

**DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

**OPCIÓN I.- TESIS**

**TRABAJO PROFESIONAL**

“Desarrollo de una aplicación para dispositivos móviles para la detección temprana de enfermedades cardiovasculares”

**QUE PARA OBTENER EL GRADO DE:  
MAESTRA EN SISTEMAS COMPUTACIONALES**

**PRESENTA:**

I.S.C ALEJANDRA MONTIEL DE JESÚS

**DIRECTOR DE TESIS:**

DR. JOSÉ LUIS SÁNCHEZ CERVANTES

**CO-DIRECTOR DE TESIS:**

M.S.C NANCY ARACELI CRUZ RAMOS

ORIZABA, VER. MÉXICO ABRIL 2022



Orizaba, Veracruz, **07/abril/2022**  
Dependencia: **División de Estudios de  
Posgrado e Investigación**  
Asunto: **Autorización de Impresión**  
OPCION: I

**C. ALEJANDRA MONTIEL DE JESÚS**  
**Candidato a Grado de Maestro en:**  
**SISTEMAS COMPUTACIONALES**  
**P R E S E N T E.-**

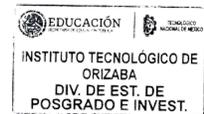
De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

" Desarrollo de una aplicación para dispositivos móviles para la detección temprana de enfermedades cardiovasculares. "

comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

**ATENTAMENTE**  
*Excelencia en Educación Tecnológica®*  
CIENCIA - TÉCNICA - CULTURA®

**DR. MARIO LEONCIO ARRIJOJA RODRÍGUEZ**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS**  
**DE POSGRADO E INVESTIGACIÓN**



OG-13-F06





Orizaba, Veracruz, **01/marzo/2022**  
Asunto: **Revisión de trabajo escrito**

**C. MARIO LEONCIO ARRIJOA RODRÍGUEZ  
JEFE DE LA DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN  
P R E S E N T E.-**

Los que suscriben, miembros del Jurado, han realizado la revisión de la Tesis del (la) C.

**ALEJANDRA MONTIEL DE JESÚS**

la cual lleva el título de:

**“Desarrollo de una aplicación para dispositivos móviles para la detección temprana de enfermedades cardiovasculares.”**

y concluyen que se acepta.

*A T E N T A M E N T E*  
*Excelencia en Educación Tecnológica®*  
*CIENCIA – TÉCNICA - CULTURA®*

PRESIDENTE: DR. JOSÉ LUIS SÁNCHEZ CERVANTES

FIRMA

SECRETARIO: M.S.C. LUIS ÁNGEL REYES HERNÁNDEZ

FIRMA

VOCAL: DRA. LISBETH RODRÍGUEZ MAZAHUA

FIRMA

VOCAL SUP.: M.S.C. NANCY ARACELI CRUZ RAMOS

FIRMA

TA-09-21



# Índice

<b>Resumen</b> .....	10
<b>Abstract</b> .....	11
<b>Introducción</b> .....	12
<b>Capítulo 1. Antecedentes</b> .....	13
<b>1.1 Marco teórico</b> .....	13
<b>1.1.1 Enfermedad cardiovascular</b> .....	13
<b>1.1.2 Aplicaciones híbridas</b> .....	13
<b>1.1.3 MHealth y EHealth</b> .....	14
<b>1.1.4 Internet de las cosas (IoT)</b> .....	14
<b>1.1.5 Monitorización remota</b> .....	15
<b>1.1.6 Dispositivos portátiles para signos vitales</b> .....	15
<b>1.1.7 Bioestadística</b> .....	16
<b>1.1.8 Adquisición de datos</b> .....	16
<b>1.1.9 Biomarcadores</b> .....	16
<b>1.1.10 Modelo predictivo</b> .....	17
<b>1.1.11 Algoritmo de estudio Framingham</b> .....	17
<b>1.1.12 Clasificación de datos</b> .....	17
<b>1.1.13 Repositorio de datos</b> .....	18
<b>1.1.14 Lenguajes de programación para aplicaciones móviles</b> .....	19
<b>1.1.15 Árboles de decisión</b> .....	20
<b>1.1.16 Series de tiempos</b> .....	20
<b>1.1.17 Algoritmos de aprendizaje</b> .....	21
<b>1.1.18 Detección temprana de enfermedades cardiovasculares</b> .....	21
<b>1.1.19 Recomendaciones para el cuidado de la salud</b> .....	21
<b>1.1.20 Visualización y graficación de datos</b> .....	22
<b>1.2 Planteamiento del problema</b> .....	23
<b>1.3 Objetivo general y específicos</b> .....	23
<b>1.3.1 Objetivo general</b> .....	23
<b>1.3.2 Objetivos específicos</b> .....	23
<b>1.4 Justificación</b> .....	24
<b>Capítulo 2. Estado de la práctica</b> .....	26
<b>2.1 Trabajos relacionados</b> .....	26

2.2	<b>Análisis comparativo de los trabajos relacionados</b>	30
2.3	<b>Propuesta de solución</b>	39
2.3.1	<b>React</b>	39
2.3.2	<b>Metodología de desarrollo Mobile-D</b>	40
2.3.3	<b>Visual Studio Code</b>	41
2.3.4	<b>Wearable</b>	42
2.3.5	<b>Descripción de la solución</b>	43
<b>Capítulo 3. Aplicación de la metodología</b>		45
3.1	<b>Metodología de desarrollo</b>	45
3.1.1	<b>Exploración</b>	45
3.1.2	<b>Iniciación</b>	54
3.1.3	<b>Producción</b>	65
3.1.4	<b>Estabilización</b>	95
3.1.5	<b>Prueba y reparación</b>	108
<b>Capítulo 4. Resultados</b>		122
4.1	<b>Caso de estudio</b>	122
4.2	<b>Evaluación y resultados</b>	131
<b>Capítulo 5. Conclusiones y recomendaciones</b>		132
5.1	<b>Conclusiones</b>	132
5.2	<b>Recomendaciones</b>	133
<b>Productos académicos</b>		134
6.1	<b>Artículo de congreso</b>	134
6.2	<b>Registro de derechos de autor</b>	134
<b>Anexos</b>		135
<b>Referencias</b>		136

## Índice de Figuras

Figura 3. 1 Diagrama de casos de uso .....	48
Figura 3. 2 Recursos físicos: Fitbit .....	55
Figura 3. 3 Recursos físicos: teléfono inteligente .....	55
Figura 3. 4 Recursos físicos: Laptop .....	56
Figura 3. 5 Diagrama de actividad para el registro de usuario .....	57
Figura 3. 6 Diagrama de actividad para realizar nuevo cálculo de probabilidad .....	57
Figura 3. 7 Diagrama de actividad para consultar resultados .....	58
Figura 3. 8 Mapeo Relacional de Objetos .....	63
Figura 3. 9 Diagrama de componentes .....	64
Figura 3. 10 Ciclo de ejecución de entregables .....	65
Figura 3. 11 Importancia de valores de los biomarcadores .....	68
Figura 3. 12 Resultados de exactitud.....	68
Figura 3. 13 Matriz de correlación de datos .....	69
Figura 3. 14 Gráfica de valores de acuerdo a frecuencia cardiaca .....	70
Figura 3. 15 Gráfica de valores de acuerdo a la presión sistólica .....	70
Figura 3. 16 Porcentaje de predicción de acuerdo al sexo.....	70
Figura 3. 17 Gráfica de datos sobre predicción de ECV .....	71
Figura 3. 18 Inicio de sesión en la aplicación móvil .....	75
Figura 3. 19 Menú lateral en la aplicación. ....	76
Figura 3. 20 Formulario para ingresar datos.....	76
Figura 3. 21 Recuperación de datos del dispositivo de monitorización .....	77
Figura 3. 22 Resultado del cálculo de probabilidad de las ECV .....	77
Figura 3. 23 Sugerencias para el usuario.....	78
Figura 3. 24 Diagrama de actividad con la API .....	86
Figura 3. 25 Interfaz del VPS .....	87
Figura 3. 26 Instalación de Postgres en el VPS .....	87
Figura 3. 27 Creación de base de datos en el VPS .....	88
Figura 3. 28 Uso de phpPgAdmin en el VPS .....	88
Figura 3. 29 Creación de tablas en el VPS .....	89
Figura 3. 30 Carpetas de archivos de la API .....	90
Figura 3. 31 Pruebas de inserción de datos desde Postman .....	94
Figura 3. 32 Ícono de la aplicación en el dispositivo móvil .....	109
Figura 3. 33 Splash Screen de la aplicación .....	109
Figura 3. 34 Registro de nuevo usuario .....	110
Figura 3. 35 Inicio de sesión.....	110
Figura 3. 36 Contraseña inhabilitada.....	111
Figura 3. 37 Contraseña habilitada.....	111
Figura 3. 38 Menú lateral para cierre de sesión.....	111
Figura 3. 39 Dashboard de la aplicación .....	112
Figura 3. 40 Menú lateral .....	113
Figura 3. 41 Datos del perfil de usuario .....	114

Figura 3. 42 Configuración de datos personales .....	114
Figura 3. 43 Formulario para datos médicos .....	115
Figura 3. 44 Formulario para la detección temprana.....	116
Figura 3. 45 Resultado favorable.....	116
Figura 3. 46 Resultado regular .....	116
Figura 3. 47 Resultado malo o de precaución .....	116
Figura 3. 48 Detección de arritmia .....	117
Figura 3. 49 Resultados de la detección de arritmia.....	117
Figura 3. 50 Detección de taquicardia .....	118
Figura 3. 51 Resultados de la detección de taquicardia.....	118
Figura 3. 52 Resultados parciales de la detección .....	118
Figura 3. 53 Sugerencias para el cuidado de la salud.....	119
Figura 3. 54 Datos del Fitbit conectado.....	120
Figura 3. 55 Descripción de dispositivo conectado.....	120
Figura 3. 56 Características del dispositivo conectado.....	120
Figura 3. 57 Resultados con alerta anterior .....	121
Figura 3. 58 Resultados con alerta actual .....	121
Figura 4. 1 Paciente 1 usando la aplicación.....	123
Figura 4. 2 Paciente 2 usando la aplicación.....	124
Figura 4. 3 Paciente 3 usando la aplicación.....	124
Figura 4. 4 Datos médicos .....	125

## Índice de tablas

Tabla 2. 1 Tabla comparativa de los trabajos relacionados .....	30
Tabla 3. 1 Establecimiento de actores .....	45
Tabla 3. 2 Requerimiento funcional RF01 .....	46
Tabla 3. 3 Requerimiento funcional RF02 .....	46
Tabla 3. 4 Requerimiento funcional RF03 .....	47
Tabla 3. 5 Requerimiento funcional RF04 .....	47
Tabla 3. 6 Requerimiento no funcional RNF01 .....	47
Tabla 3. 7 Requerimiento no funcional RNF02 .....	47
Tabla 3. 8 Requerimiento no funcional RNF03 .....	47
Tabla 3. 9 Biomarcadores para la detección de enfermedades cardiovasculares .....	49
Tabla 3. 10 Selección de variables .....	50
Tabla 3. 11 Tipo de recuperación de datos .....	51
Tabla 3. 12 Datos manuales.....	51
Tabla 3. 13 Lista de procesos .....	52
Tabla 3. 14 Diagrama de Gantt.....	53
Tabla 3. 15 Descripción de tabla User.....	58
Tabla 3. 16 Descripción de tabla ManualData .....	59
Tabla 3. 17 Descripción de tabla WearableData .....	60
Tabla 3. 18 Descripción de tabla Contacts .....	60
Tabla 3. 19 Descripción de tabla Wearable.....	61
Tabla 3. 20 Descripción de tabla Suggestions.....	61
Tabla 3. 21 Descripción de tabla PhotoUser .....	62
Tabla 3. 22 Descripción de tabla PhotoCon .....	62
Tabla 3. 23 Descripción de tabla PhotoWear .....	62
Tabla 3. 24 Descripción de tabla Results .....	62
Tabla 3. 25 Resultados de pruebas en modelos de predicción .....	67
Tabla 3. 26 Características y funcionalidades de la aplicación móvil.....	74
Tabla 3. 27 Fracción de código sobre Bibliotecas usadas .....	78
Tabla 3. 28 Fracción de código sobre Inicio de sesión.....	79
Tabla 3. 29 Fracción de código sobre validación de datos .....	80
Tabla 3. 30 Fracción de código sobre Menú lateral .....	81
Tabla 3. 31 Fracción de código sobre visualización de datos de wearable .....	82
Tabla 3. 32 Fracción de código sobre interfaz de datos médicos .....	83
Tabla 3. 33 Fracción de código sobre alertas con resultado.....	84
Tabla 3. 34 Fracción de código sobre el diseño utilizado .....	85
Tabla 3. 35 Fracción de código de la API para conexión a la Base de Datos .....	89
Tabla 3. 36 Fracción de código de la API para insertar datos .....	90
Tabla 3. 37 Fracción de código de la API para consultar datos .....	91

Tabla 3. 38 Fracción de código de la API para modificar datos .....	92
Tabla 3. 39 Fracción de código de la API para eliminar datos.....	92
Tabla 3. 40 Fracción de código para la conexión de la aplicación con la API.....	93
Tabla 4. 1 Grupo muestra para el caso de estudio .....	122
Tabla 4. 2 Resultados obtenidos por cada paciente .....	126
Tabla 4. 3 Evaluación del uso de la aplicación.....	131

## Resumen

En la actualidad, la medicina está teniendo mucha relación con la tecnología, ya que mediante él se obtienen valores de sensores, análisis clínicos, pruebas de nuevas sustancias, visualización de enfermedades en el ser humano y animales, entre otras. En cuestión de las enfermedades se está realizando seguimiento de pacientes y estadísticas computacionales para generar reportes.

Para las enfermedades cardiovasculares se realizan investigaciones sobre el origen y desarrollo de los biomarcadores durante las diferentes edades de las personas. Con los hallazgos se genera documentación, con apoyo de esa documentación se desarrollan soluciones tecnológicas como aplicaciones móviles, aplicaciones Web, simuladores, entre otras, para comprobar las teorías e hipótesis con resultados.

El término mHealth menciona la relación de la tecnología con la salud, y es ahí donde el uso de las aplicaciones móviles ha incrementado ya que accede a toda el área de la medicina, por ejemplo, hay aplicaciones que conectan a médicos y pacientes para tener citas o consultas sobre algún padecimiento.

Entonces si hablamos de aplicaciones y enfermedades cardiovasculares podemos deducir que trabajar con ambas implica apoyar las dos partes: medicina y tecnología. Por esta razón se ha investigado sobre soluciones que se enfocan a esta finalidad, la mayoría de ellos tienen como objetivo captar a personas que padecen alguna enfermedad cardiovascular y darle seguimiento en cuánto a sus tratamientos. Otros usan algoritmos que dan una predicción no tan acertada ya que solamente toman en cuenta algunos biomarcadores específicos.

Por estos motivos el proyecto que se propone tiene como objetivo desarrollar una aplicación para dispositivos móviles que realiza una detección temprana de enfermedades cardiovasculares. El desarrollo de una aplicación que realiza la detección a personas sintomáticas o asintomáticas, mayores de 18 años (que es la edad en la que los médicos ponen mayor atención a sus pacientes), apoyará en el cuidado de la salud de manera oportuna, permitiendo que se reduzcan las probabilidades y riesgos que una persona tiene ante las enfermedades.

Se busca reducir el número de mortalidad que existe por estas causas, ayudando a las personas en cambiar su estilo de vida.

## **Abstract**

Currently, the medicine is closely related to technology, since through it sensors values, clinical analyzes, tests of new substances, visualization of diseases in humans and animals, among others, are obtained. Regarding diseases, patient monitoring and computational statistics are being carried out to generate reports.

For cardiovascular diseases, research is carried out on the origin and development of biomarkers during the different ages of people. With the findings, documentation is generated, with the support of this documentation, technological solutions such as mobile applications, Web applications, simulators, among others, are developed to test the theories and hypotheses with results.

The term mHealth mentions the relationship of technology with health, and that is where the use of mobile applications has increased since it accesses the entire area of medicine, for example there are applications that connect doctors and patients to have appointments or inquiries about a condition.

So if we talk about applications and cardiovascular diseases, we can deduce that working with both implies supporting both parts: medicine and technology. For this reason, solutions that focus on this purpose have been investigated, most of them aim to capture people who suffer from cardiovascular disease and monitor their treatments. Others use algorithms that give a not so accurate prediction since they only take into account some specific biomarkers.

For these reasons, the proposed project aims to develop an application for mobile devices that performs early detection of cardiovascular diseases. The development of an application that performs the detection of symptomatic or asymptomatic people, over 18 years of age (which is the age in which doctors pay more attention to their patients), will support health care in a timely manner, allowing the probabilities and risks that a person has before the diseases are reduced.

It seeks to reduce the number of mortality that exists from these causes, helping people to change their lifestyle.

## Introducción

Es un hecho que la industria móvil está en un auge de todos los tiempos, lo que ha llevado a un fácil acceso a los teléfonos inteligentes para todos. A esto se agrega el acceso fácil y económico a conexiones rápidas a Internet. Esto ha llevado a un drástico cambio de lo que antes era la computadora de escritorio hasta el uso de teléfonos inteligentes para acceder al Internet.

Los cambios provocados en el enfoque de desarrollo de las aplicaciones móviles híbridas proporcionan un acceso general de parte de de las personas, ya que a grandes rasgos, no dependen de lo que contenga o cómo sea el teléfono para instalar las aplicaciones.

Conjuntando el trabajo mediante servicios web, se accede a los datos que brindan los dispositivos de monitorización de signos vitales que hay en el mercado, con el uso de una API se obtiene la información necesaria de las personas. Sumándole el uso de modelos de predicción enfocado a la detección temprana de enfermedades, se obtendrá un porcentaje de la probabilidad en contraer una enfermedad cardiovascular.

La solución que se propone en este anteproyecto de tesis permitirá ayudar a las personas a prevenir situaciones graves en su estado de salud, con las sugerencias se pretende motivar al usuario a cambiar sus hábitos diarios, a darle mayor importancia a las condiciones de vida.

En este documento se describe detalladamente la solución propuesta, el contenido se divide en cinco capítulos. El primero provee los conceptos básicos del dominio del problema, es decir el marco teórico, así también el planteamiento del problema, el objetivo general, los objetivos específicos y la justificación del proyecto; el segundo capítulo trata de un análisis y estudio de veinte artículos relacionados con el tema de investigación, enfocados al desarrollo de aplicaciones móviles para detectar enfermedades cardiovasculares. El tercer capítulo presenta la metodología seleccionada para el desarrollo de la solución, abarca cada una de las fases y las actividades realizadas. En el cuarto capítulo se describen los resultados obtenidos, las pruebas realizadas y el caso de estudio en la cual se aplicó la solución. Finalmente, en el capítulo cinco se realizan una conclusión general del trabajo y se anexan las recomendaciones.

# Capítulo 1. Antecedentes

En este capítulo se presentan aspectos primordiales de la investigación, tales como: el marco teórico, el problema a resolver, el objetivo general, los objetivos específicos, así como la justificación del presente trabajo.

## 1.1 Marco teórico

A continuación se definen conceptos y términos importantes para el trabajo de esta tesis.

### 1.1.1 Enfermedad cardiovascular

La enfermedad cardiovascular (ECV) es un término amplio para problemas con el corazón y los vasos sanguíneos. Esto casi siempre se debe a la aterosclerosis. Esta afección ocurre cuando la grasa, el colesterol y otras sustancias se acumulan en las paredes del vaso sanguíneo (arteria). Esta acumulación se denomina placa. Con el tiempo, las placas pueden estrechar los vasos sanguíneos y causar problemas en todo el cuerpo. Si una arteria resulta obstruida, esto puede llevar a que se presente un ataque cardíaco o un accidente cerebrovascular [1].

### 1.1.2 Aplicaciones híbridas

Las aplicaciones móviles multiplataforma o híbridas son aquellas aplicaciones desarrolladas con lenguajes de programación distintos a los lenguajes nativos de cada plataforma y que permiten ejecutar aplicaciones en diferentes sistemas con un mismo código base. Por ejemplo los Frameworks que utilizan los lenguajes web HTML, CSS y JS o que trabajan con estándares web, lo que realmente están ejecutando nuestros dispositivos es un renderizado de una aplicación web con una capa estética nativa. Frameworks como Ionic, React Native o PhoneGap son ejemplo de ello y estos se encargan de ofrecer a los desarrolladores casi todas las funcionalidades nativas de los dispositivos, como son el acceso a las cámaras, sensores, entre otras, de forma que no se tiene que programar en código nativo. Sin embargo hay que tomar en cuenta que otros Frameworks para desarrollo híbrido que no utilizan estos lenguajes web, como en el caso de Flutter y el lenguaje Dart [2].

#### 1.1.2.1 Ventajas

- El desarrollo es más ágil y sencillo, por lo tanto, más económico.
- Es más sencillo generar las actualizaciones.

- Hay posibilidad de reutilizar el código para otras soluciones.
- No se requieren permisos externos para distribuir la aplicación en las tiendas online.
- Permite abordar a un mercado de usuarios mucho más amplio.

#### 1.1.2.2 Desventajas

- No aprovechan de manera tan natural las capacidades del hardware. Aunque en la actualidad eso ya no es tan vital.
- Existe un 80% de personas que prefieren solo las aplicaciones nativas.
- En ciertos casos, las exigencias tan específicas de un dispositivo particular pueden hacer que las aplicaciones híbridas corran lentas o pesadas.

#### 1.1.3 MHealth y EHealth

El *mHealth* (*mobile Health*, salud móvil) es la práctica de la medicina y la salud pública soportada por dispositivos como teléfonos, wearables, asistentes digitales y otros dispositivos. Todo eso incluyendo aplicaciones que orientan sobre la vida saludable y el cuidado de la salud, que apoyan a comunicarse médicos con pacientes para brindar orientaciones mediante mensajes y alertas [3].

Las aplicaciones de *mHealth* ofrecen la oportunidad de recopilar una cantidad ingente de información del paciente que puede ser y, será para ayudar a la prevención e investigación de distintas enfermedades. Las aplicaciones pueden aportar en la mejora de la atención de pacientes, la experiencia del usuario y la optimización de los servicios de salud.

*EHealth* o salud digital, se habla de la manera de aplicar la tecnología en el ámbito de la salud, esto con el fin de apoyar en la gestión de servicios, en la prevención y diagnóstico de enfermedades, en la aplicación de tratamientos y los seguimientos de pacientes. *EHealth* trabaja con diversos productos, servicios y tecnologías, tales como los *wearables*, aplicaciones móviles, videojuegos orientados a la salud, sistemas de información y sistemas médicos industriales [4].

#### 1.1.4 Internet de las cosas (IoT)

IoT (*Internet of Things*, Internet de las cosas) es un concepto popular dentro del sector tecnológico. Se refiere a escenarios donde la conectividad de red y la capacidad informática se extiende a objetos, sensores y elementos cotidianos que normalmente no se consideran

computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana [5].

Una de las aplicaciones de mayor interés, recorrido e importancia del fenómeno IoT para la sociedad, es la que busca mejorar la salud de las personas. El estar conectado a determinados *wearables* proporcionará una mayor esperanza de vida gracias a los datos que los dispositivos tecnológicos recogen, analizan y generan; propiciando soluciones para la toma de decisiones saludables, tanto para el enfermo como para el médico y apoyo a la prevención de enfermedades graves.

### **1.1.5 Monitorización remota**

En [6] los autores mencionan a la monitorización remota, como la instalación de distintos dispositivos electrónicos en una vivienda o en alguna persona. Su función consiste en avisar a una central de monitorización a través de una alarma sobre posibles situaciones de necesidad o emergencia captadas.

En los últimos años se han desarrollado diversas soluciones que proporcionan un servicio global y específico en función del grado de dependencia de la persona. El término se conoce como monitorización remota avanzada, es decir, un servicio que ofrece una respuesta más amplia y que permite un seguimiento más activo del estado del usuario e incorpora servicios de intervención domiciliaria. Algunos sistemas de monitorización son: pulsadores de alarma, sistemas de detección, sensores y *wearables*.

### **1.1.6 Dispositivos portátiles para signos vitales**

Los dispositivos portátiles para signos vitales son equipos biomédicos que permiten detectar, procesar y desplegar de manera continua toda la información de los parámetros fisiológicos de un paciente. Con este tipo de dispositivos se pueden descubrir situaciones adversas o fuera del rango deseado, sirviendo como guías visibles y audibles para todos los médicos que los utilizan.

#### *1.1.6.1 Wearable*

*Wearable* [7], es una categoría de dispositivos electrónicos que se llevan como accesorios, incrustados en la ropa, implantados en el cuerpo del usuario, o incluso tatuados en la piel. Los dispositivos son aparatos de manos libres con usos prácticos, alimentados por microprocesadores y mejorados con la capacidad de enviar y recibir datos a través de Internet.

La tecnología de los *wearables* se está convirtiendo en una importante categoría del IoT, con aplicaciones que cambian la vida en la medicina y otros campos.

### **1.1.7 Bioestadística**

La bioestadística envuelve el desarrollo y aplicación de técnicas estadísticas a la investigación en campos relacionados a la salud incluyendo medicina, epidemiología, salud pública y también en áreas de estudios ambientales, investigación agrícola y biología.

Como una rama más de la estadística, la bioestadística comprende diferentes aspectos desde el desarrollo de experimentos, la recolección de datos y el análisis detallado de los resultados. Al momento de ejercer un diseño de la investigación, la bioestadística tiene una participación importante ya que a la hora de definir el número de pacientes participantes se incluye la justificación de la selección y la forma en la que estos deben ser asignados a los diferentes tratamientos o condiciones que van a investigarse. Una vez recolectada la información, la bioestadística proporciona las herramientas de análisis necesarias para representar de manera gráfica la información que se ha obtenido. Los métodos de inferencia estadística también permiten establecer las relaciones subyacentes entre variables, encontrar patrones de comportamiento y sistematizar los resultados en una muestra de sujetos a la población de la cual provienen. [8].

### **1.1.8 Adquisición de datos**

Los sistemas de adquisición de datos son productos o procesos que se usan para recoger información para documentar o analizar un fenómeno [9]. Los dispositivos enfocados en la adquisición de datos, utilizan sensores para la recopilación de datos de temperatura, flujo, nivel, nivel o presión.

### **1.1.9 Biomarcadores**

Los biomarcadores son moléculas que indican la normalidad o anormalidad de los procesos que tienen lugar en el cuerpo y es un signo de una enfermedad subyacente. Varios tipos de moléculas, tales como el ADN (Ácido Desoxirribonucleico), las proteínas o las hormonas, sirven como biomarcadores, ya que todo indica algo acerca de la salud. Los biomarcadores son producidos por tejidos, células y son encontrados en la sangre, heces, orina, tejido o fluidos corporales. Hay biomarcadores para enfermedades del corazón, esclerosis múltiple, entre otras [10].

### **1.1.10 Modelo predictivo**

El modelo predictivo es un modelo de datos, basado en estadísticas inferenciales, que se utiliza para predecir la respuesta a una promoción de marketing o a una determinada inversión [11].

El modelo predictivo es creado habitualmente por los científicos de datos y utiliza estadísticas para predecir los resultados. Frecuentemente se busca predecir algún evento del futuro, pero los modelos predictivos se usan para cualquier tipo de evento desconocido, sin importar cuándo pasó.

En muchas ocasiones el modelo a utilizar se elige de acuerdo a la teoría de detección al cuál será aplicado para encontrar la probabilidad de un resultado dada una cantidad establecida de datos de entrada, por ejemplo, dando un correo electrónico para determinar la probabilidad de que sea spam.

Los modelos llegan a utilizar entre uno y varios clasificadores al momento de determinar la probabilidad de un conjunto de datos que pertenecen a otro conjunto. Cuando se despliega comercialmente, el modelado predictivo se refiere a menudo como análisis predictivo.

### **1.1.11 Algoritmo de estudio Framingham**

El Algoritmo de estudio Framingham o Estudio Framingham del corazón (*Framingham Heart Study*) es un estudio de cohortes de larga duración sobre el riesgo cardiovascular, realizado entre los residentes de Framingham, una ciudad del estado de Massachusetts, EE. UU[12].

El cálculo del riesgo mediante el algoritmo de Framingham utiliza el valor del colesterol total, las variables que intervienen son: el sexo, la edad en años, el colesterol sérico en mg/dl, la fracción de colesterol ligado a lipoproteínas de alta densidad (HDL), la presión sistólica, padecimiento de diabetes y por último, si la persona fuma.

### **1.1.12 Clasificación de datos**

La clasificación de datos implica categorizar o estructurar datos en orden de importancia o relevancia. Generalmente la clasificación se realiza según la criticidad de los documentos o su sensibilidad (confidencialidad, restringido, uso general y público). La clasificación es el pilar dentro del proceso de análisis de la información ya que gracias a ella se pierde un mínimo de datos [13].

Existen distintos métodos de clasificación de datos disponibles, así como diferentes criterios en la disposición de datos de un repositorio. En la mayoría, los métodos son utilizados para ordenar datos no estructurados.

### **1.1.13 Repositorio de datos**

Un repositorio, depósito o archivo de datos es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

El medio de distribución de los datos almacenados es mediante el Internet, una red informática o físicamente. Los datos son de acceso público, protegido o necesitan una autenticación previa. Los repositorios más conocidos son de carácter académico e institucional. Estos depósitos cuentan con sistemas de respaldo y mantenimiento preventivo y correctivo, lo que hace que la información se pueda recuperar en el caso que la máquina quede inutilizable. A esto se le conoce como preservación digital y requiere un exhaustivo trabajo de control de calidad e integridad para realizarse correctamente [13].

#### *1.1.13.1 Base de datos*

Se denomina base de datos a un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y transmisión. Hoy en día, las bases de datos se presentan de diferentes formas y tamaños, esto de acuerdo al lugar donde son empleados, por ejemplo en una biblioteca o en cuentas de una empresa.

Las bases de datos se originaron para cubrir la necesidad de almacenar grandes cantidades de información, es decir, de preservarla contra el tiempo y el deterioro, para acudir a ella posteriormente. Es ese sentido, la aparición de la electrónica y la computación brindaron el elemento digital indispensable para almacenar enormes cantidades de datos en espacios físicos limitados, gracias a su conversión en señales eléctricas o magnéticas [14].

#### *1.1.13.2 Big Data*

El *Big Data* consta de datos que son tan grandes o complejos que no pueden manejarse con los métodos tradicionales de procesamiento. El Big Data también se conoce con las tres v: velocidad, variedad y volumen. El primero se refiere a la necesidad de procesarlos con rapidez y eficiencia; el segundo se enfoca en la amplia cantidad de formatos no estándar y la tercera al gran tamaño que maneja [15].

Uno de los objetivos principales del Big Data es brindar información en tiempo real, aunque las empresas lo usan para aportar valor a sus clientes de manera uniforme y sin

inconvenientes y es una característica fundamental del *Edge computing*. La información del *Big Data* permite reducir costos, operar con mayor eficiencia y descubrir nuevas formas de aumentar los beneficios y obtener mejores resultados a futuro.

### 1.1.14 Lenguajes de programación para aplicaciones móviles

Los lenguajes que se utilizan actualmente en el desarrollo de aplicaciones móviles son *JavaScript*, para la parte visual, CSS para el diseño de las interfaces y para la lógica del software.

#### 1.1.14.1 *JavaScript*

*JavaScript* (abreviado comúnmente JS) es un lenguaje de programación interpretado. Ha tenido diferentes definiciones: orientado a objetos, imperativo, dinámico, basado en prototipos y débilmente tipado. Se utiliza del lado del cliente, implementado como parte del navegador, de esta manera permite mejoras en la interfaz de usuario y páginas *Web* dinámicas y *JavaScript* del lado del servidor (*Server-side JavaScript* o SSJS) [16].

*JavaScript* se desarrolló similar al lenguaje C, aunque se adopta similarmente a Java. Sin embargo, Java y *JavaScript* tienen semánticas y propósitos diferentes.

A continuación se enlistan las características más comunes de acuerdo al estándar ECMAScript.

- Imperativo y estructurado: *JavaScript* es compatible con gran parte de la estructura de programación de C.
- Tipado dinámico: es decir el tipo está asociado al valor y no a la variable.
- Objetual: *JavaScript* está formado casi en su totalidad por objetos.
- Prototipos: para crear herencia usa prototipos en vez de clases. En *JavaScript* es posible imitar las características de las clases en lenguajes orientados a objetos tradicionales por medio de prototipos en *JavaScript*.
- Funciones como constructores de objetos: Es necesario prefijar una llamada a la función con la palabra clave *new* para crear una nueva instancia de un prototipo, que hereda propiedades y métodos del constructor.

#### 1.1.14.2 CSS

CSS (*Cascading Style Sheets*, Hojas de Estilo en Cascada) es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille [17].

CSS está diseñado con la finalidad de brindar la estética al sistema, darle forma a la presentación desde estilos de texto, colores y formas.

Ventajas de usar CSS:

- Trabajo individual del contenido y la presentación.
- Consistencia del sitio: si se trabaja con estilos basado a herencia, el mismo diseño se puede utilizar en todo el sitio.
- Ancho de banda: para crear el diseño se emplean clases, estas se reutilizan solamente donde sean necesarias, como este tipo de archivos se almacenan en la memoria caché se consultan en el momento, reduciendo la transferencia de datos a través de la red.
- Formateo de página: para cambios solamente basta con editar las líneas necesarias.
- Accesibilidad: sin CSS, los diseñadores tendrían que crear estilos dentro del lenguaje usado, esto conlleva un perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales.

#### 1.1.15 Árboles de decisión

De acuerdo a [11], un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento. Ayudan a tomar la sentencia más acertada, desde un punto de vista probabilístico, ante un abanico de posibles decisiones. Permiten desplegar visualmente un problema y organizar el trabajo de cálculos que deben realizarse.

#### 1.1.16 Series de tiempos

En [19] se define una serie de tiempo como una secuencia de datos u observaciones, medidos en determinados momentos y ordenados cronológicamente. Representa una curva que va cambiando en el tiempo. Siendo, también, un conjunto de observaciones sobre los valores que toma una variable (cuantitativa) a través del tiempo, por lo que es una forma estructurada de representar datos.

Los datos se comportan de diferentes maneras, como una tendencia estacional o simplemente no presentan una forma definida.

### **1.1.17 Algoritmos de aprendizaje**

Los algoritmos de aprendizaje o también conocidos como algoritmos de aprendizaje automático son fragmentos de código que apoyan a las personas a buscar y analizar conjuntos de datos y buscar el significado en ellos. Cada algoritmo es un conjunto finito de instrucciones paso a paso inequívoco que puede seguir una máquina para lograr un determinado objetivo. En un modelo de aprendizaje automático, el objetivo es establecer o detectar patrones que los usuarios pueden usar para hacer predicciones o clasificar información [20].

Los algoritmos usan parámetros basados en los datos de entrenamiento, un subconjunto de datos que representa el conjunto más grande. Conforme va aumentando el número de datos de entrenamiento, el algoritmo brinda una mayor precisión del resultado.

Los diferentes algoritmos analizan los datos de distinta manera. A menudo se agrupan con técnicas de aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje de refuerzo. Los algoritmos que se usan con mayor frecuencia utilizan la regresión y la clasificación para predecir categorías de destino, buscar puntos de datos inusuales, predecir valores y detectar similitudes.

### **1.1.18 Detección temprana de enfermedades cardiovasculares**

Los exámenes de detección temprana son pruebas que se realizan para encontrar una enfermedad antes de que comiencen los síntomas. El objetivo de la detección temprana es detectar una enfermedad en su etapa más tratable. Para que sea ampliamente aceptado y recomendado por los médicos, un programa de detección temprana debe cumplir con una serie de criterios, que incluyen la reducción del número de muertes por la enfermedad [21].

### **1.1.19 Recomendaciones para el cuidado de la salud**

Recomendar es la acción por la cual algo o alguien persuaden a una persona para algún fin, es decir se lleva a cabo entre una persona y un dispositivo, o también un texto, una alerta o cualquier otra fuente de información, que busca ayudar a la persona a quien dirige el mensaje. Es una propuesta que se hace para que el otro se beneficie en algún sentido [21].

### **1.1.20 Visualización y graficación de datos**

La visualización de datos es la representación gráfica de información y datos. Al utilizar elementos visuales como cuadros, gráficas y mapas, las herramientas de visualización de datos proporcionan una manera accesible de ver y comprender tendencias, valores atípicos y patrones en la información. En el mundo del *Big Data*, las herramientas de este tipo son esenciales para analizar grandes cantidades de información y tomar decisiones basadas en los datos.

Los gráficos se usan comúnmente para realizar resúmenes financieros, estadísticos, matemáticos, científicos, de ingeniería y económicos para realizar informes de investigación, boletines de información y otros tipos de publicaciones [22]. También son importantes en la presentación de resultados ya que depende de ellos las decisiones que se toman para cambiar algo en la realidad.

## **1.2 Planteamiento del problema**

De acuerdo a la OMS (Organización Mundial de la Salud) se calcula que en 2019 fallecieron alrededor de 17,5 millones de personas por alguna enfermedad cardiovascular, esto simboliza el 30% de las muertes registradas en todo el mundo. Las ECV son una de las principales causas de muerte en el mundo. Cada año mueren más personas por alguna enfermedad relacionada con este padecimiento que por cualquier otra cosa. De los fallecimientos, aproximadamente 7,4 millones se debieron a cardiopatías coronarias y 6,7 millones a accidentes cerebrovasculares [23].

Las ECV afectan en mayor medida a los países de ingresos bajos y medios. De aquí a 2030, casi 23,6 millones de personas morirán por alguna ECV, ya que se prevé que esta enfermedad siga siendo la principal causa.

Para erradicar con esta situación, algunas instituciones han realizado investigaciones sobre detectar y dar seguimiento a las condiciones de vida que llevan las personas, como resultado, encontraron que una de las tecnologías más usadas son las que se relacionan con mHealth y eHealth. El desarrollo de las aplicaciones móviles, en este aspecto, ha apoyado en el control de las enfermedades, pero es necesario contar con una aplicación que realice la detección temprana de las ECV, que en coordinación con dispositivos como wearables, pueda aportar a las personas en general, un estado actual de su salud, para prevenir las posibles consecuencias y evitar riesgos o muertes a futuro.

## **1.3 Objetivo general y específicos**

En este apartado se presentan el objetivo general y los objetivos específicos del proyecto de Tesis.

### **1.3.1 Objetivo general**

- Desarrollar una aplicación para dispositivos móviles que apoye a las personas en el cuidado de la salud, mediante la detección temprana de enfermedades cardiovasculares.

### **1.3.2 Objetivos específicos**

- Estudiar y analizar el estado del arte de las aplicaciones móviles para la detección temprana de enfermedades cardiovasculares, así como proyectos relacionados con enfermedades del corazón.

- Realizar un análisis comparativo de las tecnologías y propuestas encontrados en el estado de la práctica para la identificación de diferencias y establecer la principal aportación del proyecto.
- Identificar las enfermedades cardiovasculares que son susceptibles a ser detectadas de manera temprana para el diseño del modelo de detección temprana.
- Determinar los dispositivos para la extracción de factores de riesgo, signos vitales, indicadores de hábitos y biomarcadores de las personas para la detección de enfermedades cardiovasculares.
- Diseñar la arquitectura del sistema que presente la comunicación de la aplicación móvil con los dispositivos de detección de signos vitales.
- Diseñar el modelo para la detección temprana de enfermedades cardiovasculares.
- Desarrollar la aplicación móvil que permita la detección temprana de enfermedades cardiovasculares.
- Realizar al menos una prueba de concepto para la validación de resultados de la aplicación.

## **1.4Justificación**

Los exámenes de detección temprana son pruebas que se realizan para encontrar una enfermedad antes de que comiencen los síntomas. El objetivo de la detección temprana es detectar una enfermedad en su etapa más temprana y más tratable. Para tener la aprobación y recomendación de los médicos, el sistema de detección temprana debe cumplir con una serie de criterios, que incluyen la reducción del número de muertes por la enfermedad.

Las pruebas de detección pueden incluir pruebas de laboratorio para evaluar la sangre y otros fluidos, pruebas genéticas que buscan marcadores genéticos como herencia y asociados a la enfermedad. Las pruebas están abiertas públicamente, pero todo depende de las necesidades de una persona, y para ello se toma en cuenta los factores de salud, tales como edad, sexo y antecedentes familiares.

Si las pruebas resultan positivas a alguna enfermedad, entonces se deben seguir ciertos pasos para reducir el riesgo de tener un ataque al corazón o de que empeore la enfermedad cardíaca. Por ejemplo, el médico puede recomendar cambios en el estilo de vida, como: una dieta saludable, dejar de fumar y realizar ejercicio.

El creciente uso de dispositivos móviles permite el desarrollo de aplicaciones que sirven de apoyo a las personas en la prevención, cuidado y tratamiento de enfermedades. Por ello se propone el desarrollo de una aplicación para dispositivos móviles que apoye a las personas en el cuidado de su salud mediante la detección temprana de enfermedades cardiovasculares.

Se pretende llevar a cabo un extractor de datos de signos vitales y biomarcadores monitorizados mediante wearables y se contempla en caso de ser necesario, la adquisición manual de datos como análisis clínicos, todos los datos recolectados serán analizados,

procesados y encapsulados en servicios Web para ser consumidos a través de la aplicación móvil con el propósito de alertar a los usuarios asintomáticos sobre la posibilidad de sufrir alguna enfermedad cardiovascular.

## Capítulo 2. Estado de la práctica

En este capítulo se presenta el estudio de trabajos relacionados con el proyecto. Se realizó una minuciosa revisión de varios artículos que plantearon problemáticas similares a las que se pretenden resolver en este trabajo.

### 2.1 Trabajos relacionados

La enfermedad cardiovascular (ECV) es un término utilizado en la descripción de problemas del corazón y los vasos sanguíneos. Estos problemas a menudo se deben a la arterosclerosis. Este padecimiento sucede cuando la grasa y el colesterol se acumulan en las paredes de la arteria. A esto se le llama placa. Con el paso del tiempo, la placa puede disminuir el tamaño de los vasos sanguíneos y causarle problemas a todo el cuerpo. Si el vaso sanguíneo se obstruye, se ocasionan los ataques cardíacos y los accidentes cerebrovasculares.

En la actualidad se realizan diferentes proyectos para prevenir situaciones de riesgo con enfermedades cardiovasculares, así como se hacen estudios que avalan la veracidad y efectividad del uso de la tecnología. En [24] analizaron 1354 registros y revisaron 101 manuscritos de texto completo para determinar la elegibilidad y factibilidad de la tecnología *mHealth* (salud móvil). En [25] estudiaron las aplicaciones que usan SMS (*Short Message Service*, Servicio de Mensajes Cortos), aplicaciones inteligentes y redes sociales. Ambas fuentes dirigieron sus intervenciones a poblaciones con hipertensión, insuficiencia cardíaca, accidentes cerebrovasculares y rehabilitación cardíaca. Entre las características más atractivas de las aplicaciones figuraron el seguimiento de los comportamientos saludables, la autovigilancia, la educación sobre las enfermedades y un contenido personalizado y adaptable. Entre los resultados encontraron que los factores que mejoraron los usuarios de las aplicaciones fueron las tasas de rehospitalización, el conocimiento específico de la enfermedad, la calidad de vida, el bienestar psicosocial, la presión arterial, el índice de masa corporal, la circunferencia de la cintura, el colesterol y la capacidad de ejercicio.

De acuerdo a estos datos se comprobó que el desarrollo de investigaciones apoya a crear sistemas de predicción de enfermedades cardiovasculares para realizar una detección precisa de la enfermedad, en las propuestas de [26], [27] y [28] utilizaron una técnica de aprendizaje automático que se deriva del uso de varios algoritmos de aprendizaje. En [29] agregaron el uso de técnicas de minería de datos y las RNA (Redes Neuronales Artificiales) para construir un clasificador de ECG (Electrocardiograma). En [30], [31] propusieron un nuevo algoritmo de diagnóstico de enfermedades para la señal de ECG, el procesamiento se implementó en el ASIC (*Application Specific Integrated Circuit*, Circuito Integrado para Aplicaciones Específicas). El ASIC y la aplicación fueron verificados para detectar la enfermedad

utilizando una base de datos de diagnóstico de Physionet PTB (*Physikalisch Technische Bundesanstalt*). Además, para dar seguimiento al paciente durante el día, diseñaron sistemas de monitoreo de pacientes en tiempo real. En el proyecto, los médicos visualizan datos mediante el uso de aplicaciones móviles. De esta manera se controlan los síntomas previniendo el desarrollo de la enfermedad.

Al recopilar datos en el monitoreo de detección y prevención de enfermedades cardiovasculares es necesario el almacenamiento en distintas fuentes, en [32] se optó por utilizar una base de datos en la nube. En [33] los datos del ECG se recopilaron utilizando un nodo de monitoreo portátil y se transmitieron a la nube mediante WiFi. Ambos utilizaron protocolos HTTP (*Hypertext Transfer Protocol*, Protocolo de transferencia de Hipertexto) y MQTT (*Message Queuing Telemetry Transport*, MQ Transporte de Telemetría) para comunicar los dispositivos a la fuente de almacenamiento. Los resultados revelaron que el sistema es confiable para recopilar y mostrar datos de ECG en tiempo real en plataformas o aplicaciones móviles, permitiendo el diagnóstico primario de ciertas enfermedades del corazón.

La prevención y el seguimiento a los pacientes con enfermedades cardiovasculares mediante el monitoreo de datos sobre la frecuencia cardiaca son importantes en el desarrollo de soluciones [34]. Para llevar un control de la medición de pulso irregular se propuso el proyecto [35], donde llevaron a cabo un estudio de forma virtual, con la selección, el consentimiento y la recopilación de datos de manera electrónica desde una aplicación de teléfono inteligente. El objetivo de esta propuesta fue seleccionar a los participantes, consentir electrónicamente a los que calificaban para el control personalizado, notificar sobre un pulso irregular que es consistente con la FA (fibrilación o el aleteo auricular) y dirigirlos a través de los procedimientos del estudio médico. En la herramienta los participantes pueden ver el número de días que han estado en el estudio y el número total de tacografías registradas. Si se obtienen cuatro confirmaciones del pulso irregular, los pacientes reciben una notificación. Con las pruebas realizadas se confirmó un seguimiento adecuado a los pacientes y una prevención de posibles complicaciones en la salud mediante la alerta inmediata.

El uso de sensores facilita la recopilación de datos acerca de la frecuencia cardiaca, ECG y FA. En [36] las señales obtenidas de distintos sensores incorporados en teléfonos inteligentes fueron tratadas con transformaciones matemáticas, para eliminar los componentes de vagabundeo y ruido de la línea base dentro de las bandas de frecuencia. Y en [27], [37] se usaron sensores con la finalidad de inspeccionar la actividad física que realizan las personas y los movimientos bruscos que alteran la frecuencia cardiaca, para brindar acompañamiento a distancia de cada persona y sugerencias que mejoren su salud. Los objetivos planteados fueron cumplidos exitosamente, ya que con el uso de los proyectos se trataron datos de frecuencia cardiaca en personas para prevenir enfermedades cardiovasculares.

El mercado provee *wearables* que las personas adquieren para conocer su estado de salud. En [38] los *wearables* fueron revisados para conocer su funcionalidad y las distintas herramientas que contienen para ser aprovechadas en la generación de soluciones tecnológicas. En [39] de Pessemier, Cailliau y Martens realizaron una comparación de las mediciones de frecuencia cardíaca que captan los dispositivos, usando tres tipos de *wearables*: un reloj inteligente, un rastreador de ejercicios y un dispositivo especializado. En [40] describieron el desarrollo de un proyecto para la monitorización de la salud con *wearables*. Los autores generaron una plataforma de código abierto que incluye circuitos de recolección de energía, un centro de sensores modular, hardware de procesamiento, un módem inalámbrico, controladores de software y una API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones). Demostraron que los datos capturados por los *wearables* son confiables, reales y aptos para ser manipulados por otros dispositivos: teléfono inteligente, computadora, por mencionar algunos.

Actualmente, el desarrollo de tecnologías que implementan IoT muestra una conectividad en tiempo real de diferentes dispositivos electrónicos, por lo que en el caso de la monitorización de datos referentes a la salud dan pauta a la captación de irregularidades. En [32] se desarrolló un proyecto con inteligencia artificial e IoT para el análisis de ECG y detección de enfermedades cardíacas. También incluye un algoritmo que se basa en la red neuronal convencional y el promedio de precisión es de 94.96%. En [31], [33] propusieron un método para monitorización de ECG, un algoritmo para el análisis y clasificación, para el diagnóstico de latidos cardíacos. Es una propuesta basada en IoT de dispositivos Intel. Los resultados comprobaron que la implementación de IoT permite la comunicación de los teléfonos inteligentes con sensores y *wearables*, por lo que los datos que se obtienen se han usado para conocer el estado de salud de las personas y obtener gráficos generales de las causas en enfermedades cardiovasculares.

Los datos obtenidos de los dispositivos (sensores, *wearables*, ingresados manualmente) son tratados con algoritmos, condiciones y cálculos para brindar a las personas los resultados sobre su salud actual. En [39], [41] desarrollaron aplicaciones móviles en Android para recopilar, almacenar y analizar las mediciones, brindando al usuario una interfaz interpretable que marca el nivel de atención que debe brindar a su salud: a) bajo; b) medio; c) alto; d) alto riesgo. Sin embargo, también se encontraron proyectos enfocados al seguimiento de tratamientos de enfermedades cardiovasculares. En [42] se propuso una aplicación para dar seguimiento a los tratamientos médicos, este software está dividido en tres secciones: a) sección informativa; b) sección de registro; c) sección registro de medicamentos. En general, las aplicaciones se desarrollaron en Android, limitando el acceso a los usuarios, por lo que hace falta una aplicación móvil multidispositivo.

Dentro del análisis también se encontraron proyectos que desarrollaron plataformas, los cuales cuentan con conexión a datos en la nube, aplicaciones móviles y *Web*, circuitos eléctricos e implementación de algoritmos. En [40] se describió la implementación de software y hardware para la monitorización de los pacientes con trastorno de movimiento. En [27], [32], propusieron plataformas para detectar y prevenir enfermedades cardíacas. Para recolectar datos se construyó una herramienta que usa dos aplicaciones móviles [43]. En los proyectos descritos se obtuvieron resultados favorables, el uso de dispositivos electrónicos con sensores y *wearables* permitió captar datos de salud de las personas, por lo que al encontrar información irregular, los médicos pudieron intervenir para alertar al paciente a cambiar sus actividades cotidianas.

## 2.2 Análisis comparativo de los trabajos relacionados

La Tabla 2. 1 muestra la comparación de los trabajos analizados en el punto anterior.

Tabla 2. 1 Tabla comparativa de los trabajos relacionados

Artículo	Problema	Contribución	Tecnología	Resultados	Estado
Turakhia et al. [35]	Medir el pulso de las personas, detectar irregularidades, obtener un filtro de personas con datos relevantes y darles seguimiento médico.	Una aplicación móvil llamada Apple Heart Study App, para la monitorización y selección de las personas con pulso irregular, brindar un seguimiento a los pacientes y guiarlos a contactar con la compañía de servicios de tecnología de telemedicina.	Apple Watch, parche de ECG, iPhone.	Se realizó un estudio de forma virtual, con la selección, el consentimiento y la recopilación de datos por el Apple Watch en sincronía con la aplicación móvil. En el momento en que se capturaban irregularidades en el pulso de los usuarios se notificó al paciente para que acuda al médico.	Finalizado
Bhat, Deb y Ogras [40]	Dar seguimiento a las actividades cotidianas de pacientes con trastorno de movimiento (tics, coreas, babilismo).	Una plataforma de código abierto para la monitorización de la salud usando <i>Framework</i> . El proyecto incluye circuitos de recolección de energía, un centro de sensores modular, controladores de	<i>Wearables</i> , sensores, módem inalámbrico, MCU ( <i>microcontroller unit</i> ), ZigBee.	Las pruebas fueron realizadas en pacientes con desórdenes de movimiento, se demostró que la plataforma monitorea a los pacientes con apoyo de sensores y los datos se	Finalizado

		<i>software</i> e interfaces de programación de aplicaciones (API).		almacenar. Obteniendo un resultado exitoso.	
Lin et al. [32]	Análisis de electrocardiograma y detección de enfermedades cardiacas.	Un sistema que incluye <i>hardware</i> basado en IoT, una interfaz de usuario en dispositivos inteligentes, una base de datos en la nube, una plataforma de inteligencia artificial y un módulo <i>Bluetooth</i> para detectar señales de electrocardiograma.	CNN ( <i>Convolutional Neural Network</i> ), parche ECG, base de datos en la nube, <i>Bluetooth</i> , IA, MIT-BIH.	El sistema detecta y muestra los datos en tiempo real, también etiqueta señales inusuales al instante y alcanza la detección de enfermedades cardiacas.	Finalizado
Ranjan et al. [43]	Optimizar la recolección de datos en la monitorización remota de proyectos de mHealth.	Una plataforma de evaluación remota de enfermedades cardiacas y recaídas, se desarrolló como una aplicación modular, se usó el <i>Framework</i> Confluent/Apache para la transmisión de datos.	Confluent/Apache Kafka, contenedores Docker, HTTP, MongoDB, JSON, Cordova.	Con la plataforma se atiende la recolección de datos a gran escala para el monitoreo remoto. El proyecto proporcionará dos aplicaciones móviles.	En proceso
De Pessemier, Cailliau y Martens [39]	Factibilidad del uso de dispositivos portátiles para la medición de frecuencia cardiaca	Una aplicación móvil para el sistema operativo Android que recopila, almacena y analiza las mediciones de frecuencia	Reloj inteligente, Android, Google Nexus 6P, Android, <i>Bluetooth</i> .	La aplicación recopila, almacena y analiza datos. Puede distinguir si la frecuencia cardiaca es de una muñeca, un brazo o una	Finalizado

	durante la actividad física.	cardiaca obtenidas de tres dispositivos portátiles: un reloj inteligente, un rastreador de ejercicios y un dispositivo especial para frecuencia cardiaca.		mano, para analizar si el paciente ha ejecutado movimientos o está en reposo.	
Coorey et al. [44]	Tipo de resultados que brinda el uso de las aplicaciones de salud en pacientes que padecen enfermedades cardiacas.	Recopilación de documentos que respaldan resultados cualitativos y cuantitativos en el uso de aplicaciones para la salud.	Cochrane CENTRAL, Medline, Embase, CINAHL ( <i>Cumulative Index to Nursing &amp; Allied Health Literature</i> , Índice acumulativo de literatura sobre enfermería y salud afín), SCOPUS.	Análisis de 1354 registros y revisión de 101 manuscritos en las que se encontró que la mayoría de las aplicaciones realizan seguimiento de los comportamientos saludables, la autovigilancia y proveen educación sobre las enfermedades.	En proceso
Iftikhar et al. [36]	Detección de fibrilación auricular con el uso de sensores de acelerómetro y giroscopio incorporados en <i>Smartphone</i> .	Un algoritmo para eliminar componentes de vagabundo y ruido en las mediciones que se realizan con sensores de acelerómetro y girocardiografía incorporados en los teléfonos celulares.	Transformación Fourier, LOOCV ( <i>Leave-One-Out Cross-Validation</i> ), Sony Xperia Z, Matlab.	Se seleccionaron usuarios a los que se les colocó el teléfono en el pecho (en el hueso del esternón) para detectar señales mecánicas a través de la sismocardiografía articular y girocardiografía, señales obtenidas de sensores de	En proceso

				acelerómetro y giroscopio incorporados en <i>Smartphone</i> . Con los datos obtenidos se formularon algoritmos para filtrar mediciones y brindar una solución segura y objetiva.	
Prabhu et al. [37]	Suspender la ejecución de la actividad física recomendada por especialistas a pacientes con frecuencia cardíaca irregular.	Una aplicación móvil que se comunica con sensores de movimiento para inspeccionar la actividad física del paciente.	Fitbit, sensor Shimer, SMS.	La aplicación móvil permite a los usuarios elegir el tipo de ejercicio a realizar, la aplicación monitorea la actividad física, en caso de que el paciente se detenga, recibe un mensaje de texto como motivación.	Finalizado
Nashif et al. [26]	Detección, diagnóstico y seguimiento de enfermedades cardíacas.	Un sistema de predicción de enfermedades cardíacas basado en la nube. Uso de una técnica de aprendizaje automático que se deriva de algoritmos en Java. Con este trabajo se monitorea a los pacientes en tiempo real utilizando Arduino.	Base de datos en la nube, algoritmos en Java, Arduino, sensores, WEKA, GSM ( <i>Global Positioning System</i> , Sistema de Posicionamiento Global).	El sistema se validó utilizando dos bases de datos, donde 10 veces se analizaron las enfermedades cardíacas y se encontró una precisión del 97.53%. También dentro de la aplicación los médicos visualizaron datos de los sensores y comenzaron transmisiones en vídeo con el paciente.	Finalizado

Lang [34]	El monitoreo óptico de recursos humanos a través de PPG (Photoplethysmogram) actualmente no proporciona resultados precisos y confiables bajo circunstancias reales del mundo.	Una descripción concisa de la frecuencia cardiaca en la monitorización con la tecnología óptica, el estado del arte e investigación, así como los escenarios del mundo real en los que aplican.	No especificado.	Una examinación relevante y las posiciones de agencias reguladoras de aplicaciones con la salud, el estado físico y <i>wearables</i> .	En proceso
Moreno- Alsasua et al. [27]	Prevención primaria de síntomas cardiovasculares en personas de diferentes edades y condiciones.	Una plataforma para la prevención primaria de síntomas cardiovasculares. Una interfaz de usuario para garantizar fácil acceso a las funciones de uso, específicamente a un calendario de registro y una visualización de datos obtenidos de los sensores.	<i>Bluetooth</i> , CoreData, SQL, iPad, iOS App, sensores, protocolo OAuth, iOSChart, JSON.	La plataforma se comunicó con sensores para obtener y registrar datos a un calendario, brindó un seguimiento a los pacientes y mostró una visualización de los registros capturados.	En proceso
Lobelo et al. [38]	La seguridad que provee el uso de aplicaciones enfocadas a la evaluación, el asesoramiento y la actividad física.	Un modelo que proporciona la recopilación de datos de la actividad física, en aplicaciones de salud móvil y monitores de actividad ( <i>wearables</i> ).	<i>Wearables</i> , mHealth, Medline, PudMed.	Una revisión de aplicaciones de <i>software</i> de mHealth para apoyar a la evaluación, el asesoramiento y la actividad física.	Finalizado

<p>Yang et al. [33]</p>	<p>Monitorización de datos de ECG en personas con problemas cardiacos.</p>	<p>Método de comunicación entre dispositivos inteligentes basado en técnicas de IoT. Almacenamiento de datos en la nube y uso de protocolos HTTP y MQTT para consultar los registros en tiempo real.</p>	<p>WiFi, datos en la nube, protocolos HTTP y MQTT, HTML.</p>	<p>La aplicación recopiló datos del ECG utilizando un nodo de monitoreo portátil, los cuales se transmitieron directamente a la nube mediante WiFi. Los experimentos fueron realizados en voluntarios sanos para verificar la fiabilidad de todo el sistema, los resultados demostraron que esta propuesta puede ayudar en el diagnóstico primario de ciertas enfermedades del corazón.</p>	<p>Finalizado</p>
<p>De la Torre-Díez et al. [42]</p>	<p>Darle seguimiento al tratamiento que los médicos brindan a los pacientes con enfermedades cardiovasculares.</p>	<p>Diseño y desarrollo de una aplicación llamada HeartKeeper, para Android, que contiene una sección informativa, una sección de registro de actividades y una sección de registro de medicamentos.</p>	<p>Android SDK, Eclipse IDE, ADT (<i>Abstract Data Type</i>, Tipo de Datos Abstractos), SHA.</p>	<p>Una aplicación que llevó un seguimiento del tratamiento que los médicos les asignaron a sus pacientes con enfermedades cardiovasculares.</p>	<p>Finalizado</p>

Jain y Bhaumik [30]	Detectar enfermedades y dar seguimiento a síntomas relacionadas con la frecuencia cardiaca.	Un algoritmo de diagnóstico de enfermedades para la señal de ECG.	Android, ASIC, PhysionetPTB.	Las salidas de datos de los ASIC se enviaron a una aplicación de Android, con la información se generaron diagnósticos mediante el algoritmo. Los resultados fueron enviados a los cardiólogos a través del correo electrónico.	En proceso
Azariadi et al. [31]	Diagnóstico de problemas cardiacos, análisis y clasificación de ECG para detectar irregularidades en la frecuencia cardiaca.	Un algoritmo para el análisis y clasificación de ECG para el diagnóstico de enfermedades cardiacas. Una plataforma integrada basada en IoT de dispositivos Intel.	DWT ( <i>Deadweight Tonnage</i> ), dispositivos Intel, tablero Galileo, MIT-BIH, Matlab.	El <i>software</i> admitió señales de ECG para extraer características y realizar la clasificación correspondiente para el diagnóstico de problemas cardiacos.	En proceso
Urrea et al. [25]	Crecimiento de la tecnología de mHealth para tratar enfermedades cardiovasculares y el potencial que pueden tener en el futuro.	Un estudio organizado de tecnologías y empresas que usan aplicaciones para dar seguimiento a las enfermedades cardiovasculares, incluyendo fumar, actividad física y dietas poco saludables.	No especificado	Una revisión de artículos que tratan sobre el uso de la tecnología mHealth dirigidas a enfermedades cardiovasculares. Demostrando que el uso de las tecnologías beneficia el tratamiento y seguimiento de las enfermedades.	Finalizado

<p>Alshraideh et al. [29]</p>	<p>Detección de personas con problemas de arritmias cardiacas.</p>	<p>Un algoritmo de detección de las enfermedades cardiovasculares abstractas que clasifica los valores recuperados y muestra los resultados en una aplicación Android.</p>	<p>Minería de datos, clasificador CVD (<i>Cardiovascular Disease</i>, Enfermedad Cardiovascular), WEKA, RNA, Android, SMS.</p>	<p>Recopilación de datos en la aplicación móvil y clasificación de la información mediante un algoritmo que implementa el <i>software</i> WEKA. Los registros fueron almacenados en un servidor y se consultaron en la aplicación, en el caso de detección de una irregularidad se alertó al médico con un SMS.</p>	<p>Finalizado</p>
<p>Ave et al. [28]</p>	<p>Detección temprana de enfermedades cardiovasculares con señal de fotobismograma.</p>	<p>Un dispositivo que detecta la señal fotobismograma, envía datos al teléfono celular mediante <i>Bluetooth</i> y una aplicación móvil para consultar datos.</p>	<p><i>Bluetooth</i>, Android, Android SQLite Database, Arduino.</p>	<p>El dispositivo se comunicó con la aplicación móvil para detectar enfermedades cardiovasculares. Los resultados fueron presentados en código de colores (rojo, verde y amarillo).</p>	<p>Finalizado</p>
<p>Eyles et al. [41]</p>	<p>Consumo de alimentos con alto nivel de sal, que perjudica la salud.</p>	<p>Una aplicación que muestra el nivel de sal contenido en los productos, la interfaz simula un semáforo como</p>	<p>No especificado</p>	<p>La aplicación móvil permitió el escaneo de código de barras de los productos existentes en el mercado. Brindó datos que perjudican la salud,</p>	<p>En proceso</p>

		advertencia y brinda sugerencias saludables.		especialmente el nivel de sal y sugirió productos saludables.	
--	--	--	--	---	--

De acuerdo a estos autores el uso de la tecnología en el cuidado de la salud ha proporcionado un cambio en el bienestar de las personas que lo utilizan, por lo que en este proyecto de tesis se propone a los usuarios el cuidado de su salud y la prevención de enfermedades cardiovasculares, sugerencias para empezar a cuidar sus actividades diarias y un apoyo a los pacientes que tengan un foco rojo en ECV. Por lo que la herramienta que se genere tendrá aplicabilidad en la población.

A diferencia de los trabajos revisados, el proyecto que se propone abordará el desarrollo de una aplicación móvil multidispositivo enfocada en IoT, de manera que llegue a mayor número de usuarios. Como se observa en la revisión, los autores se enfocaron en el seguimiento o detección de síntomas de enfermedades cardiovasculares, pero este proyecto se enfatizará en la detección temprana con el uso de wearables existentes en el mercado, para evitar que la persona llegue a obtener alguna complicación con su salud, es decir, prevenirlo desde el momento en que empiece a utilizar la herramienta brindada y también para apoyarlo a cambiar el estilo de vida.

## 2.3 Propuesta de solución

Para dar solución al problema planteado se realizó un estudio minucioso de las tecnologías que apoyarán al desarrollo de la aplicación para dispositivos móviles etapa a etapa, por lo que se analizaron los *Frameworks* React, AngularJS, Node JS y Vue.js, entre ellos se destaca React por sus características aplicadas al desarrollo de aplicaciones móviles nativas, híbridas y *Web* progresivas.

Entre las metodologías de desarrollo se seleccionó Mobile-D por las etapas que tiene, empleadas especialmente para el desarrollo de soluciones en forma de aplicaciones móviles, se toman en cuenta los recursos y el tiempo disponible para el trabajo. Los lenguajes de programación que se eligieron son: PHP, *JavaScript* y para el diseño, la implementación de CSS. Para la codificación se seleccionó la herramienta de desarrollo Visual Studio Code, por su potente soporte a varios lenguajes.

Para seleccionar el dispositivo de monitorización, se revisaron las que actualmente se encuentran en el mercado, se tomó en cuenta las variables que manejan, el costo y el acceso a los datos.

Los sistemas gestores de base de datos analizados fueron MongoDB, Apache Cassandra y PostgreSQL, con la finalidad de seleccionar la adecuada para la manipulación de datos.

A continuación, se describe la justificación de cada elección.

### 2.3.1 React

React es un *Framework JavaScript* para la creación de componentes visuales del lado del cliente, fue creada por Jordan Walke, Ingeniero de Software de Facebook, en marzo de 2013 y actualmente es un software libre que se usa en cientos de páginas y aplicaciones de primer nivel. El desarrollo con React se basa en componentes, una arquitectura que permite una reutilización del código y la realización de aplicaciones complejas más sencillas de programar y de mantener [2].

React esta siendo una herramienta muy usada en el área de desarrollo para el lado del cliente, empezó en Facebook pero está migrando a más proyectos.

El rendimiento es su principal característica, motivo por el cual ha sido seleccionada para multitud de proyectos de gran calidad, además de que cuenta con una arquitectura mantenible y un desarrollo ágil y ordenado. Entre otras innovadoras ideas de React para mejorar el rendimiento de las aplicaciones, está la incorporación de un DOM virtual (*Document Object Model*, Modelo de Objeto de Documento).

### 2.3.1.1 Características

- Eficiencia de tiempo. Puede visualizar rápidamente lo que crea.
- Los componentes confeccionados aumentan la velocidad de desarrollo.
- Aplicaciones confiables y estables.
- Capacidad para apuntar a múltiples sistemas operativos con menos código.
- DOM virtual que permite organizar documentos en formatos HTML, XHTML (*eXtensible HyperText Markup Language*, Lenguaje Extensible de Marcado de Hipertexto) o XML (*eXtensible Markup Language*, Lenguaje de Marcado Extensible) en un árbol desde el cual es mejor aceptado por los navegadores web al analizar diferentes elementos de la aplicación *Web*.
- Biblioteca de *JavaScript* de código abierto 100%.

### 2.3.2 Metodología de desarrollo Mobile-D

Una metodología diseñada para el desarrollo de aplicaciones móviles, recibe el nombre de Mobile-D y es propuesta por Pekka Abrahamsson y su equipo del VTT (*Valtion Teknillinen Tutkimuskeskus*, en inglés *Technical Research Centre of Finland*) quienes se encuentran en Finlandia y que lideran una corriente importante sobre el desarrollo ágil dedicada a las aplicaciones móviles.

Mobile-D [45], consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de las fases se despliega en etapas, cumpliendo las diferentes tareas y prácticas asociadas. En la primera fase, explorar, el equipo de desarrollo debe generar un plan y establecer las características del proyecto. Para llevar a cabo el objetivo se trabaja con tres etapas: establecer actores, definir alcance y establecer proyectos. Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planeación inicial del proyecto y los requisitos, y el establecimiento de procesos.

En la fase de iniciación, los desarrolladores alistan todos los recursos que se van a ocupar durante el trabajo. Se diseñan y preparan los planes para las otras fases y se constituye el entorno técnico, tales como recursos físicos, tecnológicos y de comunicaciones (incluyendo la capacitación del equipo de trabajo). Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

En la fase de producción hay una iteración de actividades que consta de planificación, trabajo y liberación, que se ejecuta desde tres días hasta lo necesario para implementar todas las funcionalidades.

Primero se describe el ciclo de trabajo de acuerdo a los requisitos y tareas a elaborar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el

último día se lleva a cabo la integración del sistema (en caso de que se hayan realizado por equipos o de manera independiente) continuando con las pruebas de funcionalidad.

En la fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipos con diferentes subsistemas desarrollados por equipos distintos. En esta fase se realiza la integración de todo el sistema, generando pruebas necesarias como en la fase de “Producción”, solo que aquí ya se construye la solución de manera completa. Adicionalmente se puede considerar en esta fase la producción de documentación.

Por último, la fase de prueba y reparación, tiene como objetivo lograr una versión estable y funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

La estructura y flujo de las fases se representa en la Figura 2. 1:

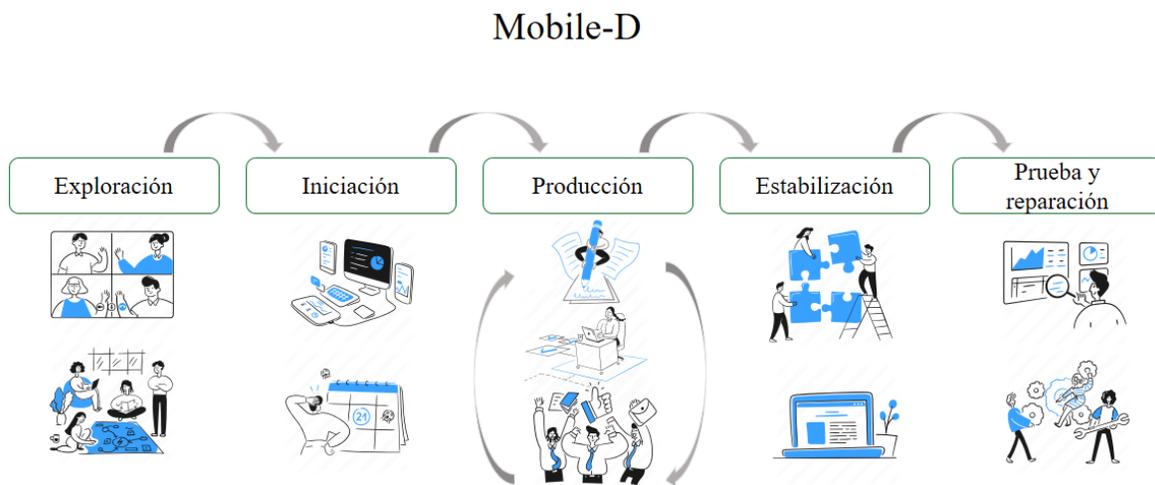


Figura 2. 1 Metodología de desarrollo Mobile-D

### 2.3.3 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código [46].

Visual Studio Code está basado en Electron, un marco de trabajo para implementar Chromium y Node.JS como software de escritorio, que se ejecuta en el motor de diseño Blink. El 18 de noviembre de 2015, Visual Studio Code fue lanzado bajo la licencia MIT (*Massachusetts Institute of Technology*) y su código fuente fue publicado en GitHub.

Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado.

Visual Studio Code puede extenderse a través de módulos que se almacenan en un repositorio, esto incluye idiomas, bibliotecas, entre otras. Una característica que lo distingue es el uso de *linters*, extensiones para analizar código, y herramientas para análisis estático, utilizando el Protocolo de Servidor de Idioma.

#### 2.3.3.1 *Características*

- Es una herramienta que ayuda a detectar y auto-completar código.
- Iniciar las aplicaciones y depurar con puntos de interrupción, pilas de llamadas y con una consola interactiva.
- Viene enlazado con Git directamente.
- Instalar las extensiones para el lenguaje de programación que se use.

#### 2.3.4 **Wearable**

Fitbit Charge 4 es una pulsera cuantificadora semejante a Fitbit Charge 3 que cuenta con sensores de: acelerómetro de tres ejes, monitor óptico del ritmo cardiaco, GPS (*Global Positioning System*, Sistema de Posicionamiento Global), motor de vibración, chip NFC para Fitbit Pay y altímetro [47].

Funciones del *wearable*:

- Monitorización continua del ritmo cardiaco.
- Ritmo cardiaco en reposo.
- Zonas de frecuencia cardiaca, cardio y pico, para optimizar el esfuerzo.
- Mapa de intensidad del entrenamiento.
- Modos de ejercicio basados en objetivos.
- GPS integrado.
- Resistencia al agua.
- Sesiones de respiración guiada.
- Notificaciones y aplicaciones.
- Fases y puntualización del sueño.
- Fitbit Pay, para realizar pagos.
- Monitorización de la actividad diaria.
- Sincronización inalámbrica con IOS y Android.

### 2.3.5 Descripción de la solución

Para solucionar el problema se propone desarrollar una aplicación para dispositivos móviles que realice una detección temprana de las ECV con la implementación de un modelo de predicción; la propuesta consiste en la implementación y comunicación de módulos. En la Figura 2. 2 se representa la comunicación de las capas.

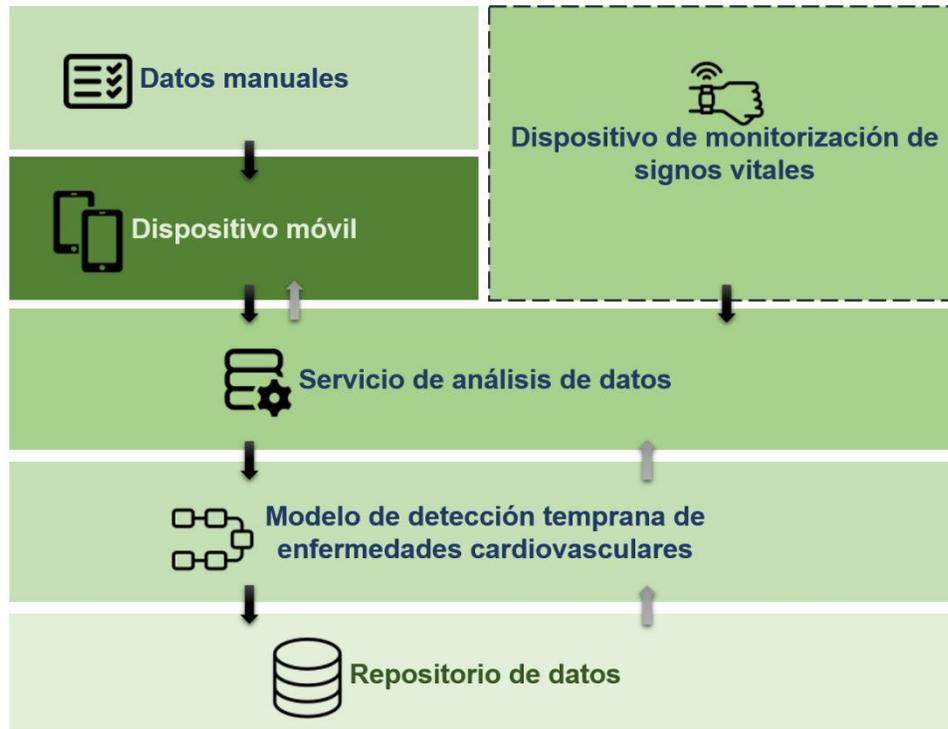


Figura 2. 2 Arquitectura de la propuesta de solución

#### 2.3.5.1 Módulo de recopilación de datos

En este módulo se obtendrán los datos de la API que usan los dispositivos de monitorización de signos vitales, los más importantes son la frecuencia cardiaca, movimientos físicos y presión arterial.

Posteriormente complementar las variables con la consulta de datos desde la base de conocimientos con apoyo de una API, correspondiente a: nivel de colesterol HDL y LDL, glucosa, medidas de cintura, estatura e índice de masa corporal; de manera manual se rescatarán mediante un formulario los datos del examen de electrocardiograma.

### 2.3.5.2 *Módulo de detección temprana de enfermedades cardiovasculares*

En este módulo se consumirán los datos almacenados en el repositorio, para manipularlos mediante el uso de un modelo de detección temprana. Con el procedimiento se pretende obtener el porcentaje de probabilidad que tiene una persona en contraer alguna ECV.

Los resultados definirán un porcentaje global y un porcentaje parcial por cada enfermedad cardiovascular relacionada con los datos del paciente.

### 2.3.5.3 *Módulo de presentación en dispositivos móviles*

Después de analizar los datos y generar el porcentaje de probabilidad a obtener una ECV, se mostrarán los datos en la aplicación. Se contempla también notificar a la persona sobre las irregularidades en su salud, brindándole en la aplicación sugerencias para el cuidado de su bienestar y de esta manera poder prevenir alguna enfermedad cardiovascular.

## Capítulo 3. Aplicación de la metodología

En este capítulo se describe el desarrollo de la aplicación para dispositivos móviles siguiendo las fases que componen la metodología Mobile-D mostrada en el capítulo 2.

### 3.1 Metodología de desarrollo

A continuación, se describe el proceso de las cinco fases de la metodología para el desarrollo de la aplicación para dispositivos móviles para la detección temprana de ECV.

#### 3.1.1 Exploración

Esta fase es la primera de la metodología por lo que se revisan todos los aspectos desde necesidades tecnológicas hasta de actores, con la finalidad de alistar todo antes de proceder al desarrollo.

##### 3.1.1.1 Establecimiento de actores

Los actores principales que participarán en el proyecto se describen en la siguiente Tabla 3.1.

Tabla 3.1 Establecimiento de actores

	Características	Funciones
 Desarrolladores	Persona que utiliza las habilidades y conocimientos que tiene para generar software.	Recaudar requerimientos del proyecto. Diseñar la aplicación para dispositivos móviles. Desarrollar la aplicación para dispositivos móviles.
 Expertos en la salud	Personal con capacidades y conocimientos de la salud principalmente en el área de enfermedades cardíacas.	Aprobar las variables que determinarán la probabilidad de tener una enfermedad cardiovascular. Aprobar los resultados que brindará la aplicación.

	Características	Funciones
 <p>Personas con síntomas y asintomáticas</p>	<p>Persona que quiere conocer su estado de salud y el riesgo que tiene sobre las enfermedades cardiovasculares, puede tener síntomas o no.</p>	<p>Brindar datos de salud para realizar test de probabilidad de ECV. Utilizar las funcionalidades de la aplicación para probar cada una de las opciones. Calificar la funcionalidad de la aplicación para dispositivos móviles.</p>

### 3.1.1.2 Definición del alcance

Se llevará a cabo un extractor de datos de signos vitales y biomarcadores monitorizados mediante *wearables* y se contempla en caso de ser necesario, la adquisición manual de datos como análisis clínicos y de electrocardiograma, dichos datos serán analizados, procesados y encapsulados en servicios Web para ser consumidos a través de una aplicación para dispositivos móviles con el propósito de alertar a los usuarios asintomáticos sobre la posibilidad de sufrir alguna enfermedad cardiovascular.

### 3.1.1.3 Requerimientos

En las Tabla 3.2 a la Tabla 3.5 se enlistan los requerimientos funcionales analizados en la reunión que se ha realizado con el cliente.

Tabla 3. 2 Requerimiento funcional RF01

<b>Requerimiento funcional:</b> RF01	<b>Prioridad:</b> Alta
<b>Nombre:</b> Detectar una enfermedad cardiovascular.	
<b>Descripción:</b> Realizar una detección temprana de enfermedades cardiovasculares.	

Tabla 3. 3 Requerimiento funcional RF02

<b>Requerimiento funcional:</b> RF02	<b>Prioridad:</b> Alta
<b>Nombre:</b> Dar una probabilidad exacta de las enfermedades cardiovasculares.	
<b>Descripción:</b> El resultado debe proporcionar una probabilidad exacta sobre el riesgo de padecer una ECV.	

Tabla 3. 4 Requerimiento funcional RF03

<b>Requerimiento funcional:</b> RF03	<b>Prioridad:</b> Media
<b>Nombre:</b> Dar información de las ECV.	
<b>Descripción:</b> La aplicación deberá proporcionar información de las posibles ECV que una persona puede contraer a determinado tiempo.	

Tabla 3. 5 Requerimiento funcional RF04

<b>Requerimiento funcional:</b> RF04	<b>Prioridad:</b> Alta
<b>Nombre:</b> Brindar sugerencias para el cuidado de la salud.	
<b>Descripción:</b> Brindar sugerencias de acuerdo a los resultados obtenidos del estudio de probabilidad.	

También se analizaron los requerimientos no funcionales para el presente proyecto de tesis, las cuales se describen en las Tabla 3.6 a la Tabla 3.8.

Tabla 3. 6 Requerimiento no funcional RNF01

<b>Requerimiento no funcional:</b> RNF01	<b>Prioridad:</b> Alta
<b>Nombre:</b> Acceso personalizado	
<b>Descripción:</b> Contar con un acceso restringido y personalizado en la aplicación.	

Tabla 3. 7 Requerimiento no funcional RNF02

<b>Requerimiento no funcional:</b> RNF02	<b>Prioridad:</b> Alta
<b>Nombre:</b> Brindar resultados de la probabilidad de contraer una ECV.	
<b>Descripción:</b> Brindar resultados de manera gráfica acompañada de una descripción, acerca de las ECV que se pueden contraer.	

Tabla 3. 8 Requerimiento no funcional RNF03

<b>Requerimiento no funcional:</b> RNF03	<b>Prioridad:</b> Alta
<b>Nombre:</b> Uso de modelos predictivos para asegurar resultados eficientes.	
<b>Descripción:</b> Para obtener resultados eficientes y brindarlos a los usuarios, es necesario el uso de un modelo predictivo.	

Con el análisis de la participación de los actores y de los requerimientos se da paso al diseño del diagrama de casos de uso, en la que se han conjuntado todas las acciones que se realizarán dentro de la aplicación.

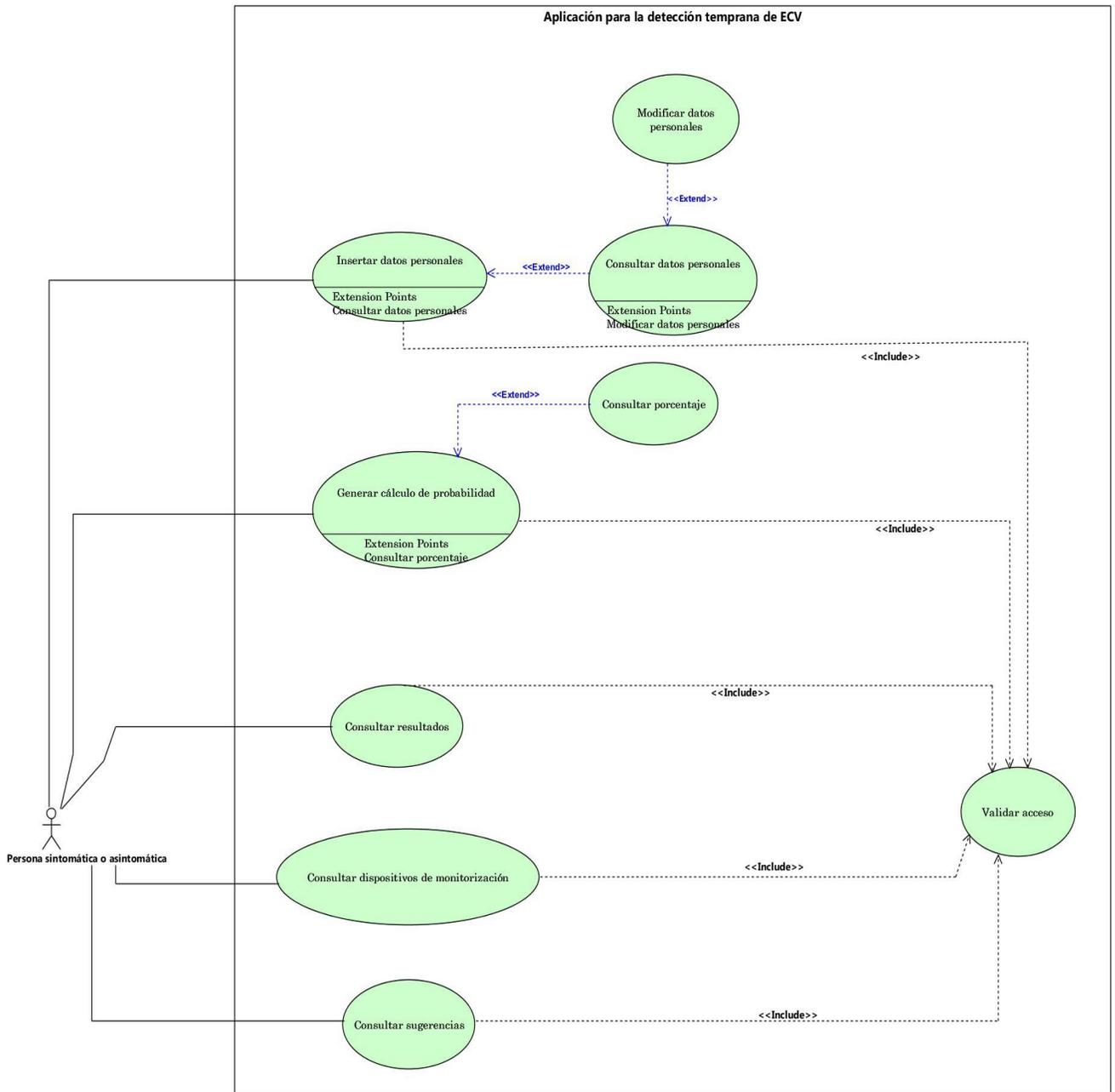


Figura 3. 1 Diagrama de casos de uso

3.1.1.4 *Establecimiento de variables para la detección de ECV*

Como primer paso se deben seleccionar las variables que permitan la valoración general de enfermedades cardiovasculares, que proporcionen resultados parciales, así como un acercamiento a alguna ECV en particular. En primera instancia se seleccionaron 24 biomarcadores que se describen en la Tabla:

*Tabla 3. 9 Biomarcadores para la detección de enfermedades cardiovasculares*

<b>Número</b>	<b>Nombre</b>	<b>Rango</b>
1	Edad	18 a 98
2	Sexo	1,0
3	Fumador	1,0
4	Alcoholismo	1,0
5	Obesidad	1,0
6	Hipertenso	1,0
7	Enfermedad renal	1,0
8	Anemia	1,0
9	Frecuencia cardiaca	60 a 150
10	Presión sistólica	60 a 180
11	Presión diastólica	50 a 120
12	Triglicéridos	100 a 500
13	Colesterol LDL	80 a 160
14	Colesterol HDL	40 a 60
15	Colesterol total	100 a 240
16	Glucosa	80 a 500
17	Creatinina	0 a 3
18	Rest_ecf	0,1
19	Old_peak	0 a 4
20	Potasio en sangre	3.5 a 5.2
21	Calcio en sangre	7 a 11
22	Sodio en sangre	135 a 145
23	Taquicardia sinusal ecg	60 a 150

Número	Nombre	Rango
24	Ejection fraction	20 a 60

De los datos iniciales, se consultarán datos desde la base de conocimientos; en la Tabla 3.10 se describen cada uno de ellos.

*Tabla 3. 10 Selección de variables*

Nombre	Variable	Descripción
Edad	Age	En años
Sexo	Sex	1 = M, 0 = F
Colesterol HDL	Chol	En mg/dl
Glucosa	Glucose	En mg
Índice de masa corporal	IMC	Kg/m
Medida de la cintura	Waist	En cm
Colesterol LDL	Chol	En mg/dl
Triglicéridos	Tri	En mg/dl
Presión sistólica	Presis	l/s
Presión diastólica	Predias	l/s
Antecedentes familiares con enfermedades	ant	0 = No, 1 = Sí
Actividad física	Physical	En minutos
Presión sanguínea	Blood_Pressure	En Mm
Frecuencia cardiaca	Heart_Rate	Frecuencia máxima

Toda la información se agrupa en dos formas de extracción organizados en la siguiente tabla.

*Tabla 3. 11 Tipo de recuperación de datos*

<b>Automático (<i>wearable</i>)</b>	<b>Semi-automático</b>
Actividad física	Edad
Presión sanguínea	Sexo
Frecuencia cardíaca	Colesterol HDL
	Glucosa
	Índice de masa corporal
	Antecedentes familiares
	Colesterol LDL
	Presión sistólica
	Presión diastólica
	Triglicéridos

Para realizar la detección temprana de ECV es necesario ingresar manualmente los datos del análisis de electrocardiograma, ver la siguiente Tabla 3.12.

*Tabla 3. 12 Datos manuales*

<b>Nombre</b>	<b>Rango</b>
Edad	1 a 98
Creatinina	0 a 3
Old_peak	0 a 4
Rest_ecf	0,1

3.1.1.5 *Procesos*

A continuación se presenta el calendario de actividades para desarrollar la aplicación para dispositivos móviles.

Tabla 3. 13 Lista de procesos

No.	Actividad
1	Analizar tipos de enfermedades que se pueden detectar de manera temprana.
2	Analizar tipos de variables usados en la detección de enfermedades cardiovasculares.
3	Analizar los modelos y algoritmos que realizan una detección o diagnóstico de enfermedades relacionadas con el corazón.
4	Realizar pruebas de efectividad de los modelos analizados para seleccionar el modelo con mayor puntaje.
5	Analizar los <i>wearables</i> que proveen los datos para el análisis de enfermedades y seleccionar la que es óptima para la finalidad del proyecto.
6	Diseñar la arquitectura del módulo de detección temprana de enfermedades cardiovasculares.
7	Diseñar el modelo de datos que almacenará la información de los biomarcadores.
8	Realizar los diseños gráficos de la aplicación mediante Mockups, contemplando matrices de diseño para aplicaciones enfocadas a la salud.
9	Desarrollar la codificación de cada una de las interfaces diseñadas.
10	Implementación del módulo de extracción de datos de los <i>wearables</i> .
11	Implementar el módulo de captura de variables para la detección temprana de enfermedades cardiovasculares.
12	Desarrollar las API's para la conexión a la base de conocimientos con la implementación de las siguientes operaciones: insertar, consultar, modificar y eliminar.
13	Implementar el módulo de tratamiento de datos mediante los modelos seleccionados.
14	Implementación del consumo de las API's generadas dentro de las interfaces gráficas de la aplicación.
15	Desarrollo del módulo de presentación de resultados en la aplicación móvil.

No.	Actividad
16	Realización de pruebas funcionales de la aplicación en al menos un caso.

Las actividades se organizan en el siguiente diagrama de Gantt.

Tabla 3. 14 Diagrama de Gantt

Actividades		2021									
		Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Sep.	Oct.	Nov.
1	P	■									
	R	■									
2	P	■	■								
	R	■	■								
3	P		■	■							
	R		■	■					■		
4	P			■							
	R			■					■		
5	P				■						
	R				■						
6	P				■						
	R				■						
7	P					■					
	R					■					
8	P						■				
	R						■	■			
9	P							■			
	R							■	■		
10	P							■			
	R							■			

Actividades		2021									
		Feb.	Mar.	Abr.	May.	Jun.	Jul.	Ago.	Sep.	Oct.	Nov.
11	R										
	P										
12	R										
	P										
13	R										
	P										
14	R										
	P										
15	R										
	P										
16	R										
	P										

### 3.1.2 Iniciación

Para llevar a cabo el desarrollo del proyecto se necesitan los recursos que a continuación se describen.

### 3.1.2.1 Establecimiento de recursos físicos

Los recursos físicos principales son el Fitbit Charge 4, el teléfono inteligente y la laptop. Las actividades a realizar con cada uno se representan en las siguientes figuras.

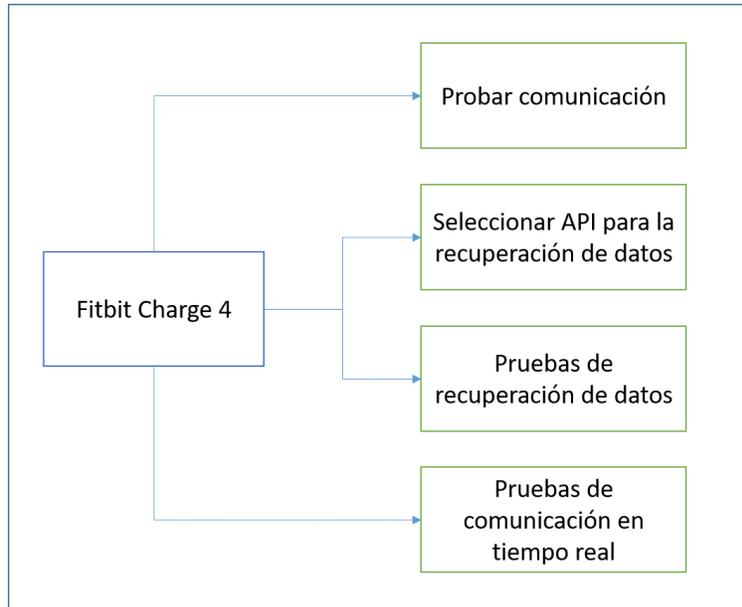


Figura 3. 2 Recursos físicos: Fitbit

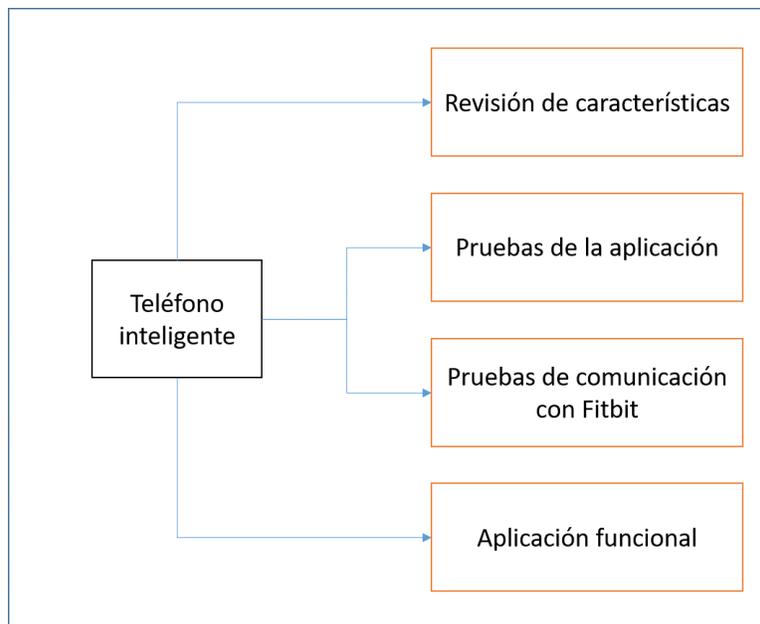


Figura 3. 3 Recursos físicos: teléfono inteligente

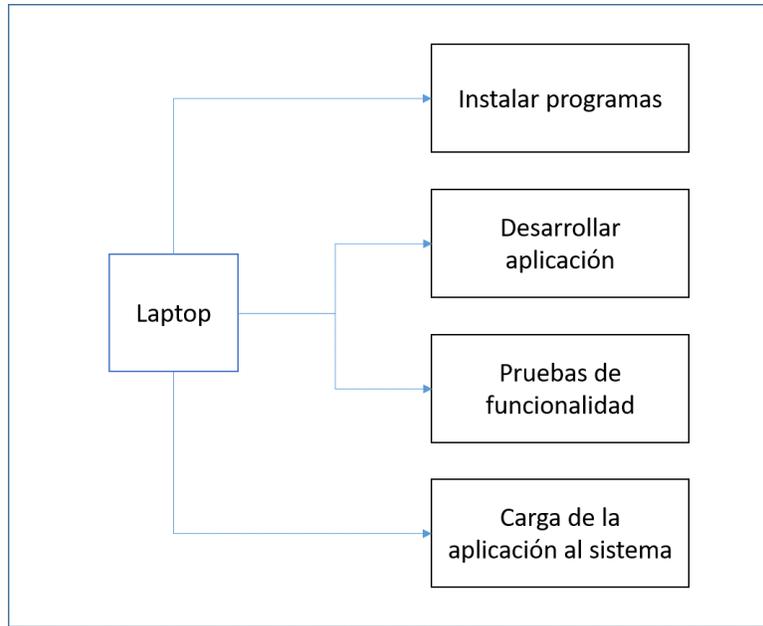


Figura 3. 4 Recursos físicos: Laptop

### 3.1.2.2 Establecimiento de comunicación

Para representar la secuencia de comunicación de cada componente en este proyecto, se diseñaron los siguientes diagramas de actividad en la que se describe la interacción.

En la Figura 3.5 se muestra la interacción entre el usuario, la aplicación móvil y la base de conocimientos. Antes de iniciar sesión, el usuario debe crear una cuenta, una vez que se almacenan los datos, permite el ingreso solo con nombre de usuario y contraseña, los cuáles se validan en la base de conocimientos.

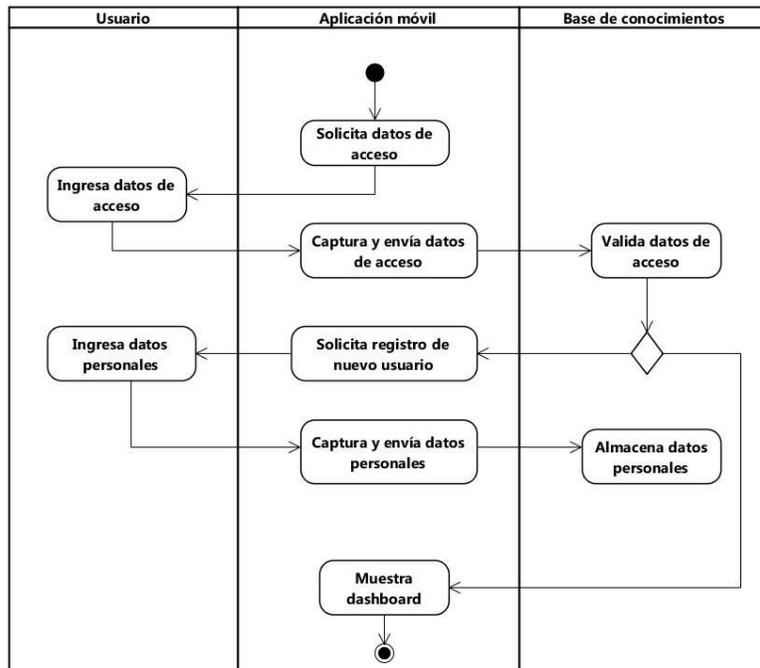


Figura 3. 5 Diagrama de actividad para el registro de usuario

En la Figura 3.6 se visualiza la comunicación entre el usuario, la aplicación móvil y la base de conocimientos, para el caso de realizar un nuevo cálculo, como se aprecia al momento en que se ingresan los datos de salud, se realiza una validación antes de almacenarlos. Los datos se resguardan hasta su consulta.

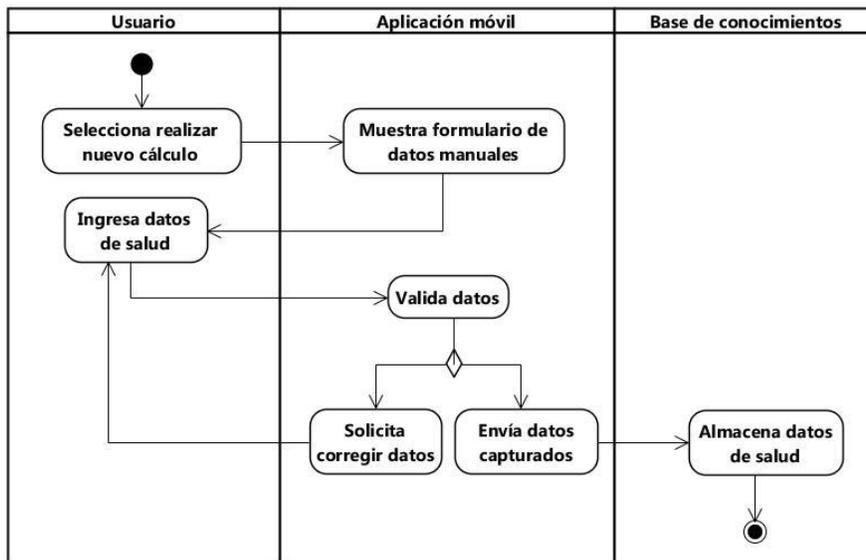


Figura 3. 6 Diagrama de actividad para realizar nuevo cálculo de probabilidad

La Figura 3.7 expone la interacción que se ejerce al momento de consultar resultados del cálculo de probabilidad, para ello aparte del usuario, la aplicación móvil y la base de conocimientos se agrega el modelo de predicción. La participación que resalta es el momento

en que se hace la solicitud y los datos pasan a ser manipulados por el modelo de predicción, en este instante es donde se define la probabilidad de padecer alguna enfermedad cardiovascular, una vez encontrado los resultados se presentan en la interfaz de la aplicación.

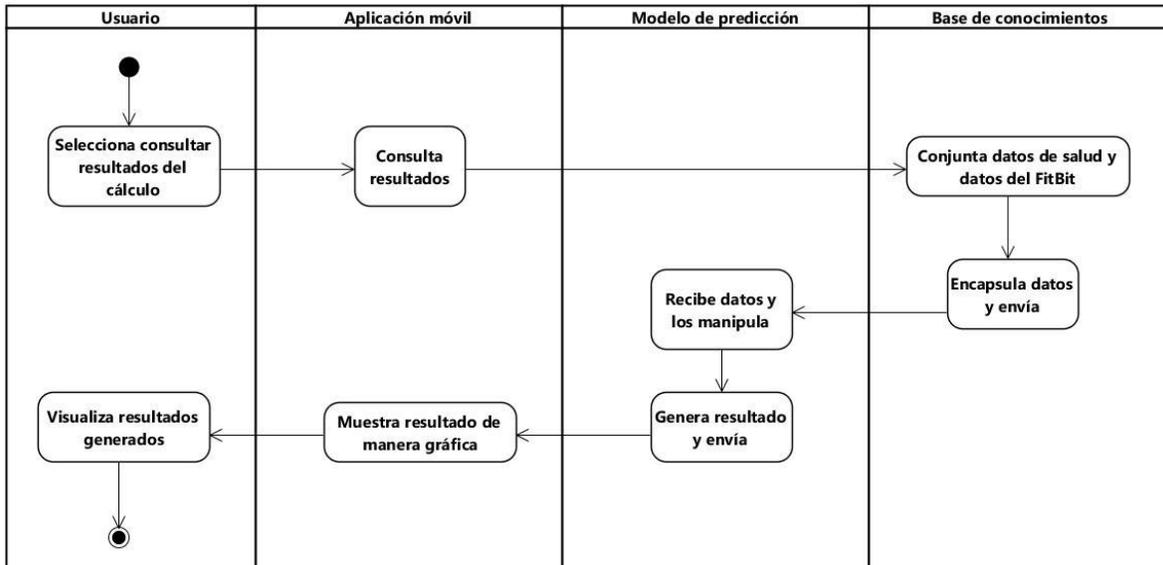


Figura 3. 7 Diagrama de actividad para consultar resultados

### 3.1.2.3 Modelado de datos

Una vez establecidas las variables que se van a recuperar, se realiza un mapeo relacional de objetos (ver Figura 3.8). Los datos se dividen en las siguientes tablas:

**User:** datos personales de los usuarios recuperados desde la aplicación y almacenados para su consulta, se ingresan para crear la cuenta por primera vez y para validar el acceso cada vez que se ingrese a la aplicación.

Tabla 3. 15 Descripción de tabla User

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Iduser	Integer	10	> 0	Identificador del registro.
Name	Varchar	50	Texto libre	Nombre (s) de la persona.
Last_name	Varchar	50	Texto libre	Apellido paterno y materno de la persona.
Email	Varchar	50	Correo válido	Correo electrónico de la persona.
Phone	Varchar	50	10 dígitos	Número de teléfono sin lada.
Birth	Date		Día/mes/año	Fecha de nacimiento.
Adress	Varchar	100	Calle, número, colonia, ciudad,	Dirección completa.

Nombre	Tipo de dato	Tamaño	Rango	Descripción
			municipio y estado.	
Employment	Varchar	50	Ocupación	Ocupación a la que se dedica.
Status	Integer	10	1 = activo, 0 = inactivo.	Estado del usuario registrado.
Username	Varchar	50	Texto libre	Nombre de usuario, para ingresar en la aplicación.
Pass	Varchar	10	Números, dígitos y texto.	Contraseña para ingresar a la aplicación.

**ManualData:** estos datos apoyarán en la detección temprana de enfermedades cardiovasculares, se recuperan con apoyo del usuario dentro de la aplicación, los datos serán tratados por el modelo de detección por lo que su valor solicitado es en números enteros.

Tabla 3. 16 Descripción de tabla ManualData

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idmdata	Integer	10	>0	Identificador del registro.
Age	Integer	10	De 1 a 100	Edad en años.
Sex	Integer	10	Hombre, mujer	Sexo
Chol	Integer	10	Entre 40 y > 60	Colesterol HDL
Glucose	Integer	10	Menos de 70 a > 250	Glucosa
Imc	Integer	10	< 18.5 a > 40	Índice de masa corporal
Waist	Integer	10	< 80 a > 100	Medida de la cintura
Ldl	Integer	10	< 100 a > 190	Colesterol LDL
Pressure	Integer	10	60 a > 120	Presión arterial diastólica
Smoker	Integer	10	1 = sí, 0 = no	Fumador
Heartattack	Integer	10	1 = sí, 0 = no	Síntoma de infarto
Diabetes	Integer	10	1 = sí, 0 = no	Padece diabetes
Palpitations	Integer	10	1 = sí, 0 = no	Palpitaciones por segundo
Fatigue	Integer	10	1 = sí, 0 = no	Síntoma de fatiga
Tired	Integer	10	1 = sí, 0 = no	Síntoma de cansancio
Dizzines	Integer	10	1 = sí, 0 = no	Síntoma de mareo
Chest	Integer	10	1 = sí, 0 = no	Dolor de pecho
Head	Integer	10	1 = sí, 0 = no	Dolor de cabeza
Breaths	Integer	10	1 = sí, 0 = no	Dificultad para respirar
Sickness	Integer	10	1 = sí, 0 = no	Enfermedad relacionada
Abdomen	Integer	10	1 = sí, 0 = no	Dolor de abdomen

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Sweating	Integer	10	1 = sí, 0 = no	Síntoma de sudoración excesiva
Threw	Integer	10	1 = sí, 0 = no	Presentación de vómito
Hyper	Integer	10	1 = sí, 0 = no	Antecedentes familiares de hipertensión
Prediabetes	Integer	10	1 = sí, 0 = no	Antecedentes familiares de diabetes
Iduser	Integer	10	>0	Identificador de la tabla User

**WearableData:** la tercera tabla almacenará datos que se recuperarán automáticamente del dispositivo de monitorización de signos vitales.

Tabla 3. 17 Descripción de tabla WearableData

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idweardata	Integer	10	>0	Identificador del registro
Physical	Integer	10	>0	Actividad física
Bloodpressure	Integer	10	<90 a >180	Presión sistólica
Heartrate	Integer	10	<60 a > 100	Frecuencia cardiaca
Iduser	Integer	10	>0	Identificador de la tabla User

**Contacts:** lista de contactos con quienes se puede acudir para alguna información.

Tabla 3. 18 Descripción de tabla Contacts

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idcontact	Integer	10	>0	Identificador del registro
Name	Varchar	50	Texto libre	Nombre o nombres completos
Mail	Varchar	50	Correo validado	Correo electrónico de la persona
Employment	Varchar	50	Profesión	Ocupación (oficio o profesión)
Relation	Varchar	50	Madre,padre, hijo,hija, médico,otro	Relación que tiene la persona con el usuario.
Iduser	Integer	10	>0	Identificador de la tabla User

**Wearable:** datos del dispositivo de monitorización enlazado con la cuenta de la persona registrada.

Tabla 3. 19 Descripción de tabla Wearable

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idwear	Integer	10	>0	Identificador del registro
Name	Varchar	50	Texto libre	Nombre del dispositivo
Serie	Varchar	50	Letras y números	Número de serie del dispositivo
Model	Varchar	50	Letras y números	Modelo del dispositivo
Date	Date		Día/mes/año	Fecha de enlace con la cuenta del usuario
Character	Varchar	150	Texto	Características del dispositivo
Data	Varchar	150	Texto	Datos que recupera el dispositivo
Descriptions	Varchar	150	Texto	Descripción del dispositivo
Iduser	Integer	10	>0	Identificador de la tabla User

**Suggestions:** tabla que resguarda las sugerencias para las personas dependiendo de los resultados obtenidos en la detección.

Tabla 3. 20 Descripción de tabla Suggestions

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idsuggestion	Integer	10	>0	Identificador del registro
Title	Varchar	150	Texto	Título de la sugerencia
Description	Varchar	150	Texto	Descripción
Date	Date		Día/mes/año	Fecha de registro
Priority	Varchar	50	Bajo,medio,alto	Prioridad de la sugerencia
Status	Integer	10	1=Activo,0=inactivo	Estado
Disease	Varchar	50	Arritmia, hipertensión, taquicardia, infarto, ninguno.	Enfermedad relacionada o enfocada

**PhotoUser:** se almacenan los datos del enlace de la imagen guardada para el perfil de usuario.

Tabla 3. 21 Descripción de tabla PhotoUser

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idpuser	Integer	10	>0	Identificador del registro
Iduser	Integer	10	>0	Identificador de la tabla User
Photo	Varchar	100	Link	Link donde se encuentra la imagen.

**PhotoCon:** se almacenan los datos del enlace de la imagen para el perfil de los contactos.

Tabla 3. 22 Descripción de tabla PhotoCon

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idpcontact	Integer	10	>0	Identificador del registro
Idcontact	Integer	10	>0	Identificador de la tabla Contacts
Photo	Varchar	100	Link	Link donde se encuentra la imagen.

**PhotoWear:** datos de almacenamiento de la imagen de los wearables.

Tabla 3. 23 Descripción de tabla PhotoWear

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idpwear	Integer	10	>0	Identificador del registro
Idwear	Integer	10	>0	Identificador de la tabla Wearable
Photo	Photo	100	Link	Link donde se encuentra la imagen.

**Results:** se guardan los resultados generados por el modelo de predicción de acuerdo al porcentaje de cada enfermedad.

Tabla 3. 24 Descripción de tabla Results

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idresult	Integer	10	>0	Identificador del registro
Number	Integer	10	>0	Número de resultado
Score	Integer	10	>0	Puntaje total
Arrhythmia	Integer	10	0, >0	Resultado obtenido de arritmia
Hyper	Integer	10	0, >0	Resultado obtenido de hipertensión
Angina	Integer	10	0, >0	Resultado obtenido de taquicardia
Heartattack	Integer	10	0, >0	Resultado obtenido de infarto
None	Integer	10	0, >0	Resultado negativo de las ECV
Iduser	Integer	10	>0	Identificador de la tabla User

Nombre	Tipo de dato	Tamaño	Rango	Descripción
Idweardata	Integer	10	>0	Identificador de la tabla WearableData
Idmdata	Integer	10	>0	Identificador de la tabla ManualData

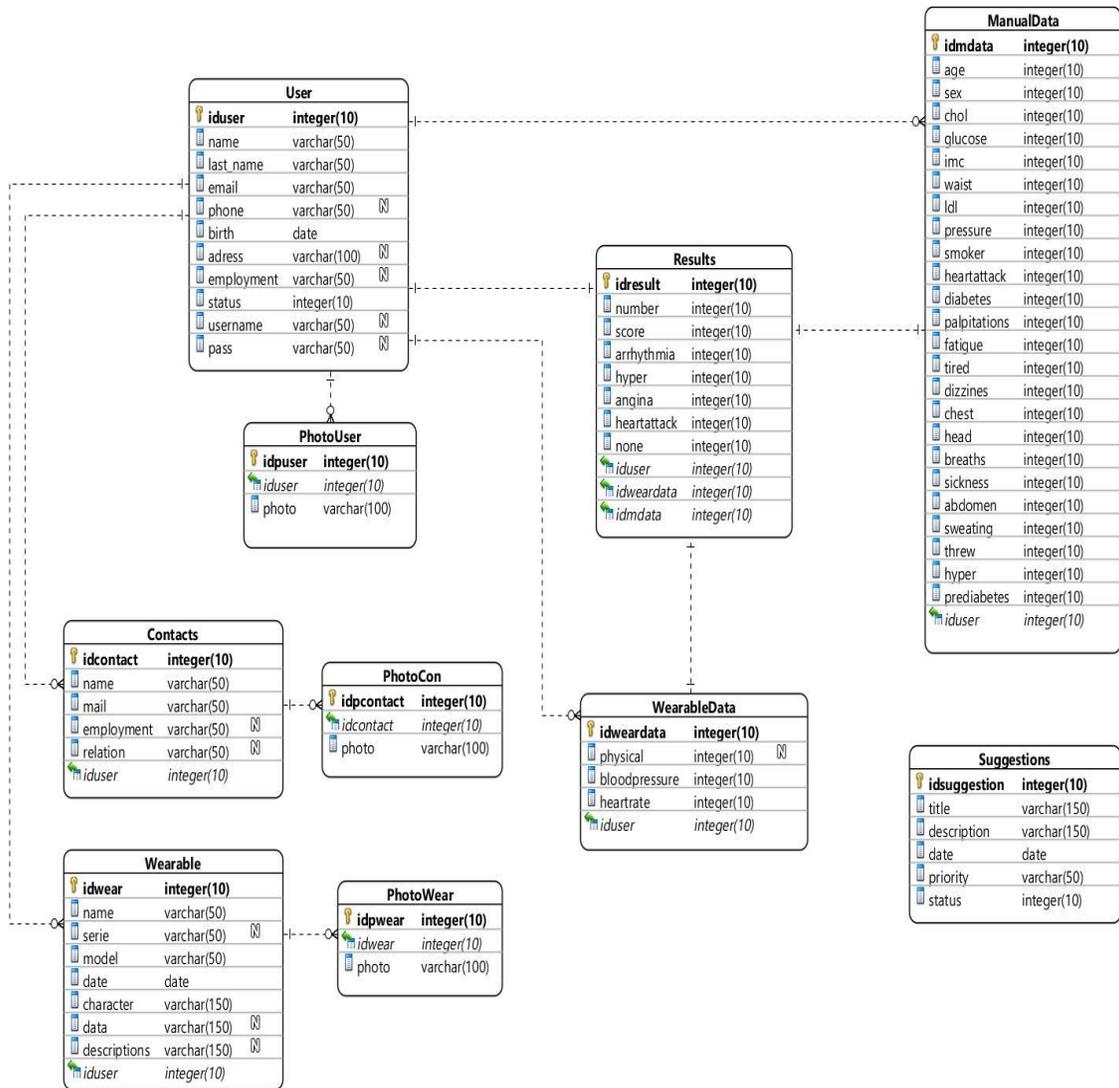


Figura 3. 8 Mapeo Relacional de Objetos

### 3.1.2.4 Modelado de componentes

Todo el proyecto se conforma de componentes que son necesarios para el buen funcionamiento del sistema. En el siguiente diagrama se presentan los componentes que a continuación se describen:

**Sistema operativo en el teléfono:** la aplicación se desplegará en sistemas operativos de Android.

**Aplicación para la detección temprana de ECV:** aplicación desarrollada como principal objetivo de este proyecto.

**SGBD (Sistema Gestor de Base de Datos):** se utilizará PostgreSQL para hacer la gestión de datos a la base de conocimientos.

**Modelo de predicción:** modelo seleccionado de acuerdo a las métricas establecidas, que será consumida en la aplicación mediante los servicios Web.

**Base de conocimientos:** almacenamiento de todos los datos recuperados.

**Módulo de extracción de datos:** ejecutable que recupera los datos de la API y selecciona la información necesaria para almacenarla en la base de conocimientos.

**API (Interfaz de programación de aplicaciones):** proporciona la conexión del Fitbit con el módulo de extracción de datos en formato JSON.

**Fitbit Charge:** dispositivo de monitorización de signos vitales.

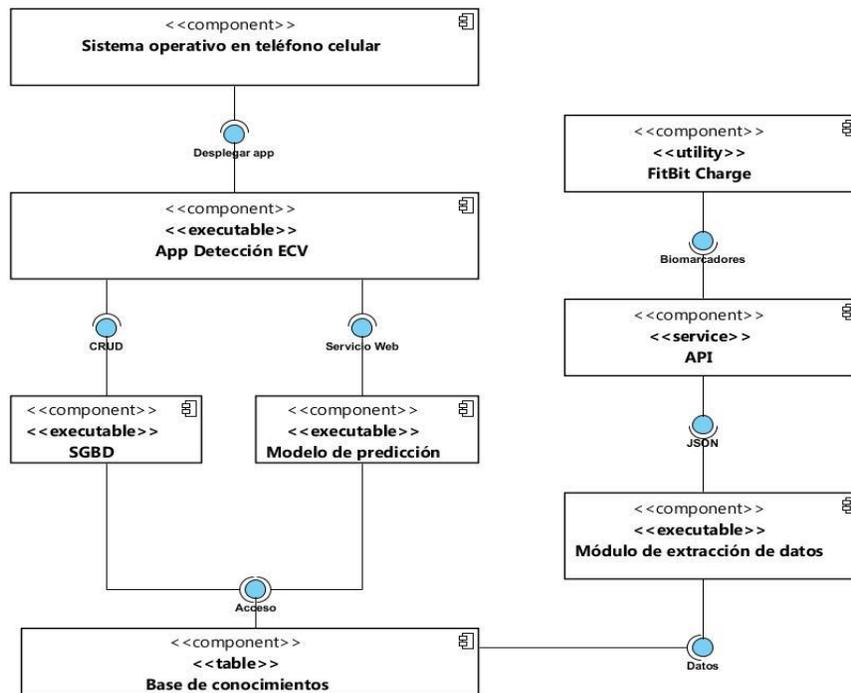


Figura 3. 9 Diagrama de componentes

### 3.1.3 Producción

Esta fase de la metodología trata del desarrollo del software, en este caso se realizará la aplicación para dispositivos móviles. La Figura 3.10 muestra el proceso cíclico de cada entregable la cual consiste en:

1. **Planeación de la tarea:** se diseña y programa la actividad a realizar, principalmente las interfaces de la solución.
2. **Inicio de la tarea:** se da inicio al desarrollo del código para la aplicación.
3. **Desarrollo de la tarea:** se codifica lo necesario para el entregable.
4. **Pruebas:** se realizan las pruebas necesarias hasta generar el resultado esperado y hacer el entregable.



Figura 3. 10 Ciclo de ejecución de entregables

En los siguientes apartados se encuentra el proceso realizado para generar la solución a la problemática de este proyecto.

#### 3.1.3.1 Planeación del estudio de modelos de predicción

Para emplear el mejor modelo de predicción dentro de la solución propuesta se contempla hacer algunas pruebas de exactitud, por lo que se utilizarán las variables seleccionadas en el apartado 3.1.1.4.

El modelo de predicción será entrenado con datos descargados de base de datos en línea, referentes a las enfermedades cardiovasculares, una vez generados los resultados se realizarán evaluaciones en la que se seleccionará el mejor modelo.

Este modelo se implementará para otorgar la detección temprana de ECV a las personas que utilicen la aplicación.

### 3.1.3.2 Inicio del estudio de modelos de predicción

Para las pruebas se seleccionaron los siguientes modelos de predicción:

**Logistic Regression (Regresión logística):** es un modelo estadístico que, en su forma básica, se utiliza una función logística a un modelo binario de la variable dependiente, a pesar de que existen extensiones más complejas. La regresión logística puede ser binomial, ordinal o multinomial. Binomial o de regresión logística binaria se ocupa en las situaciones en las que el resultado observado de una variable dependiente sólo puede estar entre 0 y 1, el primero se refiere a un resultado, por ejemplo “muerto”, el segundo a otro resultado, en este ejemplo a “vivo”. La regresión logística multinomial se ocupa en las situaciones donde el resultado puede tener tres o más tipos posibles que no están ordenadas. Ordinal de regresión logística se ocupa de las variables dependientes que se ordenan [48].

**Support Vector (Soporte vectorial):** es un algoritmo de máquina de aprendizaje supervisado utilizado para la clasificación y regresión. El objetivo del algoritmo SVM es encontrar un hyperplane en un espacio N-dimensional que claramente clasifica los puntos de datos. La dimensión de la hyperplane depende del número de funciones. Si el número de características de entrada es de dos, entonces el hyperplane es sólo una línea. Si el número de características de entrada es de tres, de la hyperplane se convierte en un 2-D del plano. Se hace difícil imaginar cuando el número de características supera los tres [49].

**KNeighbors (K-Vecinos):** es un método no paramétrico de clasificación desarrollado por primera vez por Evelyn Revisión y José Hodges en 1951, se utiliza para la clasificación y regresión. En ambos casos, la entrada consiste en la k más cercano de formación en el conjunto de datos. El resultado depende de si los k-NN se utilizan para la clasificación o regresión: a) En k-NN clasificación, el resultado es una clase de membresía. Un objeto está clasificado por la pluralidad de votos de sus vecinos, con el objeto de ser asignado a la clase más común entre sus k vecinos más cercanos (k es positivo entero, normalmente pequeño). Si  $k = 1$ , entonces el objeto es simplemente asignado a la clase del vecino más cercano. B) En k-NN regresión, la salida es el valor de la propiedad para el objeto. Este valor es el promedio de los valores de k vecinos más cercanos [50].

**Decision Tree (Árbol de decisión):** es una de las herramientas visuales de clasificación y predicción eficaz, aunque se vean más sencilla, se utiliza para la toma de decisiones. El árbol toma un problema o situación de raíz y explora todos los posibles escenarios relacionados con él con base en numerosas decisiones. Los árboles de decisión son muy utilizadas en diferentes sectores. Desde la programación hasta el análisis empresarial, los ejemplos de árboles de decisión están por todas partes [51].

**Random Forest (Bosque aleatorio):** es una potente y versátil máquina supervisada de algoritmo de aprendizaje, que crece y combina varios árboles de decisión para crear un "bosque". Puede ser utilizado para la clasificación y regresión. Es creado usando un conjunto de datos de entrenamiento. El modelo está capacitado utilizando diferentes ejemplos de varias entradas y salidas, y por lo tanto aprende a clasificar cualquier nuevo dato de entrada [52].

**GradientBoosting (Potenciación del gradiente):** es una técnica de aprendizaje automático utilizado para el análisis de la regresión y para problemas de clasificación estadística, el cual produce un modelo predictivo en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión. Construye el modelo de forma escalonada como lo hacen otros métodos de boosting, y los generaliza permitiendo la optimización arbitraria de una función de pérdida diferenciable [53].

**CatBoost (Impulso categórico gradiente):** CatBoost se basa en el aumento de gradiente, una tecnología de aprendizaje automático que funciona muy bien con datos de diferentes fuentes. El algoritmo funciona no sólo con números sino también con muchas otras "categorías" de datos, como audio, texto o imágenes, incluyendo datos históricos [54].

### 3.1.3.3 Desarrollo del estudio de modelos de predicción

Para generar resultados gráficos se ingresaron 1260 registros obtenidos de bases de datos médicos. En la que se obtuvieron los resultados presentados en la tabla, las celdas marcadas en azul representan los valores mayormente altos, como se observa el que cumple con la mayoría de precisión es la *Support Vector*.

Tabla 3. 25 Resultados de pruebas en modelos de predicción

No.	Modelo	Exactitud	Precisión	Recall	Puntuación	Roc auc
1	Logistic Regression	67.95	75.00	29.03	41.86	61.32
2	Support Vector	67.95	80.00	25.81	39.02	60.78
3	KNeighbors	66.67	72.73	25.81	38.10	59.71
4	Decision Tree	74.36	64.10	80.65	71.43	75.43
5	Random Forest	69.23	64.00	51.61	57.14	66.23
6	GradientBoosting	69.23	62.96	54.84	58.62	66.78
7	CatBoost	71.79	69.57	51.61	59.26	68.36

### 3.1.3.4 Pruebas del estudio de modelos de predicción

De acuerdo a los resultados obtenidos anteriormente, se seleccionó trabajar con dos modelos: *Support Vector* y *Random Forest*, y nuevamente se ingresaron 1260 datos nuevos para observar el comportamiento. La dispersión de datos que se obtuvo se presenta en las siguientes figuras.

La Figura 3.11 representa los valores ingresados y su mayor importancia al calcular la predicción.

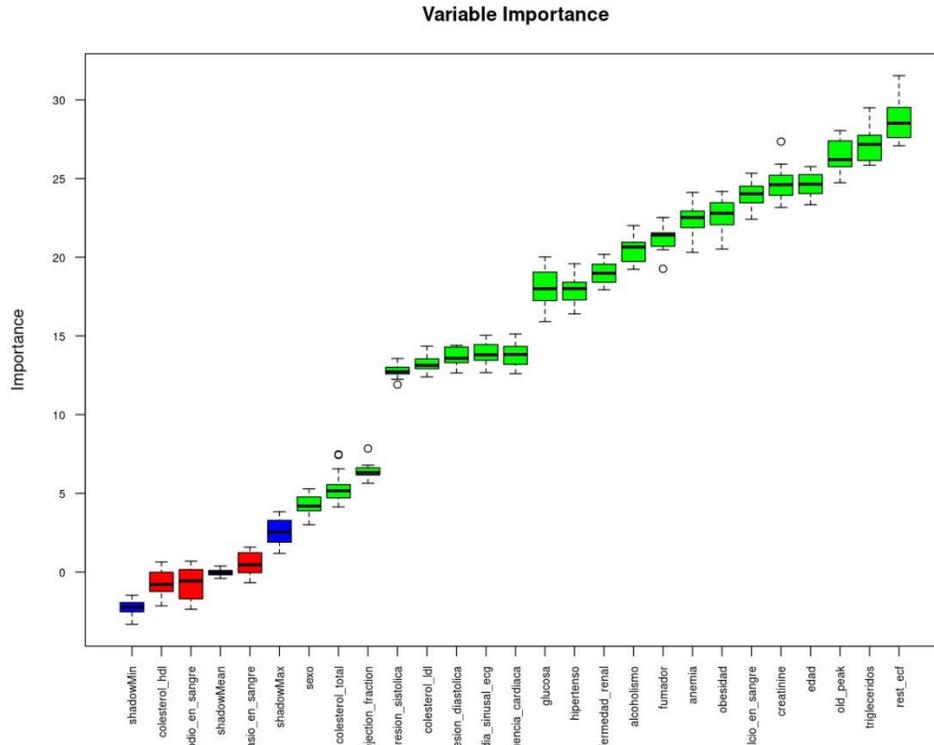


Figura 3. 11 Importancia de valores de los biomarcadores

La Figura 3.12 muestra los resultados de Accuracy, la exactitud (accuracy) mide el porcentaje de casos que el modelo ha acertado. Esta es una de las métricas más usadas y favoritas.

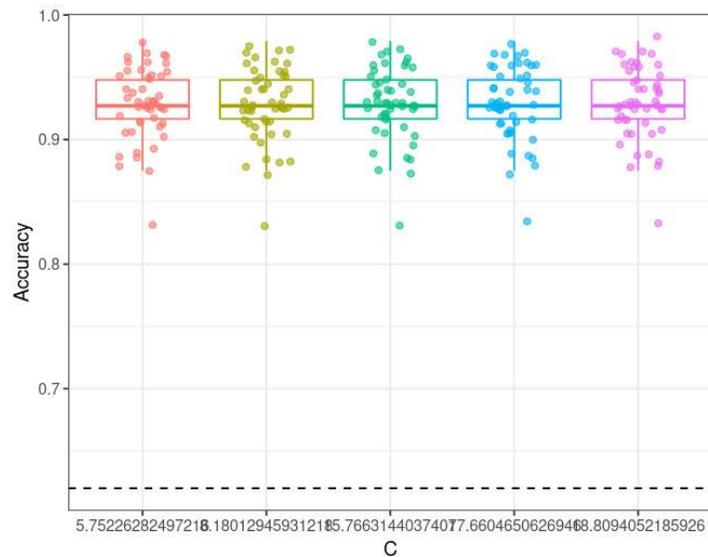


Figura 3. 12 Resultados de exactitud

A continuación en la Figura 3.13 se muestra la matriz de correlación obtenida. Con la finalidad de brindar una medida estadística de la relación entre dos variables. La medida se utiliza mejor en variables que demuestran una relación lineal entre sí. La matriz representa la correlación entre los pares de valores dentro de la tabla. Es una herramienta confiable para resumir un enorme conjunto de datos e identificar y visualizar los patrones.

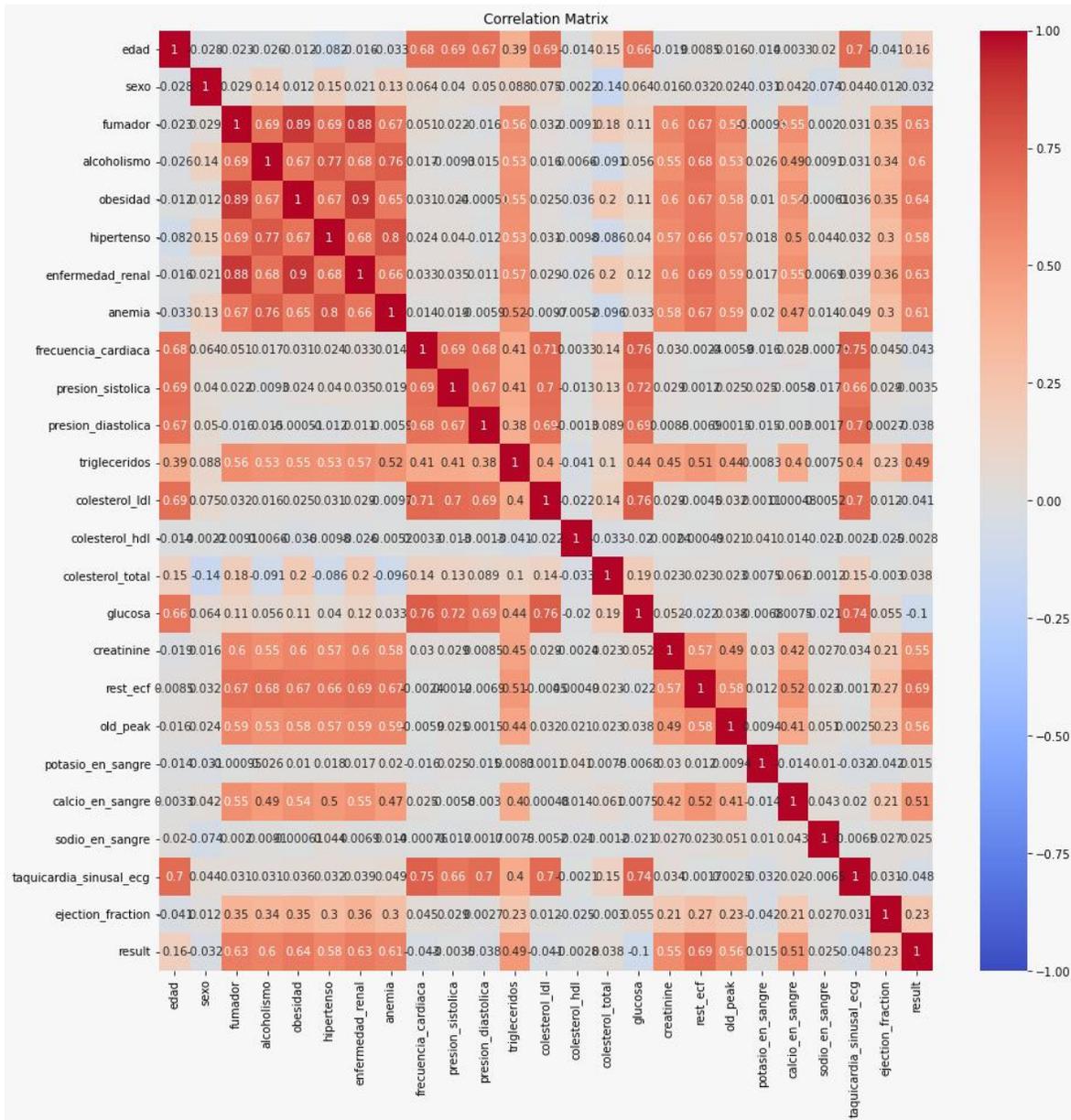


Figura 3. 13 Matriz de correlación de datos

Los resultados en cuanto a los datos ingresados se interpretan de la siguiente manera: primero se agruparon los datos de acuerdo a la frecuencia cardiaca, en la que se observa una diversidad de niveles, el dato que tiene mayor repetición es el de 70 a 75 con alrededor de

72% de personas. La figura presenta los datos sobre la presión sistólica con mayor frecuencia de 125 a 130 con 65% de personas y le continúa el rango del nivel 120 a 125 con 62% de personas.

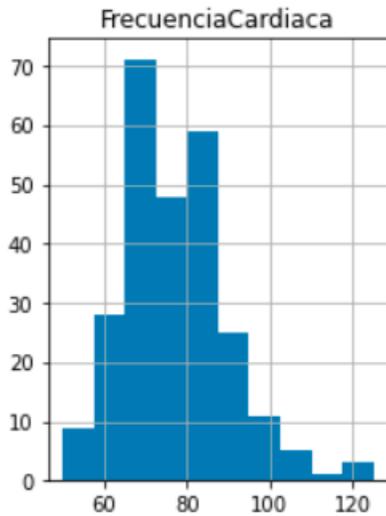


Figura 3. 14 Gráfica de valores de acuerdo a frecuencia cardiaca

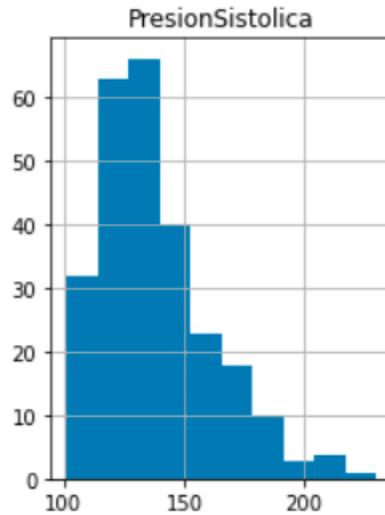


Figura 3. 15 Gráfica de valores de acuerdo a la presión sistólica

Después se organizó de acuerdo al sexo, en la Figura 3.16 se percibe que las mujeres tienen un 60.99% de no padecer alguna ECV y un porcentaje de 39.01% de padecer alguna enfermedad. Los hombres tienen un 46.22% de tener alguna enfermedad y un 53.78% de no, por lo que la mayoría no tienen riesgo en el futuro si cuidan sus condiciones de salud.

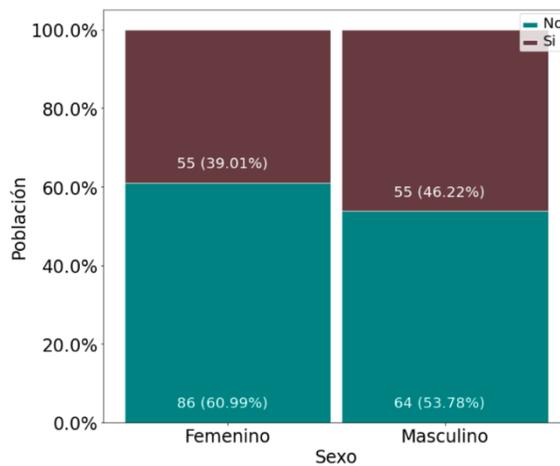


Figura 3. 16 Porcentaje de predicción de acuerdo al sexo

Con este análisis se puede definir que el modelo muestra que las personas mayormente susceptibles son los adultos mayores, a quienes se les debe tener un seguimiento a los factores de riesgo y mayor cuidado en las condiciones de salud para prevenir complicaciones futuras (ver Figura 3.17).

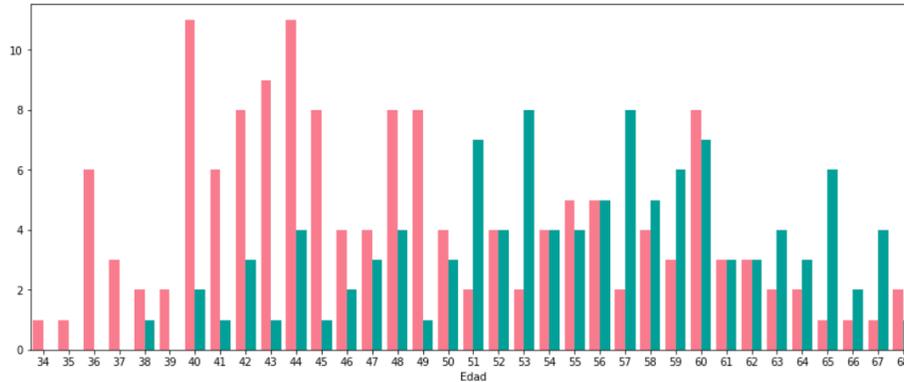


Figura 3. 17 Gráfica de datos sobre predicción de ECV

### 3.1.3.5 Planificación del desarrollo de las interfaces gráficas

Para el diseño de las interfaces de usuario, se realizó una investigación acerca de las métricas que se emplean en el desarrollo de soluciones enfocados a la salud y la medicina. En donde se encontró lo siguiente.

De acuerdo a [44], las aplicaciones relacionadas con la salud se están convirtiendo en una parte integral de práctica médica, debido a su potencial tanto mejorar la eficiencia de la prestación y superar las barreras de la distancia a los proveedores de servicios. Algunas aplicaciones se han desarrollado de acuerdo con pautas basadas en evidencia y tienen un beneficio cuantificable para mejorar los resultados clínicos.

En 2019, se identificaron 710 aplicaciones relacionadas con la cardiología disponibles para los usuarios la mayoría eran de monitores cardíacos y calculadoras médicas, muy pocos estaban dirigidos específicamente a la rehabilitación o prevención cardíaca.

Para los componentes principales y las características ideales de una aplicación móvil deben ser las siguientes:

#### 3.1.3.5.1 Sencillez

Las opciones contenidas dentro de la aplicación deben ser claras e inequívocas, debe existir un número mínimo de pasos para navegar de una función a otra. Todos los usuarios prefieren que la información presentada sea de forma clara y concisa para evitar confusiones sobre los mensajes principales [55].

#### 3.1.3.5.2 Información confiable

Que la información contenida en la aplicación sea de una fuente confiable, recibe una alta calificación por parte de los consumidores. La confiabilidad de la fuente, experiencia y autoridad percibida, veracidad y respaldo de terceros (como fundaciones) son las más valoradas. Estas características se consideran persuasivas para aumentar el compromiso y facilitar el cambio de comportamiento.

Los pacientes valoran muy positivamente la recomendación de su médico, especialmente los pacientes mayores. La probabilidad de que los pacientes con enfermedades cardíacas participen en la rehabilitación cardíaca se incrementa con una recomendación de su médico. Los mensajes claros y coherentes de los proveedores de atención médica también son importantes para facilitar el cambio de comportamiento, subrayando la importancia de profesionales de la salud que tengan suficiente información para recomendar aplicaciones basadas en evidencia y respaldado por teorías comprobadas de cambio de comportamiento [56].

#### 3.1.3.5.3 Conceptos de cambio de comportamiento

El cambio de comportamiento positivo llega cuando las personas están decididas a cambiar y empiezan a tomar decisiones propias. Entre las especificaciones que se han utilizado están: el establecimiento de metas personalizado y cuantificado, la resolución de problemas alrededor de las barreras de cambio, sugerencias informativas para cambio de comportamiento. También se encuentran las opciones de interacción con otras personas similares, grupos de competencia, uso de un avatar [57].

#### 3.1.3.5.4 Seguimiento de datos biométricos en tiempo real

Una característica para aumentar el compromiso es la capacidad de rastrear datos biométricos, como presión arterial, niveles de colesterol, niveles de actividad física y otras medidas objetivas de factores de riesgo cardiovascular en tiempo real, una función muy valorada por los usuarios de aplicaciones. Los estudios han encontrado que existe la probabilidad de que el autocontrol aumente el cambio en el comportamiento [58].

Una revisión destacó que el seguimiento en tiempo real permite reflexionar sobre hábitos actuales y fomenta la búsqueda de objetivos. Además, este tipo de seguimiento puede permitir que el usuario interactúe con una imagen más amplia de su salud, observando patrones y conexiones en los datos recopilados [59].

#### 3.1.3.5.5 Recompensas

Los beneficios de proporcionar las recompensas son reconocidas como centrales por los minoristas, con aplicaciones ahora dirigidos a los compradores en la tienda con recompensas personalizadas para aumentar el gasto. Las recompensas están dirigidas en el protocolo del estudio CONNECT, que es un ensayo controlado que incluye 2000 participantes para probar una aplicación que se integre con el sistema medico de registro de los pacientes [60].

#### 3.1.3.5.6 Personalización de la aplicación

La información personalizada es un componente central del cambio de comportamiento exitoso. En una revisión de persuasión tecnología y desarrollo de aplicaciones, la capacidad de adaptar contenido para el usuario, permitir el establecimiento de objetivos personales, y adaptar la información, aumentaron los beneficios percibidos de una aplicación [61].

La forma en que una aplicación puede acceder a información personalizada es mediante el intercambio de datos en la nube, las ventajas son que el procesamiento y almacenamiento de datos por lo que se reduce el requisito de procesamiento del dispositivo móvil y permite una funcionalidad más compleja.

#### 3.1.3.5.7 Elementos sociales

Las oportunidades de comparación social y apoyo social son importantes para aumentar el cambio de comportamiento. En un estudio cualitativo, los individuos creían que el elemento de la competencia les ayudaría a lograr sus objetivos de salud, mientras que otros consideraron que los consejos y sugerencias de personas con objetivos similares son mayormente valiosos para su motivación. Los usuarios de aplicaciones también valoraron altamente la posibilidad de establecer contacto con profesionales de la salud [62].

#### 3.1.3.5.8 Privacidad

Las preocupaciones sobre la privacidad de los usuarios potenciales son una barrera importante a la adopción de aplicaciones relacionadas con la salud, no solo en relación con el intercambio de datos, sino también en términos de seguridad personal. Las aplicaciones de atención médica pueden recopilar grandes cantidades de datos y, por lo tanto, los desarrolladores de aplicaciones deben asegurarse de no vender información del consumidor (por ejemplo, a anunciantes), especialmente porque muchas aplicaciones gratuitas generan ingresos de la publicidad. La comprobación de que existe una política de privacidad y

proporcionar una protección adecuada para los datos del usuario es un componente importante del uso de aplicaciones móviles [63].

Con base en [64], la modificación o la prevención de los factores de riesgo de ECV pueden reducir la morbilidad y mortalidad. Estos factores de riesgo son acumulativos en sus efectos deletéreos, y la idea es centrarse en la reducción integral del riesgo, utilizando valores absolutos de riesgo de ECV, en lugar de factores de riesgo individuales.

Una vez que han surgido factores de riesgo de ECV en un individuo, la prevención primaria se vuelve importante para limitar la progresión de la enfermedad. La prevención primaria se dirige específicamente factores de riesgo conocidos de ECV, como fumar, exceso de peso u obesidad o hipertensión.

De acuerdo al análisis realizado se seleccionaron las características definidas en la Tabla 3.26, contemplando la edad de los usuarios a quienes está dirigida la aplicación de detección temprana, en cada una se describe la función que tendrá la aplicación.

Tabla 3. 26 Características y funcionalidades de la aplicación móvil

Característica	Funcionalidad	Descripción de la funcionalidad	Interfaz
Personalización de la aplicación y privacidad	Datos personales del usuario.	Recuperar datos personales del usuario: nombre, apellidos, correo electrónico, fecha de nacimiento.	Registro nuevo y perfil
	Personaliza perfil de usuario.	Crear un perfil de usuario y que pueda ser modificado, para que los datos almacenados sean personales.	Editar perfil
	Cálculo de la probabilidad de contraer una enfermedad cardiovascular.	Generar un diagnóstico de los datos recuperados para calcular la probabilidad de contraer una enfermedad cardiovascular a cada usuario.	Diagnóstico
	Porcentaje general de sufrir una enfermedad cardiovascular.	Proveer el porcentaje general que tiene una persona de sufrir una	Diagnóstico

Característica	Funcionalidad	Descripción de la funcionalidad	Interfaz
		enfermedad cardiovascular de acuerdo al diagnóstico.	
Información confiable	Sugerencias para evitar las complicaciones en la salud.	Proporcionar al usuario sugerencias para evitar las complicaciones de salud en el futuro, aprobadas por especialistas.	Sugerencias
Seguimiento de datos biométricos en tiempo real	Recuperar datos.	Recuperar datos.	Dispositivos

Para la funcionalidad de la aplicación se diseñaron Mockups que a continuación se presentan.

1. Para acceder a la aplicación, el usuario debe iniciar sesión con un nombre y contraseña, en caso de que aún no tenga uno podrá crear una cuenta. Los datos insertados serán validados (Figura 3.18).



Figura 3. 18 Inicio de sesión en la aplicación móvil

2. Al ingresar, se visualizará un menú lateral con la finalidad de que se seleccione la opción deseada. También se puede cerrar sesión en la opción que se encuentra al final

del menú. En la parte superior se encuentra el ícono de notificaciones y el de perfil, para ingresar de manera sencilla a ellas (Figura 3.19).

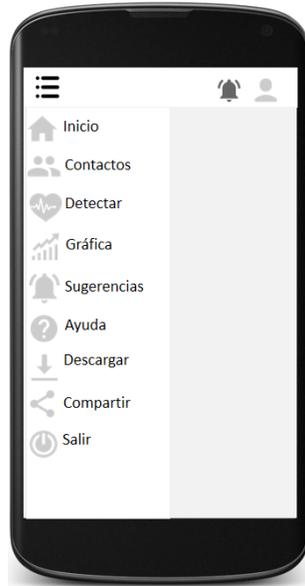


Figura 3. 19 Menú lateral en la aplicación.

3. Al momento de hacer un nuevo cálculo de probabilidad de llegar a tener una enfermedad cardiovascular se deben ingresar datos generales de la persona, datos de análisis clínicos y posibles síntomas presentados (Figura 3.20). Los datos ingresados se almacenarán en una base de conocimientos.

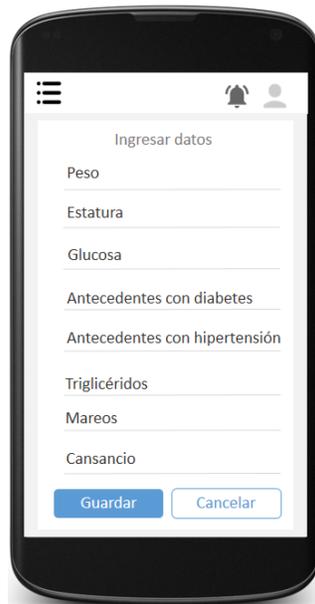


Figura 3. 20 Formulario para ingresar datos

4. La aplicación se conectará con el dispositivo de monitorización y recuperará los datos de frecuencia cardiaca, actividad física y presión arterial, esta información se presentará a manera de gráficas como se simula en la Figura 3.21.



Figura 3. 21 Recuperación de datos del dispositivo de monitorización

5. El cálculo de probabilidad se realizará con los datos ingresados manualmente y los recuperados desde el *wearable*, se usará el modelo de predicción para generar el porcentaje de probabilidad y los resultados se presentarán al usuario a manera de gráfica, en la que se representarán las posibles enfermedades que puede llegar a tener una persona (Figura 3.22).



Figura 3. 22 Resultado del cálculo de probabilidad de las ECV

6. Para apoyar a las personas a mejorar su salud, la aplicación brindará sugerencias a manera de lista (Figura 3.23).



Figura 3. 23 Sugerencias para el usuario

### 3.1.3.6 Inicio del desarrollo de las interfaces gráficas

Esta fase trata del desarrollo de las interfaces gráficas de usuario (GUI) utilizando herramientas de diseño para el frontend: React Native, CSS, HTML, JavaScript, entre otras.

La siguiente lista de código contiene las bibliotecas utilizadas dentro de las interfaces gráficas para permitir el correcto funcionamiento de los mismos:

Tabla 3. 27 Fracción de código sobre Bibliotecas usadas

```
1 import React from "react";
2 import { Dashboard, DetectionScreen, ProfileScreen, ResultScreen,
  WearableScreen, SuggestionsScreen, EditProfileScreen,
  DataUser, InfoScreen} from "./src/screens";
3 import Router from './src/routes';
4 import {View, ActivityIndicator} from 'react-native';
5 import { AuthContext } from "./src/helpers/context";
6 import { NavigationContainer } from "@react-navigation/native";
7 import {DrawerContent} from './src/components/menu';
8 import { createDrawerNavigator } from "@react-navigation/drawer";
9 import AsyncStorage from 'react-native';
10
```

## 3.1.3.7 Desarrollo de las interfaces gráficas

Al momento de crear la interfaz del inicio de sesión se ocuparon bibliotecas y código en JavaScript para la funcionalidad de los componentes, a continuación se presenta una lista de código que contiene los detalles de la interfaz.

Tabla 3. 28 Fracción de código sobre Inicio de sesión

```

104 <View style={styles.header}>
105   <Text style={styles.text_header}>Bienvenid@!</Text>
      </View>
106   <Animatable.View style={styles.footer}
107     animation="fadeInUpBig">
      <Text style={styles.text_footer}>Usuario</Text>
      <View style={styles.action}>
108       <FontAwesome
109         name="user-o"
110         color="#05375a"
111         size={20}></FontAwesome>
112       <TextInput
113         placeholder="Nombre de usuario"
114         style={styles.textInput}
115         autoCapitalize="none"
116         onChangeText={(val) => textInputChange(val)}
117         onEndEditing={(e) =>
118           handleValidUser(e.nativeEvent.text)}></TextInput>
119       {data.check_textInputChange ?
120         <Animatable.View
121           animation="bounceIn">
122           <Feather
123             name="check-circle"
124             color="green"
125             size={20}></Feather>
126         </Animatable.View>
127         : null}
128       </View>
129       {data.isValidUser ? null :
130         <Animatable.View animation="fadeInLeft"
131         duration={500}>
132         <Text style={styles.errorMsg}>Mínimo 4
133         caracteres</Text>
134       </Animatable.View>
135     }
136   </Text>

```

```

137         style={styles.text_footer, { marginTop: 35
138     ]]}>Contraseña</Text>
139         <View style={styles.action}>
140             <Feather
141                 name="lock"
142                 color="#05375a"
143                 size={20}></Feather>
144             <TextInput
145                 placeholder="Contraseña"
146                 secureTextEntry={data.secureTextEntry ? true
147             : false}
148                 style={styles.textInput}
149                 autoCapitalize="none"
150                 onChangeText={(pass) =>
handlePasswordChange(pass)}></TextInput>
151             <TouchableOpacity
152                 onPress={updateSecureTextEntry}>
153                 {data.secureTextEntry ?
154                     <Feather
155                         name="eye-off"
156                         color="grey"
157                         size={20}></Feather>
158                     :
159                     <Feather
160                         name="eye"
161                         color="grey"
162                         size={20}></Feather>
163                     }
164                 </TouchableOpacity>
165             </View>

```

Al momento de ingresar los datos cada uno se valida con un método como el siguiente:

Tabla 3. 29 Fracción de código sobre validación de datos

```

25 const handlePasswordChange = (pass) => {
26     if (pass.trim().length >= 8) {
27         setData({
28             ...data,
29             password: pass,
30             isValidPassword: true
31         });
32     } else {
33         setData({
34             ...data,
35         }
36     }

```

```

37         password: pass,
38         isValidPassword: false
39     });
40     }
41 }
42 }

```

Para el menú desplegable se realizó el siguiente código:

Tabla 3. 30 Fracción de código sobre Menú lateral

```

25 <DrawerContentScrollView {...props}>
26     <View style={styles.drawerContent}>
27         <View style={styles.userInfoSection}>
28             <View style={{flexDirection:'row',marginTop:
29 15}}>
30                 <Avatar.Image
31                     source={require('../assets/icon.png')}
32                     size={50}
33                 />
34                 <View style={{marginLeft:15,
35 flexDirection:'column'}}>
36                     <Title style={styles.title}>AIDA</Title>
37                     <Caption
38 style={styles.caption}>Bienvenid@</Caption>
39                 </View>
40                 </View>
41             </View>
42             <View>
43                 <Drawer.Section style={styles.drawerSection}>
44                     <DrawerItem
45                         icon={({color, size}) => (
46                             <Icon
47                                 name="home-outline"
48                                 color={color}
49                                 size={size}
50                             />
51                         )}
52                         label="Inicio"
53                         onPress={() =>
54                             {props.navigation.navigate('Inicio')}}
55                     />
56                 </DrawerItem>
57                 <DrawerItem
58                     icon={({color, size}) => (
59                         <Icon
60                             name="clipboard-list-outline"
61                             color={color}

```

```

60             size={size}
61           />
62         )}
63         label="Detección temprana"
64                                     onPress={() =>
65 {props.navigation.navigate('Detección de ECV')}}
66           />
67
68

```

La aplicación recibirá datos del *wearable* los cuales se mostrarán en una interfaz, la siguiente lista de código muestra la parte de desarrollo de la interfaz. Se genera una visualización puntualizando la actividad física captada y los movimientos ejecutados.

Tabla 3. 31 Fracción de código sobre visualización de datos de wearable

```

31 <View style={stylescards.div}>
32   <View style={stylescards.containerC3}>
33     <View style={stylescards.containerIcon1}>
34       <FontAwesome
35         name="heartbeat"
36         color="#ffffff"
37         size={40}></FontAwesome>
38     </View>
39     <View style={stylescards.containerTextEnd}>
40       <Text
41         style={stylescards.textBold}>{datos.bloodpressure}</Text>
42       <Text style={stylescards.colorText}>Latidos por
43         minuto</Text>
44     </View>
45   </View>
46   <View style={stylescards.containerC3}>
47     <View style={stylescards.containerIcon2}>
48       <FontAwesome
49         name="stethoscope"
50         color="#ffffff"
51         size={49}></FontAwesome>
52     </View>
53     <View style={stylescards.containerTextEnd}>
54       <Text
55         style={stylescards.textBold}>{datos.heartrate}</Text>
56       <Text style={stylescards.colorText}>mm de
57         Hg</Text>
58     </View>
59   </View>
60 </View>
61 </View style={stylescards.containerC3}>

```

```

62         <View style={stylescards.containerIcon3}>
63             <FontAwesome
64                 name="bicycle"
65                 color="#ffffff"
66                 size={32}></FontAwesome>
67         </View>
68         <View style={stylescards.containerTextEnd}>
69             <Text
70                 style={stylescards.textBold}>{datos.physical}</Text>
71             <Text style={stylescards.colorText}>Minutos de
72             actividad física</Text>
73         </View>
74     </View>
75 </View>
76

```

Para los datos médicos ingresados se generó una tabla en la que se pintarán por colores si el valor está en el rango, en color verde cuando es normal, color amarillo cuando es alerta y color rojo cuando ya se superó el límite. Para ello se creó el siguiente código.

Tabla 3. 32 Fracción de código sobre interfaz de datos médicos

```

26 <View style={stylescards.listWrapperTitle}>
27     <Text style={stylescards.row1}>Nombre</Text>
28     <Text style={stylescards.row1}>Valor</Text>
29     <Text style={stylescards.row1}>Resultado</Text>
30 </View>
31 <View style={stylescards.listWrapper}>
32     <Text style={stylescards.row}>Colesterol HDL</Text>
33     <Text style={stylescards.row}>{datos.chol}</Text>
34     <Text style={stylescards.row}>
35         {col ? null :
36             <FontAwesome
37                 name="circle"
38                 color="#228b22"
39                 size={28}
40             ></FontAwesome>
41         }{col2 ? null:
42             <FontAwesome
43                 name="circle"
44                 color="#ffd700"
45                 size={28}
46             ></FontAwesome>
47         }{col3? null:
48             <FontAwesome
49                 name="circle"
50

```

```

53         color="#ff0000"
54         size={28}
55     ></FontAwesome>
56     }
57 </Text>
58 </View>
59

```

Los resultados generados acerca de la probabilidad de llegar a tener alguna ECV se presentarán como una alerta, con descripción de las afectaciones en la salud. Cada especificación tendrá una imagen representativa.

Tabla 3. 33 Fracción de código sobre alertas con resultado

```

26 <Modal
27     animationType="slide"
28     transparent={true}
29     visible={modalVisible}
30     onRequestClose={() => {
31         Alert.alert("Modal has been closed.");
32         setModalVisible(!modalVisible);
33     }}
34 >
35     <View style={{
36         flex: 1,
37     }}>
38         <View style={{
39             margin: 20,
40         }}>
41             <Text style={{
42                 marginBottom: 0.5,
43             }}>Resultado obtenido</Text>
44             {col ? null :
45                 <Image
46                     source={require('../assets/csad.jpg')}
47                     style={{ marginBottom: 20 }}
48                 />
49                 {col2 ? null :
50                     <Image
51                         source={require('../assets/tres.jpg')}
52                         style={{ marginBottom: 20 }}
53                     />
54                     {col3 ? null :
55                         <Image
56                             source={require('../assets/chappy.jpg')}
57                             style={{ marginBottom: 20 }}
58                         />
59                     }
60                 }

```

```

61         />
62     }
63     <Text style={{
64         marginBottom: 5,
65     }}> {data.probabilidad}</Text>
66     <Text style={{
67         marginBottom: 25,
68     }}> Porcentaje: {data.porcentaje}%</Text>
69     <Pressable
70         style={{
71             borderRadius: 20,
72         }}
73         onPress={() => setModalVisible(!modalVisible)}
74     >
75         <Text style={{
76             color: "white",
77         }}>Aceptar</Text>
78
79
80
81
82
83

```

Para todas las interfaces se creó código en CSS, con la finalidad de brindar estilo a cada uno de los componentes. El código siguiente tiene una fracción de ello.

Tabla 3. 34 Fracción de código sobre el diseño utilizado

```

1  const stylescards = StyleSheet.create({
2      container: {
3          flex: 1,
4          backgroundColor: '#ffffff',
5      },
6      div: {
7          marginTop: 2,
8          marginHorizontal:20,
9          marginBottom:20,
10     },
11     containerC1: {
12         flexDirection: 'row',
13         height: 90,
14         backgroundColor: '#CD5C5C',
15         paddingHorizontal: 20,
16         paddingVertical: 20,
17         borderRadius: 5,
18         elevation: 5,
19     },
20
21

```

```

22     marginTop: 10,
23   },
24   icon: {
25     justifyContent: 'center',
26     alignItems: 'center',
27   },.....
28

```

### 3.1.3.8 Pruebas de las interfaces gráficas

Después de generar cada una de las interfaces gráficas planeadas se obtuvo un buen funcionamiento ingresando a ellas. También al momento de hacer pruebas se eliminaron variables que se crearon pero no se utilizan, así como bibliotecas que en posterior ya no se ocuparon.

### 3.1.3.9 Planificación del desarrollo de las API

Se plantea generar una *Application Programming Interfaces* (Interfaz de Programación de Aplicaciones) que permita la comunicación entre la base de conocimientos y la aplicación móvil, para que se puedan insertar, consultar, modificar y eliminar datos de usuarios, contactos, información de wearables y resultados de la detección de ECV.

Con la finalidad de ser usada en vez de una conexión directa desde la aplicación hacia la base de conocimientos. La Figura 3.24 muestra la comunicación que se espera de la API.

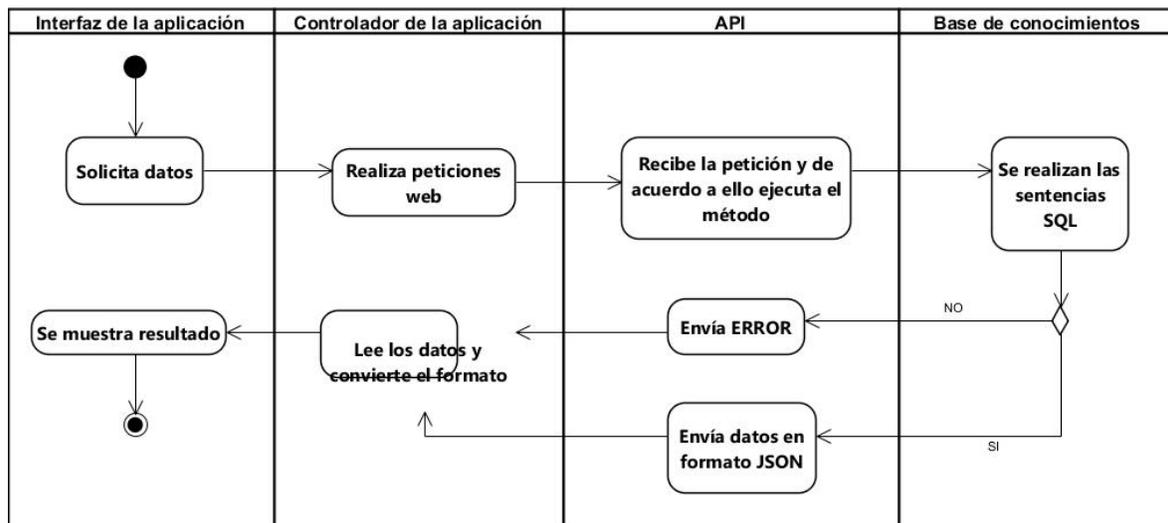


Figura 3. 24 Diagrama de actividad con la API

### 3.1.3.10 Inicio del desarrollo de las API

Para generar una comunicación adecuada se realizaron las siguientes actividades antes del desarrollo del código.

1. Primero se contrató un VPS (*Virtual Private Server*, Servidor Virtual Privado) para alojar la base de datos necesaria al proyecto. La seleccionada fue WHM que contiene las siguientes características:  
Sistema operativo: Centos 7.9  
Versión: v98.0.6  
Al ingresar se agrega nombre de usuario y contraseña (ver Figura 3.25).

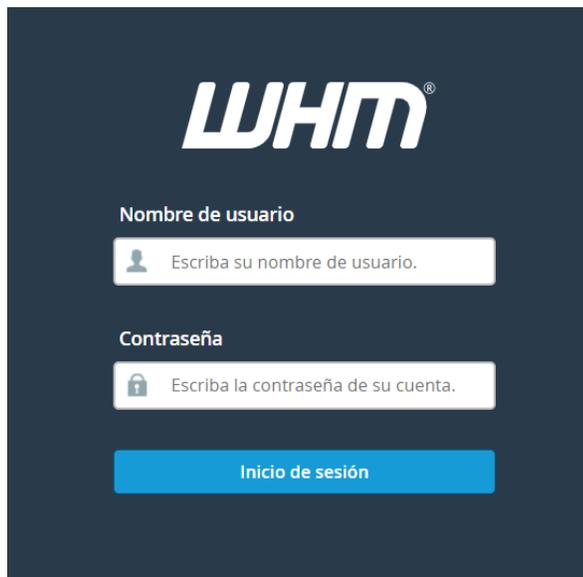


Figura 3. 25 Interfaz del VPS

2. Sucesivamente se instaló el Sistema Gestor de Base de Datos PostgreSQL desde la terminal:



This interface provides command line access to the server's "root" account.

**root@vps-383157.msc-healthlab.com:**

```
root@vps-383157.msc-healthlab.com [~]# sudo -u postgres psql
could not change directory to "/root"
Password:
psql (9.2.24)
Type "help" for help.

postgres=#
```

Figura 3. 26 Instalación de Postgres en el VPS

3. Una vez comprobado el funcionamiento de PostgreSQL, se creó una nueva base de datos con el nombre de AIDA, para la aplicación:

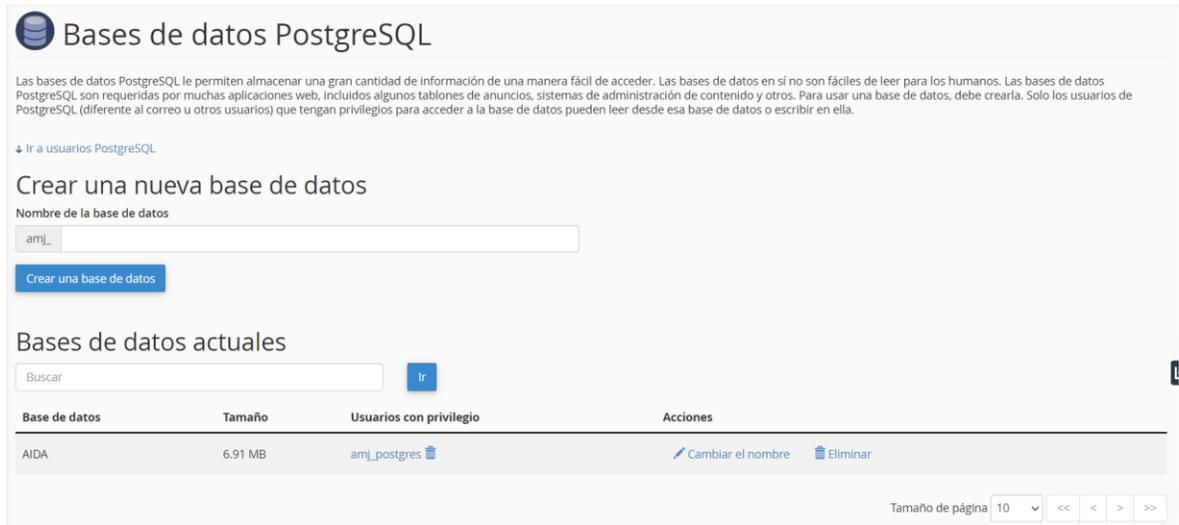


Figura 3. 27 Creación de base de datos en el VPS

4. Después de crear la base de datos, se ingresó al menú de Bases de Datos desde el CPANEL configurado y se seleccionó la opción de phpPgAdmin:



Figura 3. 28 Uso de phpPgAdmin en el VPS

- Se crearon las tablas necesarias para almacenar la información, de acuerdo al análisis realizado.



Figura 3. 29 Creación de tablas en el VPS

La importancia de estas actividades es dejar todo preparado para la comunicación y el acceso desde la API.

### 3.1.3.11 Desarrollo de las API

Al iniciar se realizó una conexión al servidor mediante el siguiente código:

Tabla 3. 35 Fracción de código de la API para conexión a la Base de Datos

```

1 private $host = " ";
2     private $db_name = "AIDA";
3     private $username = " ";
4     private $password = " ";
5
6     public $conn;
7
8     public function getConnection() {
9         $this->conn = null;
10        try {
11            $this->conn = new PDO("pgsql:host=" .
12            $this->host . ";dbname=" . $this->db_name, $this->username,
13            $this->password);
14
15            $this->conn->exec("set names utf8");

```

```

14         } catch(PDOException $exception) {
15             echo "Connection error: " . $exception-
>getMessage();
16         }
17         return $this->conn;
    
```

Sucesivamente se crearon carpetas para la gestión de datos de cada tabla, con la finalidad de tener una organización y en caso de modificar algún dato sea más fácil encontrar el directorio, estos archivos se almacenaron dentro del VPS:

Name	Size
 config	4 KB
 contact	4 KB
 manual	4 KB
 object	4 KB
 result	4 KB
 usuario	4 KB
 wearable	4 KB
 wearabledata	4 KB

Figura 3. 30 Carpetas de archivos de la API

Se trabajó con la estructura controlador y modelo, por lo que el encargado de validar la gestión y los datos ingresados o solicitados es la que se presenta en las siguientes fracciones de código, cada uno está estructurado por tres apartado, la primera para conectarse al modelo, la segunda para validar cada uno de las variables y el tercero para devolver el resultado.

La primera es para insertar nuevos datos, en este caso para la información del usuario nuevo.

Tabla 3. 36 Fracción de código de la API para insertar datos

```

1  $database = new Database();
2  $db = $database->getConnection();
3  $product = new User($db);
4  $data = json_decode(file_get_contents("php://input"));
5  if(
6      !empty($data->name) &&
    
```

```

7         !empty($data->email)
8     ) {
9         $product->iduser = $data->iduser;
10        $product->name = $data->name;
11        $product->last_name = $data->last_name;
12        $product->email = $data->email;
13        $product->phone = $data->phone;
14        $product->birth = $data->birth;
15        $product->adress = $data->adress;
16        $product->employment = $data->employment;
17        $product->status = $data->status;
18        $product->username = $data->username;
19        $product->pass = $data->pass;
20
21        if ($product->create()) {
22            http_response_code(201);
23            echo json_encode(array("message" => "Se
    ingresó un nuevo Usuario"));
24        } else {
25            http_response_code(503);
26            echo json_encode(array("status" => "No se
    registró, verifica los datos"));
27        }
28    } else {
29        http_response_code(400);
30        echo json_encode(array("error" => "Incapaz de crear
    el registro. Los datos están incompletos."));
31    }

```

En caso de realizar consultas se ejecuta el siguiente código, el ejemplo es para extraer todos los contactos del usuario.

*Tabla 3. 37 Fracción de código de la API para consultar datos*

```

1  $stmt = $obj->consultAllContacts();
2  $num = $stmt->rowCount();
3  if($num>0) {
4      $arr = array();
5      $arr["datos"]=array();
6      while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
7          extract($row);
8          $arr_cont=array(
9              "idcontact" => $idcontact
10             , "name" => $name
11             , "mail" => $mail
12             , "employment" => $employment

```

```

13         , "relation" => $relation
14         , "iduser" => $iduser
15     );
16     array_push($arr["datos"], $arr_cont);
17 }
18 http_response_code(200);
19 echo json_encode($arr);
20 } else {
21     http_response_code(404);
22     echo json_encode(array("Error" => "No se encontraron
23 Usuarios"));
    }

```

Cuando de requiera modificar algún registro se ejecutará la siguiente secuencia de código:

*Tabla 3. 38 Fracción de código de la API para modificar datos*

```

1  $database = new Database();
2  $db = $database->getConnection();
3  $xemc = new Wearable($db);
4  $data = json_decode(file_get_contents("php://input"));
5
6  $xemc->idwear = $data->idwear;
7  $xemc->name = $data->name;
8  $xemc->serie = $data->serie;
9  $xemc->model = $data->model;
10 $xemc->date = $data->date;
11 $xemc->character = $data->character;
12 $xemc->data = $data->data;
13 $xemc->descriptions = $data->descriptions;
14 $xemc->iduser = $data->iduser;
15
16 if($xemc->updateDataWearable()){
17     echo json_encode(array("message" => "Los datos del
wearable se actualizaron"));
18 } else{
19     echo json_encode(array("error" => "Error al
actualizar"));

```

Cuando se solicite eliminar un registro, el siguiente código hará la tarea.

*Tabla 3. 39 Fracción de código de la API para eliminar datos*

```

1  $database = new Database();
2  $db = $database->getConnection();
3  $wear = new Wearable($db);

```

```

4     $data = json_decode(file_get_contents("php://input"));
5     $wear->idwear = $data->idwear;
6     $wear->searchId();
7     if ($wear->idwear!=null){
8         if($wear->deleteWearable()){
9             http_response_code(200);
10            echo json_encode(array("Éxito" => "El Wearable
se eliminó"));
11        }else{
12            http_response_code(503);
13            echo json_encode(array("Status" => "No se
eliminó el Wearable"));
14        }
15    }else{
16        http_response_code(404);
17        echo json_encode(array("Error" => "No se encontró el
Wearable con dicho Identificador"));

```

Todas las carpetas presentadas en la Figura 3.30 se componen de las operaciones de: insertar nuevo registro, consultar todos, consultar por identificador, modificar y eliminar. Solo se han agregado ejemplo de cada uno de ellos en esta sección.

### 3.1.3.12 Pruebas de las API

Al finalizar el desarrollo de cada API, se realizaron pruebas de acceso a los datos, primero a datos de usuario para ello se ocupó la siguiente fracción de código:

Tabla 3. 40 Fracción de código para la conexión de la aplicación con la API

```

1     const url_api = 'http://162.214.187.236/~amj/api/usuario/consulta_id.
2     php?iduser='+id;
3
4     fetch (url_api)
5         .then(res => res.json())
6         .then(data => {
7             const name = document.createElement('p')
8             name.innerHTML = "Nombre: " + data.name
9             aplicacion.appendChild(name)
10
11            const lastname = document.createElement('p')
12            lastname.innerHTML = "Apellidos: " + data.last_name
13            aplicacion.appendChild(lastname)
14
15            const email = document.createElement('p')

```

```

16     email.innerHTML = "Correo: " + data.email
17     aplicacion.appendChild(email)
18
19     const phone = document.createElement('p')
20     phone.innerHTML = "Teléfono: " + data.phone
21     aplicacion.appendChild(phone)
22
23     const birth = document.createElement('p')
24     birth.innerHTML = "Fecha de nacimiento: " + data.birth
25     aplicacion.appendChild(birth)
26
27     const adress = document.createElement('p')
28     adress.innerHTML = "Dirección: " + data.adress
29     aplicacion.appendChild(adress)
30
31     const employment = document.createElement('p')
32     employment.innerHTML = "Ocupación: " + data.employment
33     aplicacion.appendChild(employment)
34 }
35
36 )
37 .catch(err => console.log(err))
38

```

Se obtuvo de manera exitosa los datos solicitados. A continuación se agrega la prueba ejecutada mediante Postman para insertar nuevo registro a la base de datos mediante el uso de la API.

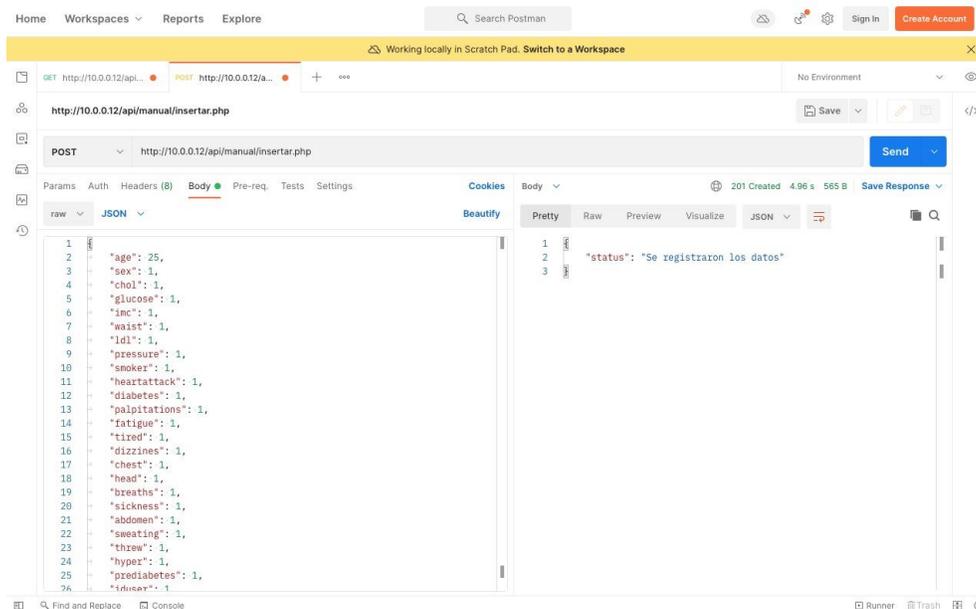


Figura 3. 31 Pruebas de inserción de datos desde Postman

Esas han sido las pruebas fundamentales para aprobar el funcionamiento adecuado de las API's generadas.

#### *3.1.3.13 Liberación*

Al concluir cada una de las tareas planeadas se realizó una entrega de archivos para su almacenamiento, quedando la carpeta con lo siguiente:

- 70 archivos de imagen
- 66 archivos de JavaScript
- 73 archivos de SCSS (Syntactically Awesome Style Sheet, hoja de estilo sintácticamente impresionante).
- 151 archivos JSX (Extensión de la sintaxis de JavaScript, usado en React).
- 3 archivos JSON
- 27 archivos PHP para el acceso a la base de datos.

### **3.1.4 Estabilización**

En esta fase se da paso a la generación de documentos para uso de lo que se ha desarrollado.

#### *3.1.4.1 Documentación*

Para el respaldo del trabajo realizado, se generaron dos documentos importantes, la primera es el manual de uso de los servicios Web mediante las API's. El cual es un documento en formato PDF, a continuación se anexan los apartados.

Para el respaldo del trabajo realizado, se generaron dos documentos importantes, la primera es el manual de uso de los servicios Web mediante las API's. El cual es un documento en formato PDF, a continuación se anexan los apartados.

# Manual - API - AIDA

Elabora: M.S.C. Alejandra Montiel de Jesús

El siguiente documento contiene la descripción del proyecto, su funcionalidad, directorios y ficheros que se involucran para una óptima funcionalidad.

## 1.1 Entidades del proyecto [- Tablas -]

- → **User**
- → **Contacts**
- → **Wearable**
- → **Result**
- → **WearableData**
- → **ManualData**
- → **Notificaactions**
- → **Suggestion**

## 1.2 Métodos

A continuación, se describen los métodos que se deberán de utilizar según la operación a realizar en la API, aunque se realice una simple consulta se debe de respetar para una respuesta en el servidor.

- **Method [- POST -]**
  - Este método se deberá de utilizar cuando se necesite **enviar datos** para crear un nuevo registro en el sistema, no se deberá de utilizar al enviar datos para su actualización, *es unicamente para crear nuevos registros.*
- **Method [- GET -]**
  - El método se deberá de utilizar en cualquier **consulta** de datos, pero no para enviar, puede ser para consulta general o por identificador.
- **Method [- PUT -]**
  - Este método se deberá de utilizar solo para enviar datos al API para **actualizar** registros ya existentes.
- **Method [- DELETE -]**
  - El método se deberá de utilizar solo cuando se solicite **eliminar** algún registro existente por medio de la API.

## 1.3 Directorios y archivos

### 1.3.1 Descripción de los directorios de la API

A continuación se describen los directorios que se encuentran dentro de la API, los documentos y una breve descripción de cada uno de ellos.

- **Carpeta: Usuario**
  - Documento: **actualizar.php**
    - El documento actualiza un registro dentro del servidor, para ello es necesario enviar los parámetros en formato JSON y el identificador del registro.
  - Documento: **buscar.php**
    - El documento realiza una petición de búsqueda de acuerdo al parámetro enviado o solicitado.
  - Documento: **consulta.php**
    - El documento no recibe argumento o dato alguno, el retorno es una lista de todos los registros encontrados en el servidor.
  - Documento: **consulta\_id.php**
    - El documento realiza una búsqueda de datos acorde al argumento o dato enviado, este deberá de cumplir con un identificador válido.
  - Documento: **eliminar.php**
    - El documento realiza una *eliminación* \* de registro conforme el argumento o identificador enviado, si no se recibe dato alguno, el documento negará la petición.
  - Documento: **insertar.php**
    - El documento tiene la finalidad de crear un nuevo registro, los datos a enviar deberán de ser acorde al argumento o datos enviados, los cuales serán aceptados en formato JSON.
- **ManualData:**
  - Documento: **consultar\_todos.php**
    - El documento realiza una búsqueda de datos acorde al identificador enviado como argumento o parámetro, si existe enviará los datos encontrados en formato JSON.

- Documento: **eliminar.php**
  - El documento realiza la eliminación del registro solicitado por medio de su identificador, si no encuentra el registro este retornará un código de error.
- Documento: **crear.php**
  - El documento tiene la finalidad de crear un nuevo registro, los datos a enviar deberán de ser acorde al argumento o datos enviados, los cuales serán aceptados en formato JSON.
- Documento: **consultar\_id.php**
  - El documento realiza una consulta de datos al servidor por medio del argumento solicitado, los datos encontrados serán enviados al solicitante en un array y en formato JSON.
- Documento: **actualizar.php**
  - El documento realiza el proceso de actualización de datos, para ello es necesario enviarlos en formato JSON, además se deberá de enviar como argumento el identificador del registro a modificar, de lo contrario enviará un código de error.
- **Wearable:**
  - Documento: **consulta\_general.php**
    - El documento recupera todos los registros encontrados en el sistema, los datos que regresa serán en un array y en formato JSON, el programador deberá de decodificarlos para su correcta lectura.
  - Documento: **consultar\_id.php**
    - El documento realiza una petición de consulta por identificador conforme el argumento enviado, si no se recibe alguno, el sistema mandará un código de error, si llega a existir enviará los datos en formato JSON.
  - Documento: **eliminar.php**
    - El documento realiza una eliminación de datos de acuerdo al identificador enviado como argumento, si no existe el registro, el sistema enviará un código de error.
  - Documento: **actualizar.php**
    - El documento enviará una petición de actualización de datos, para ello es necesario enviar el identificador y los datos a guardar, el argumento será en formato JSON. Si el registro o los datos no son

válidos, o llegara a faltar el identificador se enviará un código del error que se generó.

- Documento: **crear\_nuevo.php**
  - El documento realiza un nuevo registro en el sistema, para su óptima funcionalidad deberá de enviar los datos en formato JSON, es necesario que los campos del identificador e identificador de usuario son obligatorios, si llegara a presentar un error se notificará mediante un código.
- **WearableData:**
  - Documento: **consultar\_id.php**
    - El documento realiza una petición de consulta por identificador conforme el argumento enviado, si no se recibe alguno, el sistema mandará un código de error, si llega a existir enviará los datos en formato JSON.
  - Documento: **ultimo\_registro.php**
    - El documento enviará al solicitante los datos encontrados en un array y en formato JSON, no enviará datos de errores.
  - Documento: **ultimos\_ten.php**
    - El documento enviará los datos encontrados en el servidor ordenados de manera descendente y en respuesta JSON, además no hay respuesta de error, ya que solo envía datos y no valida.
- **Result:** [- Porcentaje de enfermedad -]
  - Documento: **consultar\_id.php**
    - El documento realiza una búsqueda de datos en el servidor de acuerdo al identificador enviado como argumento, si encuentra datos este los envía en formato JSON, si no encuentra registros entonces enviará un código de error, este deberá de ser controlado por el programador.
  - Documento: **insertar.php**
    - El documento para su óptimo funcionamiento recibe los datos en formato JSON, además se deberá de enviar como datos obligatorios el identificador del usuario, al finalizar se envía un mensaje de éxito, de lo contrario se envía un código de error.
  - Documento: **consulta.php**

- El documento envía los registros encontrados en el servidor, si no se encuentran datos enviará un error. Los datos que existan se enviarán en formato JSON.

## 1.4 Clases y Métodos

### 1.4.1 Objetos del Sistema

A continuación se describe cada objeto con sus diferentes métodos su funcionalidad para la manipulación de datos, en él se realiza una breve descripción y/o datos necesarios.

- **Usuario**
  - *function count()*
    - El método no necesita argumentos para su funcionalidad, él solo realiza un conteo de todos los registros encontrados en la base de datos.
  - *function read()*
    - El método realiza una consulta de todos los registros en el sistema, no requiere argumentos para su funcionalidad, y retorna en una variable los datos encontrados.
  - *function create()*
    - El método realiza la creación de un nuevo registro en el sistema, por ello se deberán de descifrar los datos enviados en JSON para poder registrarlo. Si la operación llega a fallar entonces envía un código de error.
  - *function readId()*
    - El método realiza una consulta de datos en el servidor, es necesario recibir el identificador del registro para su óptima funcionalidad, si no encuentra datos retornará un error y de lo contrario retorna los datos encontrados.
  - *function delete()*
    - El método realiza una actualización del parámetro “status” de 1 a 0 en la tabla de usuario, es necesario recibir el identificador del registro a modificar, si el método falló o no encontró el registro enviará un mensaje de error, de lo contrario mostrará un mensaje de éxito.
  - *function search(\$keywords)*
    - El método realiza una búsqueda de registros en la tabla señalada, para su funcionalidad se deberá de enviar como argumento el valor

y el método lo buscará en toda la tabla, los datos encontrados se retornarán de acuerdo a la tabla.

- *function updateEmployee()*
  - El método realiza una actualización de un registro a la vez, para su buena funcionalidad se deberá de enviar los datos en formato JSON, si el registro se realizó con éxito entonces mandará un mensaje de éxito, de lo contrario mandará un error.
- **ManualData:**
  - *function readAllManual()*
    - El método realiza una consulta de todos los registros existentes, una vez encontrados, se retornarán en un array.
  - *function searchId()*
    - El método realiza una búsqueda de un registro, y valida su existencia por medio de un TRUE o un FALSE, no retorna valor alguno, sólo un valor booleano según lo encontrado.
  - *function deleteManual()*
    - El método realiza la eliminación de un registro, para su funcionalidad se deberá de enviar como argumento el identificador del registro, si existe este realizará la operación y enviará un mensaje de éxito, de lo contrario mandará un mensaje de error. (Retorna un valor boolean según la operación).
  - *function crearNuevoManual()*
    - El método realiza el registro de nuevos datos, para ello deberá de recibir los datos del registro, el método realizará la operación si los campos no estan vacíos, si se ejecutó correctamente mandará una respuesta verdadera, si llega a presentar un error este mandará un código del error.
  - *function consulDataById()*
    - El método realiza una consulta de todos los datos en el sistema, para ello es necesario enviar como argumento el identificador, una vez se reciba buscará los datos, si existe entonces retorna los datos, de lo contrario mandará un mensaje de error. Si no se envió el identificador mandará un mensaje de error.
  - *function updateManualdata()*
    - El método realiza la actualización de datos por medio de su identificador, para ello es vital enviarlo además de compartir los campos en formato JSON.

## 1.4.2 Dispositivo Wearable

- **Wearable:**

- *function readAllWearable()*
  - El método realiza la consulta de todos los registros, además retorna los valores encontrados, no envía mensajes de error, el programador debe controlar los errores.
- *function wearId()*
  - El método realiza la búsqueda de datos, de acuerdo al identificador enviado como argumento, si encuentra registros el sistema los retornará en un array, de lo contrario enviará un dato falso.
- *function searchId()*
  - El método realiza la búsqueda del identificador marcado, para ello se deberá de compartir como argumento el id a buscar, una vez enviado será buscado, si se encontró este retornará el valor del identificador, mas no los datos ligados a él, si no se encuentran datos retornará un falso.
- *function deleteWearable()*
  - El método realiza la eliminación de un registro, para su funcionalidad se deberá de enviar como argumento el identificador del registro, sí existe este realizará la operación y enviará un mensaje de éxito, de lo contrario mandará un mensaje de error. (Retorna un valor boolean según la operación).
- *function updateDataWearable()*
  - El método realiza una actualización de un registro a la vez, para su buena funcionalidad se deberá de enviar los datos en formato JSON, sí el registro se realizó con éxito entonces mandará un mensaje de éxito, de lo contrario mandará un error.
- *function createNewWearable()*
  - El método realiza el registro de nuevos datos, para ello deberá de recibir los datos del registro, el método realizará la operación si los campos no estan vacíos, sí se ejecutó correctamente mandará una respuesta verdadera, si llega a presentar un error este mandará un código del error.

## 1.4.3 Datos del Wearable

- **WearableData:**

- *function consultByIdentifier()*

- El método realiza la búsqueda de todos los datos que son almacenados de cada Wearable, de acuerdo al identificador enviado como argumento, si encuentra registros el sistema los retornará en un array de los datos, de lo contrario enviará un dato falso.
- *function lastRegisterWearable()*
  - El método realiza una consulta del último registro encontrado en el sistema, no es necesario enviar argumento alguno, además no retorna datos de errores.
- *function lastTenRegister()*
  - El método enviará una array de los últimos diez registros encontrados en el servidor, ordenados de manera descendente, no es necesario argumento alguno, no retorna error alguno, así que el programador deberá de realizar su método para los errores controlados.

## 1.5 URL's y Datos en formato JSON

### 1.5.1 URL ejemplo de consultas

A continuación, se describen las consultas a realizar directamente en la API, para ello es necesario configurar las url de prueba, además de respetar los tipos de datos que interactuarán en formato JSON, aclarando lo siguiente: si se envían los datos como argumento en la url es necesario la separación por comas, y si es sólo un valor no es necesario encerrarlo entre comillas ya sea dobles o simples.

En los siguientes ejemplos se describen las url a consultar según su necesidad, además de los formatos JSON a respetar para el envío de datos y ejemplos de muestra de respuesta del servidor.

Cargar archivos de:

- **CARPETA - Configuración** : Carpeta con acceso a la base de datos
  - Documento : Database, este contiene la conexión a la base de datos.
- **CARPETA - Objetos** : Carpeta con objetos que hacen uso de cada entidad actual en el servidor.
  - **Documento : ManualData**
    - Status : Actualmente funcionando
  - **Documento : Usuario**
    - Status : Actualmente funcionando
  - **Documento : Contacts**
    - Status : Actualmente funcionando
  - **Documento : Result**
    - Status : Actualmente funcionando
  - **Documento : Wearable**

- Status : Actualmente funcionando
- **Documento : WearableData**
  - Status : Actualmente funcionando

#### Manual Data

- **CARPETA - ManualData** : Contiene los archivos para el acceso a los datos
  - : Documento - **Consultar** por identificador
    - [http://162.214.187.236/~amj/api/manual/consulta\\_id.php](http://162.214.187.236/~amj/api/manual/consulta_id.php)
  - : Documento - **Insertar** nuevo registro
    - <http://162.214.187.236/~amj/api/manual/insertar.php>
  - : Documento - **Actualizar** registro
    - <http://162.214.187.236/~amj/api/manual/extra/actualizar.php>
  - : Documento - **Consultar** todos los registros
    - [http://162.214.187.236/~amj/api/manual/extra/consulta\\_all.php](http://162.214.187.236/~amj/api/manual/extra/consulta_all.php)
  - : Documento - **Eliminar** registro por identificador
    - <http://162.214.187.236/~amj/api/manual/extra/eliminar.php>

*NOTA:* Ya está funcionando en el servidor VPS. Algunos datos deberán de ser en formato JSON, por ejemplo:

#### Usuario

- **CARPETA : Usuario** : Contiene los archivos para el control de datos
  - : Documento : **Actualizar** registro por identificador
    - <http://162.214.187.236/~amj/api/usuario/actualizar.php>
  - : Documento : **Buscar** por parametro “s”
    - <http://162.214.187.236/~amj/api/usuario/buscar.php>
  - : Documento : **Consultar todos** los registros
    - <http://162.214.187.236/~amj/api/usuario/consulta.php>
  - : Documento : **Consultar por identificador**
    - [http://162.214.187.236/~amj/api/usuario/consulta\\_id.php](http://162.214.187.236/~amj/api/usuario/consulta_id.php)
  - : Documento : **Eliminar** registro por identificador
    - <http://162.214.187.236/~amj/api/usuario/eliminar.php>
  - : Documento : **Insertar** nuevo registro

```
- http://162.214.187.236/~amj/api/usuario/insertar.php
```

**NOTA:** En las funciones a utilizar se tiene que verificar lo siguiente: - Para insertar datos mediante la API se deberá de agregar en formato **JSON**, ejemplo:

```
{
  "iduser": 2,
  "name": "Alejandra",
  "last_name": "Montiel",
  "email": "ale@gmail.com",
  "phone": "2722341234",
  "birth": "2000-02-29",
  "adress": "Mendoza, Ver.",
  "employment": "Ingeniera",
  "status": 1,
  "username": "ale",
  "pass": "1212"
}
```

- En la función de **Buscar** se tiene que pasar como argumento una **S** seguido del valor a buscar, ejemplo:

```
- /buscar.php?s=texto
```

- Para el archivo de consultar se deberá de ingresar como argumento el campo iduser, seguido de un identificador anterior ya almacenado o existente.

```
- /consulta_id.php?iduser=5
```

- En la actualización de los datos de algún registro existente se deberán de enviar los datos en formato **JSON**

```
{
  "iduser": 1,
  "name": "Alejandra",
  "last_name": "Montiel",
  "email": "ajeandra@gmail.com",
  "phone": "2721234567",
  "birth": "2000-02-29",
  "adress": "Mendoza, Ver.",
  "employment": "Maestra",
  "status": 1,
  "username": "alejandra",
  "pass": "1212"
}
```

- Para eliminar un registro se deberá de enviar como argumento el identificador del registro a eliminar.

```
- /eliminar.php?iduser=10
```

## Resultados

- **CARPETA: Result** Contiene los archivos para el control de datos sobre los contactos

- Documento: **Consulta** realizará una búsqueda de todos los datos en el sistema

```
http://162.214.187.236/~amj/api/result/result.php
```

- Documento: **Consulta por Identificador** para su correcto funcionamiento es necesario enviar como argumento el identificador.

```
http://162.214.187.236/~amj/api/result/result.php
```

Ejemplo de consulta:

```
.../api/result/result.php?idresult=5
```

- Documento: **Insertar** el archivos crear nuevos registros

```
http://162.214.187.236/~amj/api/result/result.php
```

### Ejemplo de formato JSON:

```
{
  "idresult": 6,
  "number": 2,
  "score": 2,
  "arrhythmia": 2,
  "hyper": 2,
  "angina": 2,
  "heartattack": 2,
  "none": 2,
  "iduser": 2,
  "idwearadata": 2,
  "idmdata": 2
}
```

Wearable datos que procesa con sus sensores.

- **CARPETA: Wearable\*\*:** Contiene los archivos para el control de datos cada wearable registrado.

- Documento: **Consulta** realizará una búsqueda de todos los datos de cada wearable.

```
http://162.214.187.236/~amj/api/wearable/extra/consulta.php
```

- Documento : **Consulta por Identificador** para su correcto funcionamiento es necesario enviar como argumento el identificador

```
http://162.214.187.236/~amj/api/wearable/consulta_id.php
```

Ejemplo de consulta:

```
.../api/wearable/consulta_id.php?idwear=5
```

- Documento: **Actualizar** datos existentes en el sistema.

```
http://162.214.187.236/~amj/api/wearable/actualizar.php
```

**Ejemplo de formato JSON:**

```
{  
  "idwear": 5,  
  "name": "Fitbit",  
  "serie": "",  
  "model": "Charge",  
  "date": "2021-01-03",  
  "character": "Fitbit",  
  "data": "",  
  "descriptions": "",  
  "iduser": 5  
}
```

- Documento: **Eliminar registro**, para su óptimo funcionamiento es necesario enviar el argumento de identificador.

```
http://162.214.187.236/~amj/api/wearable/eliminar.php
```

Ejemplo de argumento:

```
.../api/wearable/eliminar.php?idwear=5
```

WearableData datos de cada dispositivo registrado en la plataforma.

- **CARPETA: WearableData** Contiene los archivos para el control de cada wearable registrado.
  - Documento : **Consulta por Identificador** para su correcto funcionamiento es necesario enviar como argumento el identificador

```
http://162.214.187.236/~amj/api/wearabledata/consulta_id.php
```

Ejemplo de consulta:

```
.../api/wearabledata/consulta_id.php?idweardata=1
```

- Documento: **Consulta últimos registro** para su funcionamiento no es necesario enviar argumentos, además no retornará errores.

```
http://162.214.187.236/~amj/api/wearabledata/ultimo_registro.php
```

**Ejemplo de retorno en formato JSON:**

```
{  
  "datos": [  
    {  
      "idweardata": 1,  
      "physical": 1,  
    }  
  ]  
}
```

```
        "bloodpressure": 1,  
        "heartrate": 1,  
        "iduser": 1  
    }  
]  
}
```

- Documento: **Consulta Últimos Diez Registros** para su funcionamiento no es necesario enviar argumentos, además no retornará errores.

[http://162.214.187.236/~amj/api/wearabledata/ultimos\\_10.php](http://162.214.187.236/~amj/api/wearabledata/ultimos_10.php)

**Ejemplo de retorno en formato JSON:**

```
{  
  "datos": [  
    {  
      "idweardata": 6,  
      "physical": 0,  
      "bloodpressure": 70,  
      "heartrate": 50,  
      "iduser": 1  
    },  
    {  
      "idweardata": 1,  
      "physical": 0,  
      "bloodpressure": 120,  
      "heartrate": 60,  
      "iduser": 1  
    }  
  ]  
}
```

El segundo documento es el manual de usuario de la aplicación para la detección temprana de enfermedades cardiovasculares, el cual se anexa en formato PDF.

### 3.1.5 Prueba y reparación

En esta última fase de la metodología Mobile-D, se trata de obtener la versión final de la solución, para eliminar aquellos rasgos innecesarios y cambiar algunas características.

### 3.1.5.1 Pruebas

Al finalizar la fase de desarrollo se enlazaron todos los componentes para generar la solución, a continuación se describe la funcionalidad obtenida en cada interfaz generada.

#### 3.1.5.1.1 Logos de la aplicación

Para tener una identificación personalizada de la aplicación, se seleccionaron colores favorecedores y un nombre apropiado. En la Figura 3.32 se aprecia el ícono y el nombre de AIDA, en la Figura 3.33, se contempla el splash screen que se colocó al iniciar la aplicación.

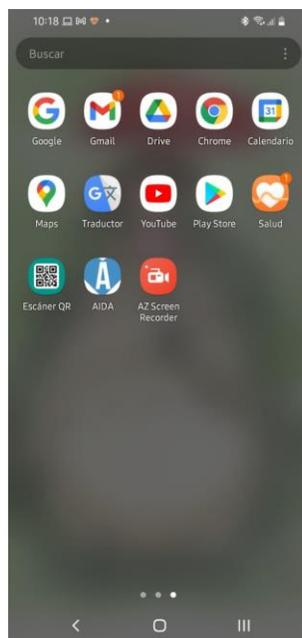


Figura 3. 32 Ícono de la aplicación en el dispositivo móvil



Figura 3. 33 Splash Screen de la aplicación

### 3.1.5.1.2 Registro de nueva cuenta

Para que el usuario ingrese a las funciones de la aplicación es necesario crear una cuenta en la que se solicitan datos importantes sobre su identidad, estas son las siguientes: Nombre, apellidos, correo electrónico, teléfono, fecha de nacimiento, dirección, ocupación, nombre de usuario (para las sesiones) y una contraseña (ver Figura 3.34). Una vez que se han ingresado todos los datos se debe presionar el botón Enviar como se muestra en la figura.



Figura 3. 34 Registro de nuevo usuario

### 3.1.5.1.3 Manejo de sesión

Las cuentas registradas ingresarán en un login (ver Figura 3.35), escribiendo nombre de usuario y contraseña, una vez llenados los espacios se selecciona el botón Acceder, se validará la existencia de los datos dentro de la aplicación, en caso de que esté el registro el usuario va a acceder a la aplicación, por el contrario, tendrá que crear cuenta nueva.



Figura 3. 35 Inicio de sesión

Para resguardar los datos de la contraseña se ha aplicado la opción de visualizar y ocultar valores, en la Figura 3.36 la contraseña está oculta, en la Figura 3.37 se habilitó el ícono de ver contraseña.



Figura 3. 36 Contraseña inhabilitada



Figura 3. 37 Contraseña habilitada

Si se ha iniciado sesión se puede cerrar desde el menú lateral seleccionando la última opción que brinda, tal como se visualiza en la Figura 3.38.



Figura 3. 38 Menú lateral para cierre de sesión

Al ingresar a la aplicación y validados los datos, se despliega el siguiente dashboard, la cual contiene los siguientes apartados:

- a. Datos de usuario
- b. Datos extraídos desde el wearable, en este caso se obtienen los latidos por minuto, la frecuencia cardiaca y los minutos de actividad física ejercida.
- c. Una tabla con datos médicos, para apoyar al usuario a visualizar el estatus de su salud, se genera una tabla señalando aquellos datos con mayor influencia y marcados con los colores del semáforo para representar en: verde = normal, amarillo = precaución y rojo = alerta.



Figura 3. 39 Dashboard de la aplicación

#### 3.1.5.1.4 Menú lateral

Para navegar dentro de la aplicación hay un menú que se encuentra en la parte lateral izquierda (ver Figura 3.40), que contiene todas las funcionalidades.

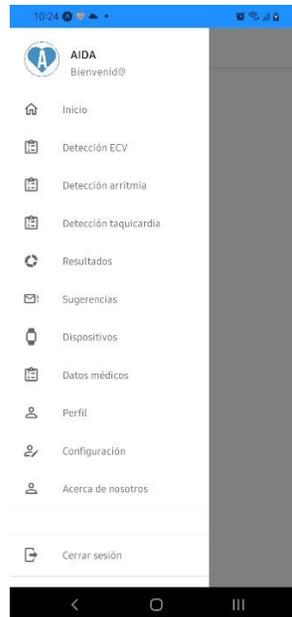


Figura 3. 40 Menú lateral

A continuación se describe la funcionalidad de las opciones:

- **Inicio:** la interfaz que se despliega al ingresar a la aplicación.
- **Detección temprana:** para realizar el cálculo de detección temprana de enfermedades cardiovasculares
- **Resultados:** en la que se presentan los resultados encontrados en los test de cálculo, que contiene información de las posibles enfermedades que son propensas las personas.
- **Sugerencias:** mensajes para el cuidado de la salud que se brindarán al usuario dependiendo de los resultados.
- **Dispositivos:** listado de dispositivos de monitorización conectados con la aplicación.
- **Datos médicos:** para nuevos usuarios es importante llenarlo antes del cálculo de probabilidad, en caso de los usuarios recurrentes solo se consultaría para actualizarlo.
- **Perfil:** acceso a datos del perfil y modificación de ellos.
- **Configuración:** para realizar modificaciones a datos del perfil.
- **Cerrar sesión:** para salir y cerrar la cuenta activa.

3.1.5.1.5 Datos del perfil

Al seleccionar Perfil aparece la interfaz que muestra la Figura 3.41, en la que se presentan datos ingresados al crear cuenta nueva.



Figura 3. 41 Datos del perfil de usuario

En caso de que el usuario requiera cambiar datos de registro deberá seleccionar Configuración en el menú del perfil o al consultar perfil puede seleccionar Modificar datos, y así editar la información necesaria (ver Figura 3.42), luego seleccionar el botón Actualizar.



Figura 3. 42 Configuración de datos personales

3.1.5.1.6 Cálculo de probabilidad

La finalidad de la aplicación AIDA es realizar una detección temprana de enfermedades cardiovasculares, pero también se ha contemplado la detección de arritmias y taquicardias, por lo que en esta sección se explicarán los componentes que realizan las funciones.

Se ingresan los datos en el formulario correspondiente (ver Figura 3.43), es importante tomar en cuenta que se debe tomar un análisis clínico para determinar los valores exactos y si es posible con apoyo de un médico ingresar los datos.

The image shows a mobile application interface for entering medical data. At the top, there is a status bar with the time 9:55 and various icons. Below that is a navigation bar with a hamburger menu icon and the text 'Datos'. The main content area has a blue header with the text 'Ingresa datos médicos'. Below the header, there are several input fields, each with a label and a value: 'Sexo' with '0', 'Colesterol HDL' with '80', 'Glucosa' with '110', 'Medida cintura' with '43', 'Colesterol LDL' with '60', and 'Estatura' with '140'. Each field has a small icon to its left. At the bottom of the form is a blue button labeled 'Enviar'. The entire form is set against a white background with a subtle shadow effect.

Figura 3. 43 Formulario para datos médicos

Los datos almacenados se consultan para generar la detección, para calcular la enfermedad cardiovascular solamente se ingresan datos obtenidos de un electrocardiograma, éstas se agregan en un formulario (ver Figura 3.44), al hacer clic en el botón enviar se realizará un cálculo y el resultado será presentado de acuerdo al nivel de probabilidad. En la Figura 3.45 se presenta un resultado del 79% que no se padecerá alguna enfermedad cardiovascular.

The screenshot shows a mobile application interface titled "Detección de ECV" and "Detección con el Modelo Random Forest". It contains a form with the following fields: "Edad" (Age) with the value 22, "Creatinina" (Creatinine) with the value 2, "Old Peak" with the value 3, and "Rest ECF" with the value 0.2. Below the form is a blue button labeled "Enviar". A note above the form states: "Los siguientes datos deben ser obtenidos del estudio de electrocardiograma".

Figura 3. 44 Formulario para la detección temprana

The screenshot shows the result screen of the application. It displays "Resultado obtenido" (Result obtained) with a percentage of "Porcentaje: 79%". The result is "No" (No). A blue button labeled "Aceptar" (Accept) is visible at the bottom. A cartoon heart character is shown with a happy expression.

Figura 3. 45 Resultado favorable

En caso de que exista una probabilidad de llegar a obtener alguna enfermedad, se distribuye el valor en medio y alto riesgo, la primera se presenta con una imagen de un corazón triste (ver Figura 3.46), la segunda se representa con una imagen de corazón enfermo (ver Figura 3.47).

The screenshot shows the result screen of the application. It displays "Resultado obtenido" (Result obtained) with a percentage of "Porcentaje: 79.5%". The result is "Si" (Yes). A blue button labeled "Aceptar" (Accept) is visible at the bottom. A cartoon heart character is shown with a sad expression.

Figura 3. 46 Resultado regular

The screenshot shows the result screen of the application. It displays "Resultado obtenido" (Result obtained) with a percentage of "Porcentaje: 100%". The result is "Si" (Yes). A blue button labeled "Aceptar" (Accept) is visible at the bottom. A cartoon heart character is shown with a sad expression.

Figura 3. 47 Resultado malo o de precaución

Para calcular la probabilidad de padecer arritmia, se solicita ingresar los siguientes datos en la interfaz específica (ver Figura 3.48).



10:24

☰ Detección de arritmia

Los siguientes datos son obligatorios

Triglicéridos  
3

Fracción de eyección  
1

Taquicardia sinusal  
1

Enviar

Figura 3. 48 Detección de arritmia

Una vez evaluados los valores se despliega el porcentaje de probabilidad y una imagen representativa, tal como se muestra en la Figura 3.49.



10:24

☰ Detección de arritmia

Los siguientes datos son obligatorios

Triglicéridos  
80

Fra  
1

Taq  
1

Resultado obtenido

¡

Sí

Porcentaje: 42.25%

Aceptar

Figura 3. 49 Resultados de la detección de arritmia

Por último, en la opción de Detectar taquicardia, se busca el porcentaje que tiene una persona en llegar a padecer taquicardia, por lo que simplemente se ingresan los datos en la Figura 3.50, y se consultan datos médicos en la base de conocimientos para encontrar el valor de sí padecer la enfermedad (ver Figura 3.51).



Figura 3. 50 Detección de taquicardia



Figura 3. 51 Resultados de la detección de taquicardia

En la siguiente opción se encuentran los cálculos parciales, en la que con una tabla se representa el riesgo ante las enfermedades cardiovasculares más frecuentes. La Figura 3.52 contiene la visualización del primer test que el usuario realizó y los resultados obtenidos.



Figura 3. 52 Resultados parciales de la detección

### 3.1.5.1.7 Sugerencias para el cuidado de la salud

De acuerdo a los resultados obtenidos en la realización del cálculo de probabilidad de padecer alguna enfermedad cardiovascular, se brindan al usuario sugerencias para el cuidado de su salud. Se consultan en menú lateral izquierdo en la opción Sugerencias.

La Figura 3.53 muestra las sugerencias recientes que han llegado a la aplicación, se representa la prioridad que tienen, la cual puede ser baja, media o alta.

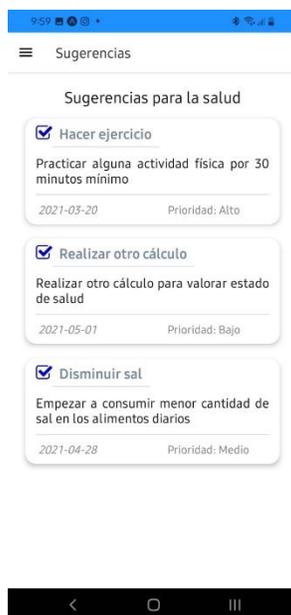


Figura 3. 53 Sugerencias para el cuidado de la salud

3.1.5.1.8 Dispositivos conectados con la cuenta del usuario

Para recuperar los datos de frecuencia cardiaca y actividad física se enlaza la cuenta en la aplicación con un wearable por lo que en el menú se ha agregado una opción para ver los dispositivos conectados. En la Figura 3.54 se presenta el dispositivo conectado actualmente con unas imágenes de referencia.



Figura 3. 54 Datos del Fitbit conectado

En la Figura 3.55 se muestra la descripción del dispositivo conectado y la fecha de enlace. Sucesivamente en la Figura 3.56 se presentan las características del Fitbit conectado.

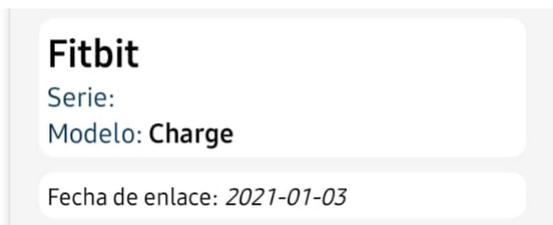


Figura 3. 55 Descripción de dispositivo conectado

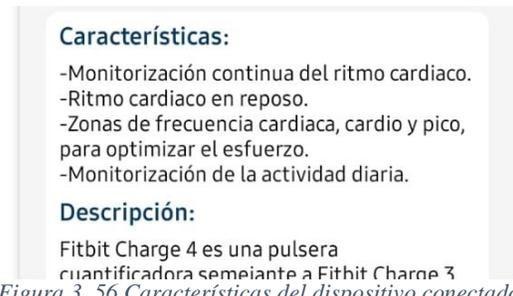


Figura 3. 56 Características del dispositivo conectado

### 3.1.5.2 Reparación del sistema

En las primeras pruebas, el resultado de probabilidad se desplegaba en forma de texto tal como se muestra en la Figura 3.57, pero se modificó la interfaz para ser mayormente comprensible ante cualquier tipo de usuario.

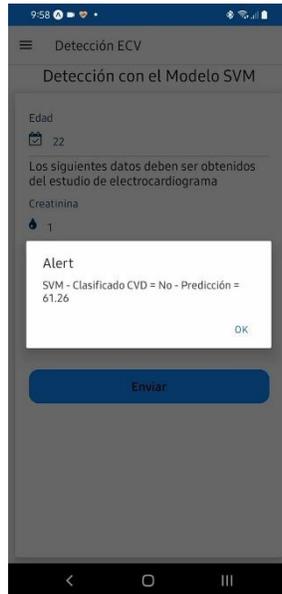


Figura 3. 57 Resultados con alerta anterior



Figura 3. 58 Resultados con alerta actual

Otra configuración que se realizó es el uso de bibliotecas, ya que al realizar la primera versión ejecutable de la aplicación, los dispositivos no reconocen las bibliotecas y por ello se quedaba en blanco la interfaz. Se tomó el acuerdo del cambio de interfaz, en el uso de otras bibliotecas y módulos de apoyo.

## Capítulo 4. Resultados

En este capítulo se describen los resultados obtenidos con la realización del presente proyecto.

Se obtuvo una aplicación para dispositivos móviles con interfaces que permiten que las personas accedan de manera sencilla a las funcionalidades de la aplicación, ya que contienen elementos cordiales: colores, formas, menús, entre otras. El ingreso de datos es fácil esto es por los menús desplegables y la escritura dentro de cada opción.

El hecho de usar las matrices como un estándar en la generación de aplicaciones para el cuidado de la salud, permitió crear funciones que ayudan a las personas a cuidar sus condiciones de vida y estar alertas para cualquier complicación.

A continuación se describen el caso de estudio en la que se llevó a cabo la prueba de funcionalidad de la solución.

### 4.1 Caso de estudio

Para llevar a cabo las pruebas de la aplicación se solicitó la colaboración de una asociación que tiene dos consultorios médicos en comunidades rurales. Una de ellas localizada en la comunidad de Los Fresnos, La Perla, Ver. que atiende actualmente 150 pacientes, de diferentes edades y condiciones de salud. Los servicios principales que ofrece el consultorio son:

- Atención básica
- Maternidad (desde los primeros meses de embarazo, parto y posparto)
- Seguimiento de enfermedades crónicas
- Seguimiento de operaciones y cirugías
- Pediatría
- Análisis clínicos

Al conversar con la médico en turno, ella autorizó aplicar la detección temprana a 10 de sus pacientes, las cuales diversificó de la siguiente manera:

*Tabla 4. 1 Grupo muestra para el caso de estudio*

<b>Número de pacientes</b>	<b>Rango</b>
3	Entre 18 y 30 años, sin enfermedades crónicas.
4	Entre 35 y 60 años, con alguna enfermedad no cardiovascular
3	Entre 25 y 40, con antecedentes familiares de enfermedades cardiovasculares.

Los pacientes fueron 3 hombres y 7 mujeres, ya que las que más acuden al consultorio son mujeres y niños.

Una vez seleccionado el grupo muestra, se continuó con la interacción de cada usuario con la aplicación. Algunos casos se describen a continuación.

En la Figura 4.1 se presenta a una paciente de 24 años de edad, sin ningún padecimiento ni malestar de alguna enfermedad crónica, cuida su alimentación y realiza ejercicio moderado. La mayoría de tiempo se la pasa sentada por cuestiones escolares.



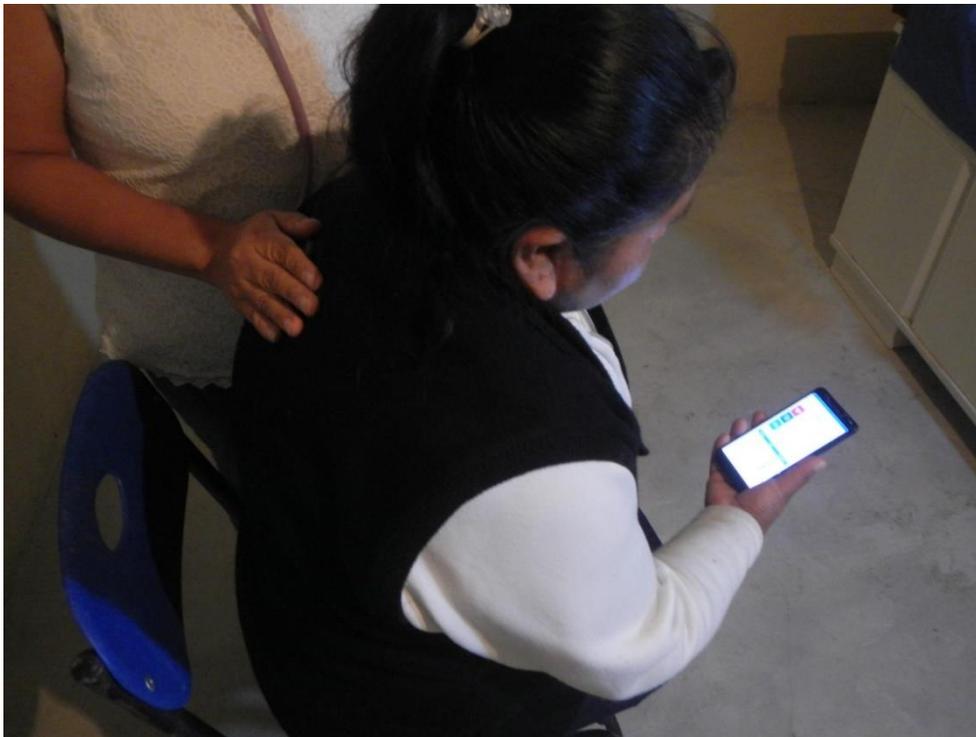
*Figura 4. 1 Paciente 1 usando la aplicación*

En la Figura 4.2 se muestra a una paciente de 18 años de edad, la cual tiene familiares con enfermedades crónicas (diabetes e hipertensión), las condiciones de salud actuales son estables, con peso adecuado, ella comúnmente realiza actividad física en casa.



*Figura 4. 2 Paciente 2 usando la aplicación*

En la Figura 4.3 se aprecia a una paciente con una enfermedad pulmonar y de 45 años de edad, con seguimiento de prediabetes, con escolaridad de primaria.



*Figura 4. 3 Paciente 3 usando la aplicación*

Cada paciente se registró dentro de la aplicación, en caso de los que no saben escribir se les apoyó. Después ingresaron con su nombre de usuario y contraseña, luego todos fueron llenando el formulario de datos médicos (Figura 4.4).

9:55 Datos

Ingresa datos médicos

Sexo  
0

Colesterol HDL  
80

Glucosa  
110

Medida cintura  
43

Colesterol LDL  
60

Estatura  
140

Enviar

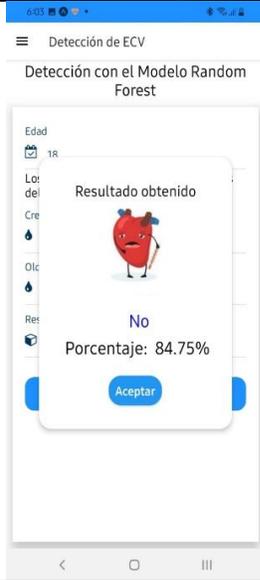
Figura 4. 4 Datos médicos

Sucesivamente con apoyo de la doctora se recuperaron datos de electrocardiograma y con ello se generó la probabilidad de tener una enfermedad cardiovascular. Cabe mencionar que para los datos del wearable, se utilizó una que posee la doctora.

Después de que los diez pacientes interactuaron con la aplicación se obtuvieron los siguientes resultados:

Tabla 4. 2 Resultados obtenidos por cada paciente

Persona	Resultado	Resultado en aplicación
<p>1: Sexo: hombre, Edad: 53 años, Ocupación: campesino, Enfermedad: ninguna, Antecedentes: cirugía</p>	<p>Sí en un 72 %</p>	
<p>2: sexo : mujer, Edad: 45 años, Ocupación: ama de casa, Enfermedad: daño en pulmones, Antecedentes: prediabetes</p>	<p>Sí en un 50.25%</p>	

Persona	Resultado	Resultado en aplicación
<p>3: sexo: mujer,                      Edad: 18 años,                      Ocupación: estudiante,                      Enfermedad: ninguna,                      Antecedentes: familiares con diabetes.</p>	<p>No en un 84.75%</p>	
<p>4: Sexo: hombre,                      Edad: 33 años,                      Ocupación: campesino,                      Enfermedad: ninguna,                      Antecedentes: familiares con hipertensión</p>	<p>No en un 80%</p>	

Persona	Resultado	Resultado en aplicación
<p>5: Sexo: mujer, Edad: 29 años, Ocupación: campesina, Enfermedad: asma, Antecedentes: ninguna</p>	<p>Sí en un 50.25%</p>	
<p>6: Sexo: mujer, Edad: 25 años, Ocupación: secretaria, Enfermedad: ninguna, Antecedentes: ninguna</p>	<p>No en un 99.75%</p>	

Persona	Resultado	Resultado en aplicación
<p>7: Sexo: hombre,                      Edad: 27 años,                      Ocupación: licenciado,                      Enfermedad: ninguna,                      Antecedentes: familiares con diabetes</p>	<p>No en un 79%</p>	
<p>8: Sexo: mujer,                      Edad: 24 años,                      Ocupación: estudiante,                      Enfermedad: ninguna,                      Antecedentes: ninguna</p>	<p>No en un 97.75%</p>	

Persona	Resultado	Resultado en aplicación
<p>9: Sexo: mujer,                      Edad: 62 años,                      Ocupación: ama de casa,                      Enfermedad: ninguna,                      Antecedentes: familiares con hipertensión</p>	<p>Sí en un 86.25%</p>	
<p>10: Sexo: mujer,                      Edad: 35 años,                      Ocupación: campesina,                      Enfermedad: colesterol elevado,                      Antecedentes: ninguna</p>	<p>Sí en un 70.75%</p>	

De acuerdo a los valores de probabilidad, la aplicación brindó sugerencias, y la doctora quedó en darle seguimiento a los que resultaron en alerta.

## 4.2 Evaluación y resultados

Para generar una evaluación acertada de la aplicación se aplicó un test a las personas que interactuaron con la aplicación. El Anexo 1. Es la que se les brindó a las personas. En la siguiente tabla se organizan las respuestas que anotaron los participantes:

*Tabla 4. 3 Evaluación del uso de la aplicación*

Pregunta	Respuesta más común	Porcentaje de aceptación
1. ¿Qué le ha parecido la aplicación?	Muy dinámica y fácil de usar; es muy sencilla	95%
2. ¿Cómo le parecieron los colores usados?	Son adecuados a la salud; son colores que no lastiman la vista; son reconocibles.	93%
3. ¿Qué opina de la detección de enfermedades cardiovasculares que hace la aplicación?	Me dice cuánto riesgo tengo; me dice que tengo que cuidarme; me dice sí o no estoy en riesgo.	100%
4. ¿Cree que cualquier persona la puede usar?	Sí; sólo que uno sepa leer y escribir.	95%
5. ¿En qué momento usted la usaría?	Cuándo me siento mal; cuándo quiera saber cómo anda mi salud.	98%
6. ¿Le facilita la forma de saber sobre su salud?	Sí	100%
7. ¿Qué le parecen las sugerencias?	Adecuadas; solo que tengo que hacer caso para mejorar mi salud.	98%
8. ¿Cómo se sintió usando la aplicación?	Tranquila; cuando me dio el resultado me sorprendí.	96%
9. ¿Tiene alguna sugerencia con respecto a la aplicación?	Que después me diga de otras enfermedades; que tenga más dibujos.	98%
10. ¿Tiene alguna duda?	¿Siempre debo apoyarme de la doctora?; ¿La doctora es quién dará seguimiento a mi salud?; ¿Dónde la puedo descargar?	60%

Respecto a las dudas y comentarios que surgieron, se aclararon en ese momento, se apoyó a aclarar cuestiones de uso de la aplicación.

## Capítulo 5. Conclusiones y recomendaciones

En este capítulo se agregan las conclusiones y recomendaciones de la solución generada durante este proyecto de tesis.

### 5.1 Conclusiones

Mediante el proceso de ejecución del proyecto de tesis, se lograron cumplir con el objetivo general y los objetivos específicos, planteados al inicio, a través del desarrollo de una aplicación para dispositivos móviles, con la finalidad de realizar una detección temprana de las enfermedades cardiovasculares. Para el desarrollo se realizaron análisis minuciosos de trabajos que tengan la misma finalidad del presente, en la que se rescataron ideas sobre el uso de tecnologías y herramientas que actualmente están en su mayor auge.

Al realizar la investigación de cada tecnología se tuvo que realizar un análisis acerca de qué tipo de resultado se esperaba, por ello se seleccionaron las adecuadas. Aparte de las tecnologías, también se estudiaron los dispositivos de monitorización, generando una lista de biomarcadores que proveen y entre ellas seleccionar la mejor.

Pasando al tema de modelos de predicción, éstas se seleccionaron y probaron con distintos valores, al final se optó por las que entregaban mejores resultados, posteriormente se tuvieron que entrenar con datos de diferentes casos, para que al realizar el cálculo se obtuviera el resultado más acertado.

La principal aportación es la aplicación llamada AIDA, que tiene como finalidad realizar una detección temprana de las enfermedades cardiovasculares, así también realiza una detección de arritmias y taquicardias valorando el porcentaje de acertación de padecer la enfermedad. Para realizar el cálculo de probabilidad se recaudan datos de manera manual en la aplicación y se complementan con datos que se obtienen del dispositivo de monitorización de signos vitales (wearable).

La aplicación se ha realizado de acuerdo a ciertas métricas para que el usuario tenga una experiencia favorable y confiable. Se contempló el cuidado de toda la información que el usuario ingrese, por lo que la seguridad y privacidad es una prioridad.

La aplicación cuenta con íconos que el usuario conoce o se ha familiarizado anteriormente, para que le sea accesible la manipulación de cada interfaz. Además de esas funcionalidades también se ha contemplado apoyar al usuario con una lista de sugerencias para el cuidado de su salud, dependiendo de sus resultados. Con todo esto se espera apoyar a las personas sintomáticas o asintomáticas a detectar las posibles enfermedades cardiovasculares a los que está expuesto a lo largo de su vida si no cuida su salud.

## **5.2 Recomendaciones**

Algunas recomendaciones para el seguimiento de este proyecto y para futuros trabajos, son las que a continuación se enumeran.

1. Crear la aplicación mayormente intuitiva, cuando se realizaron las pruebas con personas de comunidades rurales, se encontró que no todos saben leer y escribir, pero que están interesados en usar la aplicación, por ello se debe contemplar una forma de acercar el diseño a ellos.
2. Conectar con médicos especialistas con la finalidad de dar seguimiento a los resultados, para el futuro es necesario colocar un enlace con médicos.
3. Agregar opción de descarga de datos, esto con el fin de obtener un archivo para compartir o imprimir.

Estas recomendaciones son generales y algunos son de acuerdo a la observación que se hizo en el caso de estudio.

## Productos académicos

En esta sección se brinda información acerca de los productos académicos que se han generado.

### 6.1 Artículo de congreso

Detección temprana de enfermedades cardiovasculares a través de análisis de biomarcadores y modelos de predicción.

Alejandra Montiel de Jesús, Nancy Aracely Cruz Ramos, Lisbeth Rodríguez Mazahua, Luis Ángel Reyes Hernández, Luis Rolando Guarneros Nolasco, José Luis Sánchez-Cervantes



### 6.2 Registro de derechos de autor

Se realizó el trámite de registro de derecho de autor ante el Instituto Nacional de Derechos de autor (INDAUTOR) del Software: AIDA: Aplicación para dispositivos móviles para la detección temprana de enfermedades cardiovasculares. (Estatus en proceso)

Autores: Alejandra Montiel de Jesús, José Luis Sánchez Cervantes, Nancy Aracely Cruz Ramos y Giner Alor Hernández

## Anexos

### Test de interacción

1. ¿Qué le ha parecido la aplicación?
2. ¿Cómo le parecieron los colores usados?
3. ¿Qué opina de la detección de enfermedades cardiovasculares que hace la aplicación?
4. ¿Cree que cualquier persona la puede usar?
5. ¿En qué momento usted la usaría?
6. ¿Le facilita la forma de saber sobre su salud?
7. ¿Qué le parecen las sugerencias?
8. ¿Cómo se sintió usando la aplicación?
9. ¿Tiene alguna sugerencia con respecto a la aplicación?
10. ¿Tiene alguna duda?

## Referencias

- [1] Michael A. Chen, MD, PhD, Associate Professor of Medicine, Division of Cardiology, Harborview Medical Center, University of Washington Medical School, Seattle, WA. Also reviewed by David Zieve, MD, MHA, Medical Director, Brenda Conaway, Editorial Director, “Qué es la enfermedad cardiovascular,” *MedlinePlus*, 2018. .
- [2] D. Kaushik, V.; Gupta, K.; Gupta, “React Native Application Development,” *Int. J. Adv. Stud. Sci. Res.*, 2018.
- [3] C. K. Chow, N. Ariyaratna, S. M. S. Islam, A. Thiagalingam, and J. Redfern, “mHealth in Cardiovascular Health Care,” *Heart Lung and Circulation*. 2016, doi: 10.1016/j.hlc.2016.04.009.
- [4] H. Oh, C. Rizo, M. Enkin, and A. Jadad, “What is eHealth (3): A systematic review of published definitions,” *Journal of Medical Internet Research*. 2005, doi: 10.2196/jmir.7.1.e1.
- [5] J. M. Cueva Lovelle, J. I. Rodriguez Molano, and C. E. Montenegro Marin, “INTRODUCCIÓN AL INTERNET DE LAS COSAS,” *Redes Ing.*, 2015, doi: 10.14483/2248762x.8505.
- [6] “Dispositivos eléctricos y monitorización remota,” *Rev. Uruguaya Cardiol.*, 2013.
- [7] J. J. Rutherford, “Wearable technology,” *IEEE Eng. Med. Biol. Mag.*, 2010, doi: 10.1109/MEMB.2010.936550.
- [8] T. Guerra Dávila, *Bioestadística*. 2014.
- [9] C. G. QUINTERO M., J. A. O. LÓPEZ, and H. J. A. DE LA HOZ, “Sistemas de adquisición de datos,” in *Instrumentación electrónica aplicada*, 2018.
- [10] I. Torres Courchoud and J. I. Pérez Calvo, “Biomarcadores y práctica clínica,” *Anales del Sistema Sanitario de Navarra*. 2016, doi: 10.4321/S1137-6627/2016000100001.
- [11] G. Prashant, “Decision Trees in Machine Learning – Towards Data Science,” *Towar. Data Sci.*, 2017.
- [12] C. J. O’Donnell and R. Elosua, “Factores de riesgo cardiovascular. Perspectivas derivadas del Framingham Heart Study,” *Rev. Española Cardiol.*, 2008, doi: 10.1157/13116658.
- [13] P. López-Roldán and S. Fachelli, “Clasificación de las técnicas de análisis de datos,” in *Metodología de la investigación social cuantitativa*, 2015.
- [14] M. E. Raffino, “Base de Datos: Concepto, Tipos y Ejemplos,” 2020, 2020. .
- [15] A. Gandomi and M. Haider, “Beyond the hype: Big data concepts, methods, and analytics,” *Int. J. Inf. Manage.*, 2015, doi: 10.1016/j.ijinfomgt.2014.10.007.
- [16] R. Steyer and R. Steyer, “JavaScript,” in *JavaScript*, 2014.

- [17] J. Krause and J. Krause, “CSS: Cascading Style Sheets,” in *Introducing Web Development*, 2016.
- [18] J. Duckett, *HTML & CSS*. 2019.
- [19] J. Villavicencio, “Introducción a Series de Tiempo,” *Man. Metodol. Ser. tiempo*, 2010, doi: 10.1177/0266666915575167.
- [20] C. García and I. Gómez, “Algoritmos de aprendizaje: knn & kmeans,” *Universidad Carlos III de Madrid*. 2006.
- [21] A. C. of Radiology, “Detección temprana de enfermedades cardíacas (del corazón),” *Am. Coll. Radiol. Radiol. Soc. North Am.*, 2017.
- [22] J. Valero Sancho, “La visualización de datos,” *Ambitos Rev. Int. Comun.*, 2014.
- [23] S. Ginebra, “Organización Mundial de la Salud,” *Wwwwho.int*, 2011.
- [24] G. M. Coorey, L. Neubeck, J. Mulley, and J. Redfern, “Effectiveness, acceptability and usefulness of mobile applications for cardiovascular disease self-management: Systematic review with meta-synthesis of quantitative and qualitative data,” *Eur. J. Prev. Cardiol.*, vol. 25, no. 5, pp. 505–521, 2018, doi: 10.1177/2047487317750913.
- [25] B. Urrea *et al.*, “Mobile Health Initiatives to Improve Outcomes in Primary Prevention of Cardiovascular Disease,” *Curr. Treat. Options Cardiovasc. Med.*, vol. 17, no. 12, p. 59, 2015, doi: 10.1007/s11936-015-0417-7.
- [26] S. Nashif, M. R. Raihan, M. R. Islam, and M. H. Imam, “Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System,” *World J. Eng. Technol.*, 2018, doi: 10.4236/wjet.2018.64057.
- [27] L. Moreno-Alsasua, B. Garcia-Zapirain, J. David Rodrigo-Carbonero, I. O. Ruiz, S. Hamrioui, and I. de la Torre Díez, “Primary Prevention of Asymptomatic Cardiovascular Disease Using Physiological Sensors Connected to an iOS App,” *J. Med. Syst.*, vol. 41, no. 12, p. 191, 2017, doi: 10.1007/s10916-017-0840-2.
- [28] A. Ave, H. Fauzan, S. R. Adhitya, and H. Zakaria, “Early detection of cardiovascular disease with photoplethysmogram(PPG) sensor,” in *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2015, pp. 676–681, doi: 10.1109/ICEEI.2015.7352584.
- [29] H. Alshraideh, M. Otoom, A. Al-Araida, H. Bawaneh, and J. Bravo, “A Web Based Cardiovascular Disease Detection System,” *J. Med. Syst.*, vol. 39, no. 10, p. 122, 2015, doi: 10.1007/s10916-015-0290-7.
- [30] S. K. Jain and B. Bhaumik, “An Energy Efficient ECG Signal Processor Detecting Cardiovascular Diseases on Smartphone,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 2, pp. 314–323, 2017, doi: 10.1109/TBCAS.2016.2592382.
- [31] D. Azariadi, V. Tsoutsouras, S. Xydis, and D. Soudris, “ECG signal analysis and arrhythmia detection on IoT wearable medical devices,” in *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, May 2016, pp. 1–4, doi: 10.1109/MOCASST.2016.7495143.

- [32] Y. J. Lin *et al.*, “Artificial Intelligence of Things Wearable System for Cardiac Disease Detection,” 2019, doi: 10.1109/AICAS.2019.8771630.
- [33] Z. Yang, Q. Zhou, L. Lei, K. Zheng, and W. Xiang, “An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare,” *J. Med. Syst.*, 2016, doi: 10.1007/s10916-016-0644-9.
- [34] M. Lang, “Beyond Fitbit: A Critical Appraisal of Optical Heart Rate Monitoring Wearables and Apps, Their Current Limitations and Legal Implications,” *Alb. L. J. Sci. Tech.*, 2017.
- [35] M. P. Turakhia *et al.*, “Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: The Apple Heart Study,” *Am. Heart J.*, vol. 207, pp. 66–75, 2019, doi: <https://doi.org/10.1016/j.ahj.2018.09.002>.
- [36] Z. Iftikhar *et al.*, “Multiclass Classifier based Cardiovascular Condition Detection Using Smartphone Mechanocardiography,” *Sci. Rep.*, vol. 8, no. 1, p. 9344, 2018, doi: 10.1038/s41598-018-27683-9.
- [37] G. Prabhu *et al.*, “Design and Development of the {MedFit} App: A Mobile Application for Cardiovascular Disease Rehabilitation,” in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer International Publishing, 2018, pp. 20–28.
- [38] F. Lobelo *et al.*, “The Wild Wild West: A Framework to Integrate mHealth Software Applications and Wearables to Support Physical Activity Assessment, Counseling and Interventions for Cardiovascular Disease Risk Reduction,” *Prog. Cardiovasc. Dis.*, vol. 58, no. 6, pp. 584–594, 2016, doi: <https://doi.org/10.1016/j.pcad.2016.02.007>.
- [39] T. De Pessemier, E. Cailliau, and L. Martens, “Heart rate monitoring and activity recognition using wearables,” in *Sixth International Conference on Building and Exploring Web Based Environments (WEB 2018)*, 2018, pp. 10–15.
- [40] G. Bhat, R. Deb, and U. Y. Ogras, “OpenHealth: Open-Source Platform for Wearable Health Monitoring,” *IEEE Des. Test*, vol. 36, no. 5, pp. 27–34, Oct. 2019, doi: 10.1109/MDAT.2019.2906110.
- [41] H. Eyles, R. McLean, B. Neal, R. N. Doughty, Y. Jiang, and C. N. Mhurchu, “Using mobile technology to support lower-salt food choices for people with cardiovascular disease: protocol for the SaltSwitch randomized controlled trial,” *BMC Public Health*, vol. 14, no. 1, p. 950, 2