

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**  
**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS**



**TESIS**

---

**APLICACIÓN MÓVIL PARA CONTROL, MONITOREO Y APOYO MÉDICO  
EN EL TRATAMIENTO DEL COVID-19**

---

**Presentado por:**

Br. RAMIRO CHOQUE VELASQUEZ

Br. CONRAD HUMBERTO PAREDES RIOS

**Para optar al título profesional de:**

INGENIERO INFORMÁTICO Y DE SISTEMAS

**Asesor:**

Dr. RONY VILLAFUERTE SERNA

CUSCO-PERÚ  
2023

## INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, Asesor del trabajo de investigación/tesis titulada: Aplicación móvil  
Para control, monitoreo y apoyo médico en el  
tratamiento del Covid-19

presentado por: Ramiro Choque Velasquez con DNI Nro.: 45996220

presentado por: Conrad Humberto Paredes Rios con DNI Nro.: 46824030

para optar el título profesional/grado académico de Ingeniero  
Informático y de Sistemas

Informo que el trabajo de investigación ha sido sometido a revisión por 3 veces, mediante el Software Antiplagio, conforme al Art. 6° del **Reglamento para Uso de Sistema Antiplagio de la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 3%.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No se considera plagio.	X
Del 11 al 30 %	Devolver al usuario para las correcciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y adjunto la primera página del reporte del Sistema Antiplagio.

Cusco, 19 de Setiembre de 2023



Firma

Post firma Dr. Rony Villafuerte Serna

Nro. de DNI 23957778

ORCID del Asesor 0000-0003-4007-522X

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema Antiplagio: OID: 27259:264871150

NOMBRE DEL TRABAJO

**controlCovidV3.pdf**

AUTOR

**Ramiro & Conrad Choque & Paredes**

RECUENTO DE PALABRAS

**21285 Words**

RECUENTO DE CARACTERES

**128139 Characters**

RECUENTO DE PÁGINAS

**158 Pages**

TAMAÑO DEL ARCHIVO

**4.5MB**

FECHA DE ENTREGA

**Sep 18, 2023 8:07 PM GMT-5**

FECHA DEL INFORME

**Sep 18, 2023 8:09 PM GMT-5****● 3% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base c

- 2% Base de datos de Internet
- Base de datos de Crossref
- 1% Base de datos de trabajos entregados
- 0% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossr

**● Excluir del Reporte de Similitud**

- Material bibliográfico
- Coincidencia baja (menos de 20 palabras)
- Material citado
- Bloques de texto excluidos manualmente



# Dedicatoria

---

*Dedicamos principalmente a Dios por habernos permitido llegar a este momento y a pesar de todas las situaciones complicadas que nos ha tocado pasar seguimos de pie y avanzando.*

*A nuestros padres quienes con mucho esfuerzo y amor nos han dado su mano y así seguir guiando siempre nuestro camino*

# Agradecimientos

---

*Agradecemos de sobremanera a nuestro asesor el Dr. Rony Villafuerte Serna por todo el apoyo, la entrega y los sabios consejos que nos supo dar a lo largo del desarrollo de la presente Tesis.*

*A nuestros primeros revisores los Mg. Edwin Carrasco Poblete y el Mg. Dennis Candia Oviedo por sus observaciones y su ayuda a proponer este tema de tesis.*

# Resumen

---

El coronavirus o SARS-COV-2 fue reportado a nivel mundial el 7 de enero del 2020 en China, Después del 31 de diciembre de 2019, la Comisión de Salud Municipal de Wuhan anunció 27 casos de neumonía de causa desconocida. En el Perú, el primer caso de COVID-19 se reportó en marzo del 2020 y por el alto riesgo de contagio que tiene por el contacto físico con las personas infectadas se ha ido expandiendo. Aunque a este punto ya han sido aclarado mucho de los temas, aún queda mucho por investigar. La enfermedad por ser de naturaleza nueva ha traído como consecuencia muchas dificultades en el abordaje del paciente infectado por el COVID-19 no solo en nuestro país, sino también en países desarrollados, sin embargo, una de los aspectos que se logra visualizar es que la mejor ayuda a la población tanto en reducir el agravamiento de los paciente y enlentecer la expansión de la pandemia es el compromiso de la propia población por cumplir las normas, consejos y reglas que impone el gobierno central y el ministerio de salud en sus primeros niveles de atención. Nuestro objetivo es contribuir con el control de los pacientes que conllevan la enfermedad de una manera moderada o leve ya que los pacientes que están en una situación grave serán derivados a los hospitales o clínica pertinentes para su atención pues se necesita de camas UCI (Unidad de cuidados intensivos) con las implementaciones medicas pertinentes y necesarias para poder mantener con vida al paciente. Por todo lo expuesto y poder afrontar este problema, se propone crear un sistema informático para poder llevar a cabo el control, monitoreo y apoyo médico en el tratamiento del Covid-19 con la finalidad de informar al personal médico indicado, todos los avances de cada uno de los pacientes que haga uso de dicho sistema. Se implementará una aplicación móvil hecha para el sistema operativo Android la cual se podrá instalar en cualquier dispositivo móvil Android desde las versiones 5.1 Jelly Bean, elaborada de la manera más simple y sencilla para que cualquier usuario pueda realizar el uso de forma fácil e intuitiva. Esta

aplicación tendrá las opciones de permitir el seguimiento por paciente de manera totalmente remota para el médico tratante además de permitir crear un histórico de la evolución del paciente en el tiempo, para permitir evaluaciones posteriores. Asimismo, buscamos que esta aplicación permita conocer los puntos con mayor cantidad de infectados por zona en la ciudad del Cusco. A su vez buscamos dar un primer paso para la implementación de sistemas similares en caso de tener otra pandemia u otra enfermedad que requiera este tipo de manejo por su alto grado de contagio al contacto físico o al estar cerca al paciente. Este sistema según esperamos dará como resultado una mayor posibilidad de atención a más personas y con lo que se espera podría contribuir a mantener a los pacientes en constante comunicación con su médico, mostrar un mapa de las zonas más afectadas en la ciudad de Cusco y tener un mejor control de pacientes infectados.

**Palabras Clave:** SARS-COV-2, COVID-19, IDE, Android Studio, Firebase, RecyclerView, Picasso, Volley, Retrofit, API.

# Abstract

---

The coronavirus or SARS-COV-2 was reported globally on January 7, 2020 in China. After December 31, 2019, the Wuhan Municipal Health Commission announced 27 cases of pneumonia of unknown cause.

In Peru, the first case of COVID-19 was reported in March 2020 and due to the high risk of contagion due to physical contact with infected people, it has been expanding. Although at this point many of the issues have already been clarified, there is still much to be investigated. The disease, because it is a new naturalization, has resulted in many difficulties in the approach to the patient infected by COVID-19, not only in our country, but also in developed countries; however, one of the aspects that can be seen is that the best help to the population both in reducing the aggravation of patients and slowing down the spread of the pandemic is the commitment of the population itself to comply with the norms, advice and rules imposed by the central government and the Ministry of Health in their first levels of care. attention.

Our objective is to contribute to the control of patients who have the disease in a moderate or mild way, since patients who are in a serious situation will be referred to the relevant hospitals or clinics for their care, since ICU (Intensive Care Unit) beds are needed with the pertinent and necessary medical implementations to be able to keep the patient alive. For all of the above and to be able to face this problem, it is proposed to create a computer system to carry out control, monitoring and medical support in the treatment of Covid-19 in order to inform the indicated medical personnel of all the progress of each one of the patients who makes use of said system. A mobile application made for the Android operating system will be implemented which can be installed on any Android mobile device from versions 5.1 Jelly Bean, made in the simplest and easiest way so that any user can use it easily and intuitively. This application will have the options of allowing the patient to be monitored completely remotely for

the treating physician, as well as allowing the creation of a history of the patient's evolution over time, to allow subsequent evaluations. Likewise, we seek that this application allows to know the points with the highest number of infected by area in the city of Cusco. At the same time, we seek to take a first step towards the implementation of this or another similar system in the event of another pandemic or other disease that requires this type of management due to its high degree of contagion through physical contact or being close to the patient. As we hope, this system will result in a greater possibility of care for more people and with what is expected could contribute to keeping patients in constant communication with their doctor, show a map of the most affected areas in the city of Cusco and have better control of infected patients.

**Keywords:** *SARS-COV-2, COVID-19, IDE, Android Studio, Firebase, RecyclerView, Picasso, Volley, Retrofit, API.*

# Índice General

---

DEDICATORIA .....	III
AGRADECIMIENTOS .....	IV
RESUMEN .....	V
ABSTRACT .....	VII
ÍNDICE GENERAL .....	IX
ÍNDICE DE FIGURAS.....	XIII
ÍNDICE DE TABLAS.....	XIII
TÉRMINOS Y ABREVIATURAS .....	XVIII
<b>1. ASPECTOS GENERALES .....</b>	<b>1</b>
1.1. PLANTEAMIENTO DEL PROBLEMA .....	1
1.1.1. Descripción del problema.....	1
1.1.2. Identificación del problema .....	2
1.2. FORMULACIÓN DEL PROBLEMA .....	3
1.2.1. Problema General .....	3
1.2.2. Problema Específico.....	3
1.3. ANTECEDENTES.....	4
1.4. OBJETIVOS.....	10
1.4.1. Objetivo General .....	10
1.4.2. Objetivo Específico.....	10
1.5. JUSTIFICACIÓN .....	11
1.5.1. Conveniencia.....	12
1.5.2. Relevancia.....	12
1.5.3. Implicancias Prácticas.....	12
1.5.4. Valor Teórico.....	13
1.5.5. Utilidad Metodológica .....	13
1.6. ALCANCES Y LIMITACIONES .....	13
1.6.1. Alcances .....	13

1.6.2.	<i>Limitaciones</i> .....	14
1.7.	<b>METODOLOGÍA</b> .....	14
<b>2.</b>	<b>MARCO TEÓRICO</b> .....	<b>16</b>
2.1.	<b>COVID-19</b> .....	<b>16</b>
2.1.1.	<i>Síntomas de la COVID-19</i> .....	17
2.1.1.1.	Medidas de Prevención.....	18
2.1.2.	<i>Diagnóstico clínico de COVID-19</i> .....	19
2.1.3.	<i>Definiciones operativas</i> .....	20
2.1.3.1.	Cuarentena y Aislamiento.....	20
2.1.3.2.	Contacto directo. ....	20
2.1.3.3.	Caso sospechoso .....	21
2.1.3.4.	Caso confirmado:.....	21
2.1.3.5.	Pacientes de riesgo .....	22
2.1.4.	<i>Clasificación clínica</i> .....	23
2.1.4.1.	Infección asintomática o presintomática .....	23
2.1.4.2.	Enfermedad leve.....	23
2.1.4.3.	Enfermedad Moderada.....	23
2.1.4.4.	Enfermedad grave.....	24
2.1.4.5.	Enfermedad crítica.....	24
2.1.5.	<i>Tratamiento en atención primaria</i> .....	24
2.1.6.	<i>Medidas generales</i> .....	25
2.1.7.	<i>Tratamiento farmacológico sugerido para atención primaria</i> .....	25
2.2.	<b>ANDROID STUDIO</b> .....	<b>28</b>
2.2.1.	<i>Kotlin</i> .....	28
2.3.	<b>API</b> .....	<b>29</b>
2.4.	<b>FIREBASE</b> .....	<b>31</b>
2.4.1.	<i>Firebase Authentication</i> .....	32
2.4.2.	<i>Firebase Realtime Database</i> .....	32
2.4.2.1.	Funciones claves .....	33
2.4.3.	<i>Cloud Storage para Firebase</i> .....	34
2.4.3.1.	Funciones clave.....	34
2.5.	<b>VOLLEY</b> .....	<b>35</b>
2.6.	<b>PICASSO</b> .....	<b>36</b>
2.7.	<b>RECYCLERVIEW</b> .....	<b>36</b>
<b>3.</b>	<b>DESARROLLO DEL APLICATIVO</b> .....	<b>38</b>

3.1.	REQUERIMIENTOS FUNCIONALES DEL PROYECTO.....	38
3.2.	REQUERIMIENTOS NO FUNCIONALES DE LA APLICACIÓN.....	40
3.3.	RESTRICCIONES .....	41
3.4.	FASE DE DISEÑO .....	42
3.4.1.	<i>Diagrama de Casos de Uso .....</i>	<i>43</i>
3.4.2.	<i>Diagrama de clases.....</i>	<i>55</i>
3.5.	DISEÑO DE LA BASE DE DATOS .....	57
3.6.	PATRÓN DE DISEÑO DE BACKEND DE FIREBASE.....	58
3.7.	PATRÓN DE DISEÑO DE LA ARQUITECTURA DE LA APP .....	60
3.8.	DISEÑO DE LA INTERFAZ .....	60
3.8.1.	<i>Opción “Datos Perú” .....</i>	<i>62</i>
3.8.2.	<i>Opción Listado países afectados.....</i>	<i>62</i>
3.8.3.	<i>Opción Datos Globales.....</i>	<i>64</i>
3.8.4.	<i>Opción Mapa de casos.....</i>	<i>65</i>
3.8.5.	<i>Opción ¿Tienes síntomas?.....</i>	<i>66</i>
3.8.6.	<i>Opción Control de paciente.....</i>	<i>72</i>
3.8.7.	<i>Opción ¿Eres paciente?.....</i>	<i>94</i>
<b>4.</b>	<b>IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>106</b>
4.1.	DETALLES DE LA IMPLEMENTACIÓN .....	106
4.1.1.	<i>Componentes de la aplicación .....</i>	<i>106</i>
4.1.2.	<i>Estructura del proyecto.....</i>	<i>112</i>
4.1.3.	<i>Diseño .....</i>	<i>116</i>
4.1.4.	<i>Uso de adaptadores para rellenar los datos de una vista.....</i>	<i>119</i>
4.1.5.	<i>Eventos.....</i>	<i>120</i>
4.1.6.	<i>Notificaciones .....</i>	<i>123</i>
4.1.7.	<i>Lectura y escritura de la base de datos.....</i>	<i>123</i>
4.1.8.	<i>Autenticación .....</i>	<i>125</i>
4.1.9.	<i>Compatibilidad.....</i>	<i>127</i>
4.2.	VERIFICACIÓN Y PRUEBAS.....	128
4.3.	GESTIÓN DE RIESGOS.....	132
<b>5.</b>	<b>CONCLUSIONES .....</b>	<b>133</b>
<b>6.</b>	<b>RECOMENDACIONES .....</b>	<b>135</b>
<b>7.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>137</b>



# Índice de Figuras

---

Figura 1 Mascarilla sin válvula de exhalación y mascarilla con válvula de exhalación .....	19
Figura 2 Visualización de un ejemplo de RecyclerView .....	37
Figura 3 Caso de uso General.....	44
Figura 4 Caso de uso Realizar triaje .....	46
Figura 5 Caso de uso Seguimiento del paciente “sin atender” .....	48
Figura 6 Caso de uso Seguimiento del paciente “Mis pacientes “ .....	50
Figura 7 Caso de uso “Asistencia del paciente”.....	53
Figura 8 Diagrama de clase de aplicación de Control y monitoreo del COVID-19....	56
Figura 9 Arquitectura de la aplicación .....	59
Figura 10 Pantalla de inicio del aplicativo      Figura 11 Pantalla del menú desplegable .....	61
Figura 12 Listado de países afectados. ....	63
Figura 13 Información detallada sobre el COVID-19 respecto a cada país. ....	64
Figura 14 Api en Formato JSON sobre información del COVID-19.....	64
Figura 15 Datos Globales de personas infectadas con el Virus.....	65
Figura 16 Mapa de casos de pacientes COVID 19.....	66
Figura 17 Inicio del triaje para verificar los síntomas del paciente.....	67
Figura 18 Ingresar datos del usuario.....	68
Figura 19 Ingresar datos del contacto.....	68
Figura 20 Síntomas que presenta el usuario .....	69
Figura 21 Mapa del cusco para marcar su vivienda actual .....	69
Figura 22 Dirección de la vivienda actual del usuario .....	70
Figura 23 Mensaje de finalización del triaje .....	71
Figura 24 Encuesta de confirmación del virus al usuario .....	71
Figura 25 Acceso al control de paciente con las credenciales del médico.....	72
Figura 26 Pacientes que se encuentran en espera de atención.....	73
Figura 27 Información de un paciente en espera de atención .....	73

Figura 28 Información de la persona a cargo del cuidado del paciente .....	74
Figura 29 Vivienda en la actualidad del paciente .....	75
Figura 30 Confirmación del COVID-19 de un paciente .....	75
Figura 31 Realizar una llamada al paciente para confirmar COVID .....	76
Figura 32 Mensaje de alerta para clasificación de pacientes.....	77
Figura 33 Clasificación de los pacientes según otras enfermedades.....	78
Figura 34 Mensaje clasificación de paciente .....	78
Figura 35 Clasificación del paciente en los 3 grupos .....	79
Figura 36 Clasificación en grupo “Mis Pacientes”.....	80
Figura 37 Información principal del paciente .....	81
Figura 38 Manual médico para atención de pacientes con COVID .....	82
Figura 39 Manual - “Recomendaciones para pacientes en aislamiento”.....	82
Figura 40 Manual – “Medidas Generales”.....	83
Figura 41 Manual – “Tratamiento .....	84
Figura 42 Manual – “Evidencia científica para recomendar medicamentos”.....	84
Figura 43 Botón “Seguimiento del Paciente” .....	85
Figura 44 Descripción del proceso de atención del paciente.....	86
Figura 45 Primer paso para la atención de un paciente – “Atender” .....	87
Figura 46 Preguntas establecidas para el control del paciente.....	87
Figura 47 Advertencia sobre el guardado de las preguntas establecidas.....	88
Figura 48 Mensajes para agregar una receta médica .....	89
Figura 49 Receta médica preestablecida para llenar solo diagnóstico y prescripción .....	90
Figura 50 Receta médica guardada y solo visualizada en el día de la creación de la receta.....	91
Figura 51 Alerta sobre la no existencia de receta médica para el día seleccionado	91
Figura 52 Visualización de respuestas del paciente después del guardado por parte del médico. ....	92
Figura 53 Clasificación en grupo “Pacientes Derivados”.....	93
Figura 54 Detalles de un paciente en estado Crítico.....	93
Figura 55 Módulo “¿Eres Paciente?”.....	94
Figura 56 Login del paciente usando sus credenciales.....	95

Figura 57 Menú del paciente .....	96
Figura 58 Vista principal del Módulo 'Chat' .....	97
Figura 59 Detalles del módulo 'Chat' (1) .....	97
Figura 60 Detalles del módulo 'Chat' (2) .....	98
Figura 61 Lista de conversaciones y cierre de sesión .....	99
Figura 62 Detalles al seleccionar una conversación .....	99
Figura 63 Envío de mensajes mediante el chat .....	100
Figura 64 Ver perfil del médico .....	100
Figura 65 Búsqueda del médico que actualmente atiende al paciente .....	101
Figura 66 Perfil del paciente .....	102
Figura 67 Modificaciones del perfil del paciente .....	102
Figura 68 Realización de llamada telefónica al médico asignado al paciente .....	103
Figura 69 Recetas del paciente .....	104
Figura 70 Mapa de los hospitales en la Ciudad del cusco .....	105
Figura 71 Código que muestra la Herencia de AppCompatActivity .....	107
Figura 72 Código que muestra la redirección entre Actividades .....	108
Figura 73 Parte del Código XML del AndroidManifest .....	109
Figura 74 Extracto del Código XML del Style.xml .....	110
Figura 75 Extracto del Código XML de Colors.xml .....	111
Figura 76 Extracto de la estructura del proyecto en Android Studio .....	113
Figura 77 Extracto del AndroidManifest y el nombre del paquete del proyecto .....	114
Figura 78 Extracto del AndroidManifest con los componentes de la aplicación .....	115
Figura 79 Extracto de la carpeta "res" y los archivos XML .....	116
Figura 80 Jerarquía de vista, que define un diseño de IU .....	117
Figura 81 Extracto del código que muestra el diseño de los detalles del control del paciente .....	118
Figura 82 Extracto del código que muestra como añadir una vista a una Actividad .....	119
Figura 83 Código que muestra como añadir un adapter a la lista .....	120
Figura 84 Código que muestra el callback de un evento .....	122
Figura 85 Código que muestra la escritura en Firebase con setValue() .....	124
Figura 86 Código que muestra el uso de receptores en la lectura de datos de Firebase .....	125

Figura 87 Ejemplos de autenticación de usuarios con Firebase.....	126
Figura 88 Código que muestra la autenticación del usuario por correo .....	126
Figura 89 Versión mínima del SDK para la compatibilidad con los dispositivos Android .....	127
Figura 90 Transferencia de la aplicación a los móviles de los pacientes .....	129
Figura 91 Realtime Database de Firebase.....	129
Figura 92 Exportación de JSON .....	130
Figura 93 Importación de JSON a Excel .....	130
Figura 94 Limpieza de Datos usando Power Query .....	131
Figura 95 Porcentaje de personas que se hicieron prueba Covid .....	131

# Índice de Tablas

---

Tabla 1 Interpretación de resultados combinando /Prueba Rápida (IgM/IgG) y RT-PCR para COVID-19.....	22
Tabla 2 Descripción de celulares con los requisitos para la aplicación.....	42
Tabla 3 Descripción de los actores de la aplicación.....	43
Tabla 4 Especificación de Caso de uso Número 1, Ingresar a la aplicación.....	45
Tabla 5 Especificación de Caso de uso Número 2, Realizar Triage.....	47
Tabla 6 Especificación de Caso de uso Número 3, Seguimiento del paciente “sin atender”.....	50
Tabla 7 Especificación de Caso de uso Número 4, Seguimiento del paciente “Mis pacientes”.....	53
Tabla 8 Especificación de Caso de uso Número 5, Asistencia del paciente.....	54
Tabla 9 Ejemplos de eventos para interactuar con el usuario.....	122

# Términos y abreviaturas

---

**Apache 2.-** Servidor web de código abierto, multiplataforma y gratuito que permite ejecutar un sitio web seguro

**API** (Application Programming Interfaces)

**EPP** (Equipos de Protección Personal)

**Framework.** - Es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software

**GitHub.** - Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git

**Google Cloud Platform.** - Permite crear, implementar y escalar aplicaciones, sitios web y servicios en la misma infraestructura que Google.

**Gradle.** - Es una herramienta que permite la automatización de compilación de código abierto.

**IDE** (Entorno de desarrollo integrado)

**IgM** (Inmunoglobulina M)

**IgG** (Inmunoglobulina G)

**IntelliJ IDEA.** - Es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos.

**IOS.** - Es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad

**JavaScript.** - JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web

**JetBrains.** - Es una compañía de desarrollo de software cuyas herramientas están dirigidas a desarrolladores de software y gerentes de proyectos

**JVM** (Java Virtual Machine)

**Open Source.** - El software open source es un código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente.

**ListView.** - Un ListView en Android es un objeto que nos permite mostrar una lista de elementos desplazables

**AndroidManifest.xml.** - Es un archivo de configuración donde podemos aplicar las configuraciones básicas de nuestra app.

**mmHg** (milímetros de mercurio)

**NoSQL.** - Las Bases de Datos NoSQL (“Not Only SQL”) pertenecen al modelo no relacional, difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales)

**OAuth 2.0.** - Es el mecanismo de autenticación recomendado para todas las API de Google

**OMS** (Organización Mundial de la Salud)

**OpenID Connect.** - Es una capa de identidad sobre el protocolo OAuth 2.0, el cual permite a los clientes verificar la identidad de un usuario basado en la autenticación realizada por un servidor de autorización.

**PaO2** (presión arterial de oxígeno)

**FiO2** (fracción inspirada de oxígeno)

**PCR** (Polymerase Chain Reaction)

**Plantilla.** - También llamando dispositivo de interfaz, suele proporcionar una separación entre la forma o estructura y el contenido. Es un medio que permite guiar, portar, o construir, un diseño o esquema predefinido.

**RT-PCR.** - La RT-PCR en tiempo real es un método nuclear que detecta la presencia de material genético específico de los patógenos, como los virus

**SDK** (Software Development Kit)

**SpO2** (Saturación arterial de oxígeno)

**StackOverflow.** - Es un sitio de preguntas y respuestas para programadores profesionales y aficionados.

**UCI** (Unidad de cuidados Intensivos)

**VIH** (virus de la inmunodeficiencia humana)

# Capítulo 1

---

## 1. Aspectos Generales

### 1.1. Planteamiento del problema

#### 1.1.1. Descripción del problema

En la actualidad son muchas las “alternativas” tecnológicas que han salido como respuesta a esta Pandemia que la humanidad está experimentando desde fines del año 2019; sin embargo, dentro del inmenso surtido de Programas, aplicaciones, Webs especializadas, etc. La mayoría de éstas no están en un idioma de fácil acceso en el Perú, como lo es el español, por lo que aquí es muy escaso encontrar sistemas informáticos funcionales que brinden la posibilidad de tener el acceso a un sistema especializado que no sólo muestre datos, sino que permita llevar un control médico y que realmente contribuya a sobrellevar esta situación y por lo tanto permita en el tiempo dar una solución a este problema.

En Perú se lanzó el aplicativo móvil “Perú en tus manos” la cual es una aplicación para teléfonos móviles que presentó el Gobierno durante la emergencia del COVID-19 la cual ofrecía una función de triaje para los pacientes, mostrar estadísticas de las cifras que son de público conocimiento en la sala situacional del Ministerio de Salud y por último el mapa de zonas de riesgo. Esta aplicación tuvo éxito rotundo los dos primeros meses con más de un millón de descargas, pero con el paso del tiempo su uso fue decreciendo. La causa fundamental o principal es que prometió más de lo que puede cumplir. Primero fue el marco legal ya que por ser de ámbito público solo pueden hacer lo que la ley los autoriza a hacer y segundo fue la negativa del Gobierno a poner a disposición

del público el código fuente que la hace posible como software código abierto (open source). Esta decisión la tomaron muchos países con éxito con sus aplicaciones del COVID-19 como Reino Unido, India, Israel, entre otros. Hacerlo involucraría a una comunidad muy amplia de desarrolladoras y especialistas en tecnología en la tarea de detectar errores y proponer mejoras y además tener una mejor auditoria del funcionamiento del aplicativo.

### **1.1.2. Identificación del problema**

Los Coronavirus son virus que generalmente solo afecta a animales, aunque algunos de ellos son capaces de transmitirse de animales a humanos y causar problemas respiratorios, la mayoría causando síntomas leves.

El SARS-CoV-2 es un nuevo coronavirus que afecta a los humanos y apareció en China en diciembre del 2019, este virus en un 80% de los casos solo produce síntomas respiratorios leves, el otro 15% desarrolla una enfermedad grave y requieren oxígeno y el 5% llegan a un estado crítico y se requiere de cuidado intensivos.

A inicios del 2020 este virus era completamente desconocido, pero a lo largo del tiempo la comunidad científica ya ha logrado desarrollar pruebas para su diagnóstico. A mediados del 2021 se logró obtener vacunas de diferentes marcas que ayudan a desarrollar inmunidad contra el virus que causa el COVID-19, pero ésta no erradica completamente la enfermedad de la población ya que aun en estos tiempos hay gente que no cree en la enfermedad y menos en las vacunas.

## **1.2. Formulación del problema**

### **1.2.1. Problema General**

¿Una aplicación móvil para el control, monitoreo y apoyo médico podría contribuir en el proceso de tratamiento del COVID-19?

### **1.2.2. Problema Específico**

¿Será posible obtener información actualizada del avance del COVID-19 a nivel mundial?

¿Será factible establecer el algoritmo que permita visualizar en un mapa los casos de personas infectadas, las fechas de cuando se produjeron y donde se ubican dichos casos utilizando el GPS del sistema operativo Android?

¿Será posible desarrollar una aplicación móvil que determine valores confiables del número de casos y evolución de pacientes leves, moderado y graves?

¿Será posible establecer una conversación mediante un aplicativo entre el médico y el paciente para reportar dolencias o consultas o verificar el estado del paciente?

## **1.3. Antecedentes**

### **1.3.1. (Instituto Finlandés de Salud y Bienestar (THL)) 2020, “Aplicación móvil Koronavilkku”, Instituto Finlandés de Salud y Bienestar (THL), Finlandia ((THL), 2020)**

Koronavilkku es una aplicación de rastreo de contactos producida por el Instituto Finlandés de Salud y Bienestar (THL) para ayudarlo a averiguar si pudo haber estado expuesto al coronavirus. Si tiene una prueba de coronavirus y se le diagnostica una infección, puede usar la aplicación para compartir esto de forma anónima con aquellos con los que ha estado en contacto cercano. Su privacidad está fuertemente protegida.

Conclusiones: El Instituto Finlandés de Salud y Bienestar ha llevado a cabo el desarrollo de una aplicación de rastreo de contactos para poder mostrar si alguien estuvo expuesto a personas que tengan Coronavirus, así como hacer que si alguien que ya fue detectado como contagiado con el Coronavirus, que ya haya pasado una prueba de Covid-19, pueda compartir anónimamente su ubicación y estado para que otros teléfonos que estén cerca a las personas puedan saber si posiblemente hayan estado expuestas y así tratar de darles un diagnóstico temprano.

Esta aplicación se ha manejado como una aplicación oficial en Finlandia para poder evitar el avance del Covid-19, es de uso totalmente opcional, pero ha contribuido significativamente en el manejo de la enfermedad en el país.

**1.3.2. (José Miguel Renom, Fabien Corniller, Claudia Antonini, Yamilet Serrano, junto a estudiantes de Ciencia de la Computación Jose Garcia, Roosevelt Ubaldo, Carlos Cupe y Jonathan Loza), 2020, “Desarrollo de la aplicación móvil Perú en tus manos”, Universidad de Ingeniería y Tecnología (UTEC), Perú (Renom, y otros, 2020)**

Esta aplicación será un elemento clave para que los ciudadanos vuelvan paulatinamente a la actividad económica al final del período de cuarentena. La aplicación mostrará un mapa de calor a nivel de manzana con círculos rojos en las áreas donde las personas tienen el virus y con círculos naranjas en los lugares donde las personas han registrado síntomas de la enfermedad.

