dispositivo, esto se puede evidenciar en la Figura 29.

Figura 29

Código de generación del token para cada dispositivo

```
MyFirebaseMessagingService.sharedPref = getSharedPreferences( name: "sharedPref", Context.MODE_PRIVATE)
FirebaseMessaging.getInstance().token.addOnSuccessListener { it: String!
    MyFirebaseMessagingService.token = it
}
```

Nota. En esta figura se puede notar el uso de las “SharedPreferences” para guardar el token del dispositivo en el servicio `MyFirebaseMessagingService`.

Finalmente, la clase `MyFirebaseMessagingService` se encargó de generar la notificación y mostrar en el dispositivo correspondiente mediante la función `onMessageReceived` y la `onNewToken` para detectar el token como se muestra en la Figura 30. Por consiguiente, el usuario que reciba la notificación podrá presionarla y acceder al chat del usuario que le envió el mensaje.

Figura 30

Código para envío de notificaciones

```

override fun onNewToken(newToken: String) {
    super.onNewToken(newToken)
    token = newToken
}

override fun onMessageReceived(message: RemoteMessage) {
    super.onMessageReceived(message)

    val intent = Intent( packageContext: this, MessageActivity::class.java)
    intent.putExtra( name: "casa", message.data["title"])
    intent.putExtra( name: "ext", prefes.getNum())
    val notificationManager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
    val notificationID = Random.nextInt()

    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
        createNotificationChannel(notificationManager)
    }
}

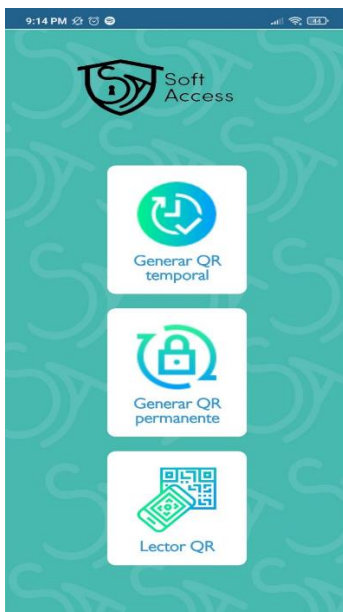
```

Nota. En la presente figura se observa como con un Intent puede acceder al *MessageActivity* del chat del usuario que envió el mensaje.

Códigos QR. Existen tres modos de operación que forman parte del menú denominados: Generar códigos QR temporales, Generar códigos QR permanentes y Lector de códigos QR tal como se muestra en la Figura 31.

Figura 31

Interfaz del menú de códigos QR



Nota. En esta figura se puede apreciar el diseño de la interfaz del menú para los códigos QR.

Para la generación de los códigos QR permanentes y lector de códigos QR, únicamente podrá acceder el personal de seguridad, mientras que, para la generación de códigos QR temporales pueda ser realizada por los usuarios finales (condóminos).

Mediante la Figura 32, se puede notar la interfaz que se despliega para la generación de códigos QR temporales con sus respectivos campos que deberán ser llenados por el usuario, con el propósito de que la persona a la que se le envíe el código QR pueda tener acceso al Conjunto habitacional.

Figura 32

Interfaz de la generación de códigos QR temporales



Nota. La figura muestra los campos a ser llenados para la generación de códigos QR temporales.

Una vez que se han escrito los datos el usuario debe presionar el botón llamado, Generar código QR, simultáneamente, de forma internamente se enviarán los datos a Firebase y la imagen del código QR generado se desplegará en la pantalla como se muestra en la Figura 33. En este código QR se guardará un conjunto de números con el nombre de ID con el objetivo de que sea único.

Figura 33

Generación de código QR temporal



Nota. En la presente figura se observa el código QR generado para que el usuario pueda enviar a la persona que requiera ingresar a la urbanización, como por ejemplo, un familiar invitado.

La librería utilizada para la generación de códigos QR fue `com.google.zxing:core:3.4.1`, la cual permitió tomar un conjunto de caracteres de texto y codificarlos a fin de convertirlos en un variable de tipo “bitmap” para luego mostrarlos como imagen, todo este proceso fue programado dentro de una función denominada `codeCreated`, como se observar en la Figura 34.

Figura 34

Código de la función codeCreated

```
private fun codeCreated() {
    val ivCodigoQR = binding.ivCodigoQR
    try {
        val barcodeEncoder: BarcodeEncoder = BarcodeEncoder()
        val bitmap : Bitmap = barcodeEncoder.encodeBitmap(
            txtDataQR,
            BarcodeFormat.QR_CODE,
            width: 750,
            height: 750
        )

        ivCodigoQR.setImageBitmap(bitmap)
        clean()
    } catch (e: Exception) {
        e.printStackTrace()
    }
}
```

Nota. La figura muestra la programación para la generación del código QR temporal.

En la generación de códigos QR permanentes se configuró la interfaz para añadir los datos de los vehículos como se observa en la Figura 35, de modo que estos códigos irán pegados en los parabrisas de los automóviles para acceder de forma rápida una vez que el personal de seguridad realice la lectura de dichos códigos.

Figura 35

Interfaz de la generación de código QR permanente



7:06 PM

Soft Access

Placa del vehículo
PDS-3567

Escriba la marca del vehículo
Chevrolet

Escriba el modelo del vehículo
Aveo

Escriba los Apellidos de la Familia
Luna Soria

Escriba el número de casa
25

GENERAR CÓDIGO QR

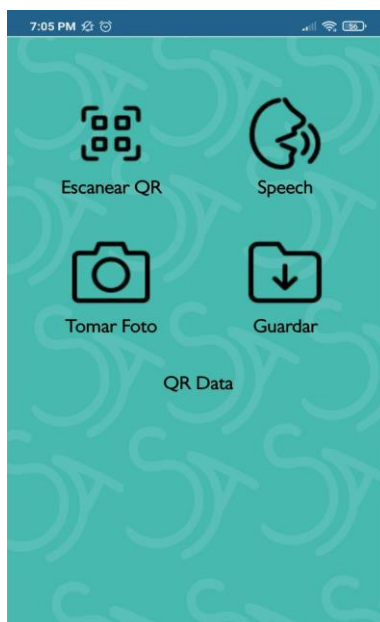
Nota. Esta figura representa la interfaz de la generación de códigos QR permanentes a la cual únicamente tuvieron acceso solo los administradores de la urbanización.

Siguiendo la programación de la generación de códigos QR temporales, se generaron los códigos QR permanentes salvo en la estructura del ID que difirió en el manejo de los datos al momento de realizar la lectura.

Para la validación del ingreso de los moradores y visitantes de la urbanización, el personal de seguridad utilizó el “Lector de códigos QR” cuyo diseño se indica en la Figura 36.

Figura 36

Interfaz del Lector de códigos QR



Nota. En esta figura se detalla cada uno de los botones del lector de códigos QR.

Al presionar el botón *Escanear QR* se activará la cámara del dispositivo móvil para tomar una foto a dicho código, luego, con la ayuda de la librería `com.google.firebase:firebase-ml-vision-barcode-model:16.1.2` se logró realizar el procesamiento de información utilizando una función denominada *processQr* como se observa en la Figura 37.

Figura 37

Función processQr

```

private fun processQr(){
    //binding.ivImageCode.visibility = View.GONE
    binding.tvResult.visibility =View.VISIBLE

    barcodeScanner.process(inputImage).addOnSuccessListener { it: (Mutable)List<Barcode!>!
        //handle barcodes list
        for (barcode:Barcode in it){
            val valueType = barcode.valueType
            when (valueType){
                Barcode.TYPE_TEXT ->{
                    val data = barcode.displayValue
                    binding.tvResult.text = "${data}"
                }
            }
        }
        qrData=binding.tvResult.text.toString()
        val ntype = qrData.length

        if (ntype > 13){
            getData(qrData)
            binding.btnSpeech.isEnabled=true
            binding.btnFoto.isEnabled=true
        }else{
            getPermaData(qrData)
        }
    }.addOnFailureListener { it: Exception
        Log.d(TAG, msg: "processQr: ${it.message}")
    }
}

```

Nota. En la Figura se observa el proceso de extracción de información desde la imagen del código QR hacia su conversión a texto.

Como anteriormente se indicó, la información que tiene el QR es un ID y como tal, esa información no muestra ningún tipo datos sobre la persona o vehículo, en consecuencia, se genera otro proceso fuera del hilo principal, es decir, una corrutina que está representada en la Figura 38 mediante la función `getData`, la cual tomará ese ID y validará con la finalidad de diferenciar entre un código QR temporal o permanente para luego hacer una petición a la *Firestore database* con el objetivo de realizar una búsqueda de todos los QR que existen y extraer información asociada al ID de ese código QR correspondiente, una vez completo este proceso, la información que aparecerá en pantalla serán los datos asociados a la persona que ingresa en ese momento.

Figura 38

Función `getData`

```
private fun getData(code:String)= CoroutineScope(Dispatchers.IO).launch{ this: CoroutineScope
    val qrdataArray = code.split(...delimiters: "\n").toArray()
    val n = qrdataArray.size
    try {
        val querySnapshot = db.collection( collectionPath: "QRcodes").whereEqualTo( field: "id", qrdataArray[n-1])
            .get().await()
        val sb = StringBuilder()
        val qrType = StringBuilder()
        for (document in querySnapshot.documents){
            sb.append("Tipo de invitado: ").append(document.data!!.getValue( key: "itemType")).append("\n")
                .append("Placa: ").append(document.data!!.getValue( key: "placa")).append("\n")
                .append("Empresa: ").append(document.data!!.getValue( key: "emp")).append("\n")
                .append("Cedula: ").append(document.data!!.getValue( key: "ced")).append("\n")
                .append("Nombre: ").append(document.data!!.getValue( key: "name")).append("\n")
                .append("Creación QR: ").append(document.data!!.getValue( key: "time")).append("\n")
                .append("Entrada: ").append(document.data!!.getValue( key: "entrada")).append("\n")
                .append("Salida: ").append(document.data!!.getValue( key: "salida")).append("\n")
                .append("Error: ").append(document.data!!.getValue( key: "error")).append("\n")
                .append("ID: ").append(document.data!!.getValue( key: "id"))
            qrType.append(document.data!!.getValue( key: "qrType"))
        }
    }
}
```

Nota. En esta figura se observa la función `getData` la cual utiliza una corrutina para acceder a Firestore Database y extraer los datos asociados al ID de cada código QR.

La información se mantiene segura a través de este ID, pues cualquier lector de códigos QR puede obtener este conjunto de números, sin embargo, no podrá acceder a los datos almacenados, puesto que, leerá un conjunto de números. De otra manera, desde la aplicación de SoftAccess se puede acceder hacia toda la información asociada al código QR, esto se puede notar en la Figura 39.

Figura 39

Datos de la lectura del código QR temporal



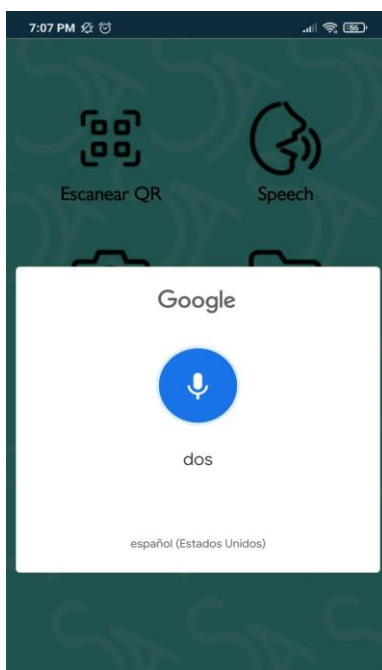
Nota. Esta figura muestra los campos se fueron llenados por medio de la lectura de un código QR temporal.

Una vez mostrados los datos en pantalla, el personal de seguridad tiene que cerciorarse de los datos presentados por la persona que desea ingresar, es por esta razón que tendrá que ejecutar el comando Speech To Text, de esta forma se realizará una doble verificación.

Librerías speech to text. Se activó por medio de la librería android.speech de Google y mediante la función `askSpeechInput` la cual fue instanciada con un intent, este se dirige a la ventana en la que se activa el micrófono, de esta manera, todas las palabras pronunciadas por la persona (visitante) se visualizarán en forma de texto, a continuación se puede notar en la Figura 40.

Figura 40

Interfaz de speech to text



Nota. En la Figura se puede observar la ventana que apareció al momento de presionar el botón Speech.

Entonces, una vez que el visitante haya hablado y si la información proporcionada que a manera de ejemplo puede ser, el número de casa a la que se dirige es la correcta, la aplicación automáticamente valida los datos e indica en pantalla un mensaje de “información correcta”, caso contrario, aparecerá “información incorrecta, intente de nuevo”. Es así, como se completa la doble verificación.

El siguiente paso es presionar el botón de guardar, por tal motivo, los datos de registro de entrada quedarán alojados en la base de datos de Firestore. Adicionalmente, existe un botón llamado Foto el cual permite hacer una captura mediante la cámara del dispositivo móvil a: personas y vehículos que deseen ingresar a la urbanización, esta opción queda a criterio del personal de seguridad, dicha foto se guardará en la base de datos. Puede existir el caso en que sea de suma importancia tomar una foto a documentos, como por ejemplo, en el caso del servicio de mudanza, es necesario tener un respaldo a manera de foto en donde se puedan evidenciar datos como por ejemplo: nombre de la empresa de mudanza, fecha, hora, casa a la que se dirigen, etc. De la misma forma, se tendrá un registro completo tanto de entrada como de salida a la urbanización.

Una vez que el guardia haya efectuado todo el proceso anteriormente explicado, es decir, el escaneo del código QR fue realizado dos veces (entrada y salida), posterior a esto, dicho código temporal se deshabilitará, cabe decir que, si una persona intenta ingresar con el mismo código QR no podrá hacerlo porque la aplicación no soporta un escaneo por tercera vez, de pasar esto la aplicación arrojará un mensaje de: “código QR vencido”, del mismo modo, se guardará la información de la hora a la que se intentó ingresar.

No obstante, para el caso de los códigos QR permanentes, el morador tendrá pegado en el parabrisa de su vehículo el código QR con el fin de que el guardia escanee por medio del lector de la app e ingrese forma inmediata, de igual modo, se guardará el registro de entrada automáticamente sin tener que realizar por el proceso de Speech To Text.

Resulta necesario recalcar que los condóminos al disponer de la aplicación serán los responsables directos de la generación de los códigos QR temporales, la cual quedará registrada en la base de datos.

Por esta razón, podrán acceder al Conjunto, vehículos, motos, camiones, etc., que ofrezcan algún tipo de servicio a los condóminos.

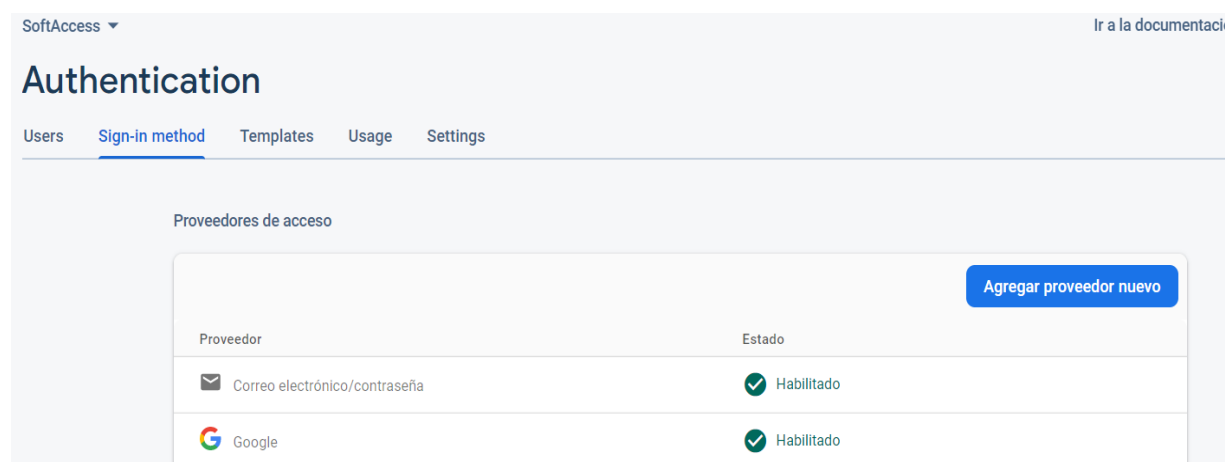
Firestore

Authentication. El primer paso que se realizó fue la sincronización entre Android Studio y la consola de Firebase por lo que se tomó el nombre del *package* del archivo *AndroidManifest* el cual se encuentra dentro del proyecto de Android Studio, luego fue puesto dentro de la consola de Firebase, simultáneamente se descargó un archivo denominado *google-services.json* para luego poder implementar dentro de Android Studio, es así, que se logró una conexión exitosa con Firebase.

Dentro de la consola de Firebase se accedió a la pestaña *authentication* y se habilitó el correo electrónico y google como proveedores de acceso como se muestra en la Figura 41.

Figura 41

Proveedores de acceso



Nota. En esta figura se pueden observar los proveedores de acceso habilitados.

La librería encargada de la autenticación se llama `com.google.firebase:firebase-auth-ktx:21.1.0`, la cual permitió configurar los registros de los usuarios mediante correo electrónico y cuenta de Google como se muestra en la Figura 42.

Figura 42

Código del botón de Google

```
binding.googleButton.setOnClickListener { it: View?
    //Configuración
    val googleConf = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(com.google.firebase.ui.auth.R.string.default_web_client_id))
        .requestEmail().build()

    val googleClient = GoogleSignIn.getClient(activity: this, googleConf)
    googleClient.signOut()

    startActivityForResult(googleClient.signInIntent, GOOGLE_SIGN_IN)
}
```

Nota. En la figura se observa que por medio del comando `GoogleSignInOptions` y con la petición de un token con `requestIdToken` se puede acceder mediante el email de Google.

De esta forma los correos electrónicos de cada usuario se guardarán en la pestaña *Users* en la consola de Firebase.

Realtime database. Se utilizó para guardar los datos de los usuarios y mensajes de los chats que forman parte de la sección de mensajería, la librería requerida fue `com.google.firebase:firebase-messaging-ktx:23.1.0` que permitió guardar información en la base de datos con el comando de la Figura 43.

Figura 43

Líneas de comando usadas para subir datos a Realtime Database

```
rtDb = FirebaseDatabase.getInstance().getReference()
rtDb.child( pathString: "user").child(username).setValue(Extension(email, username, it))
```

Nota. En esta figura se muestra como acceder a un fichero de la base de datos y subir datos mediante el comando `setValue`.

Igualmente, se pudo extraer datos por medio del comando de la Figura 44.

Figura 44

Datos extraídos del Realtime Database

```
rtDb.child( pathString: "chats").child(senderRoom!).child( pathString: "messages")
    .addValueEventListener(object: ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {
            messageList.clear()

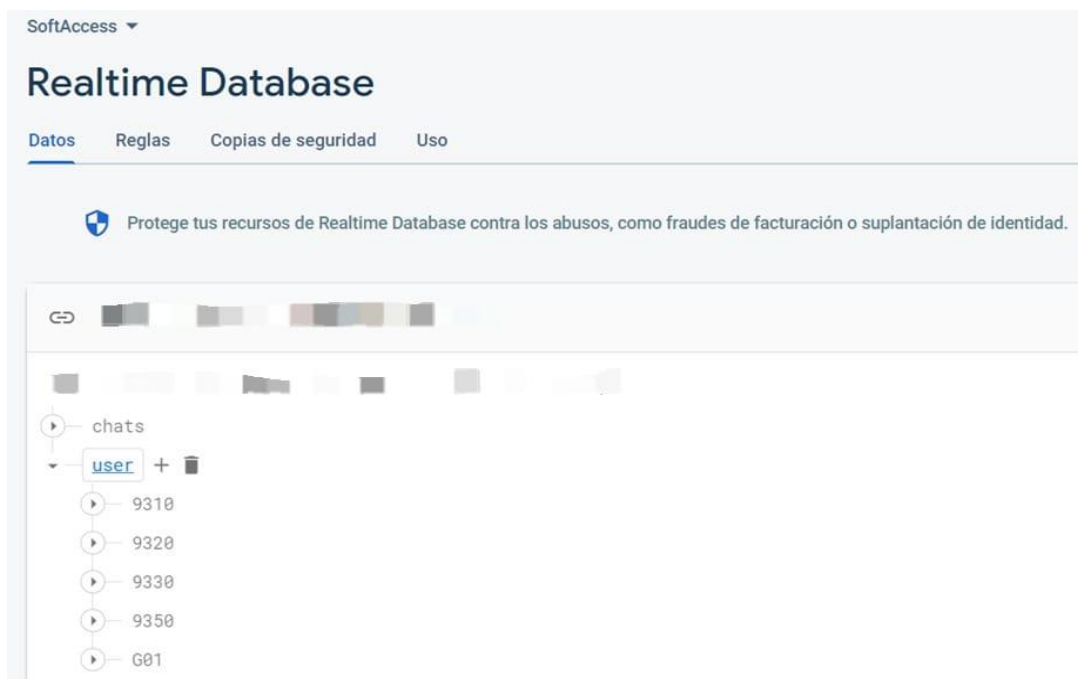
            for (postSnapshot in snapshot.children){
                val message = postSnapshot.getValue(Message::class.java)
```

Nota. Mediante esta figura se puede observar como dentro de una función `onDataChange` se puede extraer información con el comando `getValue`.

En la consola de Firebase, específicamente en la sección de Realtime Database se encuentran los datos ordenados jerárquicamente como se muestra en la Figura 45.

Figura 45

Datos almacenados en Realtime Database

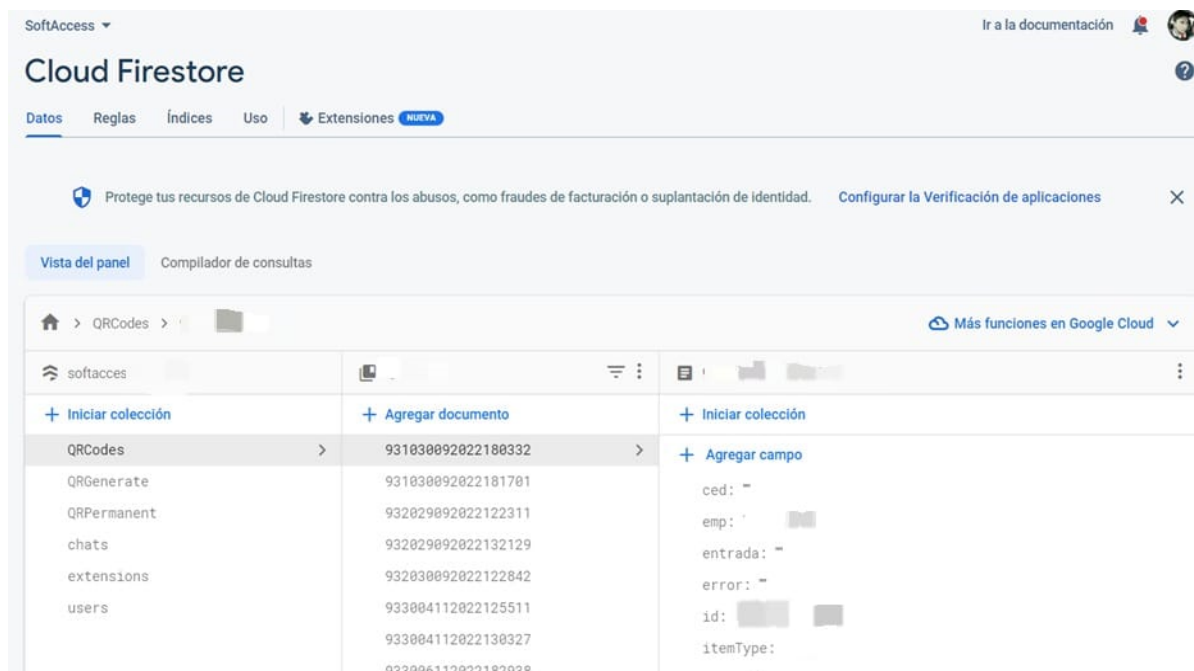


Nota. En la presente figura se detalla cada pestaña de datos en donde existe un orden de jerarquía.

Firestore database. La sección de códigos QR se configuró a través de *Firestore Database* el cual es un servicio muy parecido al Realtime Database, pero se diferencian en los comandos para subir y extraer datos, del mismo modo, al hablar de estructura, puesto que se guardan los datos a través de colecciones y documentos como se puede observar en la Figura 46. La librería que dió soporte a la configuración de esta sección fue `com.google.firebase:firebase-firestore-ktx`, al ser importante para la conexión entre la aplicación y la base de datos.

Figura 46

Consola de Firestore Database



Nota. En esta figura se aprecia la estructura de almacenamiento de datos mediante colecciones y documentos.

Para subir información a Firestore Database se debe acceder a la colección y al documento deseado con el propósito de indicar la información a subir, que en este caso es una data Class, es decir, un conjunto de variables estructurados en un arreglo de strings como se puede observar en la Figura 47.

Figura 47

Código empleado para subir datos en Firestore Database

```

val qrInfo = InfoQr(
    itemType = item,
    placa = "",
    emp = empresa,
    ced = "",
    name = "",
    time = getTime(),
    entrada = "",
    salida = "",
    error = "",
    qrType = "1",
    id = idTem,
    nCasa = casa!!
)

txtDataQR = casa + salto + idTem
codeCreated()
db.collection(collectionPath: "QRCodes").document(idTem).set(qrInfo)
db.collection(collectionPath: "extensions").document(casa!!).collection(collectionPath: "QRCodes").document(idTem).set(qrInfo)

```

Nota. En la figura se indica la data class “qrInfo”, la misma que es subida a Firestore Database mediante el comando set.

Así mismo, para obtener los datos fue necesario conocer la colección y el documento del cual se quiere extraer datos, para esto, se utilizó el comando `whereEqualTo`, el cual sirvió para buscar el documento del ID del código QR correspondiente para luego extraer toda la información relacionada a dicho ID como se puede constatar en la Figura 48.

Figura 48

Código usado para la extracción de datos de Firestore Database

```

try {
    val querySnapshot = db.collection( collectionPath: "QRCodes").whereEqualTo( field: "id", qrDataArray[n-1])
        .get().await()
    val sb = StringBuilder()
    val qrType = StringBuilder()
    for (document in querySnapshot.documents){
        sb.append("Tipo de invitado: ").append(document.data!!.getValue( key: "itemType")).append("\n")
            .append("Placa: ").append(document.data!!.getValue( key: "placa")).append("\n")
            .append("Empresa: ").append(document.data!!.getValue( key: "emp")).append("\n")
            .append("Cedula: ").append(document.data!!.getValue( key: "ced")).append("\n")
            .append("Nombre: ").append(document.data!!.getValue( key: "name")).append("\n")
            .append("Creación QR: ").append(document.data!!.getValue( key: "time")).append("\n")
            .append("Entrada: ").append(document.data!!.getValue( key: "entrada")).append("\n")
            .append("Salida: ").append(document.data!!.getValue( key: "salida")).append("\n")
            .append("Error: ").append(document.data!!.getValue( key: "error")).append("\n")
            .append("ID: ").append(document.data!!.getValue( key: "id"))
        qrType.append(document.data!!.getValue( key: "qrType"))
    }
    withContext(Dispatchers.Main){ this: CoroutineScope
        binding.tvResult.text = sb.toString()
        qrInt = qrType.toString().toInt()
    }
}

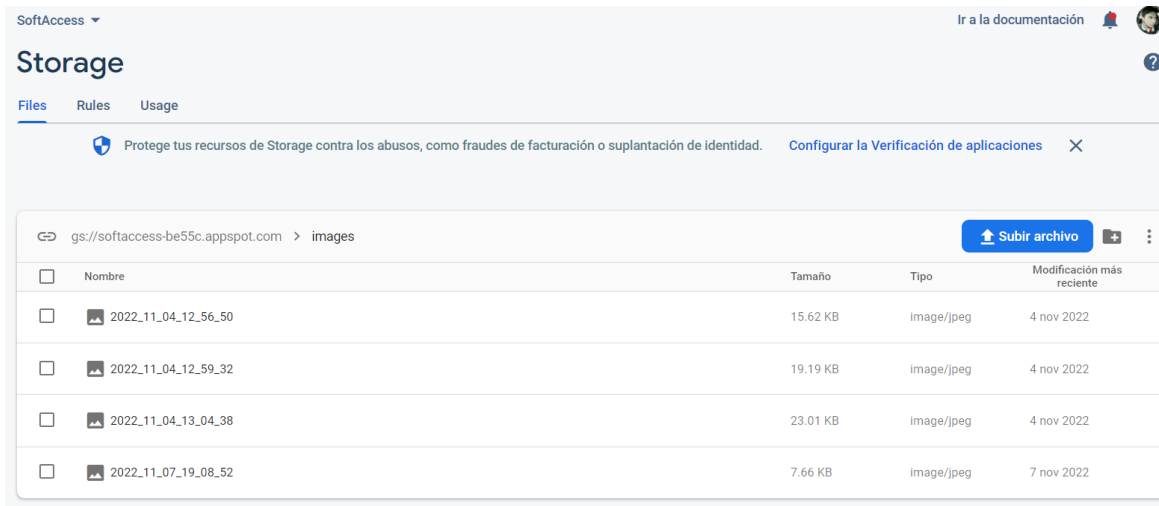
```

Nota. En esta figura se observa la extracción de datos de una colección en Firestore Database a través de un ID en específico.

Storage. Mediante la librería `com.google.firebase:firebase-storage-ktx` fue posible programar el almacenamiento de imágenes en la base de datos de Firebase. Dichas imágenes, las cuales pueden ser documentos o rostros, sirvieron como constancia de que cualquier persona ajena al conjunto quiso acceder. Por tal motivo, se habilitó este servicio para que quede registrado este tipo de eventos en la base de datos como se puede observar en la Figura 49.

Figura 49

Imágenes almacenadas en Storage



Nota. En la figura se evidencia las imágenes almacenadas en la consola del Storage.

En el caso de querer subir una imagen se utilizó la instrucción `putFile` indicando la localización de la carpeta en donde se desea guardar. A continuación, se puede evidenciar en la Figura 50.

Figura 50

Código para subir imágenes al Storage

```

val storageReference = FirebaseStorage.getInstance().getReference(location: "images/$fileName")
storageReference.putFile(imageUri)
    .addOnSuccessListener { it: UploadTask.TaskSnapshot!
        Toast.makeText(context: this@LectorQrActivity, text: "Imagen guardada correctamente", Toast.LENGTH_LONG).show()
    }

```

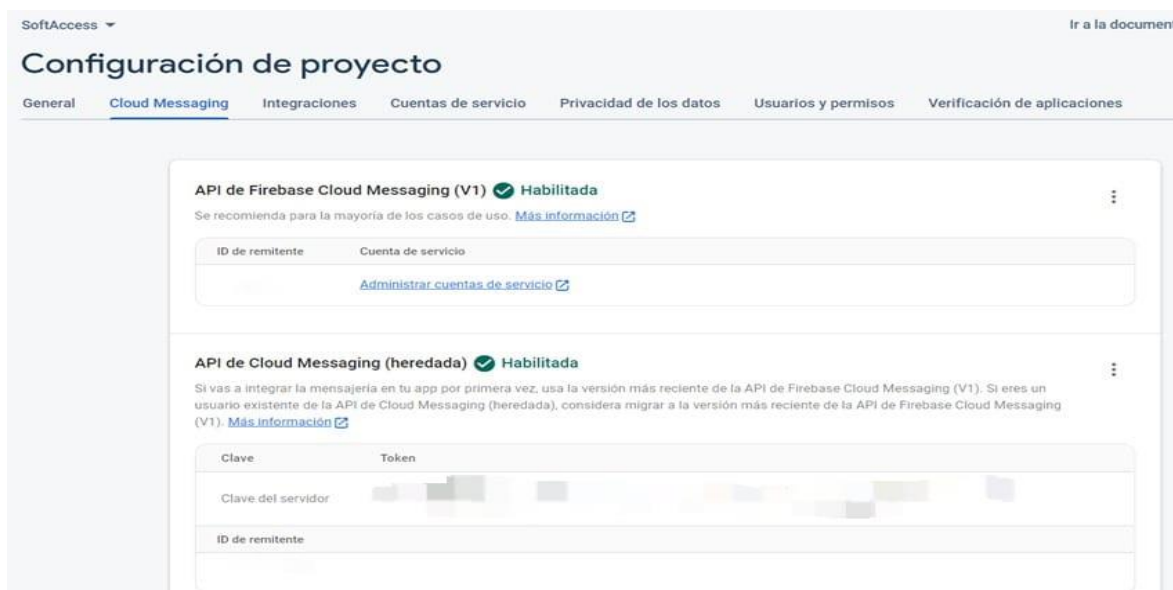
Nota. En la presente figura se observa el código para subir las imágenes al Storage.

Cloud messaging service. La librería que se utilizó para levantar el servicio de notificaciones se denomina, `com.google.firebase:firebase-messaging`, la misma que ayudó con la sincronización entre los dispositivos que se registraron en la aplicación y la base de datos.

Una variable importante que se extrajo del Cloud Messaging service es el token de la clave del servidor, como se pudo observar en el apartado de Notificaciones se creó una función que genere un token para cada dispositivo, de igual forma, la base de datos tiene su propio token, esto resulta importante a la hora de enviar notificaciones generales, es decir, para enviar a todos los dispositivos asociados a la aplicación SoftAccess. En la Figura 51, se observa como en la sección de configuración del proyecto se pudo obtener el token de la clave del servidor.

Figura 51

Vista de Cloud Messaging



Nota. En la figura se puede observar la pestaña de Configuración del Proyecto y dentro de la misma existe el Cloud Messaging que contiene los datos necesarios para la conexión con la Aplicación.

A su vez, en Android Studio se configuró dentro de una clase llamada *Constants* un *companion object*, que sirvió para guardar los datos del Cloud Messaging y a su vez para acceder a ellos desde cualquier clase o Activity del proyecto como se muestra en la Figura 52.

Figura 52

Código de la Clase Constants

```
class Constants {  
  
    companion object {  
        const val BASE_URL = "https://fcm.googleapis.com"  
        const val SERVER_KEY = "AAAA7MQ6YfY:APA91bFDKa169wkYw5IOHXtyCqAtBZqbLc1rsLih4wruYHoL"  
        const val CONTENT_TYPE = "application/json"  
    }  
}
```

Nota. La figura muestra las constantes que sirvieron para configurar el servicio de notificaciones.

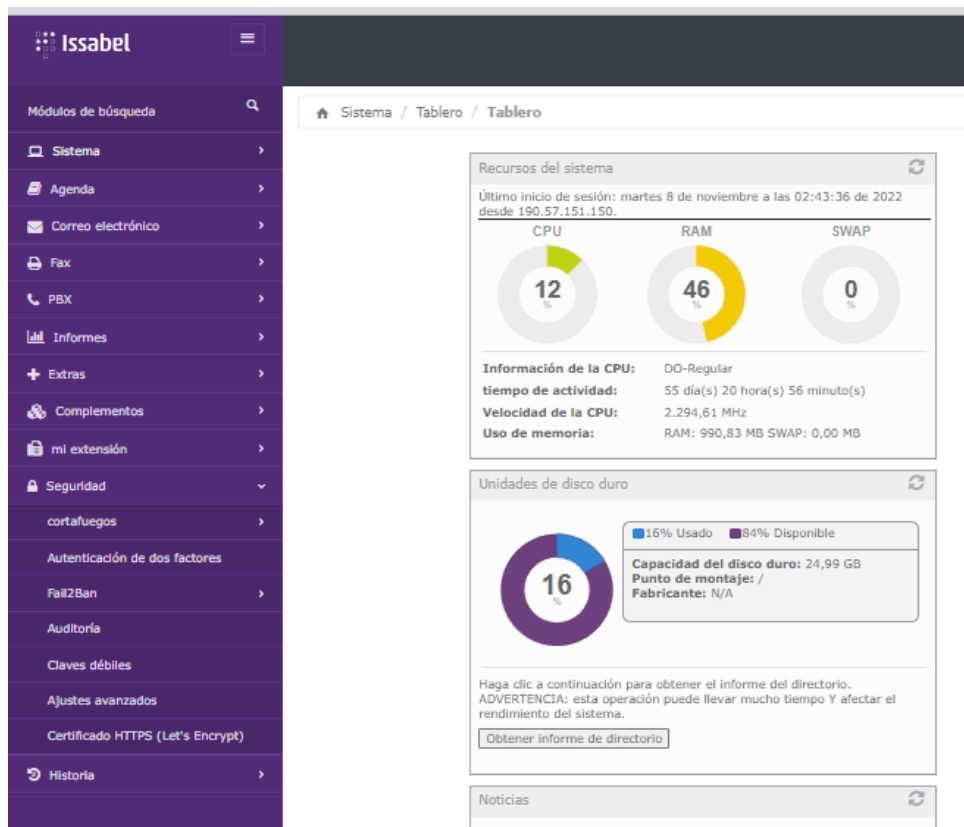
Configuraciones

Digital Ocean e Issabel

Configuración de videollamadas y colas. Mediante Issabel PBX, se seleccionó la opción de “Seguridad” que se encuentra en el lateral izquierdo, seguido de esto la opción de “Ajustes avanzados”. A continuación se indica en la Figura 53 los pasos ya mencionados.

Figura 53

Interfaz de Issabel para la habilitación de la cámara

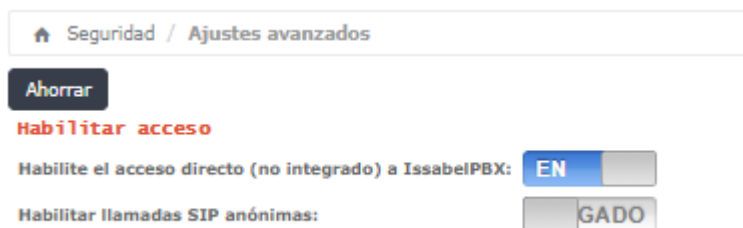


Nota. Esta figura indica el proceso de configuración para habilitar la cámara.

Dentro de “Ajustes avanzados”, se habilitó la opción de Issabel PBX, como se observa en la Figura 54.

Figura 54

Selección de la opción para la habilitación de la cámara

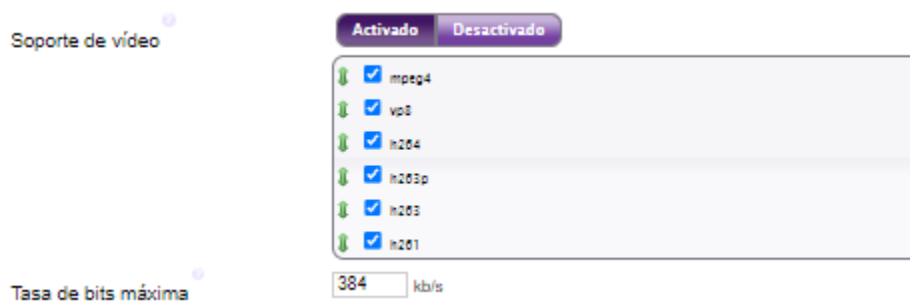


Nota. En esta figura se habilita la primera opción, la cual permite que desde Issabel PBX se pueda habilitar la cámara.

Ahora se procedió a seleccionar la opción de “Configuración de la centralita” “Issabel PBX no integrado”, “Ajustes” y finalmente “Configuración SIP de Asterisk”. De esta forma se obtuvo la siguiente interfaz como la de la Figura 55.

Figura 55

Activación de los códecs para videollamada



Nota. La figura muestra la habilitación de todos los códecs para obtener una videollamada.

Se consideró importante configurar la cola de llamadas con el objetivo de poder atender a todas las llamadas entrantes, es decir, si dos condóminos deciden llamar a una misma persona, la llamada que primero entre será atendida y la segunda tendrá que esperar, mas no será desviada o cancelada. Para dicha configuración, se seleccionó “PBX”, “Configuración de la

Centralita”, “Colas”, y se procedió a llenar los campos: Número de colas, Nombre de la cola, Selección rápida de extensiones y Destino de conmutación por error. A continuación se detalla este proceso y configuración en la Figura 56.

Figura 56

Configuración de la cola de llamadas

Básico

- Clase de servicio
- Extensiones
- Códigos de características
- Rutas de salida
- Bañador
- Control de llamadas entrantes**
- Anuncios
- Lista negra
- Control de flujo de llamadas
- Grabación de llamada
- Fuentes de búsqueda de CallerID
- DID de canal DAHDI
- Rutas Dinámicas
- Sígueme
- IVR
- Rutas entrantes
- Prioridades de cola
- Colas**
- Grupos de timbre
- Establecer identificador de llamadas
- Condiciones de tiempo
- Grupos de tiempo
- Opciones internas y configuración
- Conferencias
- Idiomas
- Aplicaciones misceláneas
- Destinos misceláneos
- Música en espera
- Conjuntos de PIN

Agregar cola

Agregar cola

Número de cola: 2000

Nombre de la cola: Cola de llamada

Contraseña de cola:

Generar sugerencias de dispositivos:

Confirmar llamada:

Llamada Confirmar Anuncio: Defecto

Prefijo del nombre CID:

Prefijo de tiempo de espera: No

Información de alerta:

Agentes estáticos:

Selección rápida de extensiones: (elegir extensión)

Miembros dinámicos:

Destino de conmutación por error

Terminar llamada Hangup

Destino de continuación de la cola

== elige uno ==

Restablecer estadísticas de cola

Correr Nunca

Nota. Por medio de esta figura se visualiza la interfaz que presenta la configuración de las colas.

Creación de extensiones. De igual forma que para la habilitación de la cámara, se debe seleccionar la opción de PBX, Configuración de la centralita, Agregar una extensión y en la opción de Dispositivo SIP Genérico, escoger Dispositivo SIP Genérico y seguido a esto se colocó la información de acuerdo a los condóminos. Este proceso se puede observar en las Figura 57 y Figura 58.

Figura 57

Creación de extensiones

Agregar extensión SIP
- Agregar extensión

Extensión de usuario

Nombre para mostrar

Alias del número de CID

Alias de SIP

- Opciones de extensión

- Agregar extensión
- Miguel Mejía<8>
- rita herrera<16>
- Raúl Olmedo<23>
- Bayron Revelo<25>
- gabriela riva<26>
- Silverio Ibarra<27>
- Irina salazar<28>
- Carlos Naranjo<50>
- marcia cevallos<69>
- Diego Vélez<70>
- Miguel Naranjo<80>
- cel1<9310>
- cel2<9320>
- cel3<9330>
- cel4<9350>

Nota. Esta figura indica el registro de la extensión de usuarios (número de casa) y nombre y apellido de un miembro de cada hogar.

Figura 58

Creación de la clave para cada condómino

Este dispositivo utiliza tecnología sip.

modo dtmf

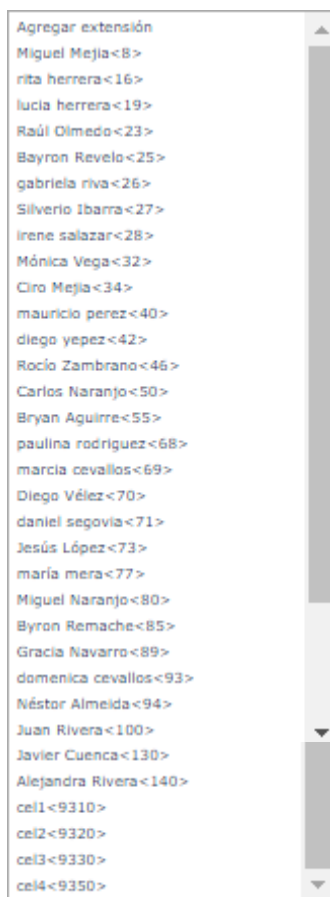
RFC

Nota. En esta figura se escribió la clave correspondiente para cada condómino.

Este proceso fue realizado para los 32 condóminos del Conjunto Benalcázar Oriental, de esta forma se pudieron realizar llamadas/videollamadas.

Figura 59

Creación total de extensiones



Nota. En la presente figura se pueden observar las 32 extensiones creadas en Issabel incluidas cuatro extensiones de prueba, para el caso de los condóminos se tomó en cuenta el nombre del propietario como su número de casa.

Fail2ban. Al usar un servidor en la nube como lo es Digital Ocean y también el software de código abierto de telefonía IP denominado Issabel, ambos se vuelven vulnerables a varios ataques (accesos indeseados o ataques de fuerza bruta) como se mencionó en el capítulo II. Con el fin de evitar este tipo de ataques y obtener seguridad se ejecutaron los siguientes comandos dentro de la consola de Digital Ocean:

- Para instalar se usó el comando `dnf install -y fail2ban`.

Figura 60

Instalación de Fail2ban

```
Installed:
  fail2ban-0.11.2-3.el7.noarch                fail2ban-firewalld-0.11.2-3.el7.noarch
Complete!
[root@softaccess ~]#
```

Nota. Tomando de referencia esta figura, ¿en la parte lateral inferior izquierda se puede observar Complete!, lo cual indica que se ha completado con éxito la instalación de Fail2ban mediante la ejecución de su respectivo comando.

- El comando `cd /etc/fail2ban` indicó los ficheros existentes

Figura 61

Ficheros de Fail2ban

```
Complete!
[root@softaccess ~]# cd /etc/fail2ban/
[root@softaccess fail2ban]# ll
total 60
drwxr-xr-x. 2 root root    4096 Sep 21 22:57 action.d
-rw-r--r--. 1 root root    2816 Nov 23 2020 fail2ban.conf
drwxr-xr-x. 2 root root     6 Oct 13 2021 fail2ban.d
drwxr-xr-x. 3 root root    4096 Sep 21 22:52 filter.d
-rw-r--r--. 1 root root  24996 Oct 13 2021 jail.conf
drwxrwxr-x. 2 root asterisk  51 Sep 27 03:46 jail.d
-rw-r--r--. 1 root root    483 Sep 21 22:57 jail.local
-rw-r--r--. 1 root root   2827 Nov 23 2020 paths-common.conf
-rw-r--r--. 1 root root    930 Nov 23 2020 paths-fedora.conf
```

Nota. Como se puede observar esta figura enlista los ficheros que posee Fail2ban, tanto el `jail.local` como el `jail.conf` fueron configurados para evitar ataques de fuerza bruta.

- Se realizó una copia de seguridad con el siguiente comando `cp -p jail.conf jail.local`.
- Se arrancó el servicio seguido de los comandos `systemctl enable fail2ban`, `systemctl start fail2ban`, `systemctl status fail2ban`.

Figura 62

Fail2ban activo

```
[root@softaccess fail2ban]# vi jail.conf
[root@softaccess fail2ban]# systemctl enable fail2ban
[root@softaccess fail2ban]# systemctl start fail2ban
[root@softaccess fail2ban]# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-09-21 23:53:35 UTC; 5 days ago
     Docs: man:fail2ban(1)
   Main PID: 1318 (fail2ban-server)
   CGroup: /system.slice/fail2ban.service
           └─1318 /usr/bin/python2 -s /usr/bin/fail2ban-server -xf start
```

Nota. En esta figura se puede observar las letras de color verde las cuales indican que Fail2ban se encuentra activo.

- Para realizar configuraciones en *bantime*, *findtime* y *maxretry* se ejecutó el comando `vi /etc/fail2ban/jail.local`. Es de suma importancia conocer que el *bantime*, indica el tiempo en segundos que va a estar bloqueada una IP, mientras que el *findtime* es el tiempo para intentar iniciar sesión, finalmente el *maxretry* señala el número de intentos permitidos. En la Figura 63 se pueden observar las configuraciones realizadas.

Figura 63

Configuración de los parámetros: *bantime*, *findtime* y *maxretry*

```
# "bantime" is the number of seconds that a host is banned.
bantime = 259200

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 120

# "maxretry" is the number of failures before a host get banned.
maxretry = 2
```

Nota. En esta figura se colocaron valores en segundos tanto en el *bantime* (259200) como en el *findtime* (120), en cuanto al *maxretry* se configuró para 2 intentos.

- El comando `tail -f /var/log/fail2ban.log` permitió visualizar en tiempo real cada vez que un usuario se autenticó, es decir se pudo ver las IPs que ingresaron ya sea por ssh o http, de igual forma se pudo visualizar los logs en tiempo real que quisieron ingresar al servicio, así como el momento en el que fueron baneadas.

Mediante la ejecución de estos comandos se logró banear IPs de acuerdo a los valores establecidos en *bantime*, *findtime* y *maxretry*. Tanto en la Figura 64 y 65 se puede evidenciar las IPs así como también el baneo.

Figura 64

IPs baneadas

```

https://cloud.digitalocean.com/droplets/317656814/terminal/ui/
2022-09-28 15:43:19,116 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.114.187
2022-09-28 15:43:19,169 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.115.122
2022-09-28 15:43:19,224 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.141.149
2022-09-28 15:43:19,282 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.154.52
2022-09-28 15:43:19,339 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.161.105
2022-09-28 15:43:19,397 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.163.159
2022-09-28 15:43:19,450 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.166.54
2022-09-28 15:43:19,509 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.197.62
2022-09-28 15:43:19,560 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.21.68
2022-09-28 15:43:19,612 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.59.135
2022-09-28 15:43:19,663 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.60.132
2022-09-28 15:43:19,714 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 128.90.61.152
2022-09-28 15:43:19,732 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.59.93 - 2022-09-28 15:43:19
2022-09-28 15:43:19,733 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.59.93 - 2022-09-28 15:43:19
2022-09-28 15:43:19,771 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 173.247.226.186
2022-09-28 15:43:19,823 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 20.0.27.220
2022-09-28 15:43:19,885 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 20.104.180.196
2022-09-28 15:43:19,936 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 209.58.190.36
2022-09-28 15:43:19,987 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 31.6.10.69
2022-09-28 15:43:20,039 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 31.6.11.120
2022-09-28 15:43:20,093 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 31.6.21.39
2022-09-28 15:43:20,145 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 31.6.22.123
2022-09-28 15:43:20,194 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 31.6.58.71
2022-09-28 15:43:20,245 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 31.6.60.70
2022-09-28 15:43:20,297 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 51.104.52.132
2022-09-28 15:43:20,350 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 51.104.53.91
2022-09-28 15:43:20,402 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 51.120.116.217
2022-09-28 15:43:20,458 fail2ban.actions [14281]: NOTICE [asterisk] Restore Ban 89.163.148.96
2022-09-28 15:43:22,146 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.182.204 - 2022-09-28 15:43:21
2022-09-28 15:43:22,147 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.182.204 - 2022-09-28 15:43:21
2022-09-28 15:43:28,761 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.166.124 - 2022-09-28 15:43:28
2022-09-28 15:43:28,762 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.166.124 - 2022-09-28 15:43:28
2022-09-28 15:43:33,971 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.117.48 - 2022-09-28 15:43:33
2022-09-28 15:43:33,972 fail2ban.filter [14281]: INFO [asterisk] Found 128.90.117.48 - 2022-09-28 15:43:33
2022-09-28 15:43:41,986 fail2ban.filter [14281]: INFO [asterisk] Found 5.62.62.199 - 2022-09-28 15:43:41
2022-09-28 15:43:41,987 fail2ban.filter [14281]: INFO [asterisk] Found 5.62.62.199 - 2022-09-28 15:43:41
2022-09-28 15:43:48,000 fail2ban.filter [14281]: INFO [asterisk] Found 5.62.58.35 - 2022-09-28 15:43:47
2022-09-28 15:43:48,000 fail2ban.filter [14281]: INFO [asterisk] Found 5.62.58.35 - 2022-09-28 15:43:47

```

Nota. En esta figura se pueden constatar las ips que han sido baneadas.

Figura 65

IPs baneadas luego de dos intentos

```

2022-09-28 17:00:19,954 fail2ban.filter [15620]: INFO [sshd] Found 157.100.171.204 - 2022-09-28 17:00:19
2022-09-28 17:00:24,690 fail2ban.filter [15620]: INFO [sshd] Found 157.100.171.204 - 2022-09-28 17:00:24
2022-09-28 17:00:24,696 fail2ban.actions [15620]: WARNING [sshd] 157.100.171.204 already banned

```

Nota. Tomando de referencia esta figura se puede recalcar que la configuración de los parámetros de bantime, findtime y maxretry están funcionando de acuerdo a lo establecido.

Pruebas de desarrollo del sistema

El principal requisito es disponer de dispositivos móviles con sistema operativo Android, internet y contar con una muestra de 32 personas.

Además, se requieren algunos datos de los habitantes del Conjunto para crear las extensiones en la central telefónica y generar los códigos QR permanentes mediante la app con la intención de colocar en los parabrisas de los vehículos. A continuación, se indican los parámetros solicitados:

- Apellidos de la familia
- Número de casa
- Placa del vehículo
- Marca del vehículo
- Modelo del vehículo
- Nombre del dueño

Para el desarrollo de las pruebas se seguirá el siguiente orden:

- Realizar una capacitación al personal de seguridad y a los administradores del Conjunto, puesto que ellos tendrán acceso a todos los servicios de la aplicación.
- Brindar una inducción a la muestra con la finalidad de que tengan conocimiento sobre los pasos a seguir para el correcto uso de la app y la generación de códigos QR.
- Una vez que tanto los guardias como los condóminos puedan interactuar con la app, se procederá a realizar el control de acceso.

Pruebas de llamadas y videollamadas

Se medirá la cantidad de llamadas y videollamadas concretadas mediante los reportes de Issabel PBX, puesto que, por medio de la pestaña *CDR Reports* de la central telefónica se tiene la opción de registro de las llamadas (estado), esto hace referencia, si la llamada fue exitosa, llamada rechazada y llamada ocupada.

Mediante el servicio de *Voip Spear* se medirán parámetros como:

- MOS
- Paquetes perdidos
- Latencia promedio
- Latencia máxima
- Latencia mínima
- Jitter

Por otra parte, se va a utilizar la aplicación *Packet Capture* para capturar los paquetes UDP, RTP y RTCP con el objetivo de analizarlos puesto que arrojará los tipos de códecs empleados.

También se realizará una encuesta específica para evaluar las llamadas para medir subjetivamente la calidad de la llamada.

Pruebas de Lectura de códigos QR

Para esta lectura de códigos QR se va a detectar el porcentaje de códigos QR leídos con éxito, para ello se verificarán los datos que se guarden en Firestore, si hay un problema en la detección de la lectura de un código QR el guardia reportará el evento y se analizarán las posibles causas, como por ejemplo, la red a la que se esté conectado el dispositivo.

Pruebas de Mensajería

Realtime Database soporta el servicio de mensajería y notificaciones por ende se analizará el porcentaje de mensajes y notificaciones que se enviaron con éxito.

Habrá una encuesta sobre la experiencia en general del usuario con el fin de valorar la aplicación. Los parámetros a evaluar son los siguientes:

- Facilidad de uso de la aplicación
- Estética de la interfaz
- Compatibilidad con el dispositivo
- Adecuada visualización del texto dentro de la aplicación
- Facilidad de instalación

Capítulo IV

Análisis de resultados

Introducción

Este capítulo contiene un análisis de los datos y resultados obtenidos una vez concluida la implementación global de este proyecto de grado. Se ha evaluado la instalación de la app, manejo, funcionamiento, calidad de llamadas y videollamadas, establecimiento de chats personales y generales, generación de códigos QR (temporales), escaneo y lectura de los mismos.

El QoS será evaluado como un parámetro fundamental para garantizar que las llamadas y videollamadas tengan una buena calidad de audio y video respectivamente. Se usará el programa Packet Capture para analizar los paquetes como por ejemplo: UDP, RTP y RTCP, así como los códecs empleados. Dicho de otra manera, mediante la obtención de los resultados se pretende evaluar la interacción entre los condóminos, guardias y app para cumplir con la satisfacción de los usuarios finales y cumplir los objetivos de este trabajo de titulación.

Cálculo de la muestra

Anteriormente en el capítulo 3 se mencionó el número total de casa dentro del Conjunto, entonces, partiendo de esta información, se debe mencionar lo siguiente:

- Total de casas: 108
- Total de casas habitadas: 106
- Condóminos que disponen de sistema operativo Android: 56%
- Condóminos que disponen de sistema operativo iOS: 44%

Teniendo en cuenta el porcentaje de condóminos que disponen únicamente de dispositivos móviles Android y el total de casas habitadas, se tiene un total de 60. De acuerdo

con Guamán y Rubio (2022), este valor vendría a ser la población total, de aquí se parte para calcular la muestra y obtener un cierto número de datos/resultados que luego serán analizados. En este caso se tiene una población finita y la fórmula a usar para calcular la muestra es la siguiente:

$$n = \frac{N * z^2 * p * q}{e^2(N - 1) + (z^2 * p * q)} \quad (1)$$

Donde:

n : Tamaño de la muestra

N : Tamaño de la población

z : Parámetro que depende del nivel de confianza

p : Probabilidad a favor

q : Probabilidad en contra

e : Error de la muestra

Se determinó que tanto la probabilidad a favor y en contra tendrán un valor del 50% debido a que el sistema de control de acceso que mantiene este Conjunto es por medio de guardias, esto quiere decir que, que el desarrollo de este proyecto es nuevo para los condóminos y al no existir otros trabajos de titulación semejantes a este, no se pueda tomar de referencia los porcentajes de p y q .

Existe una tabla de *Niveles de confianza* la cual es utilizada comúnmente por proporcionar el valor del parámetro z , se indica a continuación en la Tabla 5.

Tabla 5*Niveles de confianza NC*

Nivel de confianza NC	Porcentaje de error	Valor z
90%	10%	1.645
91%	9%	1.69
92%	8%	1.75
93%	7%	1.81
94%	6%	1.88
95%	5%	1.96

Nota. En esta tabla se pueden observar los valores correspondientes de porcentaje de error y z de acuerdo a cada NC (Guamán & Rubio, 2022).

Datos a ser reemplazados en la fórmula (1) sabiendo que se usará el NC del 90%:

N: 60

NC: 90%

z: 1.645

p: 50%

q: 50%

% *e*: 10%

$$n = \frac{(60) * (1.645)^2 * (0.5) * (0.5)}{0.1^2(60 - 1) + ((1.645)^2 * (0.5) * (0.5))} = 32.05 \approx 32 \text{ personas}$$

Análisis de la implementación

Los condóminos que habitan en el Conjunto Habitacional Benalcázar Oriental poseen algunas necesidades que involucran su propia seguridad y también la de sus bienes materiales, de modo idéntico, los moradores sienten preocupación en el momento que personas ajenas al conjunto entran por diferentes motivos.

Después de haber expuesto este proyecto de titulación a la Directiva del Conjunto se pudo acoplar y mejorar la app con el fin de que los moradores se sientan a gusto, por ende, se hicieron algunas modificaciones en Android Studio tales como:

- Generación de códigos QR para mudanzas.
- Activación de sonido para el chat general.
- Captura de fotos a personas (ajenas al Conjunto) o documentos al momento de ingresar.
- Reimpresión de los stickers (códigos QR) con fondo blanco y mejora del tipo de papel.

En otras palabras, tanto los moradores como el personal de seguridad pudieron interactuar con la app y establecer llamadas, videollamadas, envío de mensajes personales y grupales, generación de códigos QR temporales para visitantes, comida a domicilio, taxi, servicio de transporte y mudanza. Todos los servicios que ofrece esta app tienen la ventaja de ser utilizados en cualquier lugar siempre y cuando se mantenga una conexión a internet estable.

Evidentemente se logra el objetivo principal de esta tesis, “Realizar el diseño e implementación de un servidor de Telefonía IP y un softphone basado en librerías Speech to Text y generación de códigos QR para el control de acceso en una Urbanización”, cabe destacar que dicha implementación se realizó en dos fases con el fin de abarcar todos los objetivos específicos y satisfacer los requerimientos de los condóminos, lo cual implicó el

manejo de diferentes herramientas para la obtención de los datos correspondientes. En las Figura 66, 67 y 68 se indica la primera fase.

Figura 66

Evidencias de la primera fase de prueba



Nota. En estas figuras se puede notar la inducción que se dio a los moradores del Conjunto sobre el desarrollo del proyecto de tesis.

Figura 67*Inducción al personal de seguridad*

Nota. Las figuras indican la inducción brindada al personal de seguridad sobre la instalación, manejo de la app, escaneo y lectura de los códigos QR temporales.

Figura 68*Colocación del código QR*

Nota. La figura muestra el código QR colocado en el parabrisa, al ser la primera impresión de los stickers, el material usado no fue el apropiado por presentar demasiado brillo al ser escaneado.

A fin de cumplir con el primer objetivo específico “Implementar un servidor de Telefonía IP virtual alojado en la nube que permita gestionar llamadas, videollamadas y mensajes de texto entre la garita de los guardias y los habitantes en una Urbanización Residencial”, se tuvo que investigar sobre servidores de Telefonía IP alojados en la nube, Bases de datos, IDE de desarrollo, etc., así como también, la programación de los servicios básicos ofrecidos por un softphone. De esta manera se pudo instalar la app en varios dispositivos Android y comprobar que los requerimientos solicitados (sección 3.3) por parte de la Directiva del Conjunto Habitacional y su funcionamiento en general sean óptimos, en cuanto a la generación de códigos QR, en un inicio se imprimieron cerca de 10 stickers que al ser colocados en los parabrisas de los vehículos no se pudo realizar un escaneo óptimo debido a que el material del

papel brillaba y esto retardó el proceso, por lo consiguiente, se mejoró la calidad del papel y por ende el proceso de escaneo y lectura fue el adecuado. A continuación, se indica en la Figura 69 la colocación del sticker mejorado.

Figura 69

Colocación de stickers en los parabrisas de lo vehículo de los condóminos del Conjunto Habitacional Benalcázar Oriental



Nota. En la presente figura se observa la colocación de los stickers en los parabrisas de los vehículos de los condóminos.

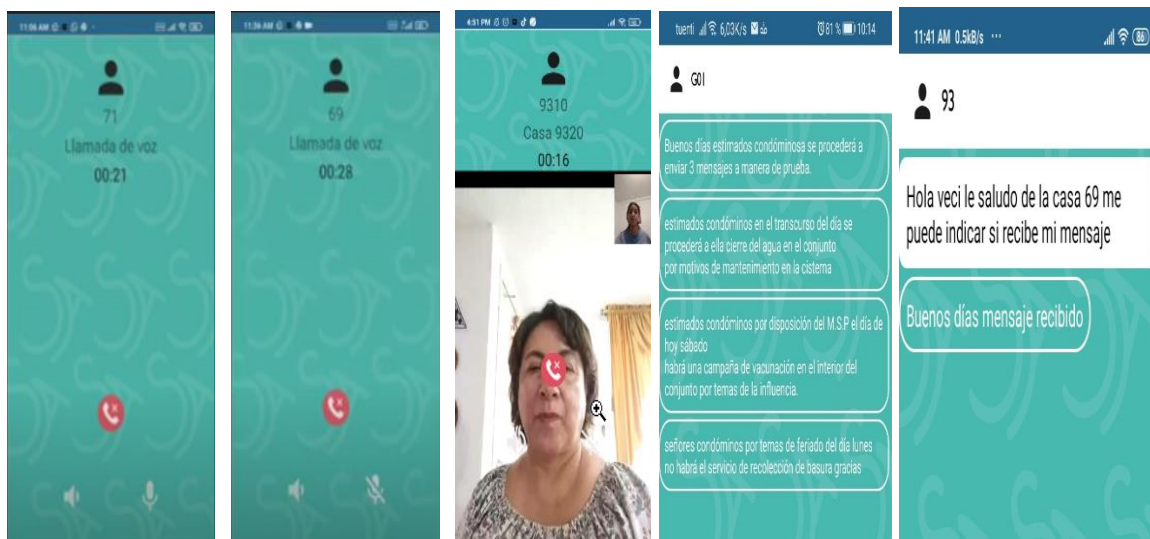
El segundo objetivo planteado fue “Diseñar y desarrollar un softphone para un sistema de control de acceso que permita la generación y lectura de códigos QR y conversiones de voz a texto mediante librerías Speech To Text” para almacenar en la base de datos una doble confirmación de las personas no residentes al Conjunto, esto se realizó en dos etapas, la primera fue que el personal de seguridad escanee el código presentado por las visitas, seguido de esto se le pidió decir el número de casa a la que se dirige, de esta forma, se desplegó un mensaje “ACCESO EXITOSO A LAS:” en donde se confirma el acceso y se registra la fecha y hora, caso contrario el visitante no podrá ingresar. Una vez concluida la visita, los guardias registraron la salida mediante el escaneo del código QR, de igual forma, fecha y hora.

El siguiente objetivo “Evaluar y realizar mediciones en sitio sobre el sistema implementado a fin de determinar el óptimo funcionamiento del sistema de control de acceso y el registro de la base de datos” va de la mano con las dos fases de pruebas anteriormente mencionadas. La primera fase resultó ser de suma importante dado que surgieron algunos inconvenientes como por ejemplo: coordinación y disponibilidad de tiempo con los condóminos, conexión a Internet, calidad del material de los stickers.

Posterior a esto en la segura fase se pudieron solventar todos los contratiempos logrando instalar y probar el sistema de forma global. En las siguientes Figuras 70, 71, 72 se puede evidenciar.

Figura 70

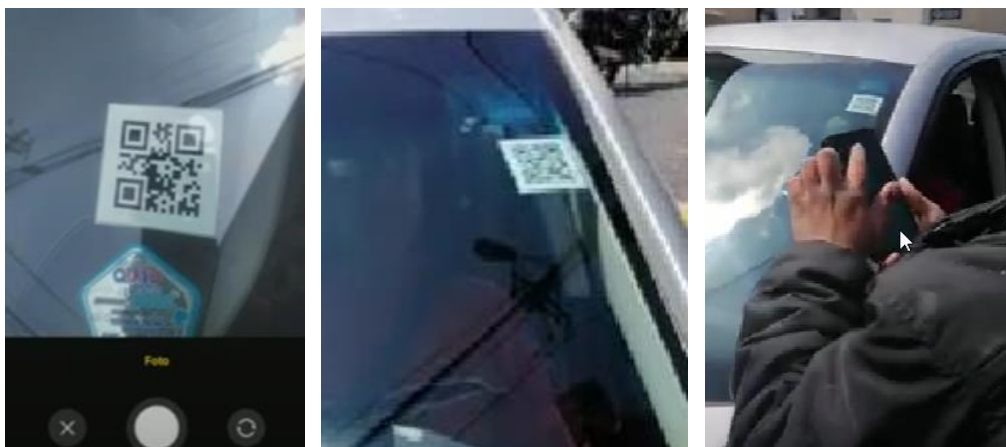
Llamadas, videollamadas y mensajes de prueba



Nota. En las figuras se puede notar el correcto funcionamiento de llamadas, mensajes y videollamadas.

Figura 71

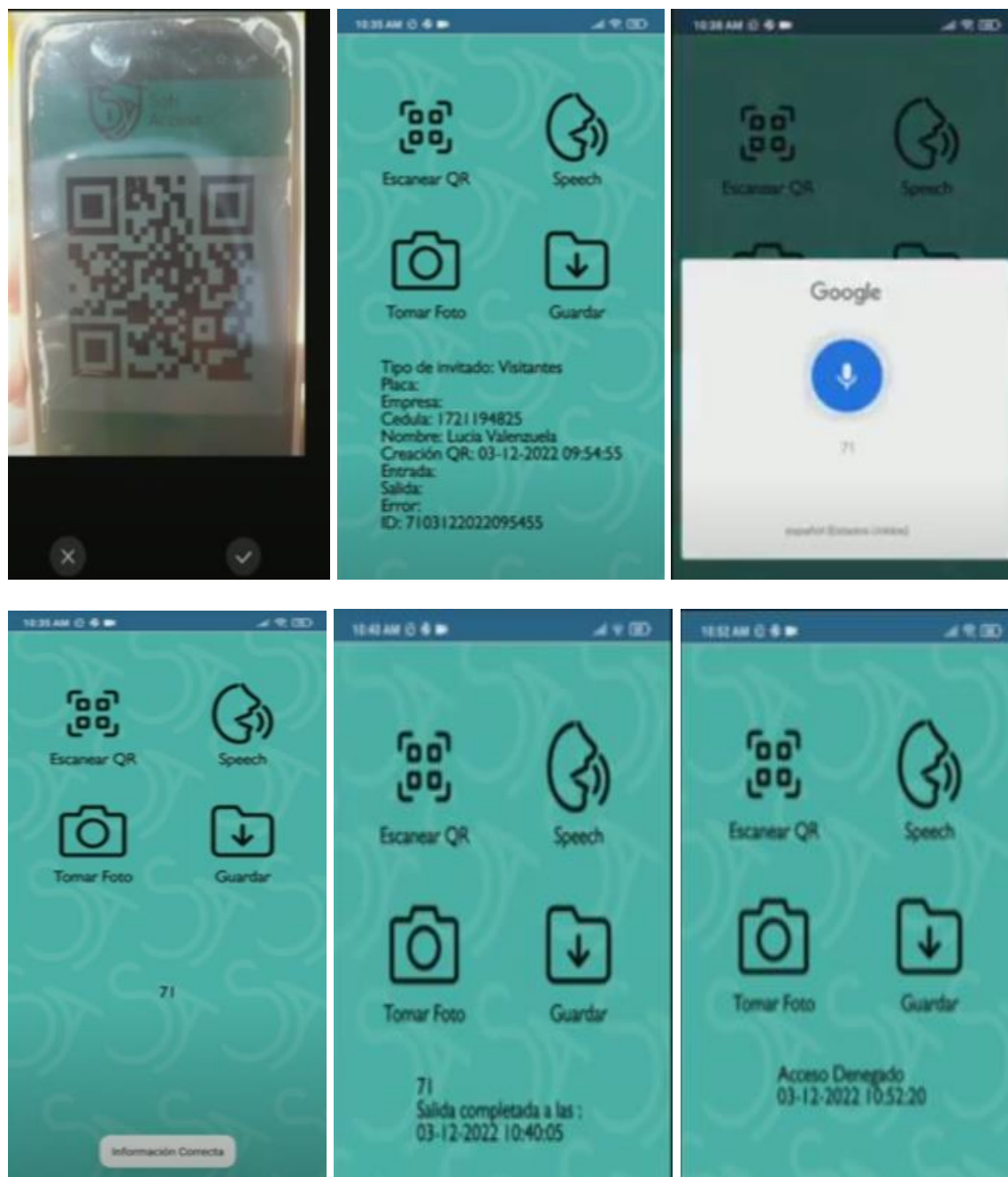
Escaneo y lectura de códigos QR permanentes por parte del personal de seguridad



Nota. En estas figuras se puede notar como el personal de seguridad realizar el proceso de lectura y escaneo de los códigos QR permanentes, a su vez, del almacenamiento de la información en la base datos.

Figura 72

Proceso de ingreso de visitantes



Nota. En estas figuras se detalla el proceso de lectura y escaneado del código QR temporal y la doble verificación que el guardia hizo al tener un visitante en la guardianaía.

La última figura indica que el visitante intentó acceder al Conjunto por tercera, es por esta razón que salió el mensaje de “Acceso denegado”, pues el código únicamente sirve para dos ocasiones (entrada y salida).

El proceso detallado para visitas es el mismo a emplear para: comida a domicilio, taxi, servicio de transporte y mudanzas.

Análisis del desempeño y funcionamiento del sistema de control de Acceso considerando QoS.

Para validar el sistema de control de acceso implementado fue imprescindible realizar una evaluación de la calidad de servicio con el fin de cumplir con el objetivo específico “Realizar el análisis de la calidad de servicio que ofrece el servidor y el softphone implementado utilizando métodos objetivos y subjetivos MOS y PSQM, PESQ respectivamente” por este motivo se utilizó el criterio MOS (Means Opinion Score) por medio de una encuesta que respondieron los condóminos. También se requirió los servicios de la plataforma Volp Spear para el monitoreo de la central telefónica.

Dicha encuesta fue realizada en Google Forms con el propósito de que los usuarios finales (condóminos y personal de seguridad) puedan evaluar la app y los servicios de la misma. Algunas preguntas fueron respondidas en base a los siguientes niveles de confianza que se detallan a continuación:

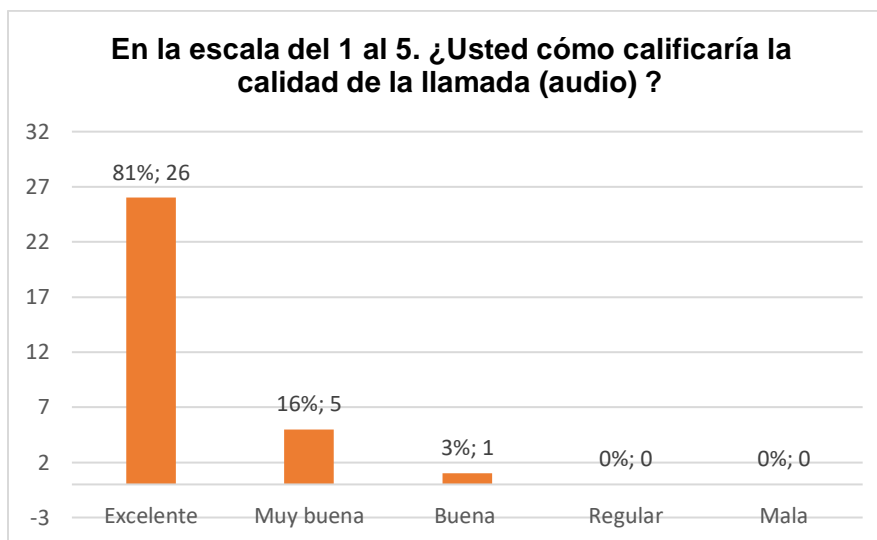
- 5 Excelente
- 4 Muy buena
- 3 Buena
- 2 Regular1 Mala

De este modo se presentan las preguntas con su respectivo análisis:

Pregunta 1. En la escala del 1 al 5. ¿Usted cómo calificaría la calidad de la llamada (audio)?

Figura 73

Calidad del audio de la llamada



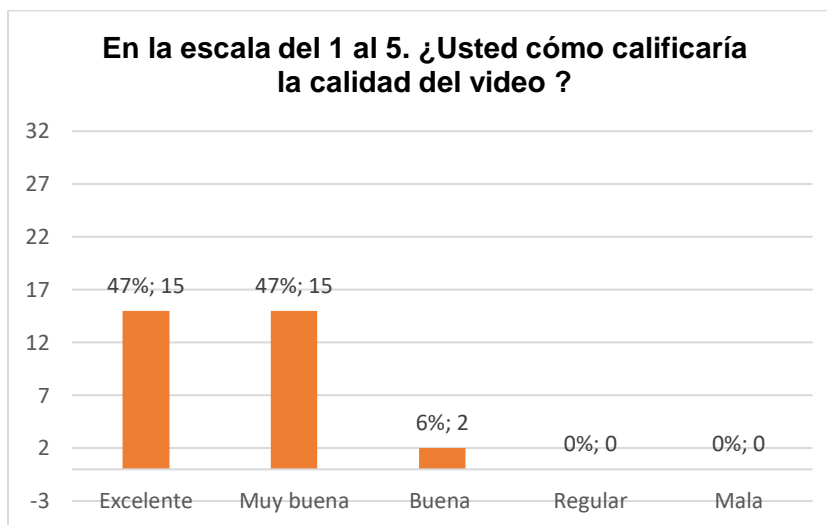
Nota. La figura indica la representación en gráfico de barras la calidad de audio que los condóminos y personal de seguridad experimentaron.

En la Figura 73 se observa la calidad de audio experimentada en las llamadas de prueba, basándose en la Tabla 4 (sección 2), la calidad es Excelente puesto que el 81% de la muestra (32 personas) mantuvieron una buena comunicación, el 16% señalaron que fue “Muy buena” y el 3% anunció que fue “Buena”. Para este último porcentaje se puede acotar que en la guardiana no existe una buena conectividad a la red WiFi del Conjunto, además la actividad del personal hace que la conexión cambie constantemente.

Pregunta 2. En la escala del 1 al 5. ¿Usted cómo calificaría la calidad del video?

Figura 74

Calidad de la videollamada



Nota. La presente figura señala por medio de un gráfico de barras las respuestas obtenidas por la muestra participante una vez experimentado la videollamada.

Mediante la Figura 74 se tiene que el 47% de la muestra indicó que la calidad de la videollamada fue “Excelente”, mientras que para el otro 47% fue “Muy buena” para el 6% fue “Buena”. De acuerdo a la Tabla 4, se puede concluir que según la Escala MOS se están cumpliendo los valores de 5,4 y 3 en donde la imagen y el audio permiten establecer una comunicación normal más no deteriorada.

Pregunta 3. Usted considera que mediante la app ha mejorado la comunicación entre el guardia y los condóminos.

Figura 75

Comunicación entre guardias y condóminos



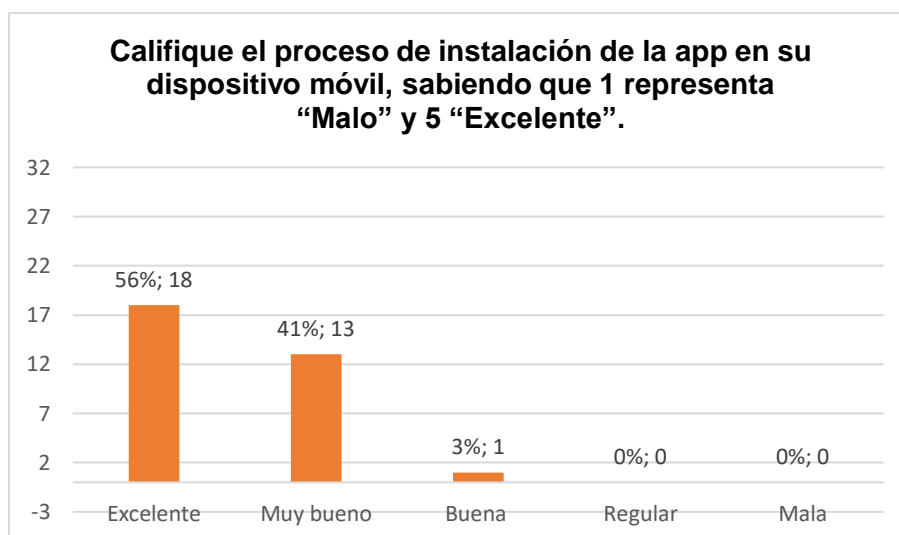
Nota. Esta figura refleja la respuesta de los condóminos en relación a la comunicación que mantuvieron con los guardias.

Por medio de la Figura 75 se puede notar que se logró mejorar la comunicación entre el personal de guardia y los condóminos que instalaron la app en sus dispositivos móviles. Tener el 100% como respuesta positiva indica que la app ayuda a establecer comunicación.

Pregunta 4. Califique el proceso de instalación de la app en su dispositivo móvil, sabiendo que 1 representa “Malo” y 5 “Excelente”.

Figura 76

Instalación de la app en los dispositivos móviles Android



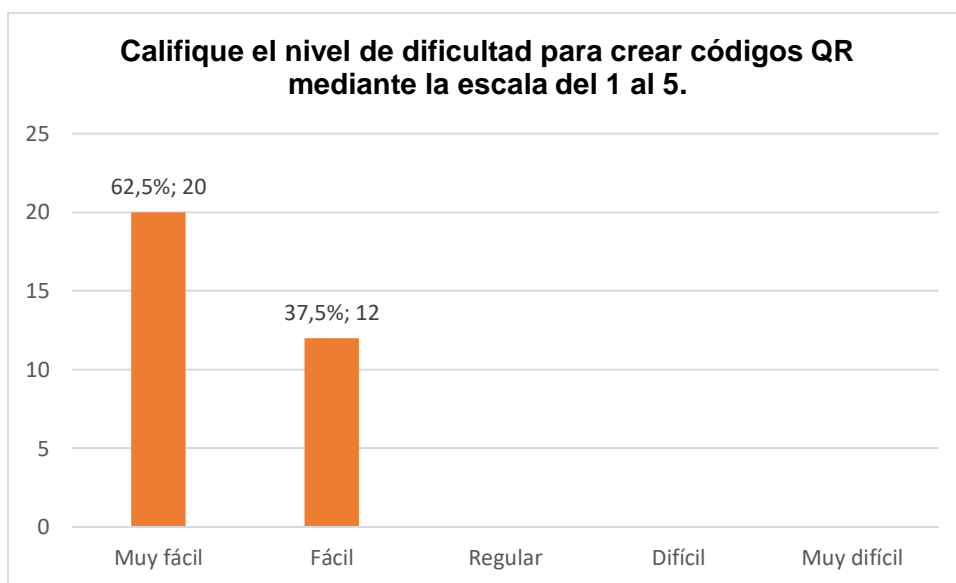
Nota. La figura muestra la opinión de los condóminos en el proceso de instalación de la app en sus respectivos dispositivos móviles.

La Figura 76 presenta a manera de diagrama de barras las respuestas proporcionadas por los condóminos manifestando que el 56% del total dijo que fue “Excelente” el proceso de instalación, el 41% comentó que fue “Bueno”, sin embargo, para el 3% fue “Bueno”. Tomando de referencia la Tabla 10, este proceso se encuentra entre los valores del 5 al 3, lo que indica que las inducciones que se dieron de forma personal a cada condómino fueron muy útiles, puesto que, las dudas que ellos tuvieron se pudieron solventar en ese momento y en la mayoría de casos se explicó como deben manejar e interactuar con la app. Sin embargo, algunas personas fueron de la tercera edad y no estaban familiarizadas con el uso de la tecnología, pese a esto, se brindó una inducción con el propósito de que dichos condóminos sepan las funcionalidades que tiene la app.

Pregunta 5. Califique el nivel de dificultad para crear códigos QR mediante la escala del 1 al 5.

Figura 77

Creación de códigos QR



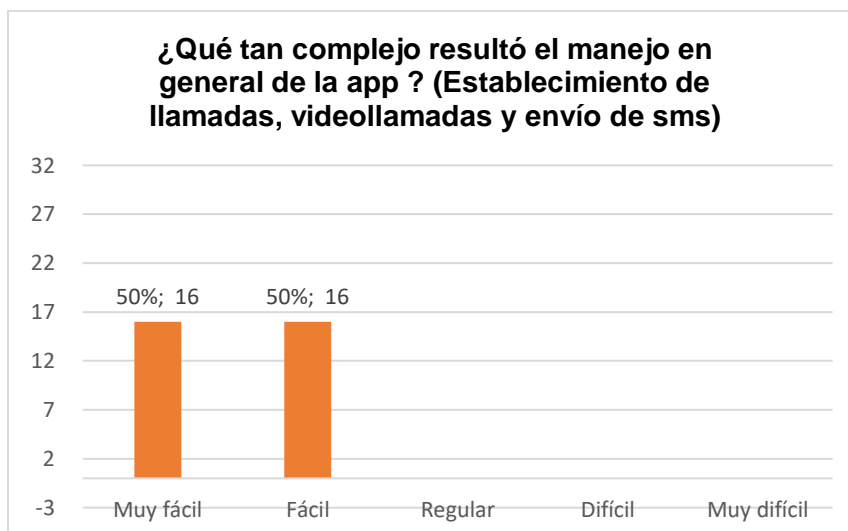
Nota. La gráfica de barras presente indica las respuestas proporcionadas por medio de la encuesta.

A través de la Figura 77 y tomando de referencia a la Tabla 4, los porcentajes que se pueden observar: 62.5% y 37.5% representan la opinión de los condóminos en cuán difícil fue crear códigos QR temporales para poder recibir visitas, comida a domicilio, servicio de transporte, etc. La tabla 4 nos indica que los valores 4 y 5 correspondientes a esta pregunta, son “Muy fácil” y “Fácil”, es decir, el manejo de la app para la generación de los códigos QR es sencillo y rápido.

**Pregunta 6. ¿Qué tan complejo resultó el manejo en general de la app?
(Establecimiento de llamadas, videollamadas y envío de sms)**

Figura 78

Manejo de la app



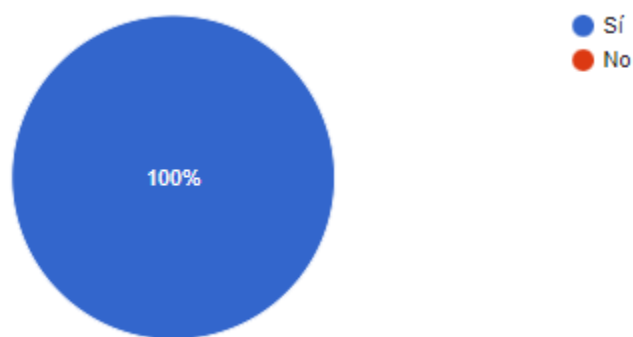
Nota. La presente figura permite visualizar las respuestas de lo complejo que resultó el manejo de la app.

En la Figura 78 se establecieron 5 opciones de respuesta teniendo de porcentaje el 50% para “Muy fácil” y el otro 50% para “Fácil”. Se puede comprobar que las respuestas son óptimas, entonces, los condóminos están en toda la capacidad de interactuar con el guardia cada vez que lo requieran de una forma eficiente.

Pregunta 7. ¿Usted considera que por medio de esta app el control de acceso de su lugar de residencia se puede monitorear mediante los servicios ofrecidos como, mensajería, llamada y videollamada desde su dispositivo móvil?

Figura 79

Control de acceso del lugar de residencia



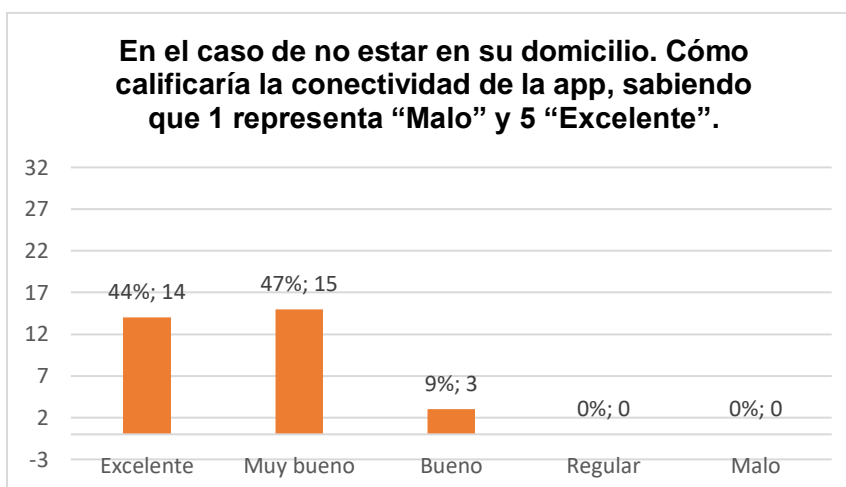
Nota. Esta gráfica que corresponde a un diagrama de pastel indica las respuestas de los 32 participantes en las pruebas piloto con el fin de saber si se puede monitorear el lugar de residencia de los condóminos por medio de todos los servicios que tiene la app.

El diagrama de pastel de la Figura 79 señala que el 100% de la muestra participante en las pruebas piloto (fase 1 y fase 2) realizadas consideran que por medio de esta app se puede monitorear su lugar de residencia, puesto que, todos los servicios ofrecidos dentro de la app, pueden ser usados libremente siempre y cuando se tenga conexión a internet.

Pregunta 8. En el caso de no estar en su domicilio. Cómo calificaría la conectividad de la app, sabiendo que 1 representa “Malo” y 5 “Excelente”.

Figura 80

Conectividad de la app



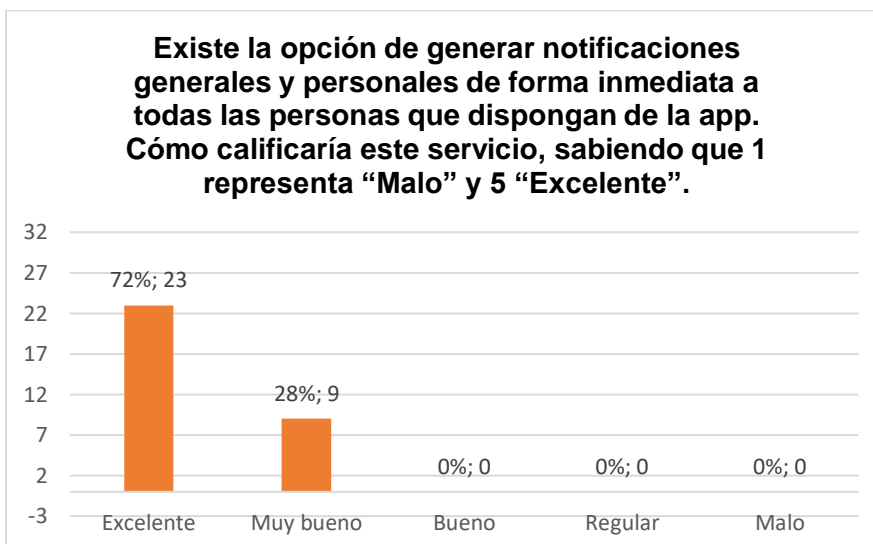
Nota. El diagrama de barras de esta figura señala los porcentajes obtenidos de la encuesta de Google Forms acerca de la conectividad de la app.

El diagrama de barras de la Figura 80 señala que el 47% del total de la muestra indicó que la conectividad de la app estando fuera del domicilio es “Muy buena”, el 44% mencionó que es “Excelente” y el 9% “Buena”. Siguiendo los valores establecidos en la Tabla 4, se puede decir que la conectividad oscila entre valores del 5 al 3. Una forma de mejorar la conectividad es brindar al personal de seguridad una conexión estable y un ancho de banda correspondiente para audio y video, además, proporcionarles dispositivos móviles adecuados para la instalación y correcto funcionamiento de la app. También se debe tener en cuenta el tipo de plan que dispone cada persona, es decir, la cantidad de megas existentes para que la app pueda funcionar completamente.

Pregunta 9. Existe la opción de generar notificaciones generales y personales de forma inmediata a todas las personas que dispongan de la app. Cómo calificaría este servicio, sabiendo que 1 representa “Malo” y 5 “Excelente”.

Figura 81

Generación de notificaciones personales y generales



Nota. En este diagrama de barras se puede observar la opinión de los condóminos en cuanto al servicio de notificaciones personales y generales.

En el diagrama de barras de la Figura 81 se plasma la opinión de los condóminos sobre la generación de notificaciones personales o grupales, que teniendo de referencia a la Tabla 4, el 72% corresponde a un valor de 5 mientras que el 28% a un 4, es decir, “Excelente” y “Muy bueno” respectivamente. Este tipo de servicio resultó muy útil en el caso de suscitarse algún tipo de emergencia o de tener que informar a todos los condóminos alguna novedad importante.

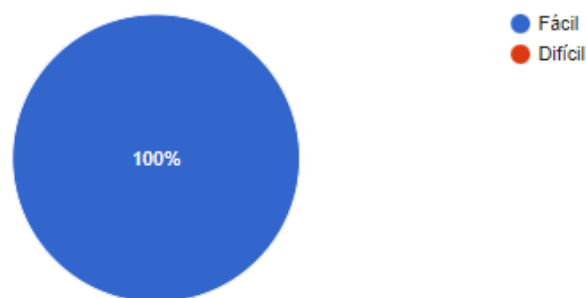
Dado que el personal de guardia instaló y manejó la app en sus dispositivos móviles, se consideró oportuno solicitarles que llenen la misma encuesta de los condóminos con la

diferencia de que se establecieron dos preguntas en relación al proceso que únicamente hicieron los guardias. En las Figuras 81 y 82 se pueden notar ambas preguntas.

Pregunta 10. Para el personal de seguridad. ¿Qué tan complejo resultó hacer la lectura de los códigos QR?

Figura 82

Nivel de complejidad de la lectura de los códigos QR



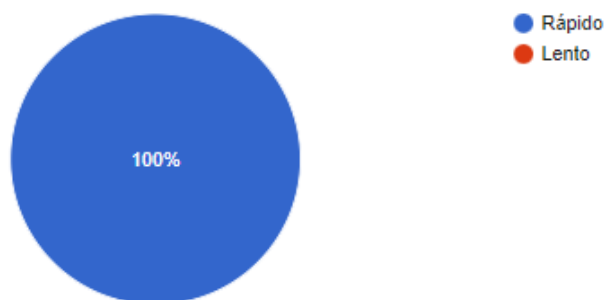
Nota. La presente figura está representada por un diagrama de pastel con las respuestas proporcionadas por el personal de seguridad del Conjunto.

En la Figura 82 se puede observar el diagrama de pastel en donde el 100% corresponde a que es fácil realizar la lectura y escaneo de los códigos QR ya sean temporales o permanentes. De este modo, el personal de seguridad evidenció que es un proceso rápido y sencillo de efectuar, debido a que el almacenamiento de datos de hora y fecha, nombre, placa, cédula, etc., es en tiempo real y de forma automática.

Pregunta 11. Para el personal de seguridad. La lectura de los códigos QR usted considera un proceso:

Figura 83

Lectura de códigos QR



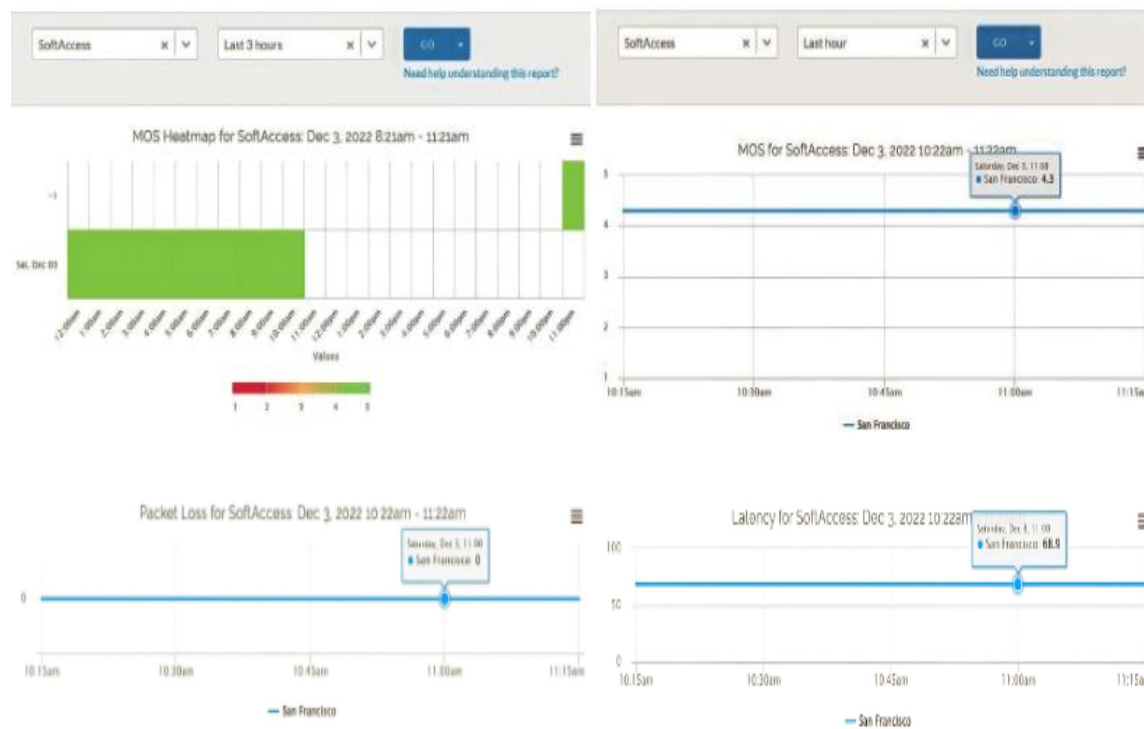
Nota. El diagrama de pastel que se puede evidenciar en esta figura indica la respuesta dada por parte del personal de seguridad a través del Formulario de Google.

La Figura 83 muestra que la lectura de los códigos QR es un proceso rápido, esto representa el 100% de las respuestas obtenidas. Con el fin de no tener varios vehículos en la calle principal haciendo cola para entrar al Conjunto se constató que el escaneo demora cerca de 4 a 5 segundos.

Por otra parte, mediante la plataforma VoIP Spear se registraron los resultados MOS obtenidos de las pruebas de llamadas y videollamadas en el intervalo de tiempo de un día (3 de Diciembre) en el horario de 8:00 a 12: 00 aproximadamente donde se pudo probar con los usuarios el funcionamiento de la app, en la Tabla 6 se puede visualizar las imágenes que indican la calidad del audio y video en una escala del 1 al 5 de la cual los mapas de calor MOS se encuentran en el valor de 4.3.

Figura 84

Mapa de calor MOS y latencia de VoIP Spear



Nota. En estas figuras se observan los mapas de calor MOS, paquetes perdidos y latencia.

En la Tabla 6 se aprecia la latencia obtenida mediante la plataforma VoIP Spear que varía entre 68 y 70 ms. Según la recomendación ITU-T para que la comunicación entre los usuarios de una llamada sea eficiente, la latencia máxima de ida y vuelta no debe superar los 300 ms, caso contrario habrá problemas de sincronización en la llamada.

Tabla 6*Parámetros de Calidad de Servicio*

Fecha	MOS	Packet Loss	Latencia (ms)	Jitter (ms)
Dec 3, 2022 8:15	4,3	0	69	0,5
Dec 3, 2022 8:30	4,3	0	69	0,6
Dec 3, 2022 8:45	4,3	0	69	0,6
Dec 3, 2022 9:00	4,3	0	69,1	0,8
Dec 3, 2022 9:15	4,3	0	69	0,8
Dec 3, 2022 9:30	4,3	0	69,2	1,1
Dec 3, 2022 9:45	4,3	0	69,1	0,8
Dec 3, 2022 10:00	4,3	0	69,1	0,8
Dec 3, 2022 10:15	4,3	0	69,1	0,7
Dec 3, 2022 10:30	4,3	0	69,1	0,8
Dec 3, 2022 10:45	4,3	0	69,1	0,9
Dec 3, 2022 11:00	4,3	0	68,9	0,5
Dec 3, 2022 11:15	4,3	0	69,1	0,8

Nota. En la presente tabla se visualizan los datos MOS, paquetes perdidos, latencia y jitter de la central telefónica (Toepoke Software, 2022).

Dentro de la central telefónica Issabel PBX se extrajeron los reportes CDR expuestos en la Tabla 7 los cuales muestran datos de cada una de las llamadas, extensiones de origen y destino junto con el estatus de las llamadas y su duración.

En total existieron aproximadamente 79 llamadas de las cuales 48 se concretaron, es decir, hubo conexión y comunicación entre las extensiones participantes, 17 no contestadas, 4 fallidas y 10 en estado de ocupado.

La razón por la cual algunas llamadas tenían el estado de ocupado fue porque la central telefónica registraba este estado para la extensión que salía de una videollamada reciente por lo que, al llamar al instante siguiente de haber colgado, la llamada no se concretaba y salía un mensaje de voz indicando que dicha extensión no está disponible ese momento. Esto se debe a las configuraciones realizadas en la central.

Otras llamadas directamente aparecieron con estado fallido debido a la conexión a internet, puesto que al hacer las pruebas en el Conjunto se hacía la llamada con diferentes tipos de redes de internet incluyendo datos móviles los cuales en dicha área en particular no eran óptimos, sin embargo, dicha llamada se conectaba y posterior a esto fallaba.

Dentro de las llamadas concretadas el promedio de tiempo en llamada es de 20 segundos.

Tabla 7

Reportes CDR Issabel PBX

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
10/12/2022 10:58	9350	9310	SIP/9350- 00000315	SIP/9310- 00000316	ANSWERED	28s
10/12/2022 10:57	9350	9310	SIP/9350- 00000311	SIP/9310- 00000312	ANSWERED	32s
10/12/2022 10:57	9310	9310	SIP/9310- 0000030f	SIP/9310- 00000310	ANSWERED	4s
10/12/2022 10:56	9350	9310	SIP/9350- 0000030d	SIP/9310- 0000030e	ANSWERED	32s
10/12/2022 10:56	9350	9310	SIP/9350- 0000030b	SIP/9310- 0000030c	ANSWERED	32s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
			SIP/9350-	SIP/9310-	NO	
10/12/2022 10:55	9350	9310	00000309	0000030a	ANSWER	0s
			SIP/9350-			
10/12/2022 10:55	9350	9310	00000308		ANSWERED	12s
			SIP/9310-			
10/12/2022 10:54	9310	9310	00000307		ANSWERED	4s
			SIP/9350-	SIP/9310-		
10/12/2022 10:49	9350	9310	00000305	00000306	ANSWERED	32s
			SIP/9350-	SIP/9310-		
10/12/2022 10:48	9350	9310	00000303	00000304	ANSWERED	32s
			SIP/9350-	SIP/9310-		
10/12/2022 10:47	9350	9310	00000301	00000302	ANSWERED	33s
			SIP/9350-	SIP/9310-	NO	
10/12/2022 10:47	9350	9310	000002ff	00000300	ANSWER	0s
			SIP/9320-	SIP/19-	NO	
9/12/2022 21:41	9320	19	000002fd	000002fe	ANSWER	0s
			SIP/19-	SIP/9320-		
9/12/2022 21:41	19	9320	000002fb	000002fc	ANSWERED	5s
			SIP/19-	SIP/9320-		
9/12/2022 21:40	19	9320	000002f9	000002fa	FAILED	0s
			SIP/19-			
9/12/2022 21:39	19	9310	000002f8		ANSWERED	6s
			SIP/19-			
9/12/2022 21:39	19	9310	000002f7		ANSWERED	4s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
			SIP/9320-	SIP/19-	NO	
9/12/2022 21:38	9320	19	000002f5	000002f6	ANSWER	0s
			SIP/19-			
9/12/2022 21:38	19	9310	000002f4		ANSWERED	5s
			SIP/19-	SIP/9310-		
9/12/2022 21:36	19	9310	000002f2	000002f3	ANSWERED	5s
			SIP/19-	SIP/9310-		
9/12/2022 21:35	19	9310	000002f0	000002f1	ANSWERED	7s
			SIP/19-	SIP/9310-		
9/12/2022 21:35	19	9310	000002ee	000002ef	ANSWERED	7s
			SIP/9320-	SIP/19-	NO	
9/12/2022 21:35	9320	19	000002ec	000002ed	ANSWER	0s
			SIP/9320-	SIP/19-		
9/12/2022 21:34	9320	19	000002ea	000002eb	ANSWERED	4s
			SIP/9320-	SIP/19-	NO	
9/12/2022 21:34	9320	19	000002e8	000002e9	ANSWER	0s
			SIP/9320-	SIP/19-		
9/12/2022 21:34	9320	19	000002e6	000002e7	FAILED	0s
			SIP/9310-	SIP/9320-	NO	
9/12/2022 21:34	9310	9320	000002e4	000002e5	ANSWER	0s
			SIP/19-	SIP/9320-	NO	
9/12/2022 21:33	19	9320	000002e2	000002e3	ANSWER	0s
			SIP/9310-	SIP/9320-	NO	
9/12/2022 21:33	9310	9320	000002e0	000002e1	ANSWER	0s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
9/12/2022 21:33	19	9310	SIP/19- 000002de	SIP/9310- 000002df	NO ANSWER	0s
9/12/2022 21:32	9320	9310	SIP/9320- 000002dc	SIP/9310- 000002dd	ANSWERED	4s
9/12/2022 21:32	9310	9320	SIP/9310- 000002da	SIP/9320- 000002db	FAILED	0s
9/12/2022 21:31	9310	9320	SIP/9310- 000002d8	SIP/9320- 000002d9	ANSWERED	19s
9/12/2022 21:31	9320	9310	SIP/9320- 000002d6	SIP/9310- 000002d7	ANSWERED	5s
9/12/2022 21:30	9320	9310	SIP/9320- 000002d4	SIP/9310- 000002d5	ANSWERED	5s
9/12/2022 21:29	9310	9320	SIP/9310- 000002d2	SIP/9320- 000002d3	NO ANSWER	0s
9/12/2022 21:28	9310	9320	SIP/9310- 000002d0	SIP/9320- 000002d1	ANSWERED	12s
9/12/2022 21:27	9310	9320	SIP/9310- 000002ce	SIP/9320- 000002cf	NO ANSWER	0s
9/12/2022 21:24	9310	19	SIP/9310- 000002cc	SIP/19- 000002cd	NO ANSWER	0s
9/12/2022 21:23	19	9310	SIP/19- 000002ca	SIP/9310- 000002cb	ANSWERED	8s
9/12/2022 21:23	9310	19	SIP/9310- 000002c8	SIP/19- 000002c9	NO ANSWER	0s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
9/12/2022 21:22	19	9310	SIP/19- 000002c6	SIP/9310- 000002c7	ANSWERED	32s
9/12/2022 21:21	19	9310	SIP/19- 000002c4	SIP/9310- 000002c5	ANSWERED	32s
9/12/2022 21:20	19	9310	SIP/19- 000002c2	SIP/9310- 000002c3	ANSWERED	32s
9/12/2022 21:19	9310	19	SIP/9310- 000002c0	SIP/19- 000002c1	FAILED	0s
9/12/2022 21:18	9310	19	SIP/9310- 000002be	SIP/19- 000002bf	NO ANSWER	0s
3/12/2022 12:36	9310	93	SIP/9310- 000002bc	SIP/93- 000002bd	ANSWERED	32s
3/12/2022 12:36	9310	93	SIP/9310- 000002ba	SIP/93- 000002bb	ANSWERED	10s
3/12/2022 12:34	9310	93	SIP/9310- 000002b8	SIP/93- 000002b9	ANSWERED	32s
3/12/2022 12:34	9310	93	SIP/9310- 000002b6	SIP/93- 000002b7	NO ANSWER	0s
3/12/2022 12:34	9310	93	SIP/9310- 000002b4	SIP/93- 000002b5	BUSY	0s
3/12/2022 12:33	9310	93	SIP/9310- 000002b2	SIP/93- 000002b3	ANSWERED	26s
3/12/2022 12:32	93	9310	SIP/93- 000002b0	SIP/9310- 000002b1	ANSWERED	32s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
3/12/2022 12:24	93	94	SIP/93- 000002ae	SIP/94- 000002af	ANSWERED	15s
3/12/2022 12:22	93	94	SIP/93- 000002aa	SIP/94- 000002ab	ANSWERED	26s
3/12/2022 12:21	94	93	SIP/94- 000002a8	SIP/93- 000002a9	ANSWERED	32s
3/12/2022 12:04	40	93	SIP/40- 000002a6	SIP/93- 000002a7	ANSWERED	26s
3/12/2022 11:54	23	73	SIP/23- 000002a3	SIP/73- 000002a4	ANSWERED	32s
3/12/2022 11:54	23	73	SIP/23- 000002a1	SIP/73- 000002a2	ANSWERED	17s
3/12/2022 11:37	93	69	SIP/93- 0000029e	SIP/69- 0000029f	BUSY	0s
3/12/2022 11:37	93	69	SIP/93- 0000029c	SIP/69- 0000029d	BUSY	0s
3/12/2022 11:37	93	69	SIP/93- 0000029a	SIP/69- 0000029b	BUSY	0s
3/12/2022 11:35	93	69	SIP/93- 00000298	SIP/69- 00000299	ANSWERED	32s
3/12/2022 11:10	71	140	SIP/71- 0000028d	SIP/140- 0000028e	BUSY	0s
3/12/2022 11:10	71	140	SIP/71- 0000028f	SIP/140- 00000290	BUSY	0s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
3/12/2022 11:10	71	140	SIP/71- 0000028b	SIP/140- 0000028c	BUSY	0s
3/12/2022 11:10	71	140	SIP/71- 00000289	SIP/140- 0000028a	BUSY	0s
3/12/2022 11:10	71	140	SIP/71- 00000287	SIP/140- 00000288	ANSWERED	32s
3/12/2022 11:10	140	71	SIP/140- 00000285	SIP/71- 00000286	BUSY	0s
3/12/2022 11:10	140	71	SIP/140- 00000283	SIP/71- 00000284	BUSY	0s
3/12/2022 11:09	71	140	SIP/71- 00000281	SIP/140- 00000282	NO ANSWER	0s
3/12/2022 11:08	140	71	SIP/140- 0000027f	SIP/71- 00000280	ANSWERED	17s
3/12/2022 11:06	140	71	SIP/140- 0000027d	SIP/71- 0000027e	ANSWERED	32s
2/12/2022 11:16	9320	9330	SIP/9320- 00000271	SIP/9330- 00000272	ANSWERED	13s
2/12/2022 11:02	9320	9330	SIP/9320- 0000026f	SIP/9330- 00000270	ANSWERED	32s
2/12/2022 12:54	9320	9330	SIP/9320- 0000026d	SIP/9330- 0000026e	ANSWERED	23s
2/12/2022 12:51	9320	9330	SIP/9320- 0000026b	SIP/9330- 0000026c	ANSWERED	26s

Date	Source	Destination	Src. Channel	Dst. Channel	Status	Duration
			SIP/9320-	SIP/9330-		
2/12/2022 12:43	9320	9330	00000269	0000026a	ANSWERED	20s
			SIP/9320-	SIP/9330-		
2/12/2022 12:39	9320	9330	00000267	00000268	ANSWERED	23s

Nota. Esta tabla muestra los registros de las llamadas y videollamadas realizadas en el intervalo que duraron las pruebas piloto de la aplicación (Issabel PBX, 2022).

Análisis de paquetes de la central telefónica

Para comprobar internamente el funcionamiento de las llamadas y videollamadas se hizo una captura de paquetes de datos en las llamadas como parte del cumplimiento del objetivo “Realizar el análisis de la calidad de servicio que ofrece el servidor y el softphone implementado utilizando métodos objetivos y subjetivos MOS y PSQM PESQ respectivamente”

Para el análisis se utilizó la aplicación Packet Capture en donde se logró capturar paquetes mediante el SPD (Session Description Protocol), el cual establece los parámetros para el intercambio de streaming siendo más amplia su aplicación dentro del protocolo SIP. En la Figura 85 se puede apreciar una captura de un paquete en el momento que se hizo una videollamada donde se tiene información como la extensión a la cual se está llamando, la dirección IP de la central telefónica. Las opciones que soporta este protocolo son: INVITE, ACK, CANCEL, BYE, NOTIFY, MESSAGE, INFO, seguido es esto, se pueden observar los códecs utilizados para audio y video. A través del protocolo RTP hay varios códecs numerados en la línea: `m=audio 17660 RTP/AVP 0 8 100` donde:

- El número 0 está definido como PCMU a 8000 Hz también llamado G711u.
- El número 8 está definido como PCMA a 8000 Hz también llamado G711a

- El número 100 es el telephone-event a 8000 Hz que es el sistema de marcación por tonos.

Los códecs para video definidos en la línea m=video 11314 RTP/AVP 96 97 donde:

- El número 96 definido como VP8 a 90000 Hz
- El número 97 definido como H264 a 90000 Hz

Figura 85

Captura de videollamada con Packet Capture

```

#32 ---> 12-03 12:35:01

SIP/2.0 200 OK
Via: SIP/2.0/UDP
10.0.8.1:40992;branch=z9hG4bK.ehnd31U6;received=157.100.142.252;rport=35906
From: "9310" <sip:9310@softaccess.cf>;tag=KAUNJQ81i
To: "93" <sip:93@softaccess.cf>;tag=as0c98d91d
Call-ID: cQgiAQHs5G
CSeq: 21 INVITE
Server: IPBX-2.11.0(11.25.3)
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
Supported: replaces, timer
Contact: <sip:93@147.182.186.72:5060>
Content-Type: application/sdp
Content-Length: 419

v=0
o=root 1495863957 1495863957 IN IP4 147.182.186.72
s=Asterisk PBX 11.25.3
c=IN IP4 147.182.186.72
b=CT:384
t=0 0
m=audio 17660 RTP/AVP 0 8 100
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:100 telephone-event/8000
a=fmtp:100 0-16
aptime:20
a=sendrecv
m=video 11314 RTP/AVP 96 97
a=rtpmap:96 VP8/90000
a=rtcp-fb:96 ccm fir
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801F
a=sendrecv

```

Nota. Esta figura muestra la información capturada por la aplicación Packet Capture de una videollamada.

Análisis de los servicios ofrecidos por el softphone implementado utilizando métodos objetivos y subjetivos MOS, PSQM y PESQ

La app Grabador de llamadas fue instalada en dispositivos móviles con el propósito de grabar las conversaciones de las llamadas que los condóminos realizaron mediante la app SoftAccess, para luego poder analizar con el programa Spek mediante métodos objetivos y subjetivos.

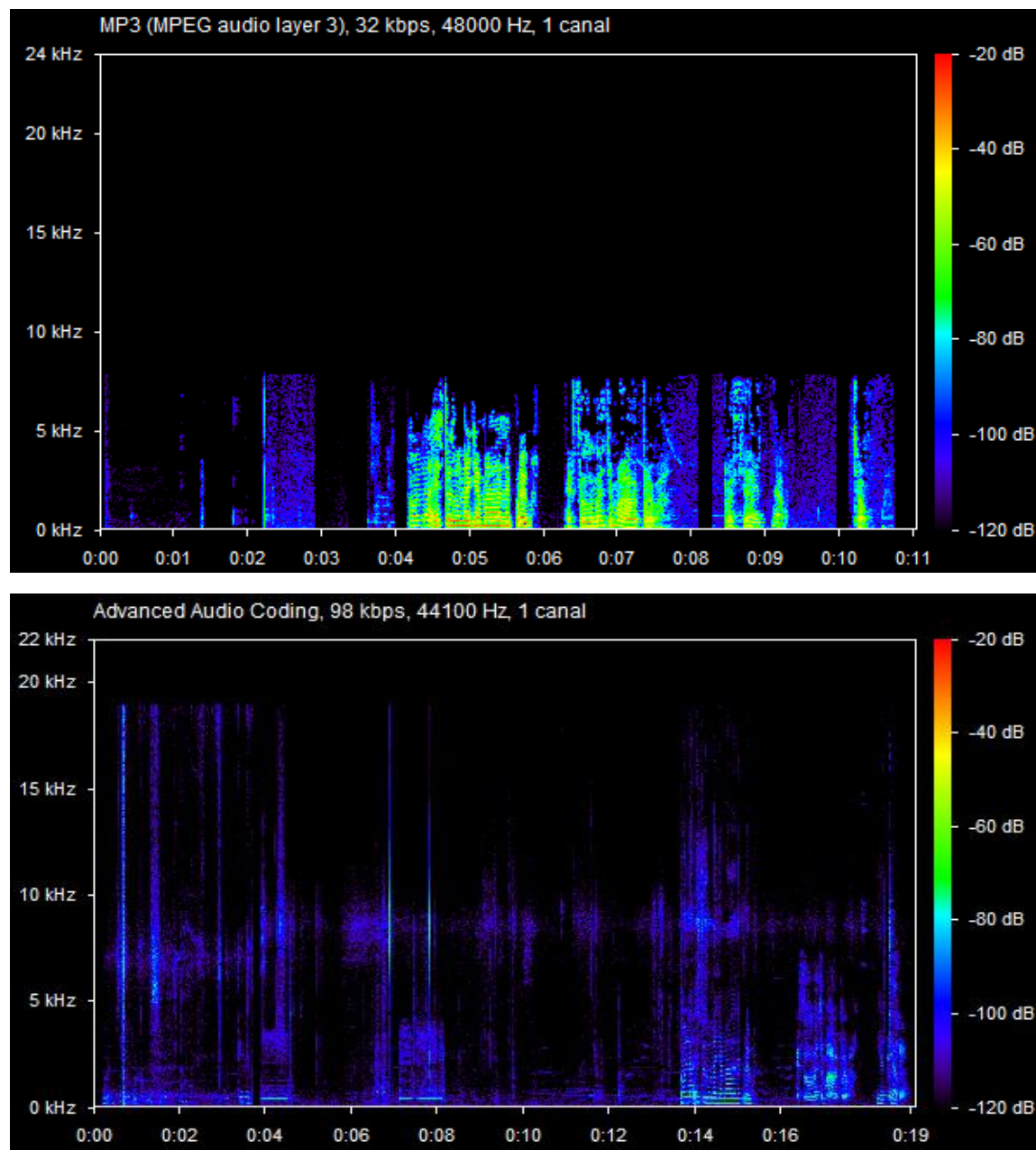
En las Figuras 86, 87 y 88 se indica el contenido espectral tanto del emisor como del receptor de la llamada experimentada entre condóminos y personal de seguridad.

Por medio del programa *Spek* se pudo observar el contenido espectral de cada llamada, en el eje x se encuentra determinado el tiempo de duración, en el lateral izquierdo están las frecuencias (KHz) y en el derecho la presión sonora (dB). En la parte superior de la gráfica se puede observar la característica técnica del archivo (llamada), sin embargo, el códec que aparece en cada gráfica es el usado en el grabador de llamadas.

De acuerdo con la UIT-T (2018), al referirse a PSQM (objetivo) y PESQ (subjetivo) se habla del método de la medida de la calidad vocal por percepción y de la evaluación de la calidad vocal por percepción respectivamente. El método PESQ mide efectos de ruido y distorsiones de voz.

Figura 86

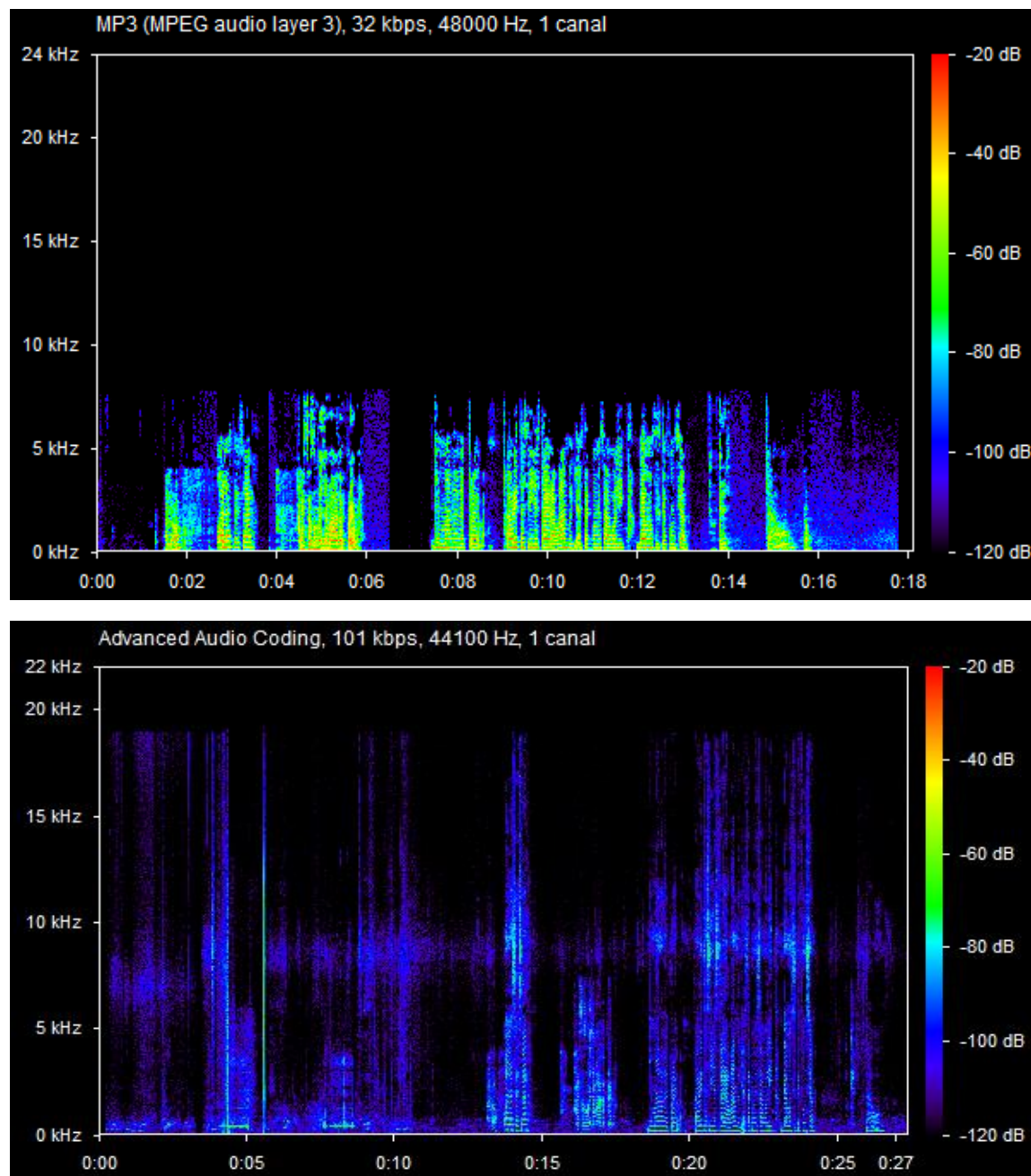
Contenido espectral de la llamada 1 mediante la app



Nota. En estas figuras se puede analizar el contenido espectral dentro de una llamada entre emisor y receptor, la llamada que dura 19 segundos corresponde al emisor puesto que se toma en cuenta desde la marcación. En la parte derecha de las figuras se indica el eje Y en dB que muestra los valores de la presión sonora.

Figura 87

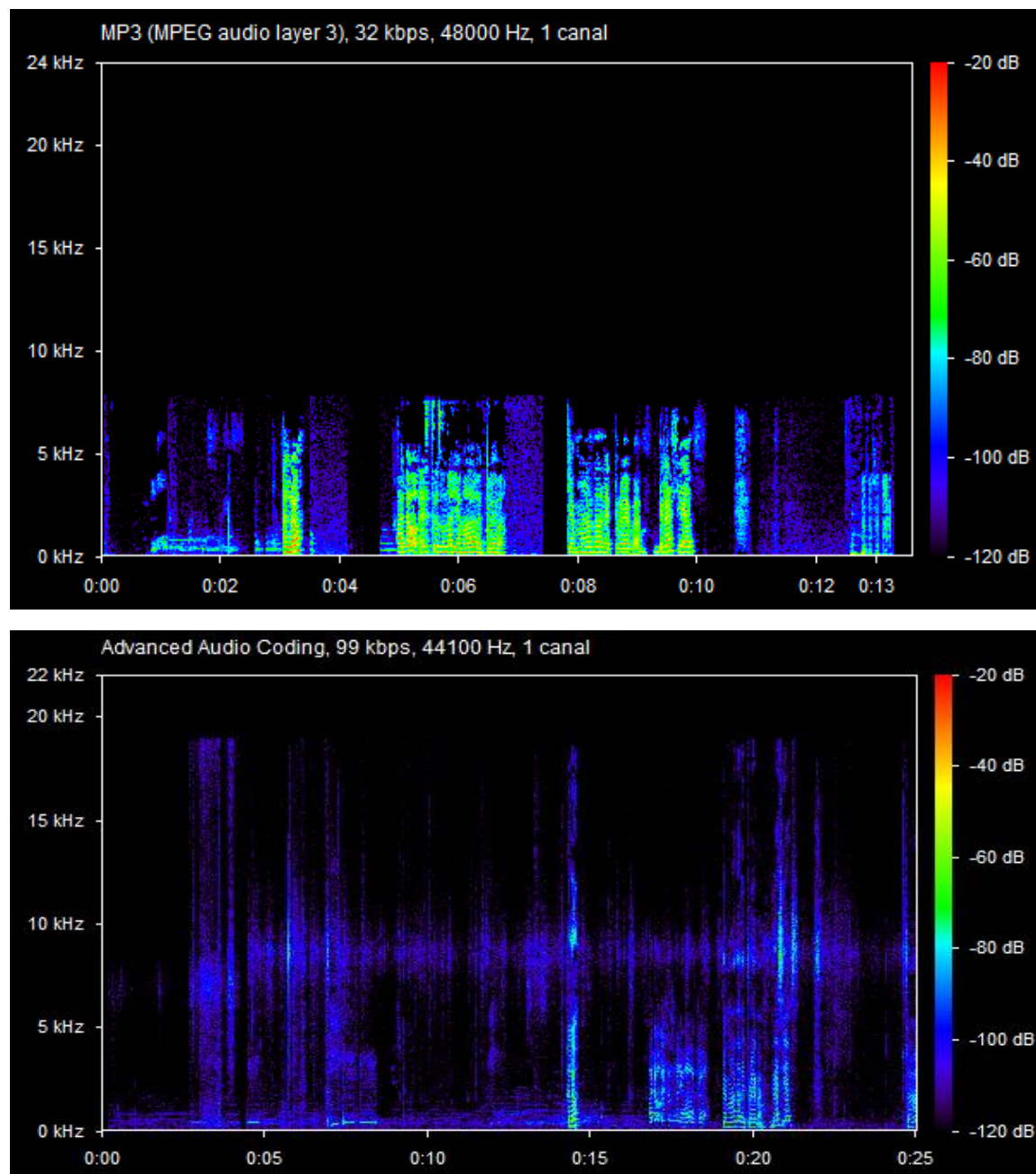
Contenido espectral de la llamada 2 mediante la app



Nota. En estas figuras se puede analizar el contenido espectral dentro de una llamada entre emisor y receptor, la llamada que dura 27 segundos corresponde al emisor puesto que se toma en cuenta desde la marcación.

Figura 88

Contenido espectral de la llamada 3 mediante la app



Nota. En estas figuras se puede analizar el contenido espectral dentro de una llamada entre emisor y receptor, la llamada que dura 25 segundos corresponde al emisor puesto que se toma en cuenta desde la marcación.

La unidad para expresar el nivel de presión sonora es el decibel (dB), se considera de suma importancia saber que el signo negativo indica la atenuación del sonido, mientras que el positivo la ganancia. Con el fin de poder interpretar las Figuras 86 – 88 se debe conocer lo siguiente:

- En el eje Y izquierdo se tiene la frecuencia (Hz), existen 3 zonas: grave, medias y agudas (de abajo hacia arriba respectivamente).
- En el eje Y derecho se tiene la presión sonora (dB), los colores verde, amarillo y rojo representan una gran cantidad de información, mientras que los colores negro y azul representan falta de información. Es decir, los colores varían de acuerdo a los dB.

En la figura 86, la primera (receptor) posee algunos espacios de color negro y azul lo cual significa pérdida de información, sin embargo, en su mayoría se pueden apreciar los colores azul, verde y amarillo lo cual representa la cantidad de información transmitida, haciendo posible una comunicación emisor y receptor. Esta comunicación se encuentra dentro del rango auditivo humano al estar entre 8 KHz y 19 KHz aproximadamente. En la segunda figura se observa que la frecuencia es mayor a la primera, pese a tener zonas de color negro.

Este análisis se realizó para las Figuras 86, 87 y 88 las cuales guardan cierta relación tanto en la frecuencia como la presión sonora. De esta manera se puede decir que en general las figuras que presentaron espacios de color negro o azul, su audio no fue nítido y hubo algún tipo de dificultad en la conversación. Tomando de referencia a la Tabla 4 el nivel MOS de estas llamadas estaría entre 3 y 4 debido a los factores ya mencionados. Entonces, desde el método PESQ y de acuerdo a la Figura 74, las llamadas se encuentran dentro del rango de la frecuencia de la voz: 300 – 3400 Hz, es decir, la calidad percibida corresponde a un valor de 5.

Cabe recalcar que el rango auditivo humano oscila entre 20 Hz – 20 KHz, entonces, se puede decir que en su mayoría las llamadas están dentro de dicho rango, razón por la cual se pudo establecer una comunicación coherente entre el emisor y receptor. Esto puede ser comprobado visualizando el lateral izquierdo de las Figuras 85, 86 y 87 en donde se encuentran los valores de la frecuencia. Adicional a esto, se puede acotar que el contenido espectral se encuentra entre la zona grave y media permitiendo escuchar de forma clara el audio generado.

Las figuras anteriormente mencionadas indican el contenido espectral en las llamadas y de acuerdo a los ejes Y se pudo determinar si están dentro del rango auditivo y frecuencia de la voz humana, lo cual es verdadero. A manera general los colores que se visualizan determinan que la información transmitida en ciertos momentos se puede perder o no, sin embargo, esto no hace que la comunicación deje de ser coherente. Los valores en dB (-50 dB a -110 dB) señalan los colores azul, celeste, verde, amarillo y naranja.

Análisis de costos

A través de las pruebas piloto se pudo tener una idea del costo inicial que se requirió con el objetivo de desarrollar las dos etapas de la implementación de este proyecto de tesis en el Conjunto Habitacional Benalcázar Oriental. En la Tabla 8 se detalla la descripción de los elementos utilizados, la cantidad, costo unitario y costo total.

Tabla 8

Costos del proyecto de tesis

Costos de implementación	Descripción	Cantidad	Costo unitario	Costo total

	Digital Ocean	4 meses	\$8	\$32
	Firebase	4 meses	\$5	\$20
	Subtotal 1			\$52
Costos de	Stickers de	60	\$1	\$10
instalación	códigos QR			
	Subtotal 2			\$10
Total				\$62

Nota. Esta tabla detalla los costos de implementación, instalación y de la aplicación que fueron solventados con el propósito de obtener resultados y datos finales verídicos para comprobar la funcionalidad de la aplicación.

Además de enlistar los costos se considera oportuno detallar los materiales usados por medio de la Tabla 9.

Tabla 9

Materiales usados para el desarrollo del proyecto de titulación

Materiales	Cantidad
Laptop	2
Dispositivos móviles Android	38

Cámara	1
Grabadora	1

Nota. En esta tabla se pueden observar los materiales que fueron necesarios para poder desarrollar y luego implementar de forma global la aplicación.

Conclusiones

En lo concerniente a la implementación del servidor de Telefonía IP y un softphone basado en librerías Speech to Text y generación de códigos QR para el control de acceso en el Conjunto Benalcázar Oriental, se pudo verificar que por medio de la app SoftAccess, los condóminos y guardias pudieron cambiar el esquema de seguridad en el transcurso de las pruebas piloto realizadas. La generación de códigos QR permanentes y temporales fue un servicio adicional de los que conlleva un softphone.

Mediante el periodo de pruebas se verificó el funcionamiento de manera general de la app y la lectura de los códigos QR, teniendo el 50% de la población en total a personas que pudieron manejar la app de forma satisfactoria, mientras que el otro 50% presentó ciertas dificultades que con explicaciones personalizadas fueron solventadas. El control de acceso tanto para moradores como para visitante fue supervisado por el personal de guardia ayudándose de los servicios ofrecidos por la app como son la lectura y escaneo de códigos QR y librerías speech to text.

En el servidor de telefonía IP implementado, la interacción entre la muestra y la app permitió probar las llamadas y videollamadas mismas que fueron registradas gracias a los Reportes CDR de Issabel PBX en donde se verificó la cantidad de llamadas totales realizadas, se puede afirmar que más del 60% fueron concretadas con una duración de 20 segundos en promedio, 22% no fueron respondidas, 13% en estado de ocupado y un 5% fueron llamadas

con error de conexión, lo que indica que en su mayoría todas las llamadas que se intentaron pudieron ser procesadas por la central telefónica cumpliendo de esta manera funcionamiento en un 95%. Para probar el servicio de mensajería se evidenció el envío y recepción de mensajes grupales y personales entre las personas que instalaron la app, la generación de un tono de alerta fue imprescindible para avisar la llegada de alguna notificación.

A través del IDE Android Studio se diseñó el softphone SoftAccess para garantizar el control de acceso con la ayuda de la plataforma Firebase Database en dónde se registraron todas las lecturas que el personal de seguridad realizó desde la aplicación que fueron 38 lecturas de códigos QR temporales y 42 de permanentes, no obstante, al principio se registraron errores debido a la calidad del material de los códigos QR pegados en el parabrisas de los automóviles, este resultó ser muy transparente, puesto que, 10 lecturas de códigos QR no fueron detectadas, por ese motivo se cambió el material a un fondo blanco, a través de este cambio el proceso de escaneo y almacenamiento de datos fue exitoso. Con relación al personal de seguridad se logró tener un registro exacto en cuanto a ingresos y salidas de vehículos con códigos QR permanentes, de esta forma el almacenamiento de información fue alojado en la base de datos de Firebase. Para el caso de los visitantes, la herramienta de speech to text resultó exitosa logrando implementar la doble verificación para brindar más seguridad y fiabilidad en el caso de que un visitante externo al conjunto requiera ingresar.

Con la finalidad de realizar mediciones sobre el sistema implementado, el establecimiento de llamadas y videollamadas con los condóminos del Conjunto Habitacional Benalcázar Oriental permitió constatar y analizar objetivamente la calidad de las llamadas se capturaron parámetros como latencia, jitter, paquetes perdidos y MOS. Gracias a la plataforma VoIP Spear se pudieron obtener estos datos directamente desde la central telefónica, los cuales están representados en la Tabla 6 y se destaca que la latencia no supera los 70 ms y el jitter es menor a un ms. En este sentido, se concluye que los valores están dentro del rango

establecido por la ITU-T en donde la latencia máxima es de 300 ms y el jitter menor a 50 ms como parámetros límite para que la calidad de una llamada sea entendible.

Las capacitaciones que se brindaron a los guardias se hicieron de manera personalizada, teniendo en cuenta, que el papel a desarrollar es fundamental en el control de acceso, así como también de mucha responsabilidad. De otro modo, los moradores captaron de diferente manera, algunos lo hicieron de forma intuitiva, otros solicitaron ayuda en el proceso, mientras que otras personas necesitaron una segunda explicación. En consecuencia, las personas participantes en las pruebas piloto lograron manejar la aplicación de forma correcta, es por esto que se recomienda motivar a los demás condóminos a experimentar un nuevo sistema de control de acceso mediante sus dispositivos móviles personales.

Las respuestas a las preguntas planteadas en el formulario de Google sirvieron para que los condóminos y guardias expresen su opinión una vez interactuado con la app, tomando de referencia la escala MOS las respuestas en promedio se dieron entre los niveles 4 y 5 señalando de modo general a la app en un estado de “Excelente” y “Buena”, sin embargo, se tuvieron preguntas en donde su valor fue de 3, es decir “Aceptable”, esto se debe a la conectividad y al proceso de instalación.

Finalmente, se aplicaron métodos objetivos (PSQM) y subjetivos (PESQ) para evaluar y medir la calidad vocal por percepción de las llamadas, para esto, se comparó la señal de entrada y de salida desde el emisor y receptor y viceversa. Tomando de referencia el método PESQ, la percepción de los usuarios finales permitió conocer que la frecuencia de la voz estuvo dentro de su rango (300 – 3400 Hz) lo cual sirvió para que la comunicación entre emisor y receptor sea entendible y el mensaje a transmitir/informar sea claro. Al hablar de PSQM los valores de las frecuencias en las llamadas de manera general oscilaron entre 8 KHz y 19 KHz estando dentro del rango del espectro audible, de igual forma, tanto emisor y receptor pudieron escuchar la conversación sin tener un ruido o distorsión que afecte su comunicación. En cuanto

a la presión sonora se puede concluir que los valores en dB oscilan entre -50 dB a -110 dB los cuales están reflejados por medio de los colores azul, celeste, verde, amarillo y amarillo, de esta manera se puede saber la cantidad de información que se tiene en la línea del tiempo.

Recomendaciones

Se recomienda tener en cuenta que la programación de una aplicación va de la mano con la compatibilidad de las librerías con las que se programa, puesto que, en este proyecto se tuvo que cambiar varias veces las librerías. Uno de los motivos fue la actualización del sistema operativo Android que al cambiar las funcionalidades las librerías quedaron obsoletas, es por esta razón, que se recomienda que sean avaladas por Google o que tengan actualizaciones constantes.

Es recomendable tener una retroalimentación de varias personas en el desarrollo de una aplicación como lo fue SoftAccess, pues en el diseño se involucran algunas funcionalidades que deben ser fáciles de usar con el objetivo de que el usuario no tenga problemas al manipular dicha aplicación. Por otro lado, es importante la estética de la interfaz de usuario, es decir, el color de fondo, tamaño de letra, botones, tiempo de respuesta, inicio de sesión, entre otros factores que la hacen más novedosa y atractiva.

Es recomendable observar el comportamiento de las personas en el periodo de pruebas para mejorar la utilidad y los servicios de la aplicación, pese a esto, no se puede dejar de lado el suceso de algún incidente como, por ejemplo, en este proyecto de tesis fue fundamental mejorar la calidad del papel de los stickers.

Se recomienda disponer de una buena conexión a internet o datos móviles para que la calidad de la comunicación sea óptima y evitar distorsión, ruido o en el peor de los casos una conexión fallida. Existen muchos métodos con los cuales se puede evaluar la calidad de la

llamada como se mencionó en la sección 4. El modelo de los dispositivos móviles Android influyen en la instalación y ejecución de la app, en consecuencia, en el funcionamiento global.

Es recomendable realizar algunas pruebas para evaluar la satisfacción que siente el usuario al utilizar la app a fin de mejorar o incrementar servicios, ya sea para personalizar de acuerdo a sus necesidades o crear nuevas aplicaciones o proyectos.

Trabajos futuros

Una vez culminada tanto la etapa de investigación, desarrollo e implementación se pudo constatar que a partir de este trabajo de titulación se pueden realizar proyectos futuros, los cuales tienen el objetivo de satisfacer de mejor manera las necesidades de los condóminos y a su vez automatizar su lugar de residencia ofreciendo un control de acceso mediante elementos, dispositivos, sensores electrónicos, etc.

A continuación, se detallan algunas ideas que podrían ser usadas en el ámbito de investigación o implementación.

- Colocar una máquina a manera de escáner que se encargue de la lectura y el escaneo de los códigos QR instalados, haciendo posible, que de manera autónoma realice este proceso sin depender del personal de guardia.
- Desarrollar este proyecto de titulación para el sistema operativo de iOS debido a que implica utilizar librerías, comandos y lenguaje de programación que estén acorde a este sistema operativo.
- Implementar un dispositivo de control facial con la finalidad de comparar la identidad de las personas a través de datos biométricos, puesto que, dicha identificación utiliza patrones matemáticos únicos de cada persona convirtiéndose en un tipo de reconocimiento muy seguro y fiable, por otro lado, mediante una cámara se puede realizar la lectura de las placas de los automóviles para brindar mayor seguridad.

- Realizar un sistema automático de control de acceso que trabaje en conjunto con cámaras de seguridad y domótica para el registro de entrada - salida de vehículos, apertura - cierre de puerta, respectivamente, para prescindir del servicio de guardianía.

Bibliografía

- Aguilar, C. (2015). *Análisis, Diseño e Implementación de un Sistema de VoIP para el Hospital Un Canto A La Vida [Tesis de Ingeniería, Universidad Politécnica Salesiana Sede Quito]*. Repositorio Institucional. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/11608/1/UPS%20-%20ST002110.pdf>
- Aguilar, I., & Chulde, A. (2020). *Desarrollo de software para la verificación y autenticación de personas a través del rasgo biométrico del Iris aplicado a una base de datos adquirida para el control de accesos de áreas industriales [Tesis de Ingeniería, Universidad de las Fuerzas Armadas]*. Repositorio Institucional. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/22645/1/T-ESPE-043959.pdf>
- Alex Guamán, K. R. (2022). *Análisis, implementación y evaluación de un asistente basado en telefonía IP para gestión de archivos e información de documentación para las Carreras del Departamento de Eléctrica, [Tesis de Ingeniería, Universidad de las Fuerzas Armadas - ESPE]*. Repositorio Institucional. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/28525/1/T-ESPE-050934.pdf>
- Ancona, J. (2015). *Desarrollo de una aplicación nativa de enseñanza para iOS con swift [Tesis de Maestría, Universidad de Montemorelos]*. Repositorio Institucional. Obtenido de <https://dspace.um.edu.mx/bitstream/handle/20.500.11972/540/Tesis%20Jos%C3%A9%20Adair%20Ancona%20Reyes.pdf?sequence=1&isAllowed=y>
- Andrade, A., & Manosalvas, S. (2014). *Desarrollo e implementación de una aplicación multiplataforma para dispositivos móviles que permita visualizar información de la Escuela Politécnica Nacional [Tesis de Ingeniería, Escuela Politécnica Nacional]*. Repositorio Institucional. Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/12032/1/CD-6618.pdf>

- Ávila, S. (2016). *Análisis de tecnologías de escaneo y reconocimiento de códigos QR y su aplicabilidad en teléfonos móviles smartphone mediante una cámara fotográfica [Tesis de Ingeniería, Universidad Tecnológica Equinoccial]*. Repositorio Institucional. Obtenido de http://repositorio.ute.edu.ec/bitstream/123456789/16855/1/67354_1.pdf
- Aviles, A., & Cobeña, K. (2015). *Diseño e Implementación de un Sistema de Seguridad a Través de Cámaras, Sensores y Alarma, Monitorizado y Controlado Teleméricamente para El Centro de Acogida "Patio Mi Pana" Perteneiente a la Fundación Proyecto Salesiano [Tesis de Ingeniería, Universidad Politécnica Salesiana]*. Repositorio Institucional. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/10401/1/UPS-GT001444.pdf>
- Ayala, A., & Lomas, J. (2010). *Análisis del desempeño de los codec's de telefonía para voz sobre IP (VoIP) [Tesis de Ingeniería, Escuela Politécnica del Ejército]*. Repositorio Institucional. Obtenido de <file:///D:/HP/Downloads/T-ESPE-030156.pdf>
- Berruecos, Maldonado, & Torres. (2015). *Implementación de un Sistema Telefónico Basado en Telefonía IP [Tesis de Ingeniería, Instituto Politécnico Nacional]*. Repositorio Institucional, México. Obtenido de https://tesis.ipn.mx/jspui/bitstream/123456789/17340/1/Implentaci%C3%B3n_de_un_sistema_telef%C3%B3nico_basado_en_telefonia_IP.pdf
- Caldera, J., & Suazo, W. (2011). *Módulo III: Telefonía IP [Universidad Nacional de Ingeniería]*. Repositorio Institucional, Nicaragua. Obtenido de <https://ribuni.uni.edu.ni/1261/1/25717-MIIITIP.pdf>
- Castorena, L. (2012). *Análisis y adaptación de codecs para la transmisión de audio cardíaco sobre redes inalámbricas con tasas de datos bajas/medias [Tesis de Maestría, Centro*

- de Investigación Científica y de Educación Superior Avanzada*]. Repositorio Institucional. Obtenido de file:///D:/HP/Downloads/188401.pdf
- Castro, R. (2019). *Implementación de una estación de pruebas entre dos centrales telefónicas basadas en Asterisk [Tesis de Ingeniería, Universidad Tecnológica Israel]*. Repositorio Institucional. Obtenido de <http://repositorio.uisrael.edu.ec/bitstream/47000/2153/1/UISRAEL-EC-ELDT-378.242-2019-059.pdf>
- Cellere, J., & Nuñez, B. (2021). *Diseño e implementación de un sistema de acceso y alarma comunitaria basado en pbx voip virtuales para áreas residenciales [Tesis de Ingeniería, Universidad de las Fuerzas Armadas]*. Repositorio Institucional. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/24965/1/T-ESPE-044567.pdf>
- Chávez, A. (2014). *Implementación de un Sistema de Comunicación con Telefonía IP en el Hotel Higuero de la ESPAM MFL [Tesis de Ingeniería, Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López]*. Repositorio Institucional. Obtenido de <https://repositorio.espam.edu.ec/bitstream/42000/49/1/CHAVEZ%20ROCA%20ADRIAN%20-%20TESIS%20DE%20GRADO.pdf>
- Choez, L., & Pérez, J. (2016). *Implementación de Telefonía IP con Elastix para un ISP con Acceso WiFi [Tesis de Ingeniería, Universidad de Guayaquil]*. Repositorio Institucional. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/18234/1/UG-FCMF-B-CINT-PTG-N.146.pdf>
- D, V. (2015). *Análisis y Diseño de Sistema para Gestión de Acceso a Urbanizaciones Privadas [Tesis de licenciatura, Universidad de Guayaquil]*. Repositorio Institucional. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/12870/1/Tesis%20Di%C3%B3genes%20Velasco.pdf>

- Enríquez, A. (2018). *Plataforma de tecnologías inteligentes para la autogestión de salud de pacientes con diabetes y asma con el apoyo de una aplicación móvil y un sistema de recomendaciones, enfocado en la optimización de la aplicación móvil en la versión 4.4.2 de Android para mejorar el diseño visual y reorganizar las pantallas de acceso [Tesis de Ingeniería, Universidad de Guayaquil]*. Repositorio Institucional. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/27635/1/B-CINT-PTG-N.274%20Enr%C3%ADquez%20Carrillo%20Alexis.pdf>
- Estrada, J., Calva, M., Rodríguez, A., & Tipantuña, C. (2016). *Seguridad de la Telefonía IP en Ecuador: Análisis en Internet [Tesis de Ingeniería, Escuela Politécnica Nacional]*. *Repositorio Institucional*. Obtenido de http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-65422016000200025
- Fiscalía General Del Estado. (2021). Obtenido de <https://www.fiscalia.gob.ec/estadisticas-de-robos/>
- Guamán, A., & Rubio, K. (2022). *Análisis, implementación y evaluación de un asistente basado en telefonía IP para gestión de archivos e información de documentación para las Carreras del Departamento de Eléctrica, Electrónica y Telecomunicaciones de la ESPE [Tesis de Ingeniería, Universidad de las Fuerzas Armadas Espe]*. Repositorio Institucional. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/28525/1/T-ESPE-050934.pdf>
- Guanga, R., & León, Y. (2012). *Análisis y evaluación de codecs de audio para transmisión de VoIP en infraestructuras de redesmpls sobre Linux [Tesis de Ingeniería, Escuela Superior Politécnica de Chimborazo]*. Repositorio Institucional. Obtenido de <http://dspace.espech.edu.ec/bitstream/123456789/2540/1/18T00516.pdf>

- Guerrero, C. (2016). *Análisis comparativo del desempeño de los diferentes CÓDECS utilizados para video conferencia en la empresa ECUAGREENPRODEX S.A de la ciudad de Guayaquil [Tesis de Maestría, Pontificia Universidad Católica del Ecuador]*. Repositorio Institucional. Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/13149/2016-TesisCarlosGuerrero%2001112016%20COMPLETO.pdf?sequence=1&isAllowed=y>
- Huallpa, J. (2017). *Análisis del tiempo de recuperación en un enlace redundante en una red con enrutamiento EIGRP y servicio de telefonía IP [Tesis de Ingeniería, Universidad Nacional del Altiplano-Puno]*. Repositorio Institucional. Obtenido de http://repositorio.unap.edu.pe/bitstream/handle/UNAP/5240/Huallpa_Vargas_Jose_Vizardo_Apaza_Quispe_Abel.pdf?sequence=1&isAllowed=y
- Issabel PBX. (2022). *Issabel PBX*. Obtenido de <https://www.issabel.org/>
- Jalca, D. (2017). *Estudio de Factibilidad para la Implementación de un Laboratorio Virtual de Telefonía IP para la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Estatal del Sur de Manabí [Tesis de Ingeniería, Universidad Estatal del Sur de Manabí]*. Repositorio Institucional. Obtenido de <http://repositorio.unesum.edu.ec/bitstream/53000/978/1/UNESUM-ECU-COMPR-16.pdf>
- Jimenez, B. (2017). *Aplicación Android para la simulación de pruebas de ingreso a la Universidad para la empresa Capacitaciones Moreano de la ciudad de Ambato [Tesis de Ingeniería, Universidad Técnica de Ambato]*. Repositorio Institucional. Obtenido de https://repositorio.uta.edu.ec/bitstream/123456789/25789/1/Tesis_t1262si.pdf
- Lavariaga, A. (2020). *Segmentación de contestadoras automáticas en marcadores para centros de llamadas con telefonía IP utilizando Asterisk [Tesis de Maestría, Centro de Tecnología Avanzada]*. Repositorio Institucional. Obtenido de

<https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/414/1/LavariegaCastellanosAndresA%20MSIM%202020.pdf>

Loarte, B. (2018). *Desarrollo de una guía metodológica para el análisis forense digital en equipos de cómputo con sistema operativo Mac OS X en el Ecuador [Tesis de Maestría, Universidad Internacional SEK]*. Repositorio Institucional. Obtenido de <https://repositorio.uisek.edu.ec/bitstream/123456789/2758/1/LOARTE-BYRON.pdf>

López, E. (2013). *Integración de Módulos de Campañas y Reportes para Sistema ContacTes [Tesis de Ingeniería, Universidad Internacional del Ecuador]*. Repositorio Institucional. Obtenido de <file:///D:/HP/Downloads/T-UIDE-061.pdf>

M, M. (2019). *Sistema de Control de Acceso a Barrios Privados [Tesis de Ingeniería, Universidad de las Fuerzas Armadas]*. Repositorio Institucional. Obtenido de <https://repositorio.uesiglo21.edu.ar/bitstream/handle/ues21/18067/MINGOLLA%20MARI A%20JOSE.pdf?sequence=1&isAllowed=y>

Montalvo, D. (2018). *Desarrollo de un sistema informático para la trazabilidad de los resultados de la materia prima (plasmas reactivos y no reactivos) en el programa de evaluación externo del desempeño para el centro de investigación para la salud en América Latina (CISEAL) [Tesis de Ingeniería, Pontificia Universidad Católica del Ecuador]*. Repositorio Institucional. Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/14629/Tesis%20Diego%20Pa%C3%BAI%20Montalvo%20P%C3%A1ez.pdf?sequence=1&isAllowed=y>

Narváez, M., Pérez, Y., & Vallecillo, L. (2019). *[Tesis de Ingeniería, Universidad Nacional Autónoma de Nicaragua UNAN-LEÓN]*. Repositorio Institucional. Obtenido de <http://riul.unanleon.edu.ni:8080/jspui/bitstream/123456789/7388/1/242488.pdf>

- Padilla, L., & Zuta, R. (2020). *Implementación de la infraestructura de red VoIP, en la Misión Nor Oriental de Tarapoto de la Iglesia Adventista del 7° día Unión Peruana del Norte, 2019 [Tesis de Ingeniería, Universidad Peruana Unión]*. Repositorio Institucional. Obtenido de https://repositorio.upeu.edu.pe/bitstream/handle/20.500.12840/4590/Luis_Tesis_Licenciatura_2020.pdf?sequence=1&isAllowed=y
- Palmieri, E. (2015). *Comparación entre sistemas Virtualizados y Paravirtualizados [Tesis de Licenciatura, Universidad del Aconcagua]*. Repositorio Institucional. Obtenido de http://bibliotecadigital.uda.edu.ar/objetos_digitales/520/tesis-5063-comparacion.pdf
- Peláez, D. (2016). *Prototipo de una central telefónica IP basado en hardware y software libre, [Tesis de Ingeniería, Escuela Politécnica Nacional]*. Repositorio Institucional. Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/16625/1/CD-7254.pdf>
- Pérez, N., & Torres, B. (2020). *Desarrollo de un prototipo para la geolocalización y monitorización de frecuencia cardíaca de mascotas en la ciudad de Quito [Tesis de Ingeniería, Universidad Politécnica Salesiana Sede Quito]*. Repositorio Institucional. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/19072/1/UPS%20-%20TTS072.pdf>
- Ponce, D. (2016). *Análisis comparativo de los entornos de desarrollo integrado (IDE): Eclipse, Netbeans y JDeveloper para el desarrollo de aplicaciones Java Enterprise Edition [Tesis de Ingeniería, Universidad de Guayaquil]*. Obtenido de Repositorio Institucional: http://repositorio.ug.edu.ec/bitstream/redug/15862/1/B_CISC_PTG_1185.Ponce%20Briones%20Darwin%20Kleber.pdf
- Quezada, N. (2020). *Desarrollo de una aplicación móvil, para el reconocimiento de Covid 19 en imágenes de radiografía de tórax [Tesis de Ingeniería, Universidad Técnica de*

Machala]. Obtenido de Repositorio Institucional:

<http://repositorio.utmachala.edu.ec/bitstream/48000/15893/1/TTFIC-2020-IS-DE00012.pdf>

Quillay, W. (2018). *Diseño e implementación de un sitio web para el control de mantenimiento de equipos tecnológicos de la unidad de soporte y mantenimiento de la Espoch desarrollado en Java con base de datos Mysql en el periodo 2018 [Tesis de Tecnología, Instituto Tecnológico Superior Particular "San Gabriel"]*. Repositorio Institucional. Obtenido de <https://www.sangabrielriobamba.edu.ec/tesis/sistemas/tesis014.pdf>

Quintana, S. (2016). *PBX Inalámbrica para la Prestación de Servicios Utilizando La Red Wi-Fi y Telefonía IP [Tesis de Ingeniería, Universidad de San Carlos de Guatemala]*. Repositorio Institucional. Obtenido de <https://core.ac.uk/download/pdf/45359269.pdf>

Ramírez, J. (2021). *Análisis y selección de una PBX IP Open-Source para mejorar el sistema de comunicación en una organización [Tesis de Ingeniería, Universidad de las Fuerzas Armadas ESPE]*. Repositorio Institucional. Obtenido de <https://repositorio.espe.edu.ec/bitstream/21000/25919/1/T-ESPESD-003168.pdf>

Sani, A. (2020). *Desarrollo de una aplicación móvil para gestionar pedidos de artículos en óptica Israel del Cantón Durán [Tesis de Licenciatura, Universidad de Guayaquil]*. Repositorio Institucional. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/52771/1/TESIS%20-%20SANI%20MORA%20ANGIE%20KATHERINE%20%281%29.pdf>

Santander, D., Rosero, K., & Libreros, M. (2020). *Reconocimiento de voz para un sistema de interacción humano máquina orientado a personas con limitaciones motrices [Tesis de Ingeniería, Fundación Universitaria Católica Lumen Gentium]*. Repositorio Institucional. Obtenido de

https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/2102/RECONOCIMIENTO_VOZ_PARA_SISTEMA_INTERACCI%C3%93N_HUMANO_M%C3%81QUINA_ORIENTADO_PERSONAS_LIMITACIONES_MOTRICES.pdf?sequence=1&isAllowed=y

Santos, S. (2020). *Propuesta de la telefonía IP segmentada para optimizar el rendimiento de red de comunicación (voz-dato) en la dirección de salud Lima Centro-2017 [Tesis Doctoral, Universidad Nacional Federico Villarreal]*. Repositorio Institucional. Obtenido de

<http://repositorio.unfv.edu.pe/bitstream/handle/UNFV/4175/SOTELO%20ANTAURO%20SANTOS%20CIRIACO%20-%20DOCTORADO.pdf?sequence=1&isAllowed=y> pag 35

Soria, H. (2013). *Propuesta para la implementación de entornos visualizados en el datacenter de la dirección de informática de la pontificia Universidad Católica del Ecuador [Tesis de Ingeniería, Pontificia Univerisdad Católica del Ecuador]*. Repositorio Institucional.

Obtenido de

<http://repositorio.puce.edu.ec/bitstream/handle/22000/12010/Tesis%20Harold%20final.pdf?sequence=1>

Toepoke Software. (Noviembre de 2022). *VoIP Spear*. Obtenido de <https://voipspear.com/>

Tomalá, D. (2020). *Sistema para el reconocimiento de notas manuscritas de las tarjetas de calificaciones de la carrera de Ingeniería Industrial de la Universidad de Guayaquil, aplicando redes neuronales artificiales y aprendizaje profundo [Tesis de*

Licenciatura, Universidad de Guayaquil]. Repositorio Institucional. Obtenido de

<http://repositorio.ug.edu.ec/bitstream/redug/51117/1/2.1.PROYECTO%20DE%20TESIS-TOMALA%20ESPIN%20DAVID%20ABRAHAM.pdf>

Torres, J. (2021). *Aplicación móvil multiplataforma para la gestión de información georeferencial y servicio técnico comunitario de plomería, aplicando geolocalización offline, en la junta*

administradora de agua potable de los barrios occidentales de Aloasí [Tesis de].

Repositorio Institucional. Obtenido de

<https://repositorio.uta.edu.ec/bitstream/123456789/32046/1/t1754si.pdf>

UIT-T. (2018). *Evaluación de la calidad vocal por percepción*. Obtenido de

<https://www.itu.int/rec/T-REC-P.862/es>

Vergara, J. (2018). *Importancia de las tecnologías de programación en el sistema gestión de procesos de depósito vehicular [Tesis de Ingeniería, Instituto Tecnológico de Colima].*

Repositorio Institucional. Obtenido de

<https://dspace.colima.tecnm.mx/bitstream/handle/123456789/1259/Joshua%20Alexander%20Vergara%20Mart%C3%ADnez%20Importancia%20de%20las%20Tecnolog%C3%ADas%20de%20Programaci%C3%B3n%20en%20el%20Sistema%20Gesti%C3%B3n%20de%20Procesos%20de%20Dep~1.pdf?sequence=1>

Villacís, F. (2011). *Estudio de factibilidad de la utilidad Asterisk en placas Alix [Tesis de*

Ingeniería, Escuela Politécnica del Ejército]. Repositorio Institucional. Obtenido de

<http://repositorio.espe.edu.ec/bitstream/21000/4515/1/T-ESPE-032660.pdf>

Villalba, J. (2013). *Implementación de control de acceso y monitorización para persona con*

discapacidad mediante un dispositivo móvil [Tesis de Ingeniería, Escuela Politécnica del

Ejército]. Repositorio Institucional. Obtenido de [file:///D:/HP/Downloads/T-ESPE-](file:///D:/HP/Downloads/T-ESPE-047043.pdf)

[047043.pdf](file:///D:/HP/Downloads/T-ESPE-047043.pdf)