Los datos se actualizarán continuamente de acuerdo con el desarrollo de Covid 19 en Perú. Este mapa debe ser la primera advertencia a los ciudadanos antes de salir a realizar cualquier tipo de actividades permitidas durante y después de la cuarentena.

De igual forma, se ha desarrollado una característica importante para los ciudadanos afectados con esta enfermedad que se encuentran en aislamiento social obligatorio, podrán informar su estado de salud y actualizar sus síntomas para que los miembros del equipo Te Cuido Perú en colaboración con el Ministerio de Salud (Minsa) puedan atender de manera oportuna.

**Conclusiones:** Este grupo de profesionales desarrolló una aplicación que permitiría a los ciudadanos retornar a las actividades de producción económica sin ningún temor al Covid-19 esto a través de la muestra de mapas de calor mostrando los diferentes lugares por donde se encuentran personas que hayan sido diagnosticadas con el virus, así como permiten a las personas contagiadas el

poder reportar sus síntomas a grupos de ayuda que, de la mano del Ministerio de Salud, pueden atender oportunamente a las personas que así lo requieran.

**1.3.3. (Ronald V. Bravo Bustillos), 2020, “Manual del Operador en Central de Urgencias y Emergencias Médicas SAMU”, Dirección Regional de Salud Huánuco - DIRESA, Perú (Bravo Bustillos, 2020)**

Recepción de la llamada de Emergencia:

1. Establecer el Número/Código de atención
2. Fecha
3. Turno de atención
4. Hora de Ingreso de la llamada
5. ¿La llamada fue Pertinente?
6. Hora de Derivación de la llamada
7. Prioridad de la Emergencia
8. Código de Diagnóstico presuntivo
9. Descripción del Diagnóstico presuntivo

**Conclusiones:** Desarrollaron una forma sistematizada de atención de Emergencias médicas y gracias a esta forma de atención han permitido un conocimiento oportuno y real de las atenciones de salud en su territorio con un adecuada y eficiente atención a los pacientes, dando el trato correcto a cada emergencia médica, mejorando así la calidad de vida y permitiendo alcanzar las expectativas del sistema de salud.

**1.3.4. (Otto Martín Díaz, Celia Llanusa Ruiz, Luis Sotillo Bent), 2016, “Sistema de gestión de información para seguimiento y alerta de casos positivos de tamizaje neonatal”, Universidad Latina, Panamá (Martín Díaz, 2016)**

“Se desarrolló un sistema de gestión de información, basado en un software, para el análisis, seguimiento y tratamiento precoz de los casos de Tamizaje Neonatal de Enfermedades Metabólicas de la Caja del Seguro Social de Panamá. Esta herramienta fue realizada a la medida para la gestión de casos, y fue más eficiente y participativa que las previamente utilizadas. La aplicación implementada permite agilizar el proceso de creación y búsqueda de pacientes, así como acelerar la información ofrecida a los gestores de salud, para garantizar el inicio precoz del tratamiento y la rehabilitación, con lo que se logra evitar o reducir las secuelas y la discapacidad producidas por estas enfermedades. La herramienta tiene además la posibilidad de elaborar un mapa que permite localizar y analizar rápidamente la distribución de los casos, lo que facilita realizar estudios epidemiológicos”.

**Conclusiones:** Estos autores desarrollaron un sistema de gestión de información para el tratamiento temprano de casos de detección neonatal de enfermedades metabólicas del Seguro Social de Panamá. Concretamente es un software que ayudó a reducir las secuelas y la discapacidad que se producen por falta de atención rápida a estas enfermedades, facilitando también los estudios epidemiológicos. A través de este software médico obtienen datos de una base de datos de laboratorios del país que aceptaron ser parte de este programa e identifican los casos positivos para poder localizar a las personas y ofrecer el tratamiento oportuno, esto se logra por los datos que se le proveen al programa para evaluar los resultados y detectar los positivos. El software se realizó de manera escalable de tal manera que da exactamente lo mismo usarlo en un centro de salud como usarlo para todo un país, este también facilita el

acceso a estadísticas actualizadas para poder estar siempre en detección de nuevos positivos.

**1.3.5. (Marulanda Echeverry, C.E.; López Trujillo, M. & López Villegas), 2016, “Desarrollo de una aplicación móvil para alerta tecnológica”, Universidad Católica del Norte, Colombia (Marulanda Echeverry & López Trujillo, 2016)**

La aplicación móvil ha sido diseñada teniendo en cuenta las necesidades de toda la comunidad académica, administrativa e investigadora, así como diversos grupos sociales, comunidades y organizaciones, el sector público y privado, que deben realizar el seguimiento de la tecnología desde el desarrollo de la investigación académica que ha realizado la Universidad, para que sea posible crear la aplicación.

**Conclusiones:** Aprovecharon la versatilidad de las tecnologías móviles para poder elaborar una aplicación que permita mejorar el acceso a la gestión de emergencias en salud, alcanzando a diversos sectores que no podían ser abarcados de una manera más tradicional, se usó perspectivas de diversos ámbitos que les permitió cubrir la mayor cantidad de necesidades de las personas.

**1.3.6. (Bedoya Suárez, Alfonso Gabriel, Laurente Rodríguez, Christopher Christian), 2015, “Análisis, diseño e implementación de un sistema de gestión del servicio de ambulancias para la atención de emergencias de una empresa aseguradora en Lima”, Pontifica Universidad Católica del Perú - PUCP, Perú (Bedoya Suárez & Laurente Rodríguez, 2014)**

Con el producto final, es posible proporcionar un conjunto de aplicaciones para todos los involucrados en el proceso de atención de urgencia. El Asegurado utilizará una aplicación móvil para agilizar el envío de alertas; los automovilistas utilizarán una aplicación web móvil para responder a las alertas y encontrar el centro médico más cercano; el personal médico utilizará una aplicación móvil para consultar registros médicos resumidos electrónicos y registrar el curso de la atención; y finalmente la compañía de seguros utilizará una aplicación web para monitorear el flujo de notificación y monitoreo de ambulancias.

El algoritmo implementado en una aplicación de escritorio en segundo plano permite al asegurado asignar automáticamente las alertas enviadas por el asegurado a las unidades de emergencia disponibles.

**Conclusiones:** Crearon un sistema que permite tener una historia clínica a medida que puede ser almacenada digitalmente que luego usan para poder alertar a las unidades de emergencia disponibles de ser necesario, hicieron este sistema de manera escalable y adaptable.

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

Desarrollar una APLICACIÓN MÓVIL PARA CONTROL, MONITOREO Y APOYO MÉDICO EN EL TRATAMIENTO DEL COVID-19

### **1.4.2. Objetivo Especifico**

1. Diseñar un sistema que haga uso de una API (*interfaz de programación de aplicaciones*) para obtener información sobre el avance actual a nivel mundial y por países del Covid-19.
2. Generar un *Data Set* de los casos y las fechas donde se produjeron, así como usar las herramientas que ofrece Android como el *GPS* (Sistema de posicionamiento global) para la captura de ubicación del usuario y mostrar un mapa de las zonas más afectadas en la ciudad de Cusco,
3. Desarrollar un Chat entre el médico y paciente para reportar molestias, realizar consultas o verificar el estado del paciente.
4. Permitir al personal médico el seguimiento del paciente mediante llamada telefónica realizada al paciente o al familiar que está a cargo de este.

## 1.5. Justificación

El presente proyecto es creado debido a la carencia de soluciones informáticas realmente especializadas que permitan atender de manera más eficiente a todos los pacientes y evitar nuevos contagios y/o detectarlos a tiempo. Se aprovechará al máximo el sistema GPS (Sistema de Posicionamiento Global) del sistema operativo Android para poder obtener la ubicación de donde existen personas contagiadas y con quienes se interrelacionaron con la finalidad de poder dar un seguimiento adecuado a todos los casos que tengamos registrados en nuestra Base de datos.

Recordemos que en las circunstancias que atravesamos como humanidad, las personas se sienten muchísimo más seguras si pueden realizar cualquier tarea sin necesidad de recurrir a contacto físico, por lo que los tratamientos “estándar” entre paciente y médico, tienen que dar un paso hacia adelante dando los primeros esbozos de telemedicina en nuestra comunidad.

Se realizará el uso del internet para poder permitir la comunicación oportuna de médico a paciente y así garantizar un control sin riesgos de contagio, además de permitir “visitas virtuales” permanentes por profesionales de salud de una manera más eficiente que una llamada telefónica.

Se hará uso de la versatilidad de Android con la variedad de librerías existentes para poder elaborar una herramienta lo suficientemente útil para permitir que todos puedan tener información adecuada y precisa, se haga un control de los contagiados que se tiene en la base de datos y evitar que nuevas personas sean infectadas, todo esto al margen del factor humano.

### **1.5.1. Conveniencia**

La aplicación móvil resulta muy conveniente ya que ayuda a solucionar un problema social que implica la reducción del contacto físico entre paciente y médico y esto en consecuencia evita el contagio del COVID-19 y brinda una mejor lectura de los casos y zonas afectadas.

### **1.5.2. Relevancia**

Para determinar la relevancia primero nos preguntamos, ¿Quiénes son los beneficiados con dichos resultados de la aplicación? Y la respuesta es tanto el médico, enfermeras, pacientes y las personas más cercanas a ellos porque no hay necesidad de ir a un centro médico u hospital a hacer colas exponiendo a los familiares que sean contagiados o simplemente no ser atendidos. Todo esto se evita ingresando a la aplicación y realizando los pasos necesarios para ser atendidos.

### **1.5.3. Implicancias Prácticas**

La pandemia del COVID-19 trajo muchos obstáculos para todas las personas a nivel mundial. Cada país tomó sus medidas para evitar que se propague esta enfermedad, en el Perú una de las medidas fue el trabajo remoto, con el que muchas personas se han beneficiado, se busca que en ese entender este aplicativo en un futuro puede ser utilizada como una herramienta pionera para la telemedicina, donde no solo se requiera la comunicación entre el médico y el paciente, además se permita rastrear o visualizar la ubicación de personas que padezcan alguna enfermedad para un seguimiento óptimo, todo ello con el consentimiento por ambas partes.

#### **1.5.4. Valor Teórico**

Aunque la aplicación no conlleva un gran valor de conocimientos ni ayuda a apoyar o desarrollar alguna teoría, ya que todo lo desarrollado ya existe en nuestro mundo digital, solamente se está reutilizando todo lo estudiado y aplicándolo por el bien de la sociedad en estos momentos de crisis mundial, también recalcar que el diseño de trabajo de dar seguimiento a los pacientes es una forma de realizar Telemedicina y esto contribuye abiertamente a gente alejada de la ciudad, que no puede costear sus gastos para ser atendido en un hospital.

#### **1.5.5. Utilidad Metodológica**

La creación de la aplicación está diseñada con una arquitectura monolítica ya que una ventaja principal es la velocidad de desarrollo gracias a su implementación sencilla, el rendimiento respecto a las API's consumidas ya que es una base de código y un repositorio centralizado, en tal sentido este tipo de aplicación puede ser utilizada como un prototipo de telemedicina ya que una vez el COVID-19 forme parte de una enfermedad común, el aplicativo puede ser fácilmente rediseñado para realizar otro tipo de seguimiento a otras enfermedades.

### **1.6. Alcances y limitaciones**

#### **1.6.1. Alcances**

El aplicativo móvil será de distribución limitada a la muestra de personas participantes de las pruebas, para nuestra investigación será solo en el departamento de Cusco, Provincia del Cusco tanto en el mapeo de casos, así como en la atención de pacientes.

## **1.6.2. Limitaciones**

- Se considerará como participantes sólo a la muestra de personas con las que trabajaremos durante las pruebas.
- El sistema dependerá de una conexión a internet activa para llevar a cabo las operaciones permitidas por el aplicativo, esto para realizar el almacenamiento de los datos.
- El sistema se realizará únicamente para el sistema operativo Android.
- El sistema no puede considerar la logística y disponibilidad de los profesionales médicos suscritos al programa.
- La falta de acceso a reuniones presenciales con personal médico y/o gubernamental necesario para la elaboración de la tesis, a razón del distanciamiento social, como se realizaría de manera natural en un momento “normal”.

## **1.7. Metodología**

El trabajo de investigación tendrá un enfoque metodológico en cuanto a la investigación y otro en cuanto al desarrollo de la aplicación móvil. En cuanto al enfoque para la investigación el trabajo tendrá un enfoque aplicado a la tecnología, puesto que se pretende investigar documentación acerca de la COVID 19, sus síntomas, su tratamiento y las formas de manejo médico para estos casos, así como la detección de casos de pacientes infectados con COVID-19 y usar todo ese conocimiento académico que se obtenga para poder plantear una solución que conlleva el monitoreo y control de dichos pacientes para tratar de sobrepasar de mejor manera la enfermedad con ayuda de un médico que se dedica a darle seguimiento al paciente. A continuación, se presenta los siguientes pasos a seguir.

- **Revisión de literatura y webgrafía.** Por ser un tema reciente se hizo el estudio de literatura previa sobre el COVID-19 y los aportes que se tiene hasta el momento. Esto permitirá un análisis de cómo funcionará el aplicativo móvil.
- **Diseño del aplicativo.** En esta etapa se realiza la creación del maquetado o interfaz de la aplicación. Al finalizar esta etapa se debe tener la visualización del aplicativo móvil en su totalidad.
- **Desarrollo del aplicativo.** Consiste en la implementación del aplicativo móvil en el lenguaje de programación *Kotlin* para el control y monitoreo de pacientes, así como el mapeo y zona de calor de pacientes infectados con COVID-19.

# Capítulo 2

## 2. Marco Teórico

Las circunstancias actuales de estar viviendo una situación tan poco ortodoxa como la pandemia actual la cual venimos atravesando como humanidad desde mediados a fines del año 2019 y en nuestro país desde febrero del 2020, situaciones que nos han llevado a vivir circunstancias totalmente atípicas como la cuarentena y cambiar muchos de los aspectos en nuestra forma regular de vivir el día a día.

Esto ha conllevado a que todas las ramas de la investigación hayan querido aportar al menos con un “granito de arena” para poder combatir a esta terrible enfermedad, es así como en la necesidad de aportar al desarrollo de una herramienta que pueda servir para apoyar al control y evitar que esta enfermedad se siga expandiendo y para poder realizarlo es necesario definir y revisar algunos conceptos que será necesario conocer para poder continuar con este tema de estudio.

Entre estos pasaremos por términos muy generales como las enfermedades en general a temas más especializados acerca de las tecnologías existentes y como nos pueden ayudar afrontar esta situación.

### 2.1. COVID-19

En diciembre de 2019, la Organización Mundial de la Salud (OMS) emitió una advertencia en la ciudad de Wuhan, provincia de Hubei, China, sobre casos de neumonía de causa desconocida. El 12 de enero de 2020, la OMS determinó que la causa de esta enfermedad respiratoria provino de un nuevo coronavirus llamado COVID-19 el 11 de febrero.

COVID-19 es una enfermedad emergente causada por un nuevo coronavirus, uno de una gran familia de virus que causan infecciones respiratorias, que van desde un resfriado común hasta enfermedades como el síndrome respiratorio de Oriente Medio (MERS) y el síndrome respiratorio agudo severo (SARS). (Universidad Anáhuac, 2021)

### **2.1.1. Síntomas de la COVID-19**

Los síntomas más comunes son:

- Fiebre
- Tos seca
- Fatiga

Síntomas menos comunes que pueden afectar a algunos pacientes:

- Pérdida del gusto u olfato
- Congestión nasal
- Conjuntivitis (ojos rojos)
- Dolor de garganta
- Dolor de cabeza
- Dolor muscular o articular.
- Varias erupciones cutáneas
- Náuseas o vómitos
- Diarrea
- Escalofríos o mareos.
- Irritabilidad
- Disminución de la conciencia (a veces combinada con convulsiones)
- Ansiedad
- Depresión
- Trastorno del sueño
- Complicaciones neurológicas más graves y raras, como apoplejía, encefalitis, estados delirantes y daño nervioso.

Los síntomas graves son:

- Falta de aire (insuficiencia respiratoria)
- Pérdida del apetito
- Confusión
- Dolor u opresión persistente en el pecho
- Temperatura alta (más de 38° C)

Cualquier persona que desarrollen fiebre, tos, dificultades para respirar, dolor u opresión en el pecho y dificultad para hablar o moverse deben consultar a un médico a la brevedad. (Cusco, 2020)

### **2.1.1.1. Medidas de Prevención**

La transmisión de persona a persona se produce a través de las secreciones de una persona infectada, principalmente a través del contacto directo con gotitas respiratorias mayores a 5 micrones (Hasta 2 metros aproximadamente) y las manos contaminadas por las mismas secreciones, luego del contacto con las membranas mucosas de la boca, nariz u ojos. (Cusco, 2020)

- Por lo tanto, el riesgo de transmisión se puede reducir haciendo lo siguiente.
- Use una máscara (sin válvula de exhalación) para cubrirse la boca y la nariz.
- Lávese las manos con agua y jabón o use frecuentemente alcohol (60 a 80%) o gel hidroalcohólico.
- Cúbrase la boca y la nariz al toser o estornudar
- Mantenga una distancia de al menos 2 metros entre personas en cualquier entorno.



*Figura 1 Mascarilla sin válvula de exhalación y mascarilla con válvula de exhalación*

*Fuente: Manual rápido para manejo del covid 19 en el primer nivel de atención- Consejo regional VI (Cusco, 2020)*

### **2.1.2. Diagnóstico clínico de COVID-19**

- El diagnóstico clínico incluye la evaluación de los signos y síntomas del paciente.
- Para un diagnóstico precoz, no se recomiendan las pruebas rápidas ya que no son muy sensibles.
- Las pruebas de imagen están dedicadas al estudio de las complicaciones en pacientes leves, moderados o graves. Estos deben solicitarse cuando la evolución clínica del paciente así lo requiera.
- Al evaluar a un paciente sospechoso o confirmado, debe indicarse el aislamiento de la habitación en su domicilio y los contactos directos deben iniciarse en el domicilio. (Cusco, 2020)

## **2.1.3. Definiciones operativas**

### **2.1.3.1. Cuarentena y Aislamiento**

La cuarentena es un procedimiento en el que se impide que una persona asintomática (o un grupo de personas) abandone su domicilio o alojamiento por un período de 14 días, aplicado bajo las siguientes condiciones:

- Contacto directo con un caso confirmado, desde el último día de contacto con el archivo.
- Ciudadanos o extranjeros provenientes de una zona o lugar de alto riesgo, contados desde la fecha de ingreso.

El aislamiento domiciliario es el proceso mediante el cual una persona sospechosa o confirmada de COVID-19 debe permanecer en una habitación de su hogar o residencia durante un período de 14 días a partir de la fecha de aparición de los síntomas.

Si el paciente no presenta síntomas, se indicará el aislamiento domiciliario por un período de 10 días a partir de la fecha del diagnóstico. (Cusco, 2020)

### **2.1.3.2. Contacto directo.**

Incluir uno de los siguientes en un caso confirmado:

- Persona que compartió o aun comparte el entorno de un caso confirmado de infección por COVID19, a menos de 2 metros (incluidos lugares de trabajo, aulas, hogares privados, hogares de ancianos, prisiones, transporte), sin mascarilla y por lo menos 15 minutos.
- Trabajadores de la salud y trabajadores esenciales corren el riesgo si no han usado el equipo de protección personal (EPP) o no siguieron los procedimientos correctos de aplicación, remoción y eliminación de EPP durante la evaluación de un caso confirmado por COVID19. (Cusco, 2020)

### **2.1.3.3. Caso sospechoso**

Se define como caso sospechoso, (Cusco, 2020) a los pacientes que presenten al menos uno o más de los siguientes síntomas:

- Fiebre/ escalofríos.
- Tos seca.
- Congestión nasal.
- Dolor de garganta.
- Malestar general.
- Astenia (cansancio o debilidad).
- Cefalea.
- Anosmia (alteraciones del olfato)
- Disgeusia (alteraciones del gusto)
- Mialgias (dolores musculares)
- Artralgias (dolores articulares)
- Diarrea.
- Náuseas o vómitos.
- Lesiones de piel.
- Dificultad para respirar.

### **2.1.3.4. Caso confirmado:**

Incluye cualquiera de las siguientes situaciones:

- Nexo epidemiológico: Caso sospechoso con síntomas que cumpla criterio de ser contacto directo de un paciente confirmado.
- Caso sospechoso con imágenes compatibles con neumonía viral por COVID-19. (Tomografía de tórax / Radiografía simple de tórax).
- Caso sospechoso con una prueba de laboratorio positiva para COVID-19: RT-PCR positiva y/o una prueba rápida de detección de IgM, IgG o IgG/IgM. (Tabla N.º 1) (Cusco, 2020)

Resultados			Significado clínico
PCR	IgM	IgG	
-	-	-	Etapa temprana negativa / falso negativo por PCR.
+	-	-	Fase inicial de la infección (etapa serológica) +
+	+	-	Etapa aguda
+	+	+	Etapa aguda (más avanzada que el anterior)
+	-	+	Etapa final de la infección
-	+	-	Etapa temprana/falso positivo/falso negativo PCR
-	-	+	Infección pasada / falso PCR negativa
-	+	+	Enfermedad actual/ falso negativo PCR

*Tabla 1 Interpretación de resultados combinando /Prueba Rápida (IgM/IgG) y RT-PCR para COVID-19*

*Fuente: Pruebas diagnósticas de laboratorio de COVID-19. Asociación Española de Pediatría de Atención Primaria – 2020, [https://www.aepap.org/sites/default/files/documento/archivos-adjuntos/pruebas\\_diagnosticas\\_de\\_laboratorio\\_de\\_covid\\_vfinal.pdf](https://www.aepap.org/sites/default/files/documento/archivos-adjuntos/pruebas_diagnosticas_de_laboratorio_de_covid_vfinal.pdf)*

### 2.1.3.5. Pacientes de riesgo

Se considera que los pacientes con las siguientes afecciones tienen un alto riesgo de sufrir complicaciones por la infección por SARCov2, independientemente de la edad:

- Obesidad (Índice de masa corporal de 30 o superior)
- Diabetes tipo 1 o 2
- Hipertensión.
- Fumadores.
- Cáncer
- Enfermedad renal crónica.
- Insuficiencia cardíaca, miocardiopatías.

- EPOC (enfermedad pulmonar obstructiva crónica), asma (moderada o severa), fibrosis pulmonar.
- Personas con inmunodeficiencia (sistema inmune debilitado) debido al trasplante de médula ósea o de órganos, VIH, el uso de corticosteroides o el uso de medicamentos que debilitan el sistema inmunológico.
- Anemia drepanocítica, talasemia.
- Embarazo
- Enfermedad cerebrovascular, demencia.
- Fibrosis quística.

## **2.1.4. Clasificación clínica**

### **2.1.4.1. Infección asintomática o presintomática**

Son pacientes con resultado positivo a PCR o una prueba serológica positiva, pero que no presentan síntomas.

### **2.1.4.2. Enfermedad leve**

El paciente presenta alguno de los siguientes signos o síntomas: fiebre / escalofríos, tos seca, congestión nasal, dolor de garganta, malestar general, astenia (fatiga o debilidad), dolor de cabeza, sensación de anosmia (trastorno del olfato), disfunción (trastorno del gusto), mialgia. (dolor muscular), artralgia (dolor articular), diarrea, náuseas o vómitos, lesiones cutáneas y sin dificultad respiratoria, disnea, taquicardia respiratoria, taquicardia, cianosis, anomalías torácicas, es decir, sin signos de lesión pulmonar o hipoxia. (Cusco, 2020)

### **2.1.4.3. Enfermedad Moderada**

Los pacientes con signos de enfermedad del tracto respiratorio inferior en la evaluación clínica (auscultación cardiopulmonar con crepitantes o estertores subpulmonares húmedos o disminución de la frecuencia respiratoria o sibilancias) o en las imágenes y la saturación de oxígeno (SatO<sub>2</sub>) < 85% (a 3400 msnm), taquipnea o disnea. (Cusco, 2020)

#### **2.1.4.4. Enfermedad grave**

El paciente tiene una frecuencia respiratoria > 30 respiraciones / min, SpO<sub>2</sub> < 85% en aire ambiente a 3400 m de altitud, relación entre la presión parcial de oxígeno arterial y la fracción inspirada de oxígeno (PaO<sub>2</sub> / FiO<sub>2</sub>) < 300 mmHg o infiltrados pulmonares superiores al 50% en los exámenes por imágenes. Es decir, el paciente necesita oxígeno o un ventilador. Es de manejo hospitalario. (Cusco, 2020)

#### **2.1.4.5. Enfermedad crítica**

Pacientes que tienen insuficiencia respiratoria, shock séptico y/o disfunción multiorgánica. (Cusco, 2020)

### **2.1.5. Tratamiento en atención primaria**

Si se decide iniciar tratamiento y aislamiento en casa es importantes tener en cuenta lo siguiente:

- Disponer de una habitación privada con la mejor ventilación posible.
- Tener un baño, si es posible, para uso privado.
- Disponer de un teléfono que permita el contacto con el personal médico.
- Las personas con factores de riesgo no vivan en casa.
- Si es posible, un oxímetro de pulso y un termómetro deben estar disponibles para monitorear y monitorear al paciente. (Cusco, 2020)

## 2.1.6. Medidas generales

- **Supervisar y estabilizar funciones críticas:** A través de los recursos disponibles.
- **Hidratación y nutrición adecuadas:** una ingesta adecuada de alimentos y líquidos evaluados por el médico tratante ayuda al cuerpo a responder mejor a las infecciones; Se debe tener cuidado en pacientes con enfermedades crónicas como diabetes mellitus, obesidad o sobrepeso, insuficiencia renal y / o cardíaca. El tipo de líquido y alimento ingerido depende del juicio médico y las preferencias del paciente. Los alimentos deben variarse para asegurar las calorías adecuadas para el paciente y para los líquidos. Estos no deben contener demasiada sal, azúcar o estimulantes.
- **Enfermedad crónica:** el tratamiento de los pacientes con enfermedades crónicas subyacentes debe continuarse según las indicaciones del médico tratante. Para problemas de obstrucción bronquial, se pueden usar broncodilatadores según sea necesario además de la atención de bioseguridad en aerosol.
- **El reposo relativo**, indicado al comienzo del caso del paciente, debe limitar las actividades físicas extenuantes, recomendar movimientos en la sala de aislamiento, realizar movimientos de estiramiento y no permitir que más de 2 no se muevan ahora. (Cusco, 2020)

## 2.1.7. Tratamiento farmacológico sugerido para atención primaria

1. **Paracetamol.** Es el medicamento más ampliamente usado y seguro en población adulta y pediátrica. Ideal para los casos leves, se debe prescribir

condicional a la presencia de síntomas. Evaluar antecedentes de problema hepático.

- Dosis recomendada: En adultos, 1 gramo cada 6 a 8 horas por vía oral. (Hasta 4gr/ día)
- En niños a partir de 10-15 mg / kg de peso corporal / dosis condicional, hasta 4 dosis por día.

**2. Metamizol.** Puede prescribirse como alternativa al uso de paracetamol.

Dosis recomendada: En adultos: 500 mg por vía oral cada 8 horas o 12 gramos por vía intravenosa o intramuscular cada 8 horas.

En niños: 15-20 mg / kg de peso corporal según los síntomas. Hasta 3 porciones al día.

**3. Antiinflamatorio (ibuprofeno).** Generalmente no se recomienda porque es dañino para el estómago, aunque en estudios consecutivos no se ha demostrado que su uso sea contraproducente en el desarrollo de COVID-19

**4. Antibióticos.** Para casos leves, no se recomienda el uso de antibióticos. En casos moderados, estará indicada la antibioterapia en función de la valoración clínica y la sospecha de sobreinfección bacteriana. Según informes preliminares, la tasa de sobreinfección bacteriana es del 38% en pacientes hospitalizados con COVID19.

**a) Azitromicina.** Actualmente, no se recomienda para uso profiláctico en pacientes con COVID-19. Puede usarse en sobreinfecciones bacterianas y está disponible con receta médica.

**b) Doxiciclina:** Su efectividad en COVID-19 no ha sido probada, pero es una alternativa antibiótica útil en infecciones respiratorias si la evaluación médica lo demuestra.

**5. Corticosteroides.** Actualmente no se recomienda el uso profiláctico en pacientes con estadios tempranos de la enfermedad, su uso está indicado en

pacientes moderados o severos que requieran oxigenoterapia. Se recomienda su uso bajo prescripción médica.

**6. Anticoagulantes.** Su uso está indicado en pacientes con inmovilización moderada o severa o aquellos con alto riesgo de tromboembolismo.

**7. Plasma de un donante convaleciente.** Actualmente, el uso de plasma de donante convaleciente está indicado en ensayos clínicos o en pacientes con indicaciones específicas de tratamiento hospitalario. Desde atención primaria podemos ayudar a inspirar a los pacientes a convertirse en donantes de plasma, informándoles que se trata de un procedimiento ambulatorio con poco riesgo para la salud.

**8. Oxigenoterapia.** Consideramos la saturación de oxígeno adecuada para Cusco, un rango entre 88%- 95 % a 3400 msnm con una FiO<sub>2</sub> de 21%. Un valor inferior al 85% justificará el uso de oxígeno suplementario. Esto es útil para pacientes sin antecedentes de enfermedad pulmonar o cardíaca crónica. El uso de oxígeno en el hogar puede estar indicado si se aseguran las condiciones de acceso al oxígeno y suministro de oxígeno en el hogar y cuando el paciente no presenta otros signos graves. (Cusco, 2020)

## 2.2. Android Studio

Android Studio es un Entorno de Desarrollo Integrado (IDE) y su función es desarrollar aplicaciones móviles para el sistema operativo Android y basadas en IntelliJ IDEA (Developer, 2021) Además del robusto editor de código y las herramientas de desarrollo que ofrece IntelliJ, Android Studio ofrece aún más funciones para aumentar su productividad mientras desarrolla aplicaciones de Android de la siguiente manera:

- Ofrece un sistema de compilación adaptable basado en Gradle.
- Emulador rápido y con muchas funciones.
- Entorno unificado para el desarrollo de todos los dispositivos Android.
- Integración de GitHub y ejemplos de código para instruir a compilar aplicaciones populares y también importar ejemplos de código libre
- Herramientas ligeras para monitorear e identificar problemas de rendimiento, usabilidad, compatibilidad de versiones y más.
- Compatible con C++ y NDK.
- Soporte de integración para Google Cloud Platform, lo que simplifica la integración con Cloud Messaging y App Engine.

### 2.2.1. Kotlin

Kotlin es un lenguaje de programación pragmático (NTT DATA, 2021), diseñado para funcionar con máquina Virtual de Java (JVM) y Android. Alternativamente, se puede compilar en código fuente de Javascript.

Una de sus características principales es su combinación de características orientadas funcionalmente a la hora de programar, con un enfoque en la seguridad, la claridad y la interoperabilidad. Los programas de Kotlin pueden utilizar los frameworks u otras librerías de Java existentes. Así mismo, su interoperabilidad no requiere capas adaptativas.

Su sintaxis también es intuitiva. Tiene una similitud con Scala; por ejemplo, el punto y coma al cerrar una oración es opcional y los saltos de línea suelen ser suficientes para que el compilador comprenda que la declaración está completa; sin embargo, es más simple y permite una compilación más rápida.

La herramienta que utiliza Kotlin para la conversión de Java a Kotlin en un click, hace notablemente su crecimiento en el mercado y más aun sabiendo que es un lenguaje de código abierto.

## **2.3. API**

El término API significa Interfaces de programación de aplicaciones, que en español significa Interfaces de programación de aplicaciones (Xakata, 2022). Es un conjunto de funciones y que se utilizan para el desarrollo y la integración de software de aplicaciones, lo que permite la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Por tanto, podemos resumir de un API como una especificación formal que establece cómo un módulo de software que interactúa con otros módulos de software para realizar una o más funciones. Todo depende de qué aplicaciones las usarán y que permisos otorgan los propietarios de la API a los desarrolladores externos.

Los APIS no son la parte visible o tampoco se encuentran en la superficie de la web, sino son los circuitos internos que solo los desarrolladores ven y conectan para operar una herramienta.

Para los usuarios normales, lo único que verá en la API es el resultado, cómo abre el juego móvil o cómo puede iniciar sesión en su cuenta de Facebook o cómo puede publicar los resultados del juego en Twitter. O cuando esta aplicación le envía notificaciones en su teléfono celular o computadora.

Una de las principales funciones de la API es facilitar a los desarrolladores el ahorro de tiempo y dinero. Por ejemplo, si está creando una aplicación que es una tienda en línea, no necesitará crear una función u otro sistema desde cero para verificar si un producto está en stock. Puede utilizar una API de un servicio de pago que ya existe, por ejemplo, PayPal, y pedirle una API a su distribuidor que le permita saber el catálogo de sus productos.

Con esto, no necesitaríamos reinventar la rueda o crear el fuego con cada servicio creado, puesto que podrá traer funciones o partes de ella que otros han creado. Solo con imaginar que cada tienda online tenga que tener su propio sistema de pago resulta tedioso, entonces para un usuario promedio es mucho más cómodo y fácil hacerlo con los servicios que el mundo utiliza actualmente.

## 2.4. Firebase

Es una plataforma en la cual se puede desarrollar aplicaciones móviles y aplicaciones web con una muy buena calidad, lanzada en 2011 y adquirida por Google en 2014 (Wikipedia, 2021).

Se trata de una plataforma basada en la nube, integrada con Google Cloud Platform, que utiliza un conjunto de herramientas para crear y sincronizar proyectos que serán de alta calidad, ayudando a incrementar el número de usuarios y generar más dinero.

Los desarrolladores cuentan con varias ventajas al usar esta plataforma:

- Sincronice de una manera sencilla los datos del proyecto sin la necesidad de tener que administrar una conexión o escribir una lógica para una sincronización compleja.
- Use de un kit de herramientas multiplataforma: fácil integración tanto para web y aplicaciones móviles. Tiene compatibilidad con las principales plataformas, como iOS, Android, aplicaciones web, Unity y C++.
- Utiliza la infraestructura de Google y realiza el escalamiento automático para cualquier tipo de aplicación, desde la más pequeña hasta la más poderosa.
- Creación un proyecto sin uso de un servidor: las herramientas están incluidas en los SDK tanto móviles como web, por lo que no es necesariamente obligatorio el crear un servidor para un proyecto.

Firebase brinda a los usuarios una documentación robusta y completa para la creación de aplicaciones utilizando dicha plataforma. Además, brinda soporte gratuito mediante correo electrónico a todos los usuarios, y, además, sus desarrolladores se encuentran activamente en plataformas como Github y StackOverflow, así como un canal de YouTube que explica cómo funciona Firebase en todas las plataformas principales.

Con todas estas características, cualquier desarrollador puede mezclar y combinar la plataforma según sus necesidades.

### **2.4.1. Firebase Authentication**

La mayoría de las aplicaciones necesitan identificar al usuario. Conocer la identidad del usuario permite que la aplicación almacene sus datos en la nube de manera segura y ofrecer el mismo servicio y experiencia personalizada en todos los dispositivos sincronizados del usuario. (Firebase Authentication, 2021)

Firebase Authentication ofrece servicios de backend, un SDK para un fácil manejo y una biblioteca de IU predefinida para autenticar en tu aplicación a los usuarios. Incorpora la autenticación mediante contraseña, número de celular o proveedores de identidad popular como Facebook, Google o Twitter, etc.

Firebase Authentication está estrechamente vinculado con otros servicios de Firebase y no desaprovecha los estándares de la industria como OAuth 2.0 y OpenID Connect, por lo que se puede enlazar sin ningún problema con su backend personalizado.

### **2.4.2. Firebase Realtime Database**

El Realtime Database es una base de datos que se encuentra en la nube. Los datos se almacenan con un formato JSON y se sincronizan en tiempo real con cada uno de los clientes conectados. Cuando se crea aplicaciones multiplataforma con nuestros SDK de Android, Apple y plataformas JavaScript, todos sus clientes compartirán una instancia de Realtime Database y recibirán de manera automática actualizaciones y/o cambios de sus datos más recientes. (Firebase Realtime Database, 2021)

Con Firebase Realtime Database, es posible la creación de aplicaciones colaborativas, que permiten el acceso seguro a las bases de datos directamente desde el código del cliente. Los datos se mantienen localmente. Además, aun estando sin conexión, los eventos se siguen activando en tiempo real, lo que

brinda una experiencia de usuario receptiva. Cuando el dispositivo vuelve a estar en línea, la base de datos en tiempo real sincroniza los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión, combinando automáticamente los conflictos.

La base de datos en tiempo real, es una base de datos NoSQL, por ende, tiene diferentes características y optimizaciones en comparación con una base de datos relacional. La API de Realtime Database está diseñada para permitir solo operaciones que se puedan ejecutar de manera rápida. Esto le permite crear una experiencia increíble en tiempo real que pueden servir a millones de usuarios sin comprometer la capacidad de respuesta. Es indispensable pensar en cómo los usuarios deben acceder a los datos y estructurarlos en consecuencia.

#### 2.4.2.1. Funciones claves

**Tiempo real:** en lugar de las solicitudes HTTP tradicionales, Firebase Realtime Database utiliza la sincronización de datos (cada vez que cambian cualquier tipo de datos, los dispositivos conectados reciben una actualización en milisegundos). Proporcione una rica experiencia de colaboración sin preocuparse por el funcionamiento de la RED.

**Sin conexión:** Todas las aplicaciones de Firebase siguen funcionando, aun cuando están fuera de línea, ya que el SDK de Firebase Realtime Database lleva tus datos directamente al disco. Cuando vuelve la conexión, el dispositivo cliente obtiene los cambios que faltan y automáticamente los sincroniza con el estado actual del servidor.

**Client Acces:** Se puede acceder a la base de datos en tiempo real de Firebase desde un dispositivo móvil o navegador web; No se requiere ningún servidor de aplicaciones. La seguridad y la validación de datos están disponibles a través de las reglas de seguridad de Firebase Realtime Database, reglas asentadas en expresiones que se establece cuando se leen o se escriben datos.

### 2.4.3. Cloud Storage para Firebase

Cloud Storage para Firebase está implementado para los desarrolladores de aplicaciones que requieran almacenar y transmitir contenido creado por el usuario, como fotos, sonidos o videos.

Este es un servicio de almacenamiento de objetos potente, simple y rentable de Google diseñado para ser escalable. El SDK de Firebase para almacenamiento en la nube le da un valor agregado de seguridad de Google a las cargas y descargas de archivos de su aplicación Firebase, libremente de la calidad de la red. (Cloud Storage para Firebase, 2021)

#### 2.4.3.1. Funciones clave

**Operaciones robustas:** El SDK de Firebase para Cloud Storage realiza carga y descarga de archivos sin importar la calidad de la red. Las cargas y descargas son sólidas, eso quiere decir que se reinician donde lo dejaron para ahorrar tiempo y ancho de banda a los usuarios.

**Seguridad sólida:** Los SDK de Firebase para almacenamiento en la nube se integra con Firebase Authentication para proporcionar autenticación fácil e intuitiva para los desarrolladores. Puede utilizar un modelo de seguridad declarativo para otorgar el acceso según el nombre del archivo, el tipo de contenido, el tamaño y otros metadatos.

**Altamente escalable:** Está diseñado para escalar hasta exabytes si su aplicación se vuelve viral. Puede pasar fácilmente del prototipo a la producción utilizando la misma infraestructura que utiliza Spotify y Google Photos.

## 2.5. Volley

Es una librería creada por Google para mejorar el envío de solicitudes Http desde las aplicaciones Android a servidores externos.

Esta librería opera como una interfaz de alto nivel, independizando al programador de tediosos análisis y administración de subprocesos, para permitir una fácil publicación de los resultados en el subproceso principal.

Volley se basa completamente en solicitudes, lo que evita la generación repetitiva de código para manejar tareas asíncronicas para cada solicitud o incluso para analizar datos del flujo externo.

Entre sus funcionalidades más potentes podemos encontrar:

- Manejo concurrente de solicitudes.
- Priorizar las solicitudes, lo que le permite definir la prioridad de cada petición.
- Cancelación de peticiones, evitando así la presentación de resultados inesperados en el hilo principal.
- Gestión automatizada de trabajos en segundo plano, sin pasar por la ejecución manual de un framework de hilos.
- Implementar caché en disco y memoria.
- Posibilidad de personalizar peticiones.

Proporciona información detallada sobre el estado y flujo de trabajo de las solicitudes en la consola de depuración. (Volley, 2021)

## 2.6. Picasso

Picasso es una librería para Android que se utiliza para administrar imágenes mediante la técnica de almacenamiento en caché. Esto consiste en que la librería es responsable de almacenar / recuperar imágenes del almacenamiento interno o volátil del dispositivo, o descargarlas de Internet si es necesario. También se encarga de su eliminación cuando ya no la usamos. (tutsplus, 2021)

## 2.7. RecyclerView

La clase RecyclerView se usa para mostrar una lista (o una cuadrícula) de elementos.

Tiene este nombre porque cuando se muestran los elementos de la lista, los elementos que ya no están visibles se reciclan para mostrar los siguientes elementos. (Android, 2021)

RecyclerView es una versión mejorada de la clase ListView, principalmente cuando el número de elementos puede cambiar y/o los datos cambian continuamente.

Es posible personalizar un ListView para hacer lo mismo, pero eso implicaría mirar diferentes detalles para obtener el mismo rendimiento.



*Figura 2 Visualización de un ejemplo de RecyclerView*

*Fuente: Develou, 2021. Crear Listas Con RecyclerView En Android,*

*<https://www.develou.com/recyclerview-en-android/>*

# Capítulo 3

## 3. Desarrollo del aplicativo

En ese capítulo se verá cada una de las fases para el desarrollo de la aplicación móvil de acuerdo con la metodología de desarrollo de software ágiles ya que los equipos que los utilizan son mucho más productivos y eficientes porque saben lo que tienen que hacer en cada momento. Además, la metodología permite adaptar el software según las necesidades que surjan en el camino, lo que facilita la construcción de aplicaciones más funcionales. La principal metodología ágil que usamos fue la Scrum ya que distribuye los requisitos en porciones mínimas. Esto se repite por tiempos cortos y fijos (de dos a cuatro semanas) para lograr un resultado perfecto en cada repetición. Los pasos son: planificación iterativa (sprint Planning), implementación (sprint), reunión diaria (daily Meeting) y presentación de resultados (sprint review).

### 3.1. Requerimientos Funcionales del proyecto

La aplicación constará de 8 apartados o módulos donde detallaremos cada opción que contará:

1. Datos Perú
  - visualizar datos sobre el covid-19
2. Lista países afectados
  - visualizar datos de cada país sobre el covid-19
3. Datos globales
  - visualizar datos globales sobre el covid-19
4. Mapa de casos
  - visualizar ubicación y zonas infectadas por covid-19
5. ¿Tienes síntomas?
  - Ingresar al triaje
    - Ingresar datos personales
      - Ingresar síntomas

- Ingresar ubicación del paciente
      - Ingresar dirección del paciente
        - Cuestionario
- 6. Control de paciente
  - Login de usuario(médico)
  - Visualizar pacientes en espera
    - Atender
      - Confirmar paciente covid-19
        - Llamar al paciente
        - Siguiete (confirmación de paciente)
        - Siguiete (clasificación del paciente)
      - Detalles de contacto
        - Llamar
      - Ubicación del paciente
    - Visualizar mis pacientes
      - Visualizar detalles
        - Seguimiento del paciente
          - Atender
            - Llamar
            - Guardar
          - Ver
          - Agregar día
            - Visualizar atención diaria del paciente
        - Manual
        - Detalles del contacto
          - Llamar
        - Ubicación del paciente
    - Visualizar mis pacientes derivados
      - Visualizar detalles
- 7. ¿Eres paciente?
  - Login de usuario(paciente)
  - Visualizar menú de paciente
    - Chat

- Chats
- Buscar a médico
  - Listado de médicos que usan la aplicación
- Perfil
  - Configuración del perfil del usuario
- Llamar al doctor
  - Realiza una llamada al doctor asignado al paciente
- Ver receta
  - Visualización diaria de la atención del paciente
    - Visualización de receta médica si es que hubiera
- Mostrar mapa
  - Muestra ruta entre dos puntos

## 8. Nosotros

## 3.2. Requerimientos no Funcionales de la aplicación

### Rendimiento de la aplicación

- La Aplicación proporciona retroalimentación (Datos) al usuario en tiempo real.
- El tiempo medio de respuesta de la aplicación no debe exceder los 15 segundos según la evaluación de Google para el 2018. (Google Inc., 2018)

### Seguridad

Conexión necesaria (Para médicos), es necesario verificar que el usuario creado sea exclusivamente para la aplicación por lo que se consulta directamente en la base de datos para acceder a la aplicación.

### Disponibilidad

- La aplicación estará disponible todo el tiempo, ya que es una aplicación nativa que se instalará en el dispositivo móvil.

- La aplicación es dependiente de la conexión a internet o plan de datos del operador del usuario para mostrar toda la información

### **Mantenibilidad**

El sistema estará en constante mantenimiento por un período de 6 meses ya que se podrá agregar nuevas funcionalidades o realizar modificaciones o correcciones.

### **Portabilidad**

Compatibilidad con plataformas: El sistema se llevará a cabo desde la versión de Android 6.0, sólo para Smartphones.

### **Operabilidad**

La aplicación podrá ser utilizada por cualquier persona, pero en los apartados de “Control del Paciente” y “¿Eres paciente?” sólo podrá ser utilizado por médicos y los pacientes que cuenten con un usuario y contraseña respectivamente.

## **3.3. Restricciones**

Para un correcto funcionamiento de la aplicación se debe tener los siguientes requisitos mínimos

- Procesador: Octa-Core 64 bits
- Memoria Ram: 4GB
- Almacenamiento: 50Mb
- Pantalla: 6” Pulgadas en adelante
- Sistema Operativo: Android 6.0 o Superior
- WiFi
- Plan de Datos: 3G o Superior

Se detalla a continuación algunas marcas y modelos de celulares más reconocidos, las cuales cumplen los requisitos mínimos y que no tendrán problemas en la ejecución.

Marca/Modelo	Procesador	Memoria RAM	Versión de Android
Sony Xperia Z3	Qualcomm Snapdragon 801	3 GB	Android 4.4.4 (KitKat)
Samsung Galaxy S7	Exynos 8890	4 GB	Android 6.0.1 Marshmallow
Samsung Galaxy J5	Snapdragon 410 1.2GHz.	1.5GB.	Android 5.1 Lollipop
Huawei Mate 8	Kirin 950 2.3GHz	3GB/4GB	Android 6.0 Marshmallow
LG G4 BEAT	Snapdragon 615 1.5GHz	1.5GB de RAM	Android 5.1.1 Lollipop
Motorola Moto G3	Snapdragon 410 1.4GHz	1GB/2GB	Android 5.1 Lollipop
Huawei P8	KIRIN 930	3GB	Android 5.0.2 Lollipop
HTC Desire 10	Qualcomm Snapdragon 400 quad-core	2GB/3GB	Android 6.0 Marshmallow

*Tabla 2 Descripción de celulares con los requisitos para la aplicación*

*Fuente: Elaboración Propia*

Sin embargo, debe tenerse en cuenta que todas las limitaciones descritas y analizadas anteriormente son solo sugerencias para un óptimo rendimiento de la aplicación, es decir que sea rápida y tenga buena visualización y con ello no se restrinja que la aplicación pueda funcionar en dispositivos de menor gama.

### **3.4. Fase de diseño**

Una de las prácticas fundamentales de la metodología de programación extrema (XP) es utilizar diseños tan simples como sea posible, en tal sentido se elaborarán diseños breves que sirven de referencia para la implementación.

En esta fase usaremos el modelado UML porque se puede usar para modelar diferentes tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real. Los programadores han elegido el modelado

UML porque que es un lenguaje para discutir problemas y soluciones relacionados con la construcción de sistemas y principalmente porque es un lenguaje de modelado de software muy utilizado.

Para este proyecto utilizaremos los diagramas más adecuados las cuales son:

- Diagramas de Casos de Uso
- Diagrama de Clases

Para un mejor entendimiento de parte de las personas se utilizará aparte de los modelados UML los modelo como el **Arquitectónico** con el objetivo ofrecer una visión simplificada del sistema y pueda entender de una sola pasada lo que se quiere conseguir o desarrollar. Y el de **Navegación** la cual nos ayudara a comprender el orden en el que se presenta las pantallas de nuestra aplicación con el contenido y enlaces que existen en cada una de ellas

### 3.4.1. Diagrama de Casos de Uso

A continuación, identificaremos a los Actores.

Actores de la Aplicación	
Paciente	Usar la aplicación, Realizar triaje para verificar si es paciente Covid-19
Médico	Usar la aplicación, Monitorear pacientes Covid-19
Administrador	Usar la aplicación, crear nuevos usuarios para médicos, acceso total a la aplicación.

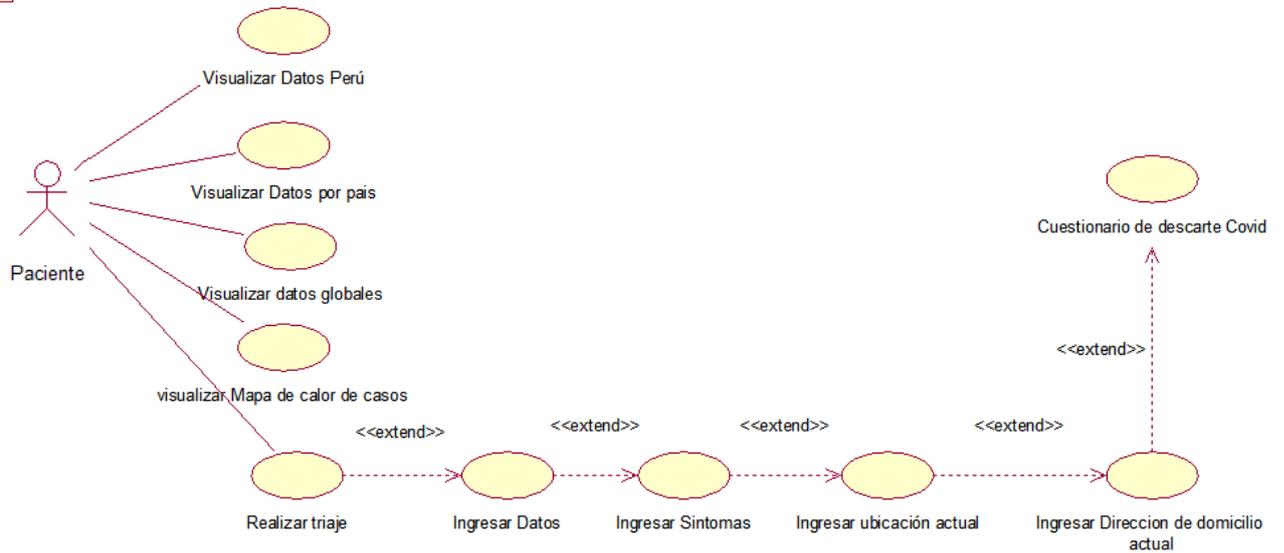
*Tabla 3 Descripción de los actores de la aplicación*

*Fuente: Elaboración Propia*

## Casos de Uso General

Veremos primero como el usuario interactúa en el escenario de la aplicación.

Aplicación de  
monitoreo



*Figura 3 Caso de uso General*

*Fuente: Elaboración propia*

### Especificación de Caso de Uso:

<b>CASO DE USO NUMERO 1</b>																					
<b>Nombre</b>	Ingresar a la aplicación																				
<b>Descripción</b>	Paciente ingresa a la aplicación para su uso																				
<b>Actor</b>	Paciente																				
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Conexión a internet</li> <li>• Que la aplicación se haya ejecutado</li> </ul>																				
<b>Flujo del Sistema</b>	<table border="1"> <thead> <tr> <th><b>Paso</b></th> <th><b>Acción</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>La aplicación despliega ventana de inicio</td> </tr> <tr> <td>2</td> <td>El paciente selecciona cualquier opción del menú</td> </tr> <tr> <td>3</td> <td>El paciente selecciona ¿Tienes síntomas? (triaje)</td> </tr> <tr> <td>4</td> <td>El paciente ingresa toda la información solicitada</td> </tr> <tr> <td>5</td> <td>La aplicación muestra interfaz de llenado de datos correctos</td> </tr> <tr> <td>6</td> <td>La aplicación agrega los datos ingresados en la Base de datos</td> </tr> <tr> <td>7</td> <td>La aplicación Muestra interfaz de llenado correcto de datos.</td> </tr> <tr> <td>8</td> <td>La aplicación muestra botón de confirmación de registro</td> </tr> <tr> <td>9</td> <td>La aplicación regresa al menú principal</td> </tr> </tbody> </table>	<b>Paso</b>	<b>Acción</b>	1	La aplicación despliega ventana de inicio	2	El paciente selecciona cualquier opción del menú	3	El paciente selecciona ¿Tienes síntomas? (triaje)	4	El paciente ingresa toda la información solicitada	5	La aplicación muestra interfaz de llenado de datos correctos	6	La aplicación agrega los datos ingresados en la Base de datos	7	La aplicación Muestra interfaz de llenado correcto de datos.	8	La aplicación muestra botón de confirmación de registro	9	La aplicación regresa al menú principal
	<b>Paso</b>	<b>Acción</b>																			
	1	La aplicación despliega ventana de inicio																			
	2	El paciente selecciona cualquier opción del menú																			
	3	El paciente selecciona ¿Tienes síntomas? (triaje)																			
	4	El paciente ingresa toda la información solicitada																			
	5	La aplicación muestra interfaz de llenado de datos correctos																			
	6	La aplicación agrega los datos ingresados en la Base de datos																			
	7	La aplicación Muestra interfaz de llenado correcto de datos.																			
8	La aplicación muestra botón de confirmación de registro																				
9	La aplicación regresa al menú principal																				
<b>Post Condición</b>	<ul style="list-style-type: none"> <li>• El paciente está a la espera de la llamada del Médico</li> <li>• El paciente no puede realizar otro triaje mientras sea paciente en espera de atención o en atención</li> </ul>																				

*Tabla 4 Especificación de Caso de uso Número 1, Ingresar a la aplicación*

*Fuente: Elaboración Propia*

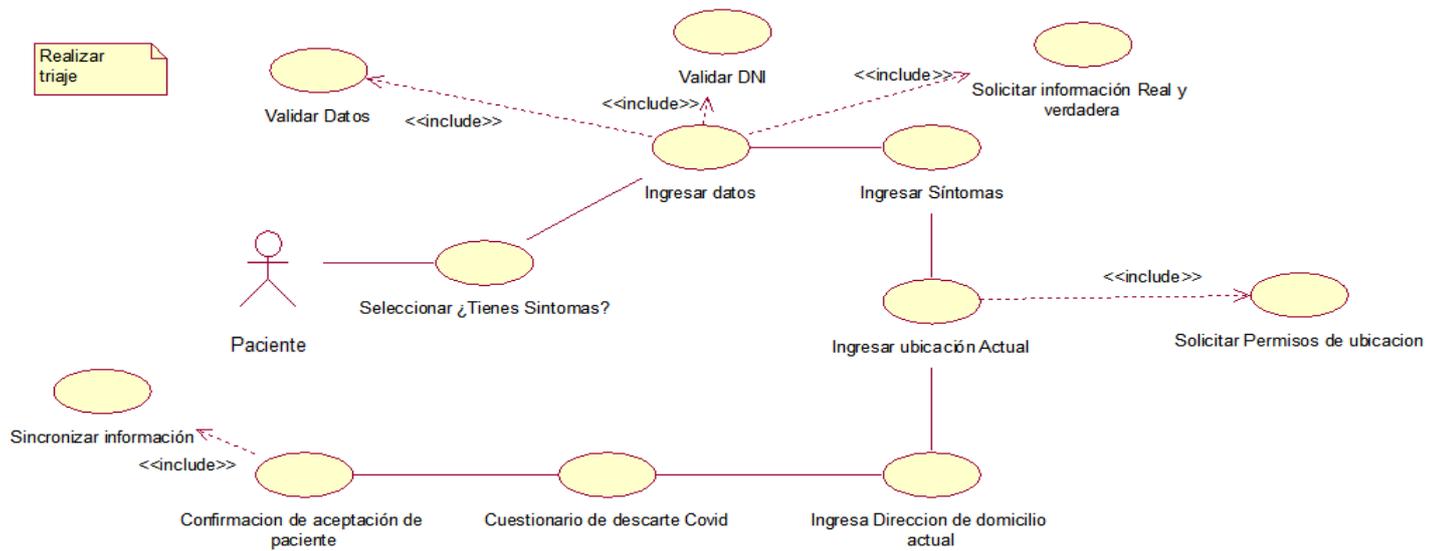


Figura 4 Caso de uso Realizar triaje

Fuente: Elaboración propia

CASO DE USO NUMERO 2																	
<b>Nombre</b>	Realizar Triaje																
<b>Descripción</b>	Registrar al paciente si considera que tiene síntomas de COVID																
<b>Actor</b>	Paciente																
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Conexión de internet</li> <li>• Tener síntomas COVID</li> </ul>																
<b>Flujo del Sistema</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El paciente elige la opción ¿Tienes síntomas?</td> </tr> <tr> <td>2</td> <td>La aplicación pide que llene los datos principales del paciente.</td> </tr> <tr> <td>3</td> <td>El paciente llena los datos solicitados</td> </tr> <tr> <td>4</td> <td>El paciente ingresa su DNI</td> </tr> <tr> <td>5</td> <td>La aplicación valida la información brindada</td> </tr> <tr> <td>6</td> <td>La aplicación pide que selecciones los síntomas del paciente</td> </tr> <tr> <td>7</td> <td>El paciente selecciona los síntomas que tiene.</td> </tr> </tbody> </table>	Paso	Acción	1	El paciente elige la opción ¿Tienes síntomas?	2	La aplicación pide que llene los datos principales del paciente.	3	El paciente llena los datos solicitados	4	El paciente ingresa su DNI	5	La aplicación valida la información brindada	6	La aplicación pide que selecciones los síntomas del paciente	7	El paciente selecciona los síntomas que tiene.
	Paso	Acción															
	1	El paciente elige la opción ¿Tienes síntomas?															
	2	La aplicación pide que llene los datos principales del paciente.															
	3	El paciente llena los datos solicitados															
	4	El paciente ingresa su DNI															
	5	La aplicación valida la información brindada															
	6	La aplicación pide que selecciones los síntomas del paciente															
7	El paciente selecciona los síntomas que tiene.																

	8	La aplicación solicita ubicación actual del paciente
	9	El paciente brinda los permisos para obtener su ubicación actual
	10	La aplicación solicita dirección de domicilio actual
	11	El paciente ingresa su domicilio Actual
	12	La aplicación solicita llenado de cuestionario para descarte COVID-19
	13	El paciente llena el cuestionario
	14	La aplicación solicita confirmación de la aceptación del usuario
	15	El paciente confirma el llenado de datos
	16	La aplicación regresa al menú principal
<b>Post Condición</b>		<ul style="list-style-type: none"> <li>• Si el paciente no accede a los permisos de ubicación, no se dará el flujo normal del proceso</li> </ul>

*Tabla 5 Especificación de Caso de uso Número 2, Realizar Triage*

*Fuente: Elaboración Propia*

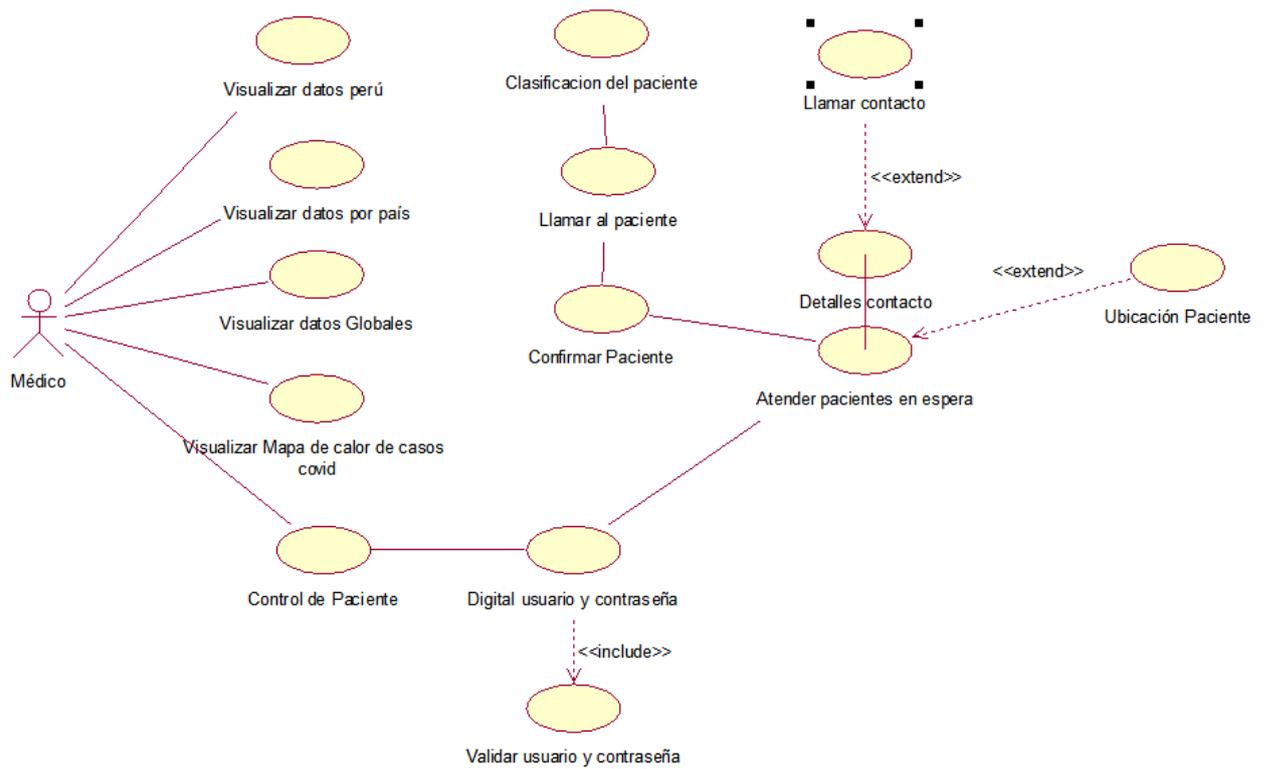


Figura 5 Caso de uso Seguimiento del paciente “sin atender”

Fuente: Elaboración Propia

CASO DE USO NUMERO 3							
<b>Nombre</b>	Realizar seguimiento de pacientes “sin atención”						
<b>Descripción</b>	Dar atención a los pacientes que aún están en espera de atención						
<b>Actor</b>	Médico						
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• CONEXIÓN A INTERNET</li> <li>• PACIENTE HABER REALIZADO TRIAJE</li> </ul>						
	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El Médico elige la opción Control del paciente</td> </tr> <tr> <td>2</td> <td>La aplicación pide que complete sus credenciales: usuario y contraseña</td> </tr> </tbody> </table>	Paso	Acción	1	El Médico elige la opción Control del paciente	2	La aplicación pide que complete sus credenciales: usuario y contraseña
Paso	Acción						
1	El Médico elige la opción Control del paciente						
2	La aplicación pide que complete sus credenciales: usuario y contraseña						

<b>Flujo del Sistema</b>	3	El médico llena los datos solicitados
	4	La aplicación valida la información brindada
	5	El médico visualiza los pacientes sin atender
	6	El médico selecciona la opción "Atender"
	7	La aplicación muestra las operaciones permitidas al médico:
	8	a) EL MÉDICO SELECCIONA LA OPCIÓN "DETALLES DEL CONTACTO"
	9	La aplicación muestra los datos del contacto y la opción de llamar al contacto
	10	b) EL MÉDICO SELECCIONA LA OPCIÓN "UBICACIÓN DEL PACIENTE"
	11	La aplicación muestra un mapa y el domicilio del paciente.
	12	c) EL MÉDICO SELECCIONA LA OPCIÓN "CONFIRMAR PACIENTE COVID"
	13	El médico llama al paciente y llena un cuestionario de preguntas
	14	La aplicación indica si el paciente tiene Covi-19, faltan datos o no tiene Covid-19
	15	<ul style="list-style-type: none"> <li>• SI ES PACIENTE COVID, LLENA UN CUESTIONARIO DE PREGUNTAS RELACIONADAS A OTRO TIPO DE ENFERMEDAD DEL PACIENTE</li> <li>• SI EL PACIENTE NO ES COVID, TERMINA EL PROCESO.</li> </ul>

	16	La aplicación clasifica al paciente según sus otras enfermedades del paciente.
	17	El médico confirma la clasificación
	18	La aplicación regresa al paso 5
<b>Post Condición</b>	<ul style="list-style-type: none"> <li>EL PACIENTE ESTARÁ EN ESTADO NO ATENDIDO HASTA QUE EL MÉDICO REALICE UNA LLAMADA Y CONTESTE EL PACIENTE.</li> </ul>	

Tabla 6 Especificación de Caso de uso Número 3, Seguimiento del paciente “sin atender”

Fuente: Elaboración Propia

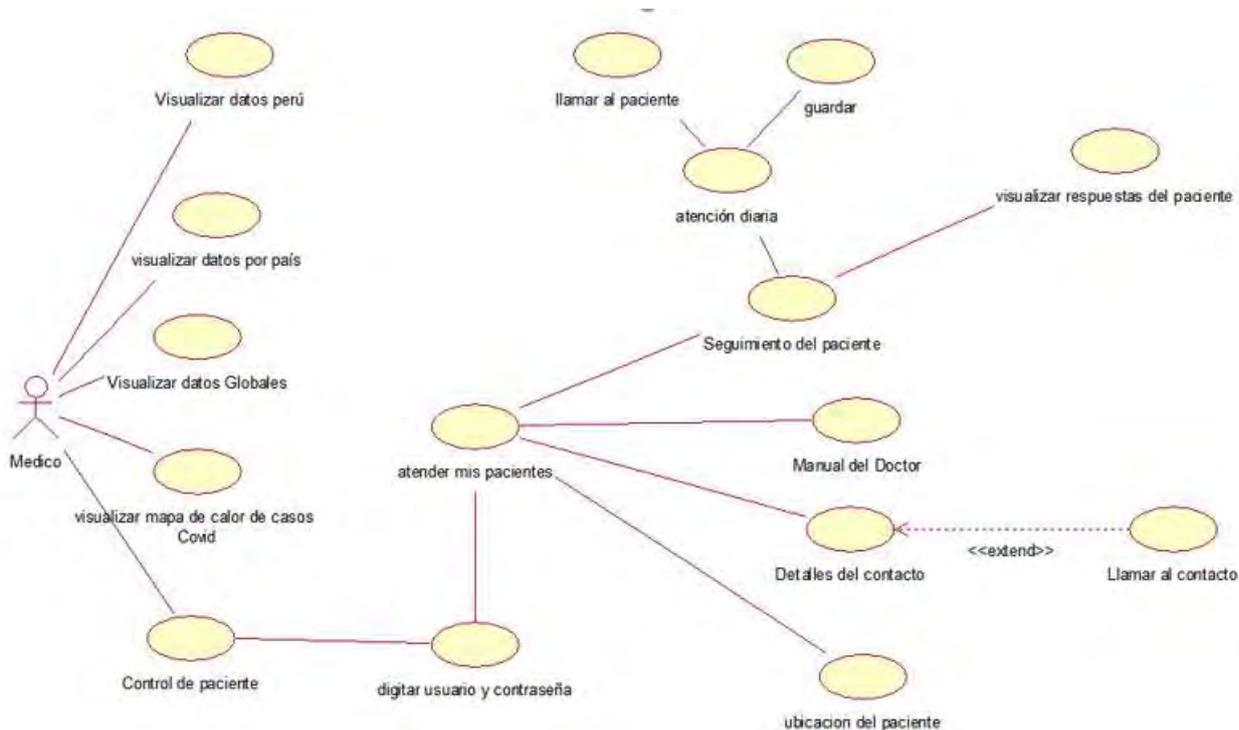


Figura 6 Caso de uso Seguimiento del paciente “Mis pacientes”

Fuente: Elaboración Propia

<b>CASO DE USO NUMERO 4</b>																											
<b>Nombre</b>	Realizar seguimiento de pacientes "Mis pacientes"																										
<b>Descripción</b>	Dar atención a los pacientes que fueron asignados al médico																										
<b>Actor</b>	Médico																										
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• CONEXIÓN A INTERNET</li> <li>• MÉDICO HABER REALIZADO LA ATENCIÓN AL PACIENTE "SIN ATENDER"</li> </ul>																										
<b>Flujo del Sistema</b>	<table border="1"> <thead> <tr> <th><b>Paso</b></th> <th><b>Acción</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El Médico elige la opción Control del paciente</td> </tr> <tr> <td>2</td> <td>La aplicación pide que complete sus credenciales: usuario y contraseña</td> </tr> <tr> <td>3</td> <td>El médico llena los datos solicitados</td> </tr> <tr> <td>4</td> <td>La aplicación valida la información brindada</td> </tr> <tr> <td>5</td> <td>El médico selecciona la opción "mis pacientes"</td> </tr> <tr> <td>6</td> <td>El médico visualiza todos los pacientes que son atendidos por el</td> </tr> <tr> <td>7</td> <td>El médico selecciona la opción "Detalles"</td> </tr> <tr> <td>8</td> <td>La aplicación muestra las operaciones permitidas al médico:</td> </tr> <tr> <td>9</td> <td>a) EL MÉDICO SELECCIONA LA OPCIÓN "DETALLES DEL CONTACTO"</td> </tr> <tr> <td>10</td> <td>La aplicación muestra los datos del contacto y la opción de llamar al contacto</td> </tr> <tr> <td>11</td> <td>b) EL MÉDICO SELECCIONA LA OPCIÓN "UBICACIÓN DEL PACIENTE"</td> </tr> <tr> <td>12</td> <td>La aplicación muestra un mapa y el domicilio del paciente.</td> </tr> </tbody> </table>	<b>Paso</b>	<b>Acción</b>	1	El Médico elige la opción Control del paciente	2	La aplicación pide que complete sus credenciales: usuario y contraseña	3	El médico llena los datos solicitados	4	La aplicación valida la información brindada	5	El médico selecciona la opción "mis pacientes"	6	El médico visualiza todos los pacientes que son atendidos por el	7	El médico selecciona la opción "Detalles"	8	La aplicación muestra las operaciones permitidas al médico:	9	a) EL MÉDICO SELECCIONA LA OPCIÓN "DETALLES DEL CONTACTO"	10	La aplicación muestra los datos del contacto y la opción de llamar al contacto	11	b) EL MÉDICO SELECCIONA LA OPCIÓN "UBICACIÓN DEL PACIENTE"	12	La aplicación muestra un mapa y el domicilio del paciente.
	<b>Paso</b>	<b>Acción</b>																									
	1	El Médico elige la opción Control del paciente																									
	2	La aplicación pide que complete sus credenciales: usuario y contraseña																									
	3	El médico llena los datos solicitados																									
	4	La aplicación valida la información brindada																									
	5	El médico selecciona la opción "mis pacientes"																									
	6	El médico visualiza todos los pacientes que son atendidos por el																									
	7	El médico selecciona la opción "Detalles"																									
	8	La aplicación muestra las operaciones permitidas al médico:																									
	9	a) EL MÉDICO SELECCIONA LA OPCIÓN "DETALLES DEL CONTACTO"																									
	10	La aplicación muestra los datos del contacto y la opción de llamar al contacto																									
11	b) EL MÉDICO SELECCIONA LA OPCIÓN "UBICACIÓN DEL PACIENTE"																										
12	La aplicación muestra un mapa y el domicilio del paciente.																										

	13	c) EL MÉDICO SELECCIONA LA OPCIÓN “MANUAL”
	14	La aplicación muestra una guía para la atención de los pacientes con COVID
	15	d) EL MÉDICO SELECCIONA LA OPCIÓN “SEGUIMIENTO DEL PACIENTE COVID”
	16	La aplicación muestra las opciones de “atender”, “ver respuestas” y “agregar día”
	17	<ul style="list-style-type: none"> <li>• EL MÉDICO LE DA LA OPCIÓN DE “ATENDER”</li> </ul>
	18	LA APLICACIÓN MUESTRA LA OPCIÓN DE LLAMAR AL PACIENTE
	19	EL MÉDICO LLAMA AL PACIENTE Y LE HACE UNA SERIE DE PREGUNTAS DE CONTROL DEL PACIENTE
	20	EL MÉDICO GUARDA LAS RESPUESTAS
	21	LA APLICACIÓN MUESTRA UNA ALERTA PARA INDICAR AL MÉDICO QUE YA NO SE PODRÁ MODIFICAR LAS RESPUESTAS
	22	<ul style="list-style-type: none"> <li>• EL MÉDICO LE DA LA OPCIÓN DE “VER RESPUESTAS”</li> </ul>
	23	EL MÉDICO VISUALIZA LAS RESPUESTAS DEL PACIENTE
	24	<ul style="list-style-type: none"> <li>• EL MÉDICO LE DA LA OPCIÓN DE “AGREGAR DÍA”</li> </ul>
	25	LA APLICACIÓN AGREGA UN DÍA MÁS PARA LA ATENCIÓN DEL PACIENTE
<b>Post Condición</b>		<ul style="list-style-type: none"> <li>• SOLO PODRÁ PASAR AL SIGUIENTE DÍA DE ATENCIÓN, MIENTRAS EL DÍA ACTUAL SEA ATENDIDO</li> <li>• UNA VEZ SE LE HAYA DADO AL PACIENTE UN SEGUIMIENTO DE 15 DÍAS, LA APLICACIÓN</li> </ul>

	MOSTRARÁ UN MENSAJE INDICANDO QUE EL PACIENTE YA PUEDE SER DADO DE ALTA
--	---

Tabla 7 Especificación de Caso de uso Número 4, Seguimiento del paciente “Mis pacientes”

Fuente: Elaboración Propia

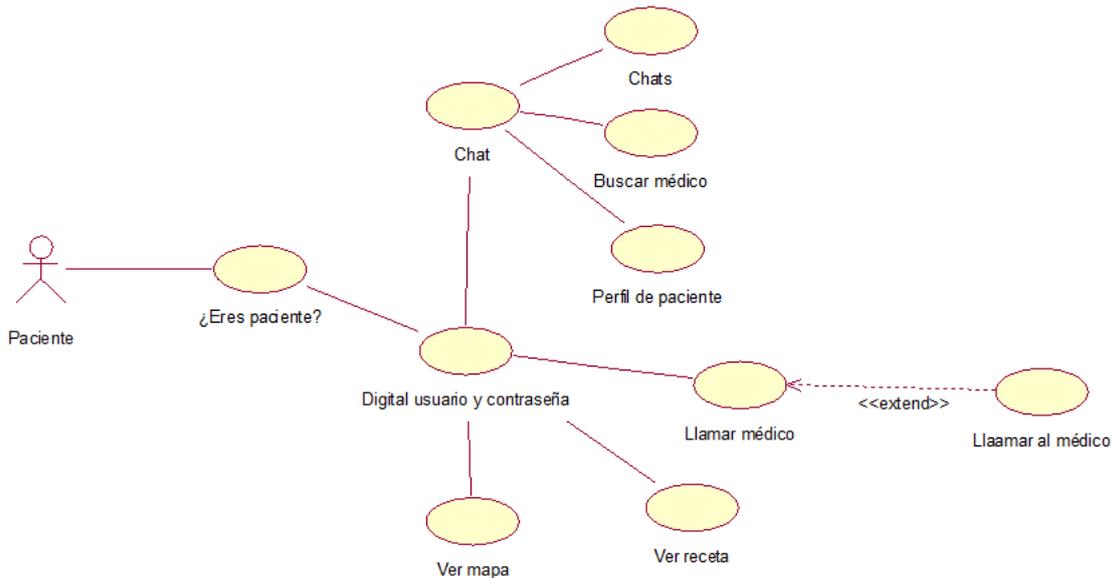


Figura 7 Caso de uso “Asistencia del paciente”

Fuente: Elaboración Propia

CASO DE USO NUMERO 5					
<b>Nombre</b>	Asistencia del paciente				
<b>Descripción</b>	Paciente ingresa a la aplicación y se comunica con el médico en caso que el paciente lo requiera				
<b>Actor</b>	Paciente				
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• CONEXIÓN A INTERNET</li> <li>• HABER REALIZADO EL TRIAJE Y DERIVADO A UN MÉDICO</li> </ul>				
	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El paciente elige la opción “¿Eres paciente?”</td> </tr> </tbody> </table>	Paso	Acción	1	El paciente elige la opción “¿Eres paciente?”
Paso	Acción				
1	El paciente elige la opción “¿Eres paciente?”				

<b>Flujo del Sistema</b>	2	La aplicación pide que complete sus credenciales: usuario y contraseña
	3	El paciente llena los datos solicitados
	4	La aplicación valida la información brindada
	6	La aplicación muestra las operaciones permitidas al paciente:
	7	El paciente selecciona la opción "Chatear con el doctor"
	8	La aplicación muestra los chats, el perfil de usuario y Buscar médico
	9	El paciente selecciona la opción "Llamar al médico"
	10	La aplicación llama al médico
	11	El paciente selecciona la opción "Ver receta"
	12	La aplicación muestra la receta del paciente por día
	13	El paciente selecciona la opción "Mostrar mapa"
	14	La aplicación muestra un mapa en la cual puede realizar la ruta entre dos puntos
	<b>Post Condición</b>	El paciente deberá dar los permisos y accesos a información personal

*Tabla 8 Especificación de Caso de uso Número 5, Asistencia del paciente*

*Fuente: Elaboración Propia*

### **3.4.2. Diagrama de clases**

Para el presente proyecto se ha manejado una base de datos NoSQL para almacenar todos los datos de la aplicación, es decir, toda la información se almacena en nodos independientes y no en tablas relacionales. Sin embargo, la lectura y escritura en los nodos realizan muchas veces a partir de información previa tomada de otros nodos, por lo que si podemos hacer un diagrama que represente la estructura de información del sistema para que tengamos una visión general de dicha información y sea más fácil de entender si solo consideramos la estructura JSON. El diagrama de clase se muestra a continuación.



### 3.5. Diseño de la base de datos

Estamos haciendo uso una base de datos NoSQL para guardar toda la información de la aplicación. El almacenamiento de datos sincronizados con Firebase permite acceder a él desde cualquier dispositivo, ya sea móvil o web. Cuando los datos se actualizan desde un dispositivo, se guardan en la nube de Google y se notifica al usuario paralelamente durante un periodo de varios milisegundos. El uso *offline* de la base de datos es otra posibilidad, ya que, si por algún motivo el usuario pierde la conexión a internet, el SDK almacenará los datos en caché localmente hasta que el usuario restaure su conexión y los datos locales se sincronizarán de manera automática.

Firebase también indica que tiene un mecanismo para mantener la seguridad de la base de datos y esto se basa en normas de seguridad que establecen y especifican que usuarios puede acceder a que datos y como se debería estructurar la base de datos

Una de sus otras mayores ventajas es que por estar alojada en la nube de Google, no hay necesidad de mantenimiento de servidores, y el SDK de Android facilita la sincronización.

Con el fin de estructurar correctamente los datos, se ha seguido las pautas provistas por Firebase en su documentación, en la cual indican que todos los datos de Firebase Realtime Database se guardan en formato JSON existente. También dejan en claro que es importante que los datos no tengan demasiados niveles de anidamiento, porque al permitir el acceso de lectura o escritura en la base de datos, también se les otorga acceso a los nodos que hay por debajo. Por esta razón es mejor mantener la estructura simple, incluso cuando haya repetición de la información.

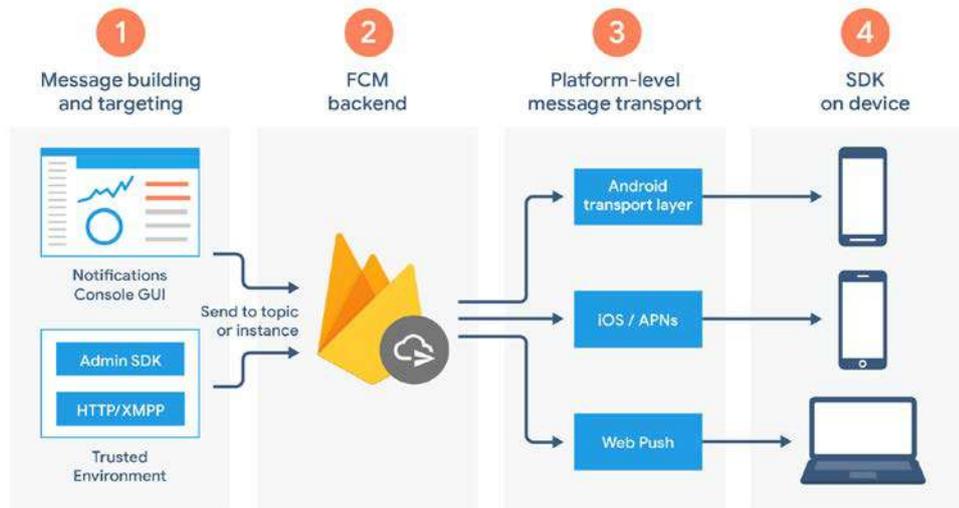
La Figura 8 muestra el diagrama de clases de la base de datos que usará la aplicación la cual fue implementada en Firebase.

### 3.6. Patrón de diseño de BackEnd de Firebase

Según la información obtenida de la documentación de Google Cloud Platform, lo que significa la creación de un *backend* para una aplicación móvil es:

- Considerar el menor uso de memoria del dispositivo.
- Conceder la sincronización de datos entre muchos dispositivos móviles.
- Administrar datos Offline (Sin conexión).
- Enviar mensajes y notificaciones.
- Minimizar costos de batería.

Para esta aplicación se ha optado por un patrón de diseño que utiliza la nube de Google para generar servicios *backend* que cumplan con los requisitos anteriores. En la figura 9 podemos ver un esquema de dicho patrón y la cual podemos describir que, si se realizan cambios desde un terminal que afecten a la base de datos, por ejemplo, Actualización de datos del usuario, registrar a un nuevo Doctor o registrar un nuevo paciente, etc. Estos se sincronizan instantáneamente y se reflejan en el resto de los dispositivos, por lo que la información se observa en tiempo real. La autenticación y la seguridad de los datos se administran de forma especial, a través de reglas declaradas mediante la interfaz de Firebase. En toda la creación de la aplicación solo se utiliza Firebase como único *backend* y eso significa que hay limitaciones de configuración, pero a cambio se obtiene muchos servicios.



*Figura 9 Arquitectura de la aplicación*

*Fuente: Google Cloud Platform, <https://firebase.google.com/docs/cloud-messaging/fcm-architecture?hl=es-419>*

Es por lo que la arquitectura de la aplicación se compone de dos partes:

- Una aplicación frontend diseñada para Android con el lenguaje Kotlin.
- Un backend en tiempo real realizado con Firebase.

En resumen, Firebase es un backend muy potente que brinda estas características:

- Sincronización de datos en tiempo real: La base de datos NoSQL de Firebase proporciona almacenamiento de datos sincronizados en tiempo real entre usuarios.
- Escalabilidad de los datos: Firebase proporciona un backend escalable horizontalmente eliminando la necesidad del mantenimiento del servidor.
- Menos código para mantener: Dado que se hace menos código en el backend, el programa puede enfocarse en una mejor experiencia del cliente y se reduce el tiempo de desarrollo.
- Gestión de usuarios: Cada solicitud enviada a Firebase contiene la información necesaria para que el dispositivo sea reconocido por Firebase y el usuario sea reconocido. Todo esto se realiza haciendo una copia de seguridad de los tokens (identificadores) del dispositivo,

además, cuando la seguridad del token se ve comprometida, se actualiza automáticamente. Firebase permite acceder a la información de acuerdo con las normas definidas en la propia base de datos.

- Almacenamiento de archivos: Las imágenes fijas de la aplicación, ya sea los íconos del menú o el icono de la aplicación, se guardan localmente. Las animaciones que dependen de que consulta se esté realizando a la base de datos, como ver la imagen de perfil de un usuario o Doctor, o las imágenes enviadas a través el chat, son guardadas en el almacenamiento de Firebase. Este servicio proporciona una URL de descarga para cada una de ellas y estas url se almacenan en la base de datos.

### **3.7. Patrón de diseño de la arquitectura de la app**

El diseño de la arquitectura utilizada para el desarrollo de software es la arquitectura monolítica ya que consiste en crear una aplicación autónoma que contenga absolutamente todas las funciones necesarias para realizar la tarea para la que está diseñada sin agregar dependencias externas a su funcionalidad.

En la arquitectura monolítica, podemos tener una serie de paquetes bien organizados y un código muy limpio, donde cada paquete puede tener una funcionalidad específica y estar desacoplado entre sí (recuerde, esto es independiente del estilo arquitectónico). Pero cuando se compila el código, todo se empaqueta en un solo programa.

A lo largo del tiempo, se les ha dado una mala fama a las aplicaciones monolíticas por las diferentes desventajas, pero una de sus mayores ventajas es que son significativamente más rápidos ya que todo se corre de manera local y no requiere consumir procesos distribuidos para realizar una tarea.

### **3.8. Diseño de la interfaz**

A continuación, se detalla el aspecto final de la aplicación tal y como ha sido implementada en este proyecto.

En la figura 10 se puede ver el splash de la aplicación, esto es lo primero que ve el usuario al momento de abrir la aplicación. En esta pantalla se muestra la información actualizada de los casos concernientes al Covid-19 en Perú, también se aprecia en la figura 11 un menú en la parte superior izquierda desplegable y la cual podemos ir seleccionando las distintas opciones que brinda la aplicación.

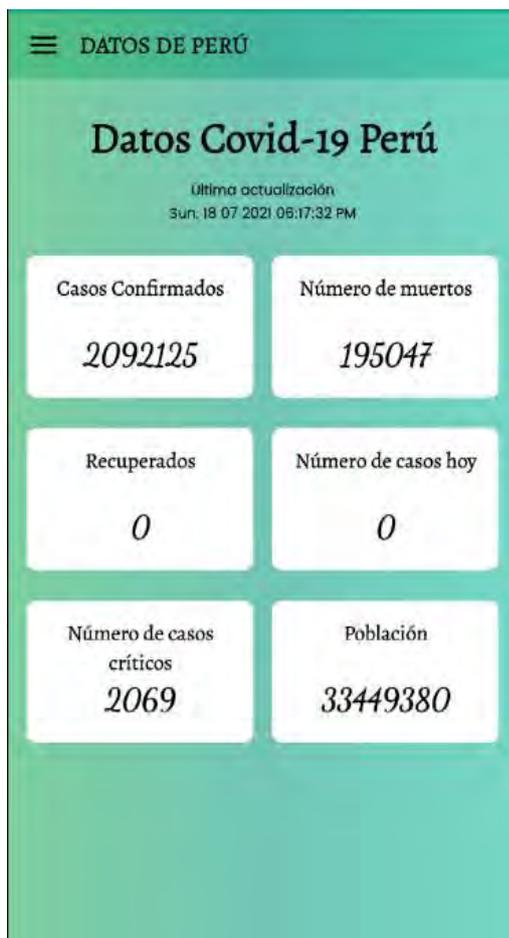


Figura 10 Pantalla de inicio del aplicativo

Fuente: Elaboración Propia



Figura 11 Pantalla del menú desplegable

Fuente: Elaboración Propia

La aplicación muestra en su menú 8 opciones en la cual detallaremos a continuación.

### **3.8.1. Opción “Datos Perú”**

Este apartado muestra como ya se dijo anteriormente, toda la información relacionada al covid1-19 según la figura 10, estos pueden ser los casos confirmados, número de muertos, recuperados, número de casos diarios, número de casos críticos y la población actual de Perú. toda esta información se obtiene de un API (application programming language) que es una interfaz que permite interactuar con código de terceros. en este caso utilizamos el api de la siguiente página <https://covid-19-apis.postman.com/> la cual es una herramienta que se ejecuta como un sitio web que usa páginas de GitHub, proporcionando un enfoque simple, de código abierto y con documentación adjunta y colecciones de Postman para ayudar a los desarrolladores y que cuentan con una actualización aproximadamente cada 2 horas.

### **3.8.2. Opción “Listado países afectados”**

Una vez ingresado a la opción “lista países afectado” dentro del menú podemos observar la figura 12 en la cual apreciamos un Recyclerview de todos los países afectados por el Covid-19 con una vista previa de todos los casos por cada país y también en la parte superior derecha el botón de Searchview para hacer una búsqueda por cada país dentro de la lista.

← Listado de países		🔍
Afghanistan		Casos 141499
Albania		Casos 132686
Algeria		Casos 153309
Andorra		Casos 14273
Angola		Casos 10805
Anguilla		Casos 112
Antigua and Barbuda		Casos 1268

Figura 12 Listado de países afectados.

Fuente: Elaboración Propia

Al hacer clic en cada país, nos muestra todos los datos relacionados al COVID-19 de cada país como se observa en la figura 13. Esto al igual que la información mostrada en “datos Perú”, también utiliza un Api de la misma página, pero con otro tipo de información JSON.

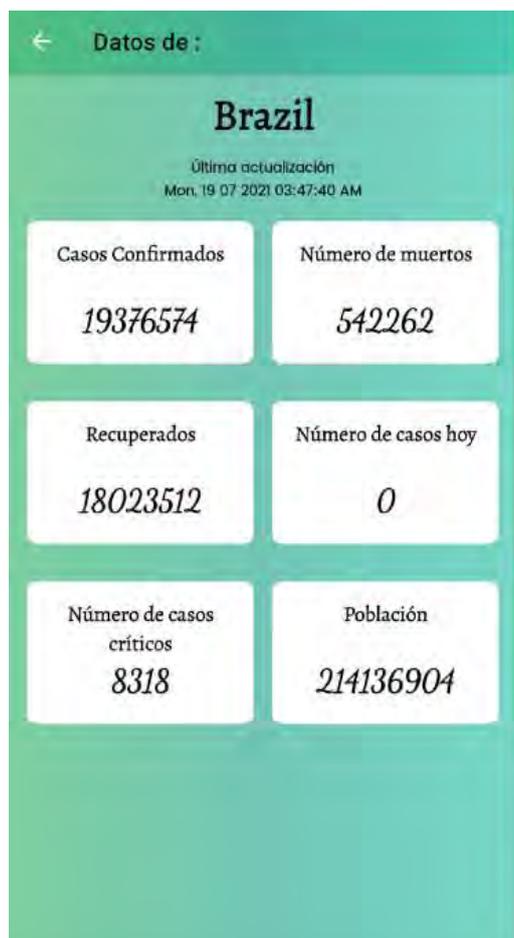


Figura 13 Información detallada sobre el COVID-19 respecto a cada país.

Fuente: Elaboración Propia

### 3.8.3. Opción “Datos Globales”

Al ingresar a la opción “datos globales” nos abre una ventana como nos muestra en la figura 15 sobre la situación actual a nivel mundial del COVID 19, la cual nos muestra información de la cantidad de casos confirmados, total de muertos y total de recuperados. Toda esta información se obtiene de una página que se describe en la visualización de la figura 14

```
[{"updated":1626810507020,"cases":192072150,"todayCases":372882,"deaths":4116967,"todayDeaths":4329,"recovered":174779461,"todayRecovered":230558,"active":13175722,"critical":81711,"casesPerOneMillion":24641,"deathsPerOneMillion":528.2,"tests":2904784741,"testsPerOneMillion":370585.1,"population":7838374349,"oneCasePerPeople":0,"oneDeathPerPeople":0,"oneTestPerPeople":0,"activePerOneMillion":1680.93,"recoveredPerOneMillion":22297.92,"criticalPerOneMillion":10.42,"affectedCountries":222}]
```

Figura 14 Api en Formato JSON sobre información del COVID-19

Fuente: API, <https://corona.lmao.ninja/v2/all>



*Figura 15 Datos Globales de personas infectadas con el Virus*

*Fuente: Elaboración Propia*

#### **3.8.4. Opción “Mapa de casos”**

La figura 16 muestra un mapa de todos los casos positivos COVID-19 con un mapa de calor sobre las zonas afectadas. Con este instrumento podemos ver cómo va el crecimiento del virus y cuáles son las zonas más afectadas.

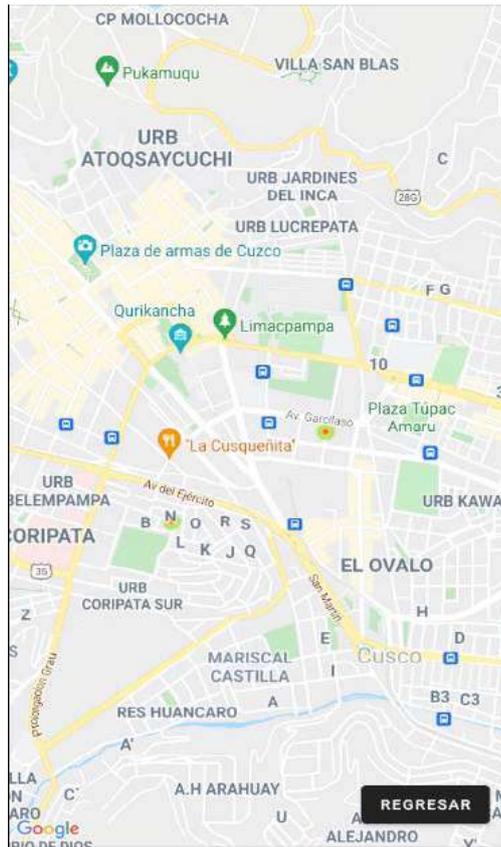


Figura 16 Mapa de casos de pacientes COVID 19

Fuente: Elaboración Propia

### 3.8.5. Opción “¿Tienes síntomas?”

Una de las principales funciones de la aplicación es este apartado. Cuando un usuario presenta síntomas, tiene sospecha de estar infectado o es un usuario que ya tiene un diagnóstico COVID 19, entonces ingresamos al ítem “¿Tienes síntomas?”. A continuación, detallaremos el proceso a seguir para registrar a un usuario.

Como primera imagen nos da una bienvenida (figura 17) para poder realizar un triaje ya sea para el usuario o un familiar cercano al usuario.



*Figura 17 Inicio del triaje para verificar los síntomas del paciente*

*Fuente: Elaboración Propia*

La figura 18 muestra un ejemplo de los datos principales solicitados al usuario como son los nombres, apellidos, DNI o documento de identidad, etc. El usuario puede hacer una consulta mediante su DNI (solo para peruanos) y de este modo llenar los campos principales automáticamente. También se le pide al usuario un correo y un número de celular, estos dos servirán más adelante para el login del paciente ya diagnosticado con COVID-19. Por último, se le pide al usuario poner los datos (Nombres, apellidos y número de celular) de la persona o contacto que estará a cargo de la atención al paciente mientras éste esté completamente aislado (figura 19).

Una vez finalizado el llenado de los datos principales en el triaje, pasamos a los síntomas del usuario (figura 20), estos síntomas son los que más se alinean al comportamiento de un paciente infectado (según consejo regional – Cusco y colegio médico del Perú).

← Datos Personales

### Ingrese sus datos personales

Nacionalidad  
Perú

Tipo de Documento  
DNI

Nro. de documento

**CONSULTAR**

Nombres

Apellidos

Celular

Edad

Correo

Figura 18 Ingresar datos del usuario

Fuente: Elaboración Propia

← Datos Personales

Día  
01

Mes  
01

Año  
2021

**Sexo**

Masculino

Femenino

**Datos de contacto de un familiar**

Nombres

Apellidos

Numero de celular

**CONTINUAR**

Figura 19 Ingresar datos del contacto

Fuente: Elaboración Propia

← Síntomas

**¿Presenta al menos unos de estos síntomas?**

Marque todas las casillas que describa su estado

- Fiebre o escalofríos
- Tos seca
- Congestión nasal
- Dolor de garganta
- Malestar general
- Dificultad para respirar
- Cansancio o debilidad
- Dolor de cabeza
- Pérdida del olfato
- Pérdida del gusto
- Dolor Muscular
- Diarrea
- Náuseas o vómitos

Figura 20 Síntomas que presenta el usuario

Fuente: Elaboración Propia

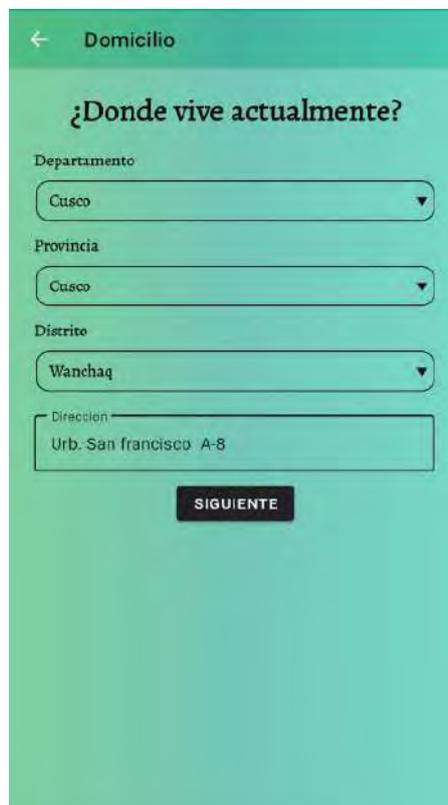
La siguiente pantalla nos muestra un mapa del Cusco, en la cual pedimos al usuario marcar la ubicación actual de su vivienda o donde reside actualmente (ver figura 21).



Figura 21 Mapa del cusco para marcar su vivienda actual

Fuente: Elaboración Propia

Al finalizar, pasamos a la siguiente pantalla en la cual pedimos al usuario indicar donde vive actualmente (figura 22), los campos a llenar son: Departamento, Provincia, Distrito y dirección actual. Como se mencionó en las limitaciones de la aplicación, solo aplicará para la provincia de Cusco.



The screenshot shows a mobile application interface for entering a user's current address. The title bar at the top is teal and contains a back arrow and the word 'Domicilio'. Below the title bar, the question '¿Donde vive actualmente?' is displayed in bold black text. There are four input fields, each with a label above it: 'Departamento' (Cusco), 'Provincia' (Cusco), 'Distrito' (Wanchaq), and 'Dirección' (Urb. San francisco A-8). Each field has a dropdown arrow on the right side. At the bottom of the form is a black button with the white text 'SIGUIENTE'.

*Figura 22 Dirección de la vivienda actual del usuario*

*Fuente: Elaboración Propia*

Por último, pedimos al usuario llenar una breve encuesta en la cual se confirma si el paciente está infectado (figura 23). Estas preguntas se almacenan en la base de datos y se las muestra al médico que atenderá al paciente en caso sea un paciente confirmado con el virus.



Figura 23 Mensaje de finalización del triaje

Fuente: Elaboración Propia

Toda la información brindada por el usuario será almacenada en la base de datos y el usuario pasará a ser un Paciente en espera de atención y el médico se estará contactando con el paciente en el transcurso de tiempo de espera. (figura 24)

A teal-colored screen titled "Cuestionario" with a back arrow on the left. The main heading is "Confirmación de paciente Covid". The first question is "¿Tuvo contacto directo con un paciente confirmado?" with radio buttons for "Si" and "No". The second question is "¿Se hizo la prueba molecular o Una Prueba rápida? ¿Que le devolvió el resultado de la prueba?" with radio buttons for "Si, Positivo", "Si, Negativo", and "No". The third question is "Tiene imágenes (Tomografía de tórax / Radiografía simple de tórax). ¿Cual fue el resultado de las imágenes?" with radio buttons for "Si, Positivo", "Si, Negativo", "Si, Pero no me explicaron el resultado", and "No". A dark grey button with the word "SIGUIENTE" in white capital letters is at the bottom.

Figura 24 Encuesta de confirmación del virus al usuario

Fuente: Elaboración Propia

### 3.8.6. Opción Control de paciente

Otra de las funciones principales del aplicativo móvil es el apartado de control de paciente. Esta función es exclusiva y única para el personal médico (Doctores), Los médicos acceden mediante un usuario y contraseña (Figura 25).



*Figura 25 Acceso al control de paciente con las credenciales del médico*

*Fuente: Elaboración Propia*

Estas credenciales se la otorga el área de desarrollo con el objetivo de validar la información del médico (Verificación del CMP). Luego de acceder mediante sus credenciales se detalla el proceso del médico para:

- Llevar a cabo la confirmación de un paciente covid-19: Todos los usuarios que acceden a la aplicación y realizan su triaje, almacenan sus datos en la base de datos y pasan a un estado “en espera de atención”. Los médicos al acceder con sus credenciales pueden visualizar en un RecyclerView a todos los usuarios registrados, con sus datos principales y ordenados por fecha de registro de triaje (Figura 26). Cuando un médico presiona el botón “Atender” pasa a un nuevo Activity y puede visualizar todos los datos que el usuario ingreso al momento de realizar su triaje (figura 27). También podemos visualizar 3 nuevos botones (Confirmar Paciente COVID, Detalles de contacto y Ubicación del paciente).



Figura 26 Pacientes que se encuentran en espera de atención

Fuente: Elaboración Propia



Figura 27 Información de un paciente en espera de atención

Fuente: Elaboración Propia

- Detalles del contacto: En ella podemos ver los datos de la persona que estará a cargo del cuidado del paciente, así como su número telefónico. (Figura 28)



*Figura 28 Información de la persona a cargo del cuidado del paciente*

*Fuente: Elaboración Propia*

- Ubicación del paciente: En esta opción visualizamos mediante un mapa de Google la ubicación de su vivienda del paciente (Figura 29).



Figura 29 Vivienda en la actualidad del paciente

Fuente: Elaboración Propia

- Confirmar Paciente COVID: Esta función lo que realiza es como su nombre lo indica, confirmar si un paciente está infectado por el virus o no (Figura 30).

Confirmación de paciente Covid

Confirmación de paciente Covid

¿Tuvo contacto directo con un paciente confirmado?

Sí  No

¿Se hizo la prueba molecular o Una Prueba rápida?  
¿Que le devolvió el resultado de la prueba?

Sí, Positivo  Sí, Negativo  No

**Paciente con Covid**  
El paciente pasará a sus pacientes en atención. ¿Desea continuar?

NO  SI

Si, Pero no me explicaron el resultado

No

Observaciones:

LLAMAR A PACIENTE SIGUIENTE

Figura 30 Confirmación del COVID-19 de un paciente

Fuente: Elaboración Propia

Accediendo al botón nos mostrará nuevamente una serie de preguntas que el médico las llenará con las repuestas del usuario realizando una llamada telefónica (Figura 31).

← Confirmación de paciente Covid

**Confirmación de paciente Covid**

¿Tuvo contacto directo con un paciente confirmado?

Si  No

¿Se hizo la prueba molecular o Una Prueba rápida ?  
¿Que le devolvió el resultado de la prueba?

Si, Positivo  Si, Negativo  No

Tiene imágenes (Tomografía de tórax / Radiografía simple de tórax), ¿Cual fue el resultado?

Si, Positivo

Si, Negativo

Si, Pero no me explicaron el resultado

No

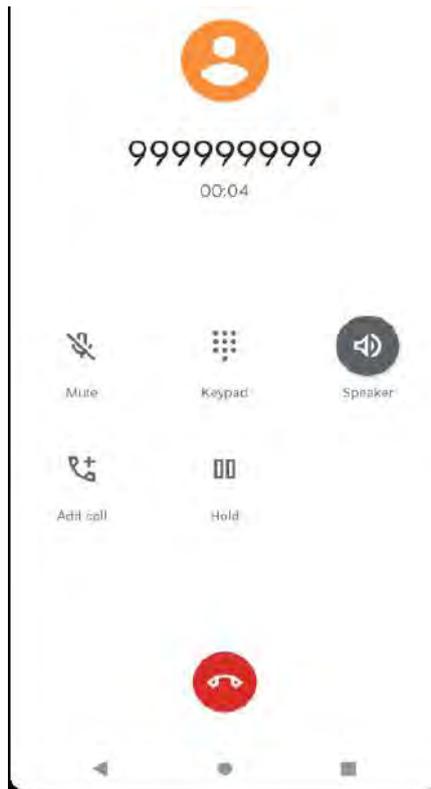
Observaciones:

**LLAMAR A PACIENTE** **SIGUIENTE**

*Figura 31 Realizar una llamada al paciente para confirmar COVID*

*Fuente: Elaboración Propia*

Una vez llenado las preguntas y presionando el botón siguiente, el sistema nos dirá automáticamente si es un paciente COVID y dará una alerta en la cual el médico confirmará la acción del sistema (Figura 32).



*Figura 32 Mensaje de alerta para clasificación de pacientes*

*Fuente: Elaboración Propia*

Para finalizar existe dos caminos, la primera es que el paciente no esté infectado por el virus y este pasa a ser un paciente atendido, pero sin COVID. La segunda es que el paciente si tiene la enfermedad, por consiguiente, pasa a otro cuestionario sobre otro tipo de enfermedades que padece el paciente y el médico le realiza las preguntas para clasificarlo de acuerdo con la gravedad del covid-19 (Figura 33 y 34).

### Clasificación de Paciente

Aparte de los síntomas descritos anteriormente por favor contestar las siguientes preguntas.

**El Paciente tiene:**

- Obesidad
- Diabetes
- Hipertensión
- Neumopatía Crónica
- Vómitos
- Convulsiones
- Desmayo
- insuficiencia respiratoria

**SIGUIENTE**

Figura 33 Clasificación de los pacientes según otras enfermedades

Fuente: Elaboración Propia

### Clasificación de Paciente

Aparte de los síntomas descritos anteriormente por favor contestar las siguientes preguntas.

**El Paciente tiene:**

- Obesidad
- Diabetes

**Paciente con Covid**

Su paciente es un paciente crítico, pasará a Pacientes derivados

OK

- Convulsiones
- Desmayo
- insuficiencia respiratoria

**SIGUIENTE**

Figura 34 Mensaje clasificación de paciente

Fuente: Elaboración Propia

Al finalizar el cuestionario, también el sistema clasifica al paciente automáticamente en 3 grupos las cuales son, “Paciente leve”, “Paciente moderado” y “Paciente crítico” (Figura 35).



*Figura 35 Clasificación del paciente en los 3 grupos*

*Fuente: Elaboración Propia*

Los médicos solo podrán dar atención a los pacientes leves y moderados. Los pacientes críticos se les deriva directamente a un centro de salud con la ayuda del médico para que él pueda realizar la llamada a emergencia y enviar un vehículo de emergencia a la vivienda del paciente.

Al realizar todo este proceso, el médico regresa al menú donde se encuentran pacientes “en espera de atención”, “Mis Pacientes” y “pacientes Atendidos”.

Dar seguimiento a sus pacientes: La segunda opción “Mis pacientes” (Figura 36), muestra un RecyclerView con una lista de pacientes que están siendo atendidos por el mismo médico que les hizo el seguimiento para determinar si el paciente está infectado o no.

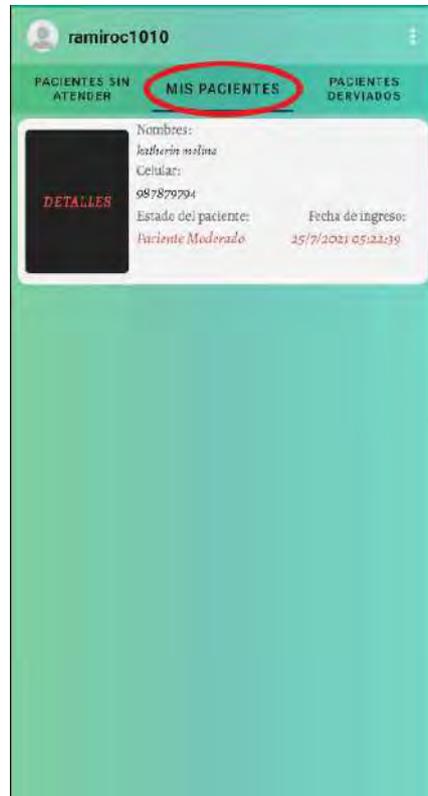


Figura 36 Clasificación en grupo “Mis Pacientes”

Fuente: Elaboración propia

Cuando un médico presiona el botón “Detalles” pasa a un nuevo Activity y puede visualizar todos los datos del usuario (figura 37), a la vez podemos visualizar 4 nuevos botones (Detalle del contacto, ubicación del paciente, Manual y Seguimiento del paciente).

- Detalles del contacto: Como ya se precisó anteriormente, en ella podemos ver los datos como el nombre y celular de la persona que estará a cargo del cuidado del paciente.
- Ubicación del paciente: Se visualiza mediante Google Maps donde vive actualmente el paciente.

← Detalles de mis paciente

Paciente  
*katherin molina*

Fecha de Nacimiento: *2021/01/01*      Estado: *Paciente Moderado*

DNI: *46779276*      Celular: *987879794*      Sexo: *Femenino*

SEGUIMIENTO DEL PACIENTE      MANUAL

DETALLES DE CONTACTO      UBICACION DEL PACIENTE

Fiebre o escalofríos

Tos seca

*Figura 37 Información principal del paciente*

*Fuente: Elaboración propia*

- Manual: En este apartado mostramos cuatro guías generales según el “Manual Rápido para el manejo del COVID 19 en el primer nivel de atención” presentado por colegio médico del Perú y el Consejo Regional del Cusco (Figura 38). A continuación, nombraremos las cuatro guías.
- Recomendaciones Para pacientes en Aislamiento Domiciliario (Figura 39).

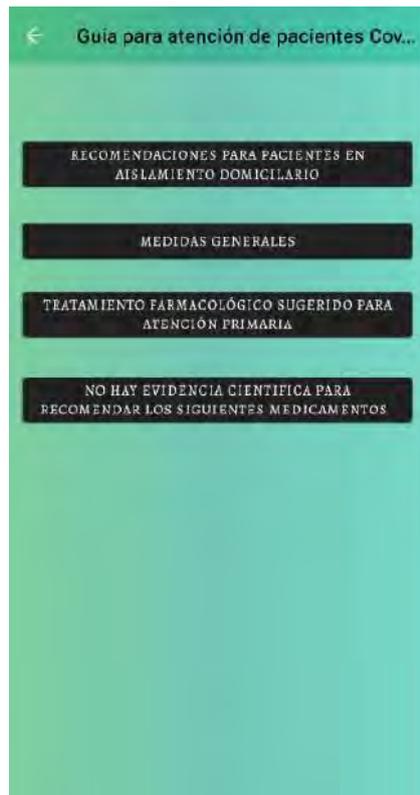


Figura 38 Manual médico para atención de pacientes con COVID

Fuente: Elaboración propia

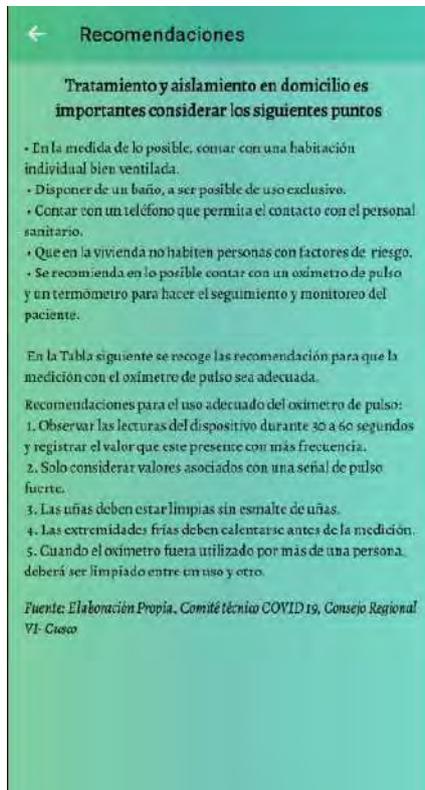
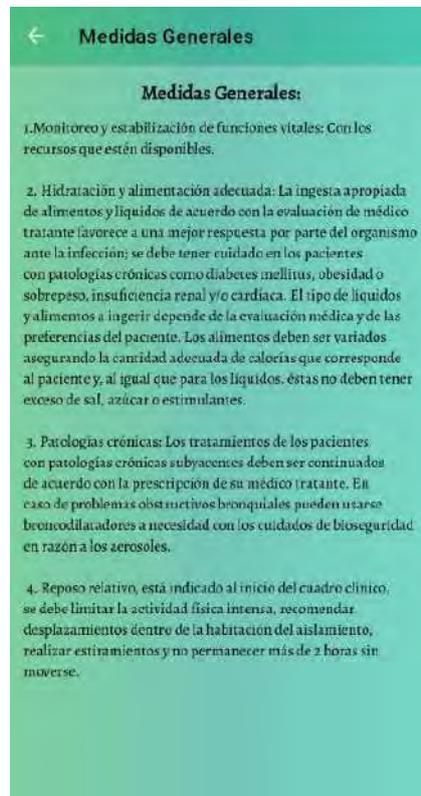


Figura 39 Manual - “Recomendaciones para pacientes en aislamiento”

Fuente: Elaboración propia

- Medidas Generales (Figura 40).
- Tratamiento Farmacológico sugerido para atención primaria (Figura 41).
- Evidencia científica para recomendar Medicamentos (Figura 42).



*Figura 40 Manual – “Medidas Generales”*

*Fuente: Elaboración Propia*

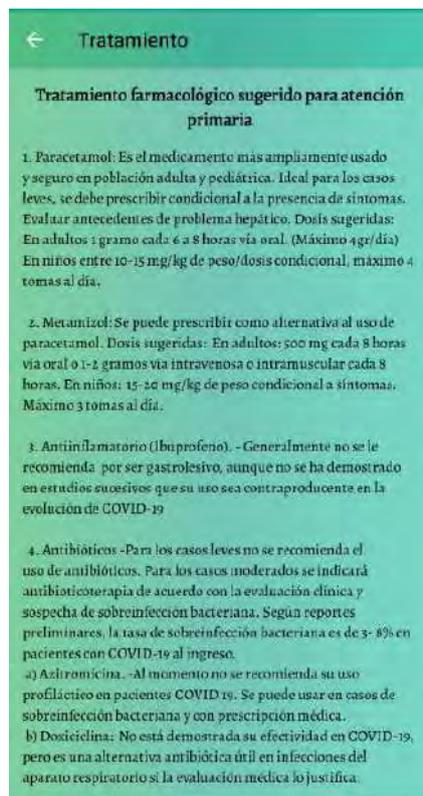


Figura 41 Manual – “Tratamiento

Fuente: Elaboración Propia

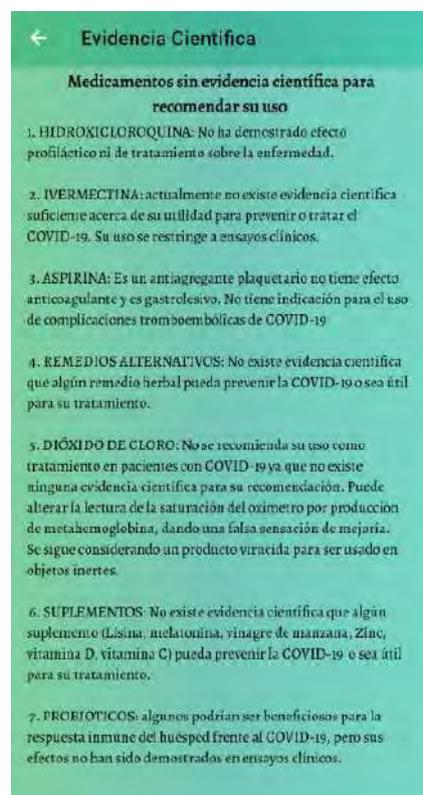


Figura 42 Manual – “Evidencia científica para recomendar medicamentos”

Fuente: Elaboración Propia

Seguimiento del paciente: Otro apartado muy importante dentro de la aplicación, en la cual podemos darle un chequeo médico diario (virtual) a los pacientes asignados a cada médico (Figura 43) y dar una receta médica si el médico ve por conveniente en cualquier día del proceso de chequeo del paciente es accediendo al seguimiento de Paciente. A continuación, daremos una descripción más profunda sobre el funcionamiento de este módulo.



*Figura 43 Botón “Seguimiento del Paciente”*

*Fuente: Elaboración Propia*

Una vez accedamos al botón “Seguimiento del Paciente” tendremos una descripción del proceso de atención del paciente. En ella visualizamos los datos principales del paciente y los días que está siendo observado por su médico (Figura 44). Dentro de cada ítem diario existe el estado de atención y tres botones, las cuales son:



Figura 44 Descripción del proceso de atención del paciente

Fuente: Elaboración Propia

Atender: El inicio de la atención diaria es accediendo a esta opción (figura 45) cuando accedamos, encontraremos una serie de preguntas que el médico le realiza al paciente a través de una llamada telefónica (Figura 46).



Figura 45 Primer paso para la atención de un paciente – “Atender”

Fuente: Elaboración Propia

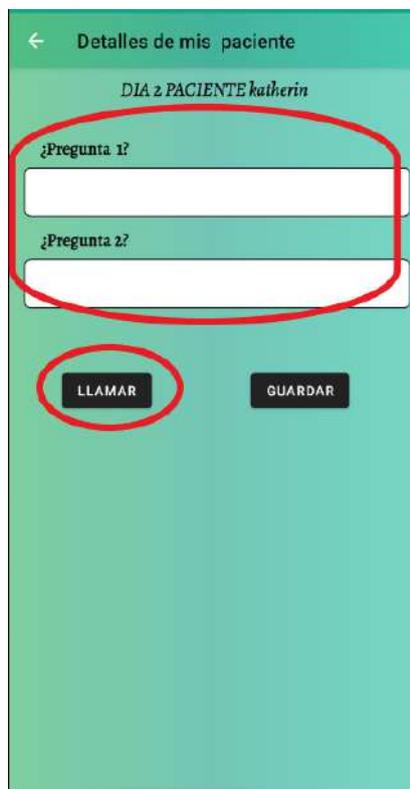
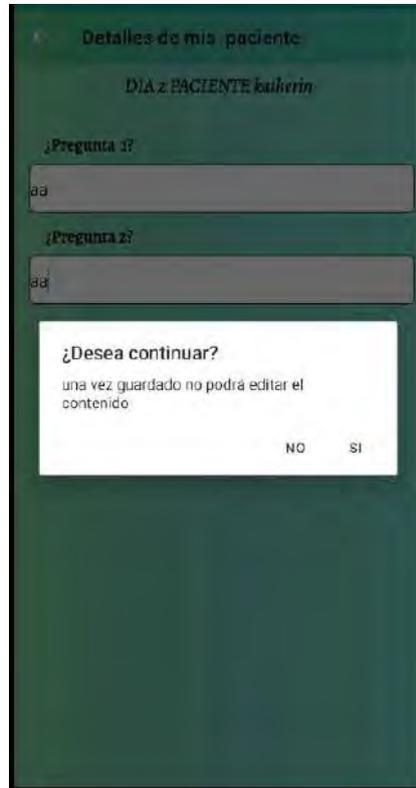


Figura 46 Preguntas establecidas para el control del paciente

Fuente: Elaboración Propia

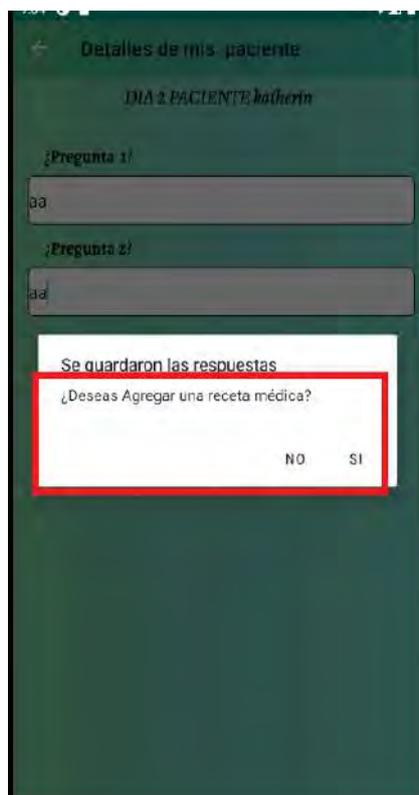
Cuando el médico finalice el chequeo solo tendrá que guardar las respuestas del paciente no sin antes mostrar un mensaje de alerta en la cual indica que una vez guardado no podrá realizar ninguna modificación a las respuestas del paciente (Figura 47).



*Figura 47 Advertencia sobre el guardado de las preguntas establecidas*

*Fuente: Elaboración Propia*

Una vez guardada la información, el sistema también le preguntará si desea agregar una receta médica para el día de la atención (Figura 48). Esta decisión es opcional.



*Figura 48 Mensajes para agregar una receta médica*

*Fuente: Elaboración Propia*

En caso de que el médico vea por conveniente que no se necesita una Receta médica, el sistema volverá a la figura 41, en caso de que si se necesite una recete médica, se abrirá una nueva ventana en la cual el médico solo ingresará el diagnóstico y la prescripción médica ya que los demás datos ya se encuentran precargados (Figura 49). La atención termina cuando el médico finaliza la receta o cuando completa las preguntas y no requiere una receta médica.

Realizar Receta Médica

DR. ramiro choque  
CMP: 101010

Paciente: Karherin molina

DNI: 45778276 Edad: 31

Diagnóstico

R.P.

GUARDAR

*Figura 49 Receta médica preestablecida para llenar solo diagnóstico y prescripción*

*Fuente: Elaboración Propia*

Receta: Esta opción solo estará habilitada cuando el médico termine la atención del día seleccionado del paciente (Figura 50), y solo mostrara información si es que el médico hizo una receta médica para el día específico, caso contrario mostrará una alerta para indicar que no se realizó una receta médica en el día seleccionado (Figura 51).

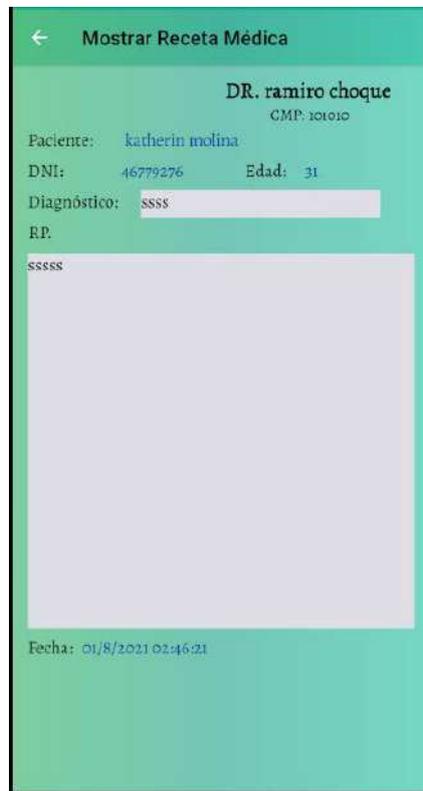


Figura 50 Receta médica guardada y solo visualizada en el día de la creación de la receta

Fuente: Elaboración Propia

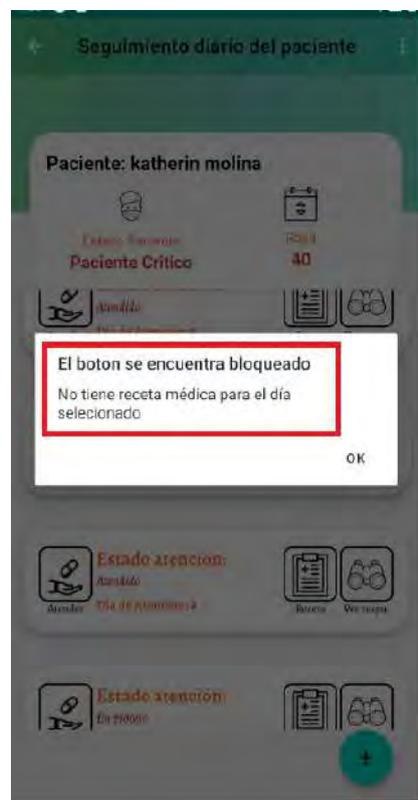
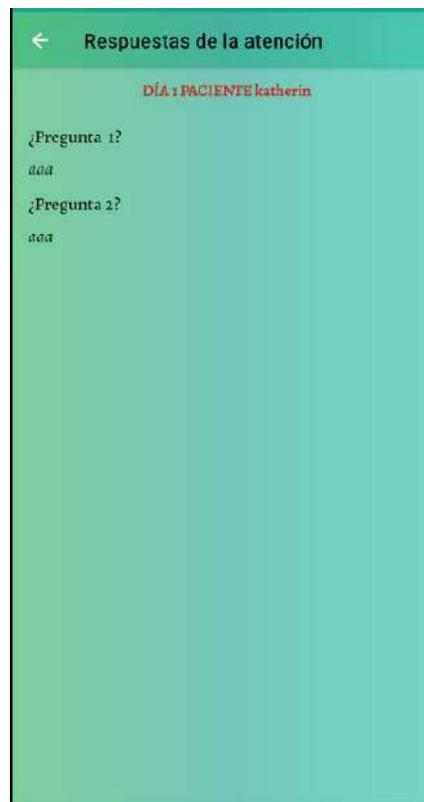


Figura 51 Alerta sobre la no existencia de receta médica para el día seleccionado

Fuente: Elaboración Propia

Ver Respuestas: Esta opción también solo estará habilitada cuando el médico termine la atención del día seleccionado del paciente. Con esto evitamos que el médico pueda cambiar o modificar las respuestas del usuario (Figura 52).



*Figura 52 Visualización de respuestas del paciente después del guardado por parte del médico.*

*Fuente: Elaboración Propia*

Pacientes Derivados: En esta opción (Figura 53), solo visualizaremos a los pacientes que fueron derivados por el médico que analizó al paciente y concluyó que debido a sus otras enfermedades o la situación en la que se encontraba el paciente, la mejor opción fue derivar a un hospital o centro de atención más cercano. El paciente seguirá contando con las mismas propiedades que un paciente leve o moderado (Figura 54).



Figura 53 Clasificación en grupo “Pacientes Derivados”

Fuente: Elaboración Propia



Figura 54 Detalles de un paciente en estado Crítico

Fuente: Elaboración Propia

### 3.8.7. Opción “¿Eres paciente?”

Es uno de los módulos más importantes dentro de la aplicación (Figura 55), una vez el usuario finalice su triaje en el módulo “¿Tienes síntomas?” y luego el médico determine si el paciente tiene el virus en el módulo “Control del Paciente”, el usuario pasa a ser un paciente y puede acceder a este módulo con sus credenciales (Figura 56). A continuación, detallaremos el proceso y/o funcionamiento del módulo.

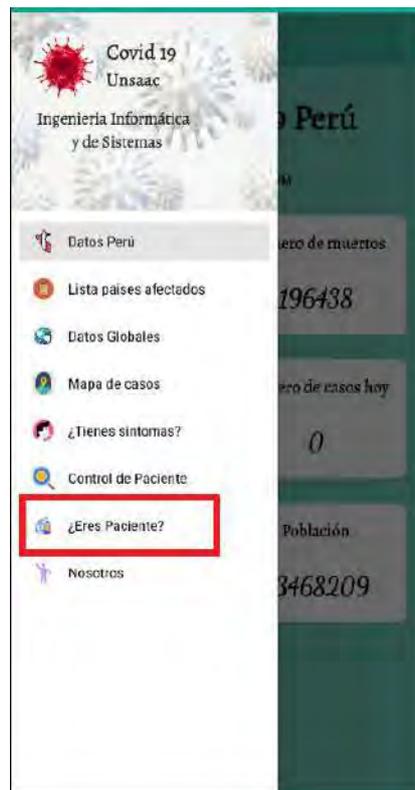


Figura 55 Módulo “¿Eres Paciente?”

Fuente: Elaboración Propia



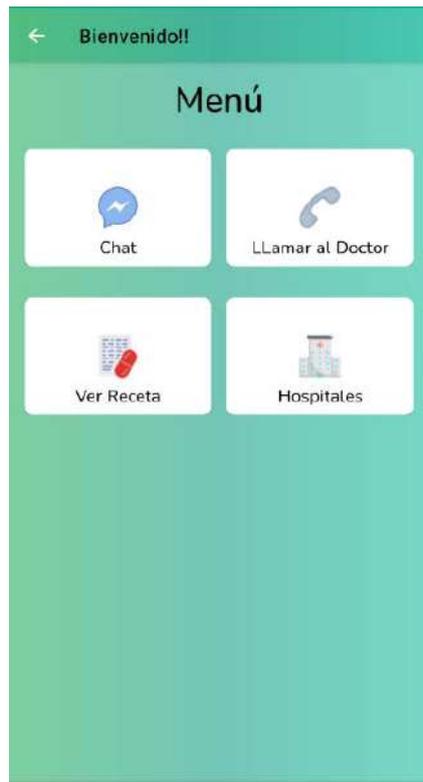
*Figura 56 Login del paciente usando sus credenciales*

*Fuente: Elaboración Propia*

Cuando el paciente ingresa con sus credenciales, accede a una nueva vista llamada "Menú" y podemos ver 4 opciones (Figura 57):

- Chat
- Llamar al Doctor
- Ver receta
- Mostrar mapa de hospitales

Detallaremos cada una de las opciones de la siguiente manera.



*Figura 57 Menú del paciente*

*Fuente: Elaboración Propia*

Chat: Este apartado como su nombre lo indica, es un chat entre el paciente y el médico que le está atendiendo (figura 58).

Dentro del módulo Visualizamos en la parte superior izquierda el nombre de usuario del paciente, en la parte superior derecha la opción de cerrar sesión y por debajo 3 opciones las cuales son: *Chats*, *Buscar* y *Perfil*. (Figura 59 y 60).

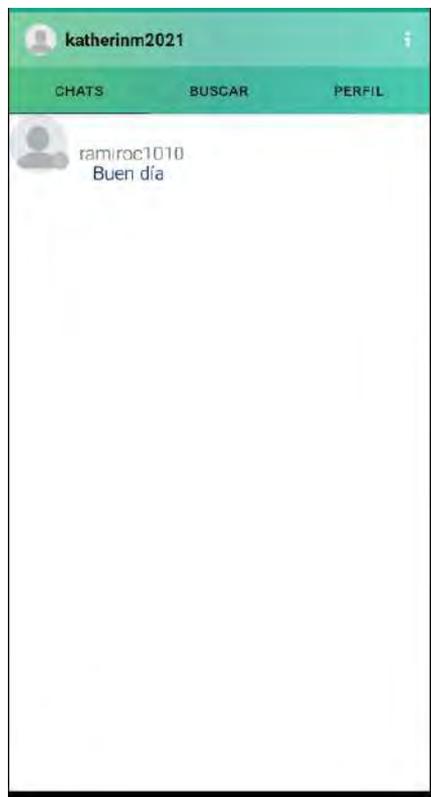


Figura 58 Vista principal del Módulo 'Chat'

Fuente: Elaboración Propia



Figura 59 Detalles del módulo 'Chat' (1)

Fuente: Elaboración Propia



*Figura 60 Detalles del módulo 'Chat' (2)*

*Fuente: Elaboración Propia*

- Chats: En ella podemos visualizar dentro de una RecyclerView la lista de conversaciones que se tiene con el médico. (Figura 61). También podemos acceder al chat y poder seguir conversando (Figura 62 y 63) o visitar el perfil del médico (Figura 64).

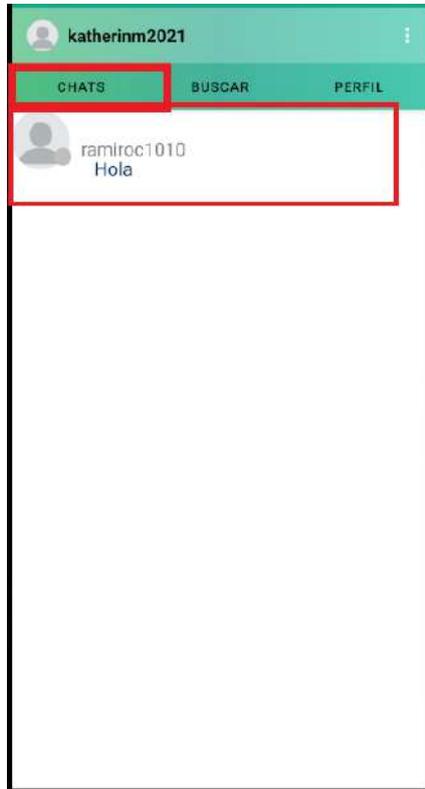


Figura 61 Lista de conversaciones y cierre de sesión

Fuente: Elaboración Propia



Figura 62 Detalles al seleccionar una conversación

Fuente: Elaboración Propia



Figura 63 Envío de mensajes mediante el chat

Fuente: Elaboración Propia

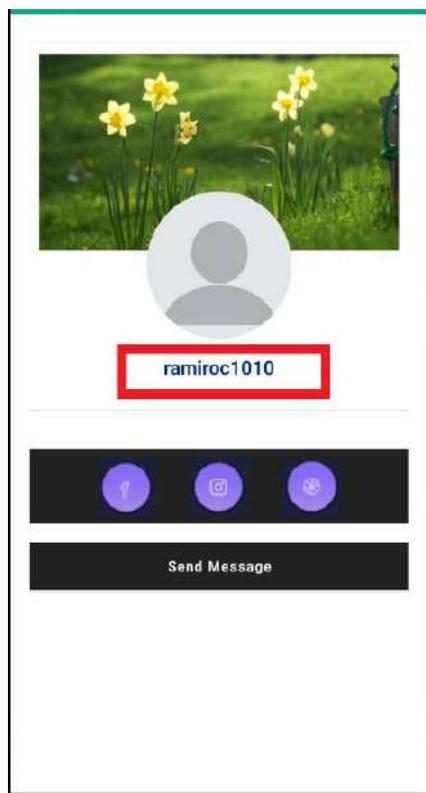
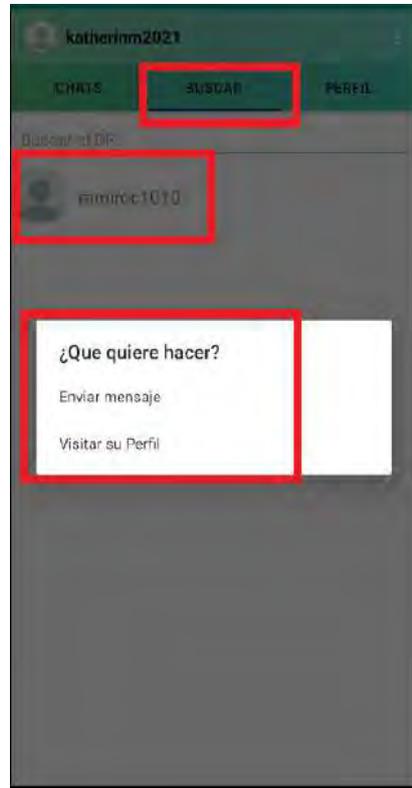


Figura 64 Ver perfil del médico

Fuente: Elaboración Propia

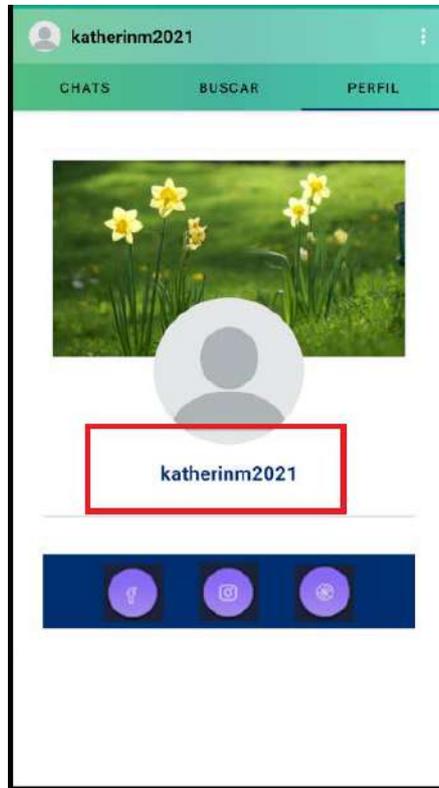
- Buscar: Aquí podemos realizar una búsqueda del médico que está atendiendo al paciente, una vez encontrado empezar una conversación o visitar su perfil (Figura 65). Al entrar a la opción “enviar mensaje”, nos cargará la pantalla como la figura 63.



*Figura 65 Búsqueda del médico que actualmente atiende al paciente*

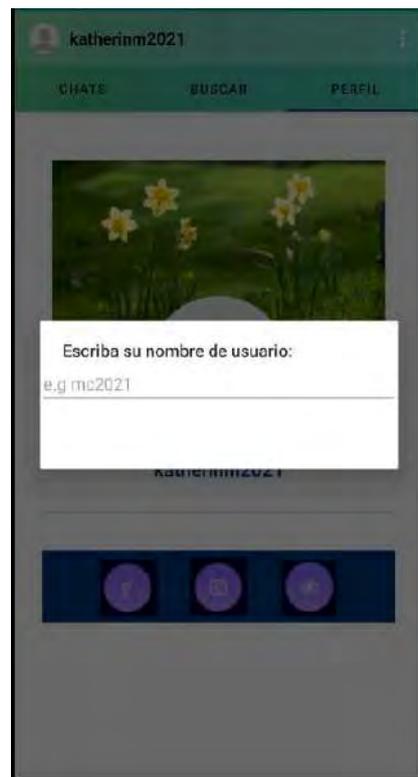
*Fuente: Elaboración Propia*

- Perfil: El paciente puede compartir información personal, por ejemplo, sus fotos de perfil y fondo, redes sociales como Facebook o Instagram (Figura 66 y 67).



*Figura 66 Perfil del paciente*

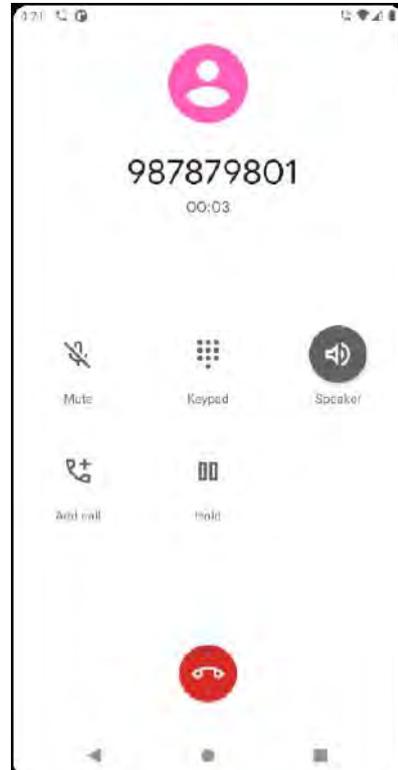
*Fuente: Elaboración Propia*



*Figura 67 Modificaciones del perfil del paciente*

*Fuente: Elaboración Propia*

- **Llamar al Doctor:** El paciente solo puede realizar la llamada a su médico asignado, la aplicación obtiene el número telefónico del médico de la base de datos. (Figura 68)



*Figura 68 Realización de llamada telefónica al médico asignado al paciente*

*Fuente: Elaboración Propia*

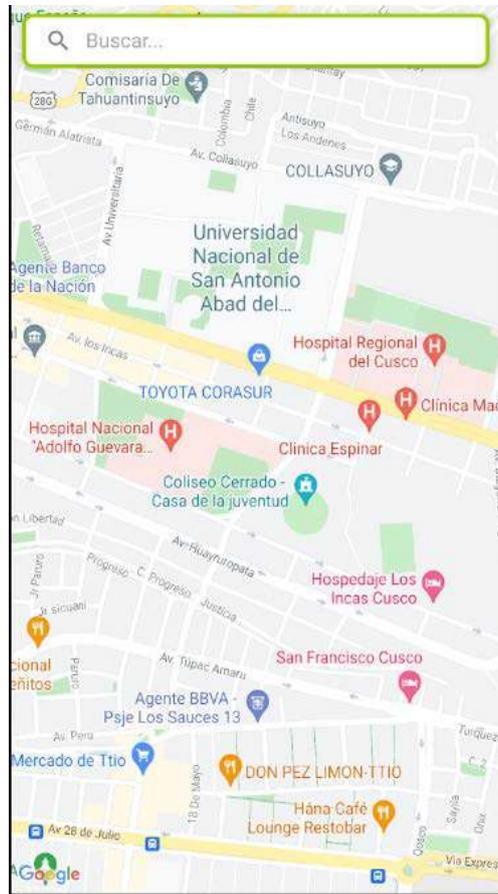
- **Ver receta:** El paciente podrá visualizar las recetas médicas asignadas por su médico, en este apartado se podrá visualizar cuando fue que se generó la receta médica y podrá guardar la receta para luego poder imprimir o mostrar en alguna farmacia. (Figura 69)



*Figura 69 Recetas del paciente*

*Fuente: Elaboración Propia*

**Mostrar mapa de hospitales:** El paciente podrá visualizar mediante un mapa de Google maps, todos los hospitales que se encuentran en la ciudad del Cusco, con el fin de poder guiar al paciente la ruta más cercana del lugar que se encuentra al hospital que desee el paciente. (figura 70).



*Figura 70 Mapa de los hospitales en la Ciudad del cusco*

**Fuente:** *Elaboración Propia*

## 4. Implementación y prueba

### 4.1. Detalles de la implementación

#### 4.1.1. Componentes de la aplicación

Los mecanismos principales de una aplicación Android son los Activities y Fragments, como se define en la guía de desarrolladores Android (Developer, 2021). Además, las actividades, aunque puedan estar relacionadas entre sí, transmitiendo información y llamándose unas tras otras, estas operan de forma independiente unas de otras. En cambio, el Fragmento representa una parte de la interfaz de usuario en una `FragmentActivity`. Se puede combinar varios fragmentos en una sola actividad para crear una interfaz de usuario de varios fragments y reutilizar el código en diversas actividades. Para crear una actividad, se debe escribir una clase Java que hereda del `Activity`. En un fragmento de nuestro código mostrado en la Figura 71, se puede visualizar cómo la clase usada hereda de `AppCompatActivity`, que es la clase base para todas las actividades que desean usar algunas características más recientes de la plataforma en dispositivos Android más antiguos.

```

2
3 import ...
25
26
27 class MainActivity : AppCompatActivity() {
28
29     private lateinit var auth : FirebaseAuth
30
31     override fun onCreate(savedInstanceState: Bundle?) {
32         super.onCreate(savedInstanceState)
33         setContentView(R.layout.activity_main)
34
35         auth = FirebaseAuth.getInstance()
36
37         val drawerLayout = findViewById<DrawerLayout>(R.id.drawerLayout)
38         findViewById<View>(R.id.imageMenu).setOnClickListener { it: View!
39             drawerLayout.openDrawer(
40                 GravityCompat.START
41             )
42         }
43         //Muestra los graficos a colores de los iconos
44         val navigationView: NavigationView = findViewById(R.id.navigationView)
45         navigationView.itemIconTintList = null
46
47         val navController = Navigation.findNavController( activity: this, R.id.navHostFragment)
48         NavigationUI.setupWithNavController(navigationView, navController)
49

```

*Figura 71 Código que muestra la Herencia de AppCompatActivity.*

**Fuente:** *Elaboración Propia*

La Guía del desarrollador también establece que las llamadas a otros componentes se realizan a través de mensajes asincrónicos, vinculando los componentes en tiempo de ejecución. Además, la información se puede transferir de una actividad a otra almacenándola en un mensaje. Un claro ejemplo de este tipo son los intents donde se puede almacenar información en una variable y pasarla a otra actividad a través de ese intent (Figura 72). En la línea 265, el mensaje se inicializa y la información se agrega a la línea 266. La actividad se inicia en la línea 267 con la función startActivity (intent)

```

261
262 fun guardar(){
263     val objetoIntent : Intent =intent
264     val idPacienteSintomas = objetoIntent.getStringExtra( name: "idPacienteSintomas")
265     val intent = Intent(applicationContext, TriajeDireccionDatosActivity::class.java)
266     intent.putExtra( name: "idPacienteSintomas", idPacienteSintomas)
267     startActivity(intent)
268 }
269 }

```

*Figura 72 Código que muestra la redirección entre Actividades*

*Fuente: Elaboración Propia*

Para que los componentes se inicialicen, deben declararse en *AndroidManifest*, y este es un archivo XML de la aplicación. Cuando instalamos una app de Android en un dispositivo, se le solicita al usuario los permisos necesarios para su funcionamiento. En este caso, se requieren diversos permisos como conexión a Internet, acceso a la ubicación, permiso para realizar llamadas, acceso a la ubicación en segundo plano, etc. para el normal funcionamiento de la aplicación. (Figura 73, línea 1020).

```
AndroidManifest.xml x
1 |<?xml version="1.0" encoding="utf-8"?>
2 |<manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 |  xmlns:tools="http://schemas.android.com/tools"
4 |  package="com.example.appcovidv5">
5 |
6 |  <uses-permission android:name="android.permission.WAKE_LOCK" />
7 |
8 |  <uses-feature android:name="android.hardware.type.watch" />
9 |
10 |  <uses-permission android:name="android.permission.INTERNET" />
11 |  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
12 |  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
13 |  <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
14 |  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
15 |  <uses-permission android:name="android.permission.WRITE_INTERNAL_STORAGE" />
16 |  <uses-permission android:name="com.androidtutorialpoint.mymapsappsdirection.permission.MAPS_RECEIVE" />
17 |  <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
18 |  <uses-permission android:name="android.permission.CALL_PHONE" />
19 |  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
20 |  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
21 |
22 |  <application
23 |    android:allowBackup="false"
24 |    android:icon="@mipmap/ic_launcher"
25 |    android:label="Covid 19"
26 |    android:roundIcon="@mipmap/ic_launcher_round"
27 |    android:supportsRtl="true"
28 |    android:theme="@style/AppTheme">
29 |    <activity android:name=".EresPaciente.VisitUserProfileActivity"></activity>
30 |    <activity android:name=".EresPaciente.ViewFullImageActivity" />
31 |    <activity android:name=".EresPaciente.EresPacienteMenuChatMessage" />
32 |    <activity
33 |      android:name=".ControlPacienteAll.ControlDePaciente"
34 |      android:label="ControlDePaciente"
35 |      android:theme="@style/AppTheme.NoActionBar" />
```

Figura 73 Parte del Código XML del AndroidManifest

Fuente: Elaboración Propia

También en la aplicación se utilizaron imágenes como recursos, estilos y colores para cada una de las pantallas de la aplicación. En la figura 74 mostramos los estilos definidos en el archivo XML y la Figura 75 mostramos los colores definidos en el XML Colors.

```
styles.xml
1 <resources>
2
3 <!-- Base application theme. -->
4 <style name="AppTheme" parent="Theme.MaterialComponents.Light.NoActionBar">
5     <!-- Customize your theme here. -->
6     <item name="colorPrimary">@color/colorPrimary</item>
7     <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8     <item name="colorAccent">@color/colorAccent</item>
9     <item name="android:statusBarColor">#17A589</item>
10
11 </style>
12
13 <style name="LoginTextInputLayoutStyle" parent="Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense">
14     <item name="boxStrokeColor">#fff</item>
15     <item name="boxStrokeWidth">2dp</item>
16 </style>
17
18
19 <style name="spinner_style">
20     <item name="android:layout_width">match_parent</item>
21     <item name="android:layout_height">wrap_content</item>
22     <item name="android:background">@drawable/gradient_spinner</item>
23     <item name="android:layout_margin">10dp</item>
24     <item name="android:paddingLeft">8dp</item>
25     <item name="android:paddingRight">20dp</item>
26     <item name="android:paddingTop">5dp</item>
27     <item name="android:paddingBottom">5dp</item>
28     <item name="android:popupBackground">#DFFFFFFF</item>
29 </style>
30
31 <style name="ToolbarTheme" parent="ThemeOverlay.AppCompat.ActionBar">
32     <item name="colorControlNormal">@color/colorWhite</item>
33     <item name="titleTextColor">@color/colorWhite</item>
34 </style>
35
36
```

Figura 74 Extracto del Código XML del Style.xml

Fuente: Elaboración Propia

```
colors.xml x
1 |<?xml version="1.0" encoding="utf-8"?>
2 |<resources xmlns:tools="http://schemas.android.com/tools">
3 |   <color name="colorPrimary">#222222</color>
4 |   <color name="colorPrimaryDark">#002d72</color>
5 |   <color name="colorAccent">#03dac5</color>
6 |   <color name="colorPrimaryText">#212121</color>
7 |   <color name="colorWhite">#FFFFFF</color>
8 |   <color name="colorGreen">#32cd32</color>
9 |   <color name="mtrl_textinput_default_box_stroke_color" tools:override="true">#fff</color>
10 |  <color name="colorTotalConfirmed">#e60000</color>
11 |  <color name="colorTotalDeaths">#999999</color>
12 |  <color name="colorTotalRecovered">#7bb974</color>
13 |  <color name="ligh_medium_gray">#f3f3f3</color>
14 |
15 |
16 |  <!-- reds -->
17 |  <color name="md_red_50">#FFEBEE</color>
18 |  <color name="md_red_100">#FFCDD2</color>
19 |  <color name="md_red_200">#EF9A9A</color>
20 |  <color name="md_red_300">#E57373</color>
21 |  <color name="md_red_400">#EF5350</color>
22 |  <color name="md_red_500">#F44336</color>
23 |  <color name="md_red_600">#E53935</color>
24 |  <color name="md_red_700">#D32F2F</color>
25 |  <color name="md_red_800">#C62828</color>
26 |  <color name="md_red_900">#B71C1C</color>
27 |  <color name="md_red_A100">#FF8A80</color>
28 |  <color name="md_red_A200">#FF5252</color>
29 |  <color name="md_red_A400">#FF1744</color>
30 |  <color name="md_red_A700">#D50000</color>
31 |
32 |  <!-- pinks -->
33 |  <color name="md_pink_50">#FCE4EC</color>
34 |  <color name="md_pink_100">#F8BBD0</color>
35 |  <color name="md_pink_200">#F48FB1</color>
36 |  <color name="md_pink_300">#F06292</color>
37 |  <color name="md_pink_400">#EC407A</color>
38 |  <color name="md_pink_500">#E91E63</color>
```

Figura 75 Extracto del Código XML de Colors.xml

Fuente: Elaboración Propia

### **4.1.2. Estructura del proyecto**

Cuando inicias Android Studio y creas un nuevo proyecto, automáticamente separa las clases de Java de los recursos que consisten en imágenes y archivos XML. Las clases de Java se dividen en 2 tipos, la parte visual o interfaz de usuario y la parte del modelo de datos que contiene la lógica de la aplicación. En la Figura 76 vemos la estructura de árbol del proyecto en Android Studio.

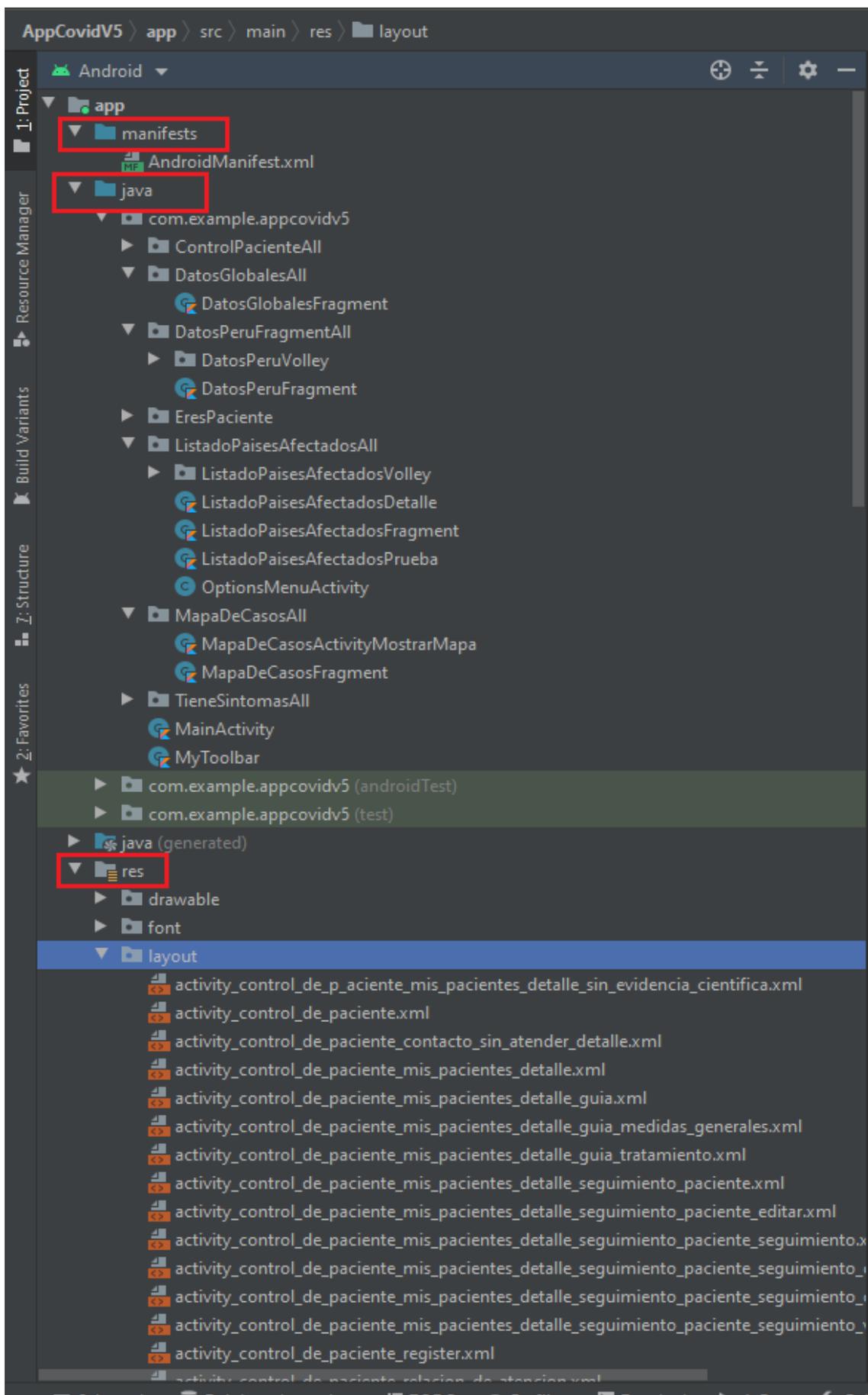


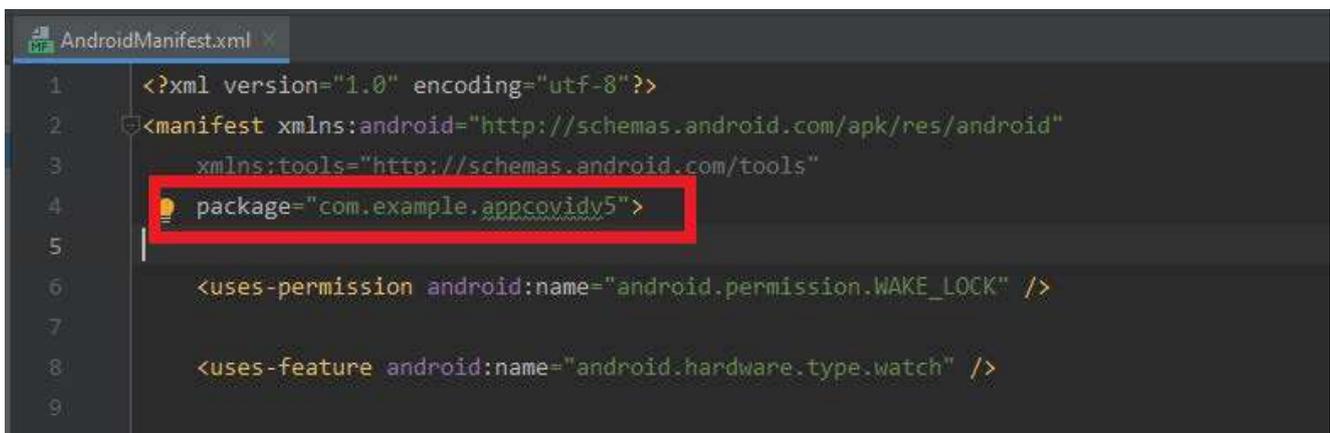
Figura 76 Extracto de la estructura del proyecto en Android Studio

Fuente: Elaboración Propia

Todo lo desarrollado está en la carpeta "app" y esta carpeta está dividida en 3 grupos respectivamente: "manifest", "java" y "res".

El primero es el archivo de configuración donde podemos aplicar las configuraciones básicas de nuestra aplicación, podemos encontrar el nombre del paquete y el ID de la aplicación (Figura 77), los componentes de la aplicación dicen que para cada componente de la aplicación generada el cual debe declarar un XML correspondiente (Figura 78), también encontramos los permisos que describimos anteriormente.

En segundo lugar, como sugiere el nombre, todo el código se desarrolla en Kotlin. Las carpetas están organizadas por módulos y están organizadas por capas según sean operaciones de control de pantalla de aplicación, capas de modelo o capas de servicio, clases que funcionan como adaptadores para mostrar elementos en *RecyclerViews*.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="com.example.appcovidv5">
5
6     <uses-permission android:name="android.permission.WAKE_LOCK" />
7
8     <uses-feature android:name="android.hardware.type.watch" />
9
```

*Figura 77 Extracto del AndroidManifest y el nombre del paquete del proyecto*

*Fuente: Elaboración Propia*

```

19 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
20 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
21
22
23 <application
24     android:allowBackup="false"
25     android:icon="@mipmap/ic_launcher"
26     android:label="Covid 19"
27     android:roundIcon="@mipmap/ic_launcher_round"
28     android:supportRtl="true"
29     android:theme="@style/AppTheme">
30     <activity android:name=".EresPaciente.ViewUserProfileActivity" />
31     <activity android:name=".EresPaciente.ViewFullImageActivity" />
32     <activity android:name=".EresPaciente.EresPacienteMenuChatMessage" />
33     <activity
34         android:name=".ControlPacienteAll.ControlDePaciente"
35         android:label="ControlDePaciente"
36         android:theme="@style/AppTheme.NoActionBar" />
37     <activity
38         android:name=".EresPaciente.EresPacienteMenuChat"
39         android:label="EresPacienteMenuChat"
40         android:theme="@style/AppTheme.NoActionBar" />
41     <activity android:name=".ControlPacienteAll.ControlDePaciente.MisPacientesDetalleSeguimientoPacienteSeguimientoDarRecetaVen" />
42     <activity android:name=".ControlPacienteAll.ControlDePaciente.MisPacientesDetalleSeguimientoPacienteSeguimientoDarReceta" />
43     <activity android:name=".EresPaciente.EresPacienteMenuMapaHospitales" />
44     <activity android:name=".EresPaciente.EresPacienteMenuReceta" />
45     <activity android:name=".EresPaciente.EresPacienteMenu" />
46     <activity android:name=".ControlPacienteAll.ControlDePacienteClases.ControlDePacienteSinAtenderConfirmarLoginDoctor" />
47     <activity android:name=".ListadoPaísesAfectadosAll.ListadoPaísesAfectadosPrueba" />

```

Figura 78 Extracto del AndroidManifest con los componentes de la aplicación

Fuente: Elaboración Propia

La tercera que es la carpeta “res” encontramos los archivos XML donde se diseñan las pantallas (carpeta “layout”), así como las imágenes, recursos dibujados en XML, fuentes para el tipo de letras, colores, estilos y temas. También encontramos una carpeta de nombre “values” con los valores de constantes al igual que la carpeta “strings” (Figura 79).

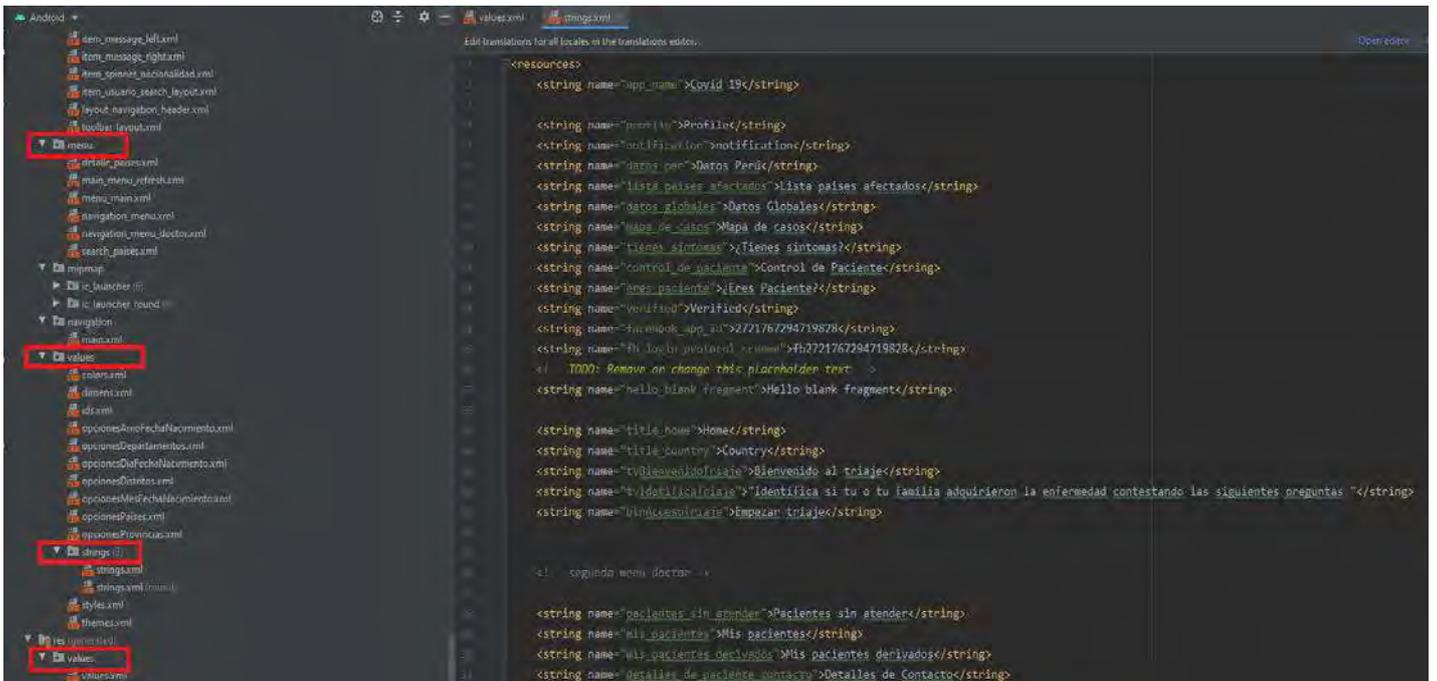
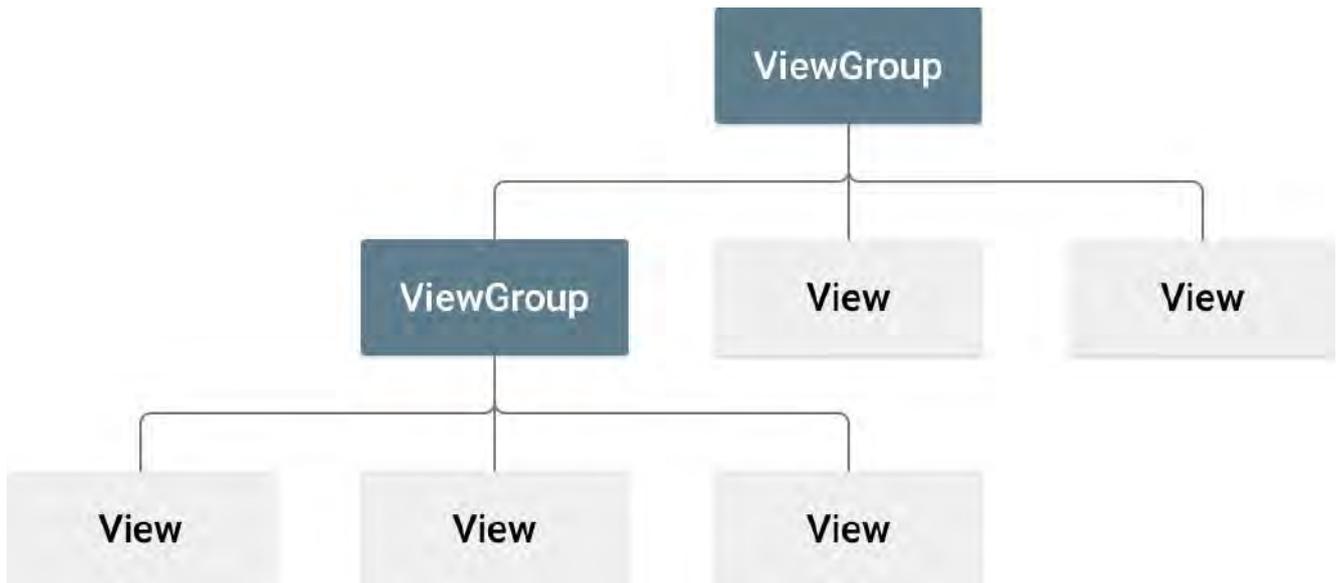


Figura 79 Extracto de la carpeta “res” y los archivos XML

Fuente: Elaboración Propia

### 4.1.3. Diseño

Según la Guía de desarrolladores de Android (Developer, 2021) explica que todos los elementos de diseño se crean utilizando una jerarquía de objetos como *View* y *ViewGroup*. Una vista normalmente muestra un elemento que el usuario puede ver e interactuar con él, en su lugar, *ViewGroup* es un contenedor invisible que contiene otras *Views* (Figura 80).



*Figura 80 Jerarquía de vista, que define un diseño de IU*

*Fuente: Elaboración Propia*

Puedes declarar un diseño de dos formas, la más sencilla es diseñar una pantalla usando un archivo XML porque Android proporciona un vocabulario XML simple que corresponde a las clases y subclases de vistas, como el ejemplo de la figura 81. La otra forma es inicializando los objetos de vista en el código de operaciones y expandiendo el árbol de vista.

```
activity_control_de_paciente_mis_pacientes_detalle.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:orientation="vertical"
7      android:layout_height="match_parent"
8      android:background="@drawable/gradient"
9      tools:context=".ControlPacienteAll.ControlDePacienteMisPacientesDetalle">
10
11      <include
12          layout="@layout/toolbar_layout"
13          android:id="@+id/toolbar_pacientes_mis_pacientes_detalle"/>
14
15      <LinearLayout
16          android:layout_marginTop="8dp"
17          android:layout_width="match_parent"
18          android:layout_height="100dp"
19          android:orientation="horizontal">
20
21          <androidx.cardview.widget.CardView
22
23              android:layout_width="160dp"
24              android:layout_height="80dp"
25              android:elevation="2dp"
26              android:layout_weight="1"
27              app:cardCornerRadius="8dp"
28              android:layout_marginLeft="8sp"
29              android:layout_marginRight="8dp">
30
31              <RelativeLayout
32
33                  android:layout_width="match_parent"
34                  android:padding="8dp"
35                  android:layout_height="match_parent">
```

Figura 81 Extracto del código que muestra el diseño de los detalles del control del paciente

Fuente: Elaboración Propia

Para poder utilizar los archivos de diseño, deben cargarse desde la actividad asociada. Esto se realiza llamado al `setContentView()` donde se pasan las referencias a los recursos(Figura 82 línea 43). Cuando se invoca una nueva Actividad, el *framework* de Android indica al método *callback* `onCreate()` donde se realiza la llamada al recurso (Figura 82 línea 41).

```
41 override fun onCreate(savedInstanceState: Bundle?) {
42     super.onCreate(savedInstanceState)
43     setContentView(R.layout.actividad_control_de_paciente_register)
44
45     EtNombre = findViewById(R.id.edittext_nombres_doctor )
46     EtApellidos = findViewById(R.id.edittext_apellidos_doctor )
47     EtCodigoCmp = findViewById(R.id.edittext_nro_de_colegiatura )
48     EtCorreo = findViewById(R.id.edittext_correo_doctor )
49     btnRegistrar = findViewById(R.id.registrar_btn_register)
50     EtCelular = findViewById(R.id.edittext_celular_doctor)
51
52     mAuth = FirebaseAuth.getInstance()
53     mDatabase = FirebaseDatabase.getInstance().reference
54
55     btnRegistrar?.setOnClickListener { view: View? -> {
56         nombres = EtNombre?.text.toString()
57         apellidos = EtApellidos?.text.toString()
58         cmp = EtCodigoCmp?.text.toString()
59         correo = EtCorreo?.text.toString()
60         celular = EtCelular?.text.toString()
61         if(nombres.isNotEmpty() && apellidos.isNotEmpty() && cmp.isNotEmpty() && correo.isNotEmpty() && celular.isNotEmpty()){
62             if(cmp.length >=6){
63                 registrarUsuario()
64             }
65             else{
66                 Toast.makeText(context this, text "Debe ingresar minimo 6 digitos de su CMP", Toast.LENGTH_SHORT).show()
67             }
68         }
69     }
70 }
71 }
```

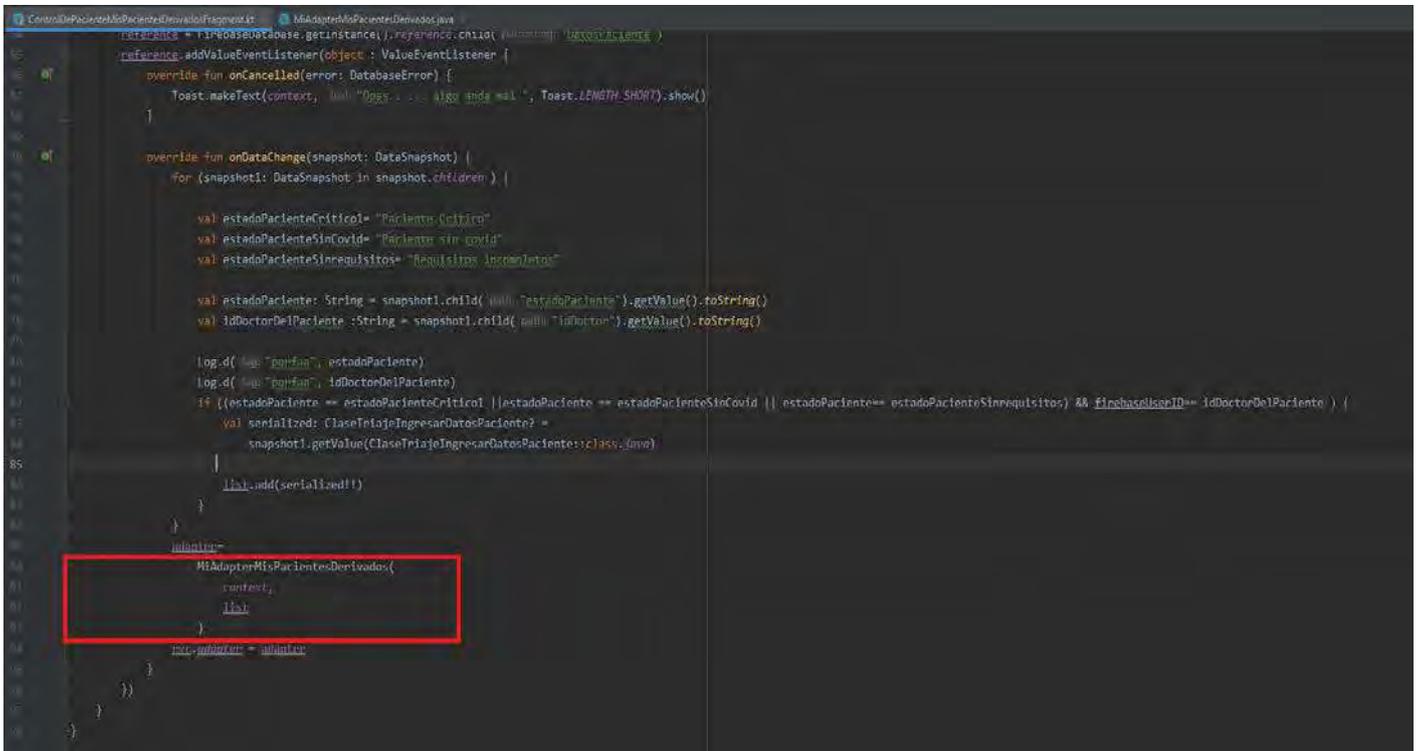
Figura 82 Extracto del código que muestra como añadir una vista a una Actividad

Fuente: Elaboración Propia

#### 4.1.4. Uso de adaptadores para rellenar los datos de una vista

Para poblar los datos de determinadas vistas, como una lista de usuarios en espera de atención, es necesario vincular la instancia de dicha lista en la Actividad a un adaptador porque este adaptador se encarga de recuperar los datos y mostrarlos. En el caso de nuestra aplicación, recuperamos datos de una base de datos de Firebase y cada clase de adaptador ha sido programada para cumplir con un diseño determinado porque las vistas son

diferentes y contienen diferentes tipos de información. Para la construcción del adaptador, los constructores se pasan como parámetros: el contexto de la aplicación y la lista de datos a adaptar (Figura 83 línea 90-92)



```

90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 83 Código que muestra como añadir un adapter a la lista

Fuente: Elaboración Propia

### 4.1.5. Eventos

De acuerdo con las pautas de Android, los eventos de usuario son donde dejamos que el usuario interactúe con nuestra aplicación, generalmente se les llama *listeners* porque están escuchando o esperando que ocurra el evento especificado. Hay muchos eventos disponibles en la clase *View*, como la tabla 9.

La figura 84 muestra un extracto del código de la aplicación que ilustra este caso.

<b>Evento</b>	<b>Método de la interfaz</b>	<b>Descripción</b>
onClick	View.OnClickListener	Llamado cuando el usuario hace clic en el elemento en cuestión
onLongClick	View.OnLongClickListener	Esto es cuando un usuario presiona y sostiene un elemento durante más de un segundo.
onFocusChange	View.OnFocusChangeListener	Se utiliza cuando el usuario se mueve de un elemento a otro en la vista.
onKey	View.OnKeyListener	Se utiliza cuando se presiona una tecla en el teclado del dispositivo.

onTouch

View.OnTouchListener

Se activa cuando se presiona, suelta o se desliza en la pantalla táctil

Tabla 9 Ejemplos de eventos para interactuar con el usuario

Fuente: Elaboración Propia

```
69 //Detalle contacto
70 btnDetalleContacto= findViewById(R.id.btnDetalleContactoPacienteMisPaciente)
71 btnSeguimientoPaciente = findViewById(R.id.btnSeguimientoPaciente)
72
73
74 //listview para sintomas
75 sintomas= findViewById(R.id.lvSintomasPaciente)
76
77 //inicializar el btn de ubicacion
78 btn_ubicacion_paciente = findViewById(R.id.btn_ubicacion_paciente_detalle_mis_pacientes)
79
80 mostrarDatosPacienteSinAtender()
81 mostrarDatosPacienteSinAtenderSintomas()
82
83 btnDetalleContacto?.setOnClickListener{ it: View!
84     val extras :Bundle = intent.extras!!
85     val value = extras.getString( key: "idCodeControlDePacientes")
86     val value3 = extras.getString( key: "tvCelularControlDePacientesSinAtenderContacto")
87
88     val intent = Intent( packageContext: this, ControlDePacienteContactoSinAtenderDetalle::class.java)
89     intent.putExtra( name: "idCodeControlDePacientes", value.toString())
90     intent.putExtra( name: "tvCelularControlDePacientesSinAtenderContacto", value3.toString())
91     startActivity(intent)
92 }
93
94
```

Inicializamos el botón

Figura 84 Código que muestra el callback de un evento

Fuente: Elaboración Propia

### **4.1.6. Notificaciones**

Una notificación es un mensaje que Android muestra al usuario incluso fuera de la interfaz de usuario (aplicación en segundo plano) para notificar al usuario sobre un recordatorio, mensaje de otra persona u otra información específica de la aplicación. Hay muchas formas de mostrarlo en un dispositivo, ya sea en la barra de estado, las notificaciones de emergencia o la pantalla de bloqueo.

En el caso de Firebase, existe Firebase Cloud Messaging (FCM), que proporciona una variedad de opciones y funciones de mensajería. Con FCM, puede enviar dos tipos de mensajes a los clientes que son los mensajes de notificación y mensajes de datos.

Si queremos que FCM (Firebase, 2021) sea responsable de mostrar mensajes en nombre de la app cliente, usamos notificaciones. Si queremos procesar los mensajes en tu app cliente, se utilizará el mensaje de datos.

### **4.1.7. Lectura y escritura de la base de datos**

En la documentación de Firebase (Firebase Realtime Database, 2021) nos indica que si queremos hacer operaciones de escritura básicas, se puede usar `setValue()` para guardar datos en la base de datos de referencia. De esta manera los datos en formato JSON se cambiarán en el destino. Cuando se obtiene un objeto JAVA personalizado, la clase de modelo solo necesita un constructor sin argumentos predeterminado, de esta manera las propiedades del objeto se asignarán a ubicaciones secundarias en los datos anidados de la base de datos. En la Figura 85, podemos ver un ejemplo de `setValue()`.

```

461 //esto es para el usuario y el chat
462 appCovidV5Usuario.child( pathString: "UsuariosChat").child(id).setValue(userHashMap)
463
464 //appCovidV5Usuario.updateChildren(userHashMap)
465
466 //esto es para lo demas
467 appCovidV5.child( pathString: "Paciente" + idPaciente).setValue(pacienteCovid)
468
469 appCovidV5.child( pathString: "Paciente" + idPaciente).child( pathString: "pacienteSeguimientoPorDias")
470 .child( pathString: "1")
471 .setValue(diasPacienteSeguimiento)
472

```

Figura 85 Código que muestra la escritura en Firebase con `setValue()`

**Fuente:** Elaboración Propia

También indica la documentación de Firebase que para recuperar datos el receptor asíncrono debe estar adjunto a una referencia de tipo *FlrebaseDatabase*, de esta manera, el receptor se activará una vez para recopilar el estado inicial de los datos y cualquier otro dato siempre que cambien en la base de datos.

Existen diferentes tipos de receptores, cada uno con varios callbacks de llamada de eventos:

- ValueEventListener: Se utiliza para mostrar cambios en el contenido de una ruta.
- ChildEventListener: Se utiliza para detectar cambios en los elementos de la lista.

Si se quiere un uso continuo, se usan los métodos *addValueEventLsitener()* o *addChildEventListener()*. La figura 86 muestra un ejemplo de lo expuesto.

```
137 fun adapterMisPacientesFragment() {
138
139     val firebaseUserID = FirebaseAuth.getInstance().currentUser?.uid
140     reference = FirebaseDatabase.getInstance().reference.child(pathName "DatosPacientes")
141     reference.addValueEventListener(object : ValueEventListener {
142         override fun onCancelled(error: DatabaseError) {
143             Toast.makeText(context, text = "Osss...!! algo anda mal", Toast.LENGTH_SHORT).show()
144         }
145
146         override fun onDataChange(snapshot: DataSnapshot) {
147             for (snapshot1: DataSnapshot in snapshot.children) {
148
149                 val estadoPaciente1 = "Paciente leve"
150                 val estadoPacienteModerado1 = "Paciente Moderado"
151                 val estadoPaciente: String =
152                     snapshot1.child(path = "estadoPaciente").getValue().toString()
153                 val idDoctorDelPaciente: String =
154                     snapshot1.child(path = "idDoctor").getValue().toString()
155
156                 if ((estadoPaciente == estadoPaciente1 || estadoPaciente == estadoPacienteModerado1) && firebaseUserID == idDoctorDelPaciente) {
157                     val serialized: ClaseTriageIngresarDatosPaciente? =
158                         snapshot1.getValue(ClaseTriageIngresarDatosPaciente::class.java)
159                     list.clear()
160                     list.add(serialized!!)
161                 } else {
162                     Toast.makeText(context, text = "No tienes pacientes", Toast.LENGTH_SHORT).show()
163                 }
164             }
165         }
166     })
167     adapter =
168         MIAdapterMisPacientes(
169             context,
170             list
171         )
172     recyclerView.adapter = adapter
173 }
```

Figura 86 Código que muestra el uso de receptores en la lectura de datos de Firebase

Fuente: Elaboración Propia

#### 4.1.8. Autenticación

De acuerdo con los documentos de Firebase (Firebase Authentication, 2021) para conocer la identidad del usuario, usamos Firebase Authentication con bibliotecas SDK y servicios preparados para autenticar al usuario. Usado con diferentes proveedores de identidad. En el caso de la aplicación, solo se utiliza autenticación basada en correo electrónico y contraseña, aunque existen muchas otras como Facebook, Twitter, GitHub, Yahoo!, Google, Telefónica o proveedores de identidad de Microsoft (Figura 87).

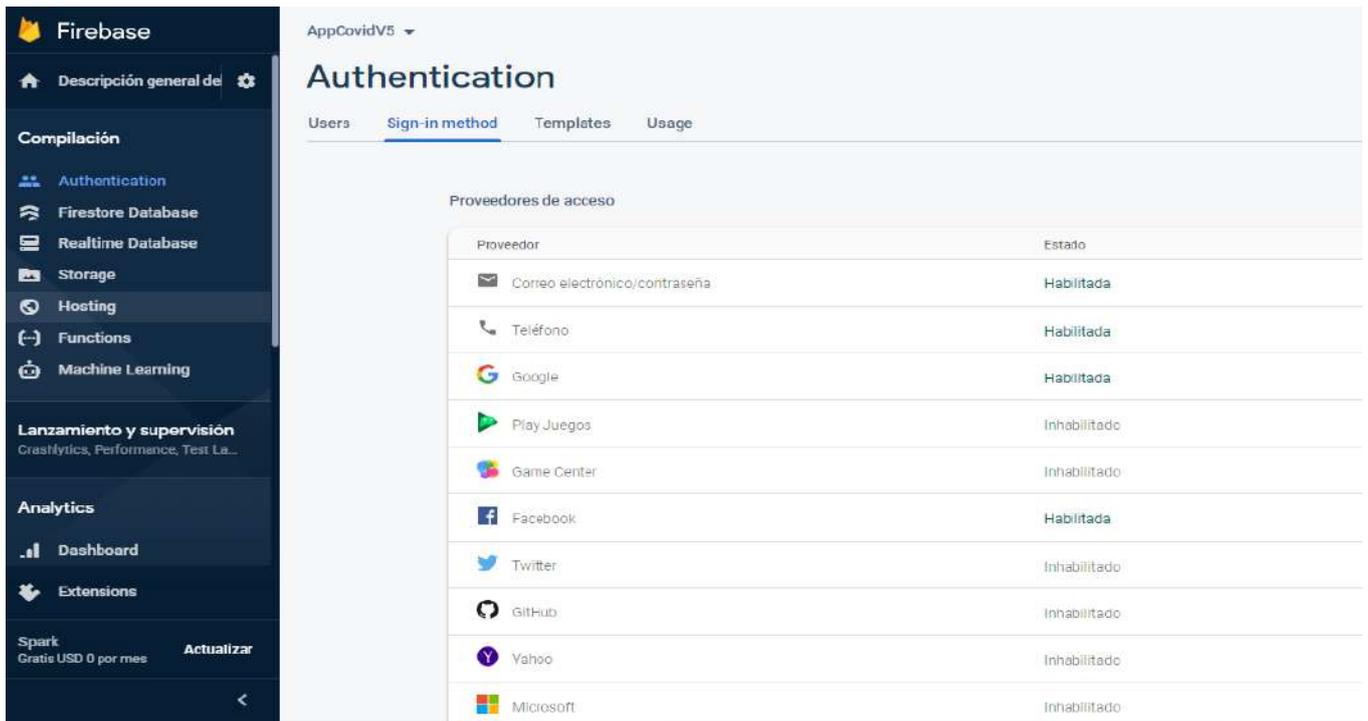


Figura 87 Ejemplos de autenticación de usuarios con Firebase

**Fuente:** Elaboración Propia

Cuando los usuarios inician sesión con sus credenciales, se pasan al SDK de Autenticación Firebase y el backend de Firebase las verifica y envía una respuesta al cliente. (Figura 88)

```

141 fun logeoMedico(){
142     val codigoColegiatura1 = codigoCorreo!!.editText?.text?.trim().toString()
143     val password = password_edittext_login!!.editText?.text?.trim().toString()
144
145     if (codigoColegiatura1.isEmpty() || password.isEmpty()) {
146         alertar()
147     }
148
149     auth.signInWithEmailAndPassword(codigoColegiatura1, password)
150         .addOnCompleteListener {task ->
151             if(task.isSuccessful){
152                 val id = auth.currentUser!!.uid
153                 reference = FirebaseDatabase.getInstance().reference
154                 reference.child( pathString: "DatosDoctor").child(id).addValueEventListener(object :ValueEventListener{
155                     override fun onDataChange(snapshot: DataSnapshot) {
156                         if (snapshot.exists()) {
157                             val estado: String =snapshot.child( path: "estadoDoctor").value.toString()
158                             Log.d( tag: "estadoDoctor", estado)
159                             if (estado == "Activo" && estado.isNotEmpty() ) {
160                                 //val extras :Bundle = intent.extras!!
161                                 //val IdPaciente = extras.getString("IdPaciente")
162                                 val intent =Intent(activity, ControlDePaciente::class.java)
163                                 intent.putExtra( name: "codigoDoctor", id)
164                                 intent.addFlags( flags: Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK)
165                                 startActivity(intent)
166                             }
167                         }
168                     }
169                 })
170             }
171         }
172     }
173 }

```

Figura 88 Código que muestra la autenticación del usuario por correo

**Fuente:** Elaboración Propia

## 4.1.9. Compatibilidad

Android funciona de tal manera que las aplicaciones más antiguas siempre son compatibles con las versiones más nuevas de Android. En el caso de la aplicación, se creó para la versión mínima del SDK 23, lo que significa que todos los dispositivos Android que utilicen el nivel de API 23 o superior (Android 5.1.1 Marshmallow y posterior) seguirán siendo compatibles con la aplicación. (Figura 89)

```
1
2  apply plugin: 'com.android.application'
3  apply plugin: 'kotlin-android'
4  apply plugin: 'kotlin-android-extensions'
5  apply plugin: 'com.google.gms.google-services'
6
7  android {
8      compileSdkVersion 30
9      buildToolsVersion "30.0.2"
10
11     defaultConfig {
12         applicationId "com.example.appcovidv5"
13         minSdkVersion 23
14         targetSdkVersion 30
15         versionCode 1
16         versionName "1.0"
17
18         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
19     }
20
21     buildTypes {
22         release {
23             minifyEnabled false
24             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
25         }
26     }
27     compileOptions {
28         sourceCompatibility JavaVersion.VERSION_1_8
29         targetCompatibility JavaVersion.VERSION_1_8
30     }
31     kotlinOptions {
32         jvmTarget = '1.8'
33     }
34 }
35 }
```

*Figura 89 Versión mínima del SDK para la compatibilidad con los dispositivos Android*

**Fuente:** *Elaboración Propia*

## 4.2. Verificación y pruebas

La validación y verificación de la aplicación se realiza durante cada uso de la aplicación con un paciente nuevo,

Se hizo la revisión del código desarrollado línea por línea y ejecutando la aplicación. También se hizo las pruebas denominadas caja negra, verificando que la aplicación cumpla con los requisitos funcionales como los no funcionales, de modo que la interfaz de usuario se adhiere a las pautas de Material Design y no crear confusión al usuario sino más bien la aplicación está desarrollada para ser utilizada de manera intuitiva. Para todos los casos en los que el usuario necesita introducir información, se verificó que todas las opciones estén completamente validadas para ingresar los valores respectivos.

Las pruebas funcionales y no funcionales se realizaron de forma manual con un dispositivo físico One Plus 6, que cuenta con la versión de Android 10.0 y también se hizo pruebas con el emulador de Android Studio para probar diferentes resoluciones.

Para realizar las pruebas se ha tomado una muestra de personas reales a las cuales se les ha brindado la aplicación a sus celulares por descarga directa para poder ingresar su información y permitir realizar las pruebas funcionales con la aplicación.

Esto ha permitido elaborar un Dataset de las personas con esos pacientes con el cual se realizará la limpieza de data para poder tratarla y visualizarla como se explicará posteriormente.

Si bien es cierto que ha sido complicado poder interactuar con las personas mencionadas por diversos factores de cuidado se ha conseguido que éstas participen de nuestra investigación por medio de la afinidad existente.

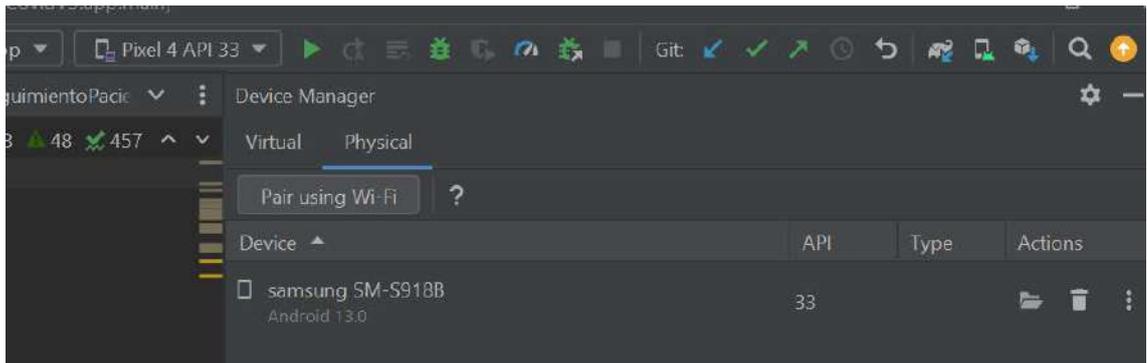


Figura 90 Transferencia de la aplicación a los móviles de los pacientes

Fuente: Elaboración Propia

### 4.3. Obtención del DataSet

Firestore almacena la información alimentada por la aplicación móvil en un archivo compuesto de formato JSON, este archivo suele tener una estructura algo complicada para su tratamiento pero existen métodos para poder hacer uso de este dataset y extraer la información que este contiene, transformarla y luego cargarla para poder usarla para la visualización de los resultados en un software como Power Bi o Excel que nos permita mostrar estos datos y dar a conocer los resultados usando estrategias de Datastorytelling.

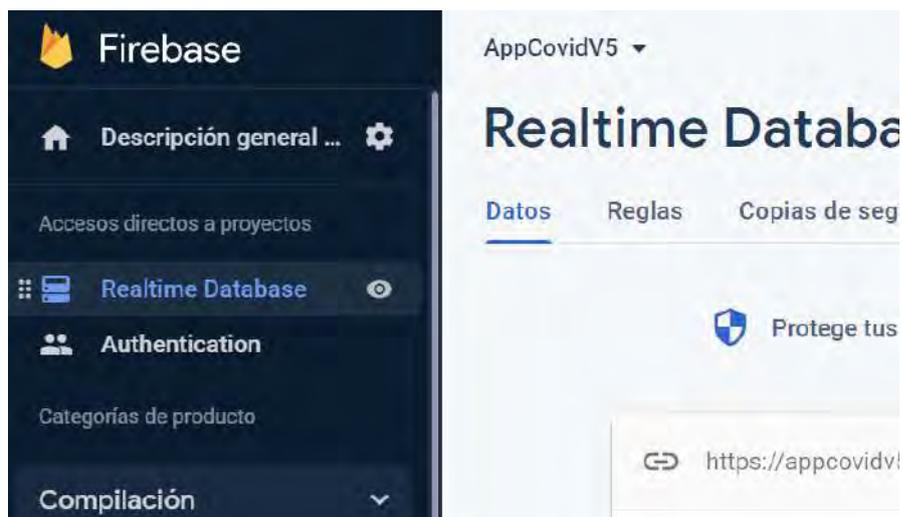


Figura 91 Realtime Database de Firebase

Fuente: Elaboración Propia

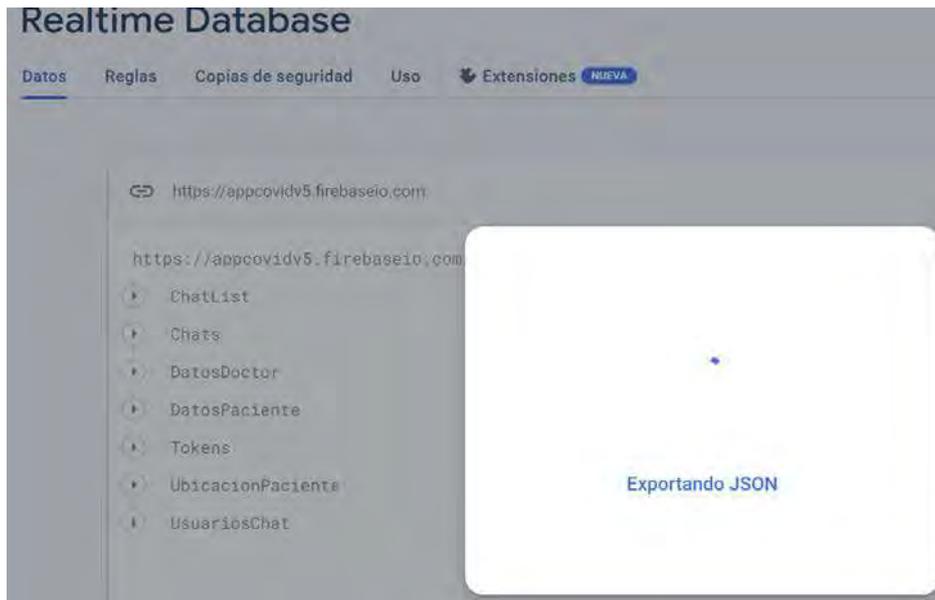


Figura 92 Exportación de JSON

Fuente: Elaboración Propia

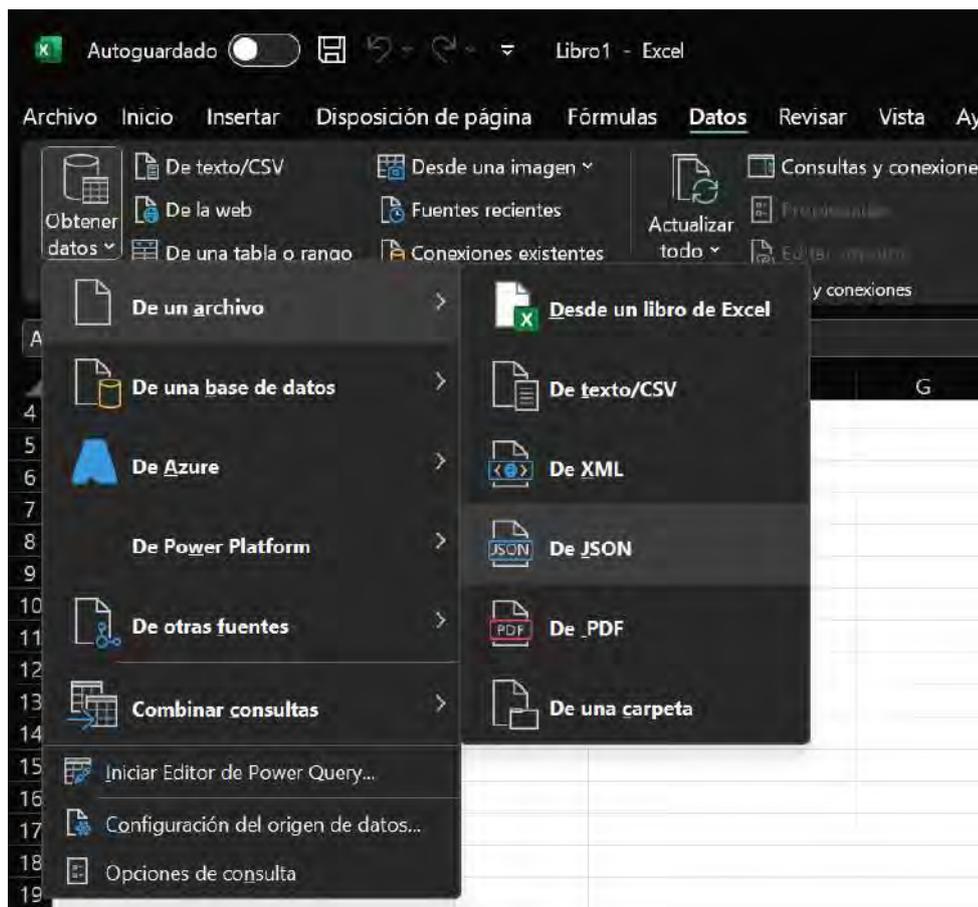


Figura 93 Importación de JSON a Excel

Fuente: Elaboración Propia

Valor_id	Valor_nombrePaciente	Valor_apellidoPaciente	Valor_estadoPaciente	Valor_estado_contado	Valor_correoPaciente
1	CAMILLA	CONCHA DUCCE	27942522	94099537	93277855
2	MIGUEL ESCOBAR	ALBERTO LLANAS	42409232	67390925	32795332
3	MARINA ALFARO	MARTIN RIVERA	43394377	89294138	69652422
4	LUIS FELIX	ARMANDO ALLENDE	47624101	93278303	93277422
5	RODRIGO	ENRIQUETA TORO	8553555	33510499	63248221
6	ANDREA	ANDREA PACHECO	36348797	98989280	43357293
7	IVAN	SEGUNDO CHICA	49968777	88797999	85667382
8	WILY MORALES	RODRIGO CLAUDIO	46353568	98268235	60248487
9	IVAN RAMIRO	IVAN RAMIRO	4137777	97950388	97248245
10	IVAN RAMIRO	IVAN RAMIRO	7139888	97790345	58255563
11	MARIA GABRIELA	IVAN RAMIRO	4754923	120988437	5463485
12	IVAN RAMIRO	IVAN RAMIRO	8148018	99892923	4948275
13	PEDRO ANTONIO	PEDRO ANTONIO	8143045	80022021	88295527
14	IVAN RAMIRO	IVAN RAMIRO	3348959	43294233	43294233
15	IVAN RAMIRO	IVAN RAMIRO	4888828	1812233	18274523
16	IVAN RAMIRO	IVAN RAMIRO	3348959	43294233	43294233

Figura 94 Limpieza de Datos usando Power Query

Fuente: Elaboración Propia

Se ha conseguido un Dataset de poco más de 54 pacientes con datos como número de teléfono, correo electrónico, así como los síntomas que presentan, entre otros datos, es con estos que podemos realizar gráficos para poder visualizar de manera sencilla por ejemplo cuál es el porcentaje de las personas que se hicieron prueba covid.

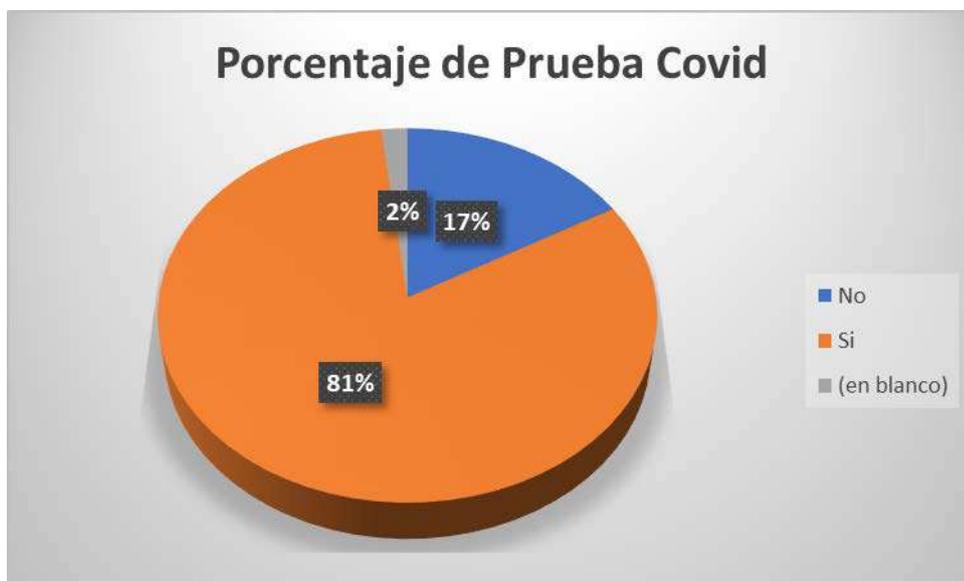


Figura 95 Porcentaje de personas que se hicieron prueba Covid

Fuente: Elaboración Propia

## 4.4. Gestión de riesgos

Para este tipo de aplicación se tienen en cuenta ciertos riesgos, y por su naturaleza no son tan numerosos. Comenzando con los servidores de aplicaciones, debe tenerse en cuenta que, dado que Firebase es un servicio de backend que brinda la capacidad de escalar y mantener los servidores, no generará una cantidad incontrolable de solicitudes. Sin embargo, existen algunos riesgos relacionados con:

- Errores en el análisis de datos: Se trató de obtener la mejor información por parte del usuario, pero la enfermedad hasta el día de la presentación del proyecto tuvo muchas variantes y por ende otro tipo de síntomas y eso no se está considerando dentro de la aplicación.
- La calidad de interfaz de usuario: Para el desarrollo de la aplicación solo se utilizó lo que Android Studio proporciona y ciertas librerías. Pero existen otras librerías que brindan una mejor diseño y calidad.

## 5. Conclusiones

- Conclusiones en base al objetivo general:

Se concluye el desarrollo de la aplicación móvil utilizando el IDE de desarrollo Android Studio, con lenguaje de programación Kotlin así como el uso de RecyclerViews, ViewBinding, Layouts, Geolocalización, librerías como Volley, Picasso, Firebase y consumo de APIs para el control de pacientes. El resultado evidencia que el médico puede dar atención a un paciente sin la necesidad del contacto físico.

- Conclusiones en base al primer objetivo específico:

Se realizó la construcción de APIs para el consumo de información en formato JSON sobre los datos relacionados a nivel mundial del COVID-19 usando las librerías como Retrofit, HttpClient, Volley y convertidores de objetos Java a JSON como la librería Gson.

- Conclusiones en base al segundo objetivo específico:

Se logró construir un dataset de 50 pacientes de los cuales 28 fueron casos leves para el Covid-19, 10 fueron casos críticos y lo demás fueron resfriados, alergias o gripe. También se logró utilizar la herramienta del GPS de los dispositivos Android para obtener la ubicación del usuario y mostrar en un mapa de calor las zonas más afectadas por el Covid-19 en la ciudad del Cusco.

- Conclusiones en base al tercer objetivo específico:

Se realizó la implementación de un chat para la comunicación entre el paciente y el médico a cargo mediante las librerías de Firebase y así el paciente informar sobre cualquier molestia, realizar consultas o indicar su estado al médico.

- Conclusiones en base al cuarto objetivo objetivo específico:

Se implementó el uso de llamadas con nuestra propia interfaz de usuario en lugar de usar la interfaz predeterminada de la app del celular. El framework de Android contiene un paquete llamado `android.telecom`, este contiene clases que facilitan la implementación.

## 6. Recomendaciones

- Se recomienda utilizar una arquitectura más robusta y escalable como la arquitectura limpia (Clean Architecture) y un patrón de diseño MVVM, dado que la aplicación está realizada con una arquitectura monolítica, Todo ello para que cualquier modificación del código o ingreso de nueva funcionalidad sea más sencillo y no requiera reestructurar nuevamente la lógica de negocio.
- Se recomienda usar la plataforma denominada HUAWEI AppGallery para las notificaciones a dispositivos Huawei ya que esta tiene sus propias librerías que conllevan a la creación de tokens para la obtención de notificaciones. las notificaciones PUSH están realizadas netamente por Firebase para dispositivos Android, pero no se consideró para el caso de los dispositivos Huawei que también utilizan el sistema operativo Android, pero no utilizan la plataforma de Google.
- Se recomienda para la monetización de la aplicación, se debe realizar la pasarela de pagos mediante la implementación de los SDKs ya desarrollados por empresas de desarrollo de software como Niubiz o Izipay y juntamente con su documentación respectiva.
- Se recomienda para mayor seguridad se debe implementar el uso de biometría facial y así evitar el introducir un usuario y contraseña ya que estas pueden ser filtradas mediante filtraciones de datos o recopilación de datos personales, con ello garantizar que la persona que intenta acceder a la aplicación, realizar una transacción o acceder a un servicio es real.
- Se recomienda la ampliación de parámetros e ingresar nuevos síntomas para la elaboración de nuevos filtros para enfrentar las nuevas variantes del Covid-19 así como la creación de un nuevo aplicativo móvil que ayude a combatir nuevas enfermedades.

- Se recomienda realizar modificaciones al algoritmo de detección del COVID-19, incluir más variables que contenga datos de nuevos síntomas y así obtener con más precisión sobre si un paciente tiene o no Covid-19.
- Se propone utilizar algoritmos más avanzados y nuevos servicios web para la determinación y confirmación de los datos personales que brinda el usuario y así obtener un *dataset* más robusto y confiable.

## 7. Bibliografía

- (THL), I. F. (11 de Setiembre de 2020). *Descarga la app Koronavilkku*. Obtenido de Descarga la app Koronavilkku: <https://koronavilkku.fi/en/>
- Android. (2021). *RecyclerView, Android: Listas dinámicas usando*. Obtenido de <https://programacionymas.com/blog/listas-dinamicas-android-usando-recycler-view-card-view>
- Bedoya Suárez, A. G., & Laurente Rodríguez, C. C. (Noviembre de 2014). *Análisis, Diseño e Implementación de un sistema de gestión del servicio de ambulancias para la atención de emergencias de una empresa aseguradora en Lima*. Lima, Lima, Perú.
- Bravo Bustillos, R. V. (10 de Agosto de 2020). *Manual del Operador en Central de Urgencias y Emergencias Médicas SAMU*. Obtenido de <https://www.diresahuanuco.gob.pe/portal/cpced/documentos/directivas/>
- Cloud Storage para Firebase, G. (2021). *Cloud Storage para Firebase*. Obtenido de <https://firebase.google.com/docs/storage?hl=es-419>
- Cusco, C. r. (septiembre de 2020). *MANUAL RÁPIDO PARA MANEJO DEL COVID 19 EN EL PRIMER NIVEL DE ATENCIÓN*. Obtenido de [https://docs.google.com/document/d/1pX3db34n8pcb-plFpfx3dhSelObKL1\\_m/edit](https://docs.google.com/document/d/1pX3db34n8pcb-plFpfx3dhSelObKL1_m/edit)
- Developer, G. (2021). *Android Studio*. Obtenido de <https://developer.android.com/?hl=es-419>
- Firebase Authentication, F. (2021). *Firebase Authentication*. Obtenido de <https://firebase.google.com/docs/auth?hl=es-419>
- Firebase Realtime Database, F. (2021). *Firebase Realtime Database*. Obtenido de <https://firebase.google.com/docs/database?hl=es-419>
- Firebase, M. F. (2021). *Mensajes FCM*. Obtenido de <https://firebase.google.com/products/cloud-messaging?hl=es-419>
- Google Inc. (Febrero de 2018). *Think With Google*. Obtenido de Think With Google: <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-page-speed-new-industry-benchmarks/>

- INEI Censo Cusco. (2007). *Características de la población*. Obtenido de <http://proyectos.inei.gob.pe/web/biblioineipub/bancopub/Est/Lib0838/libro26/cap01.pdf>
- Martín Díaz, O. (Diciembre de 2016). Sistema de gestión de información para seguimiento y alerta de casos positivos de tamizaje neonatal. *Revista Cubana de Información en Ciencias de la Salud*. Obtenido de [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2307-21132016000400008](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2307-21132016000400008)
- Marulanda Echeverry, C., & López Trujillo, M. &. (2016). Desarrollo de una aplicación móvil para alerta tecnológica. *Revista Virtual - Universidad Católica del Norte*, 1-15.
- NTT DATA, I. g. (2021). *¿Que es Kotlin?* Obtenido de <https://ifgeekthen.nttdata.com/es/que-es-kotlin-y-por-que-deberias-empezar-aprenderlo-ya>
- Renom, J. M., Corniller, F., Antonini, C., Serrano, Y., junto a estudiantes de Ciencia de la Computación García, J., Ubaldo, R., & Cupe, C. y. (07 de Mayo de 2020). Desarrollo de la aplicación móvil "Perú en tus manos". *Gestión*.
- Saunders. (1994). *Diccionario Médico Ilustrado de Dorland (Dorland's Illustrated Medical Dictionary)*. Phila: McGraw Hill - Interamericana.
- tutsplus. (2021). *Picasso, Aplicación Android*. Obtenido de <https://code.tutsplus.com/es/code-an-image-gallery-android-app-with-picasso--cms-30966t>
- Universidad Anáhuac, M. (2021). *COVID-19, ¿qué es, cómo se propaga y cómo se previene?* Obtenido de <https://www.anahuac.mx/mexico/noticias/COVID-19-que-es-y-como-se-propaga#:~:text=El%20COVID%2D19%2C%20es%20una,y%20el%20S%C3%ADndrome%20Respiratorio%20Agudo>
- Volley, D. (2021). *Realizar Peticiones Http Con La Librería Volley En Android*. Obtenido de <https://www.develou.com/android-volley-peticiones-http/>
- Wikipedia, F. (septiembre de 2021). *Firestore*. Obtenido de <https://es.wikipedia.org/wiki/Firebase>

Wilkins, W. &. (1995). *Diccionario Médico de Stedman (Stedman's Medical Dictionary)* (26th ed ed.). Baltimore: Panamericana.

Xakata, A. q. (2022). *¿Que es una API?* Obtenido de <https://www.xataka.com/basics/api-que-sirve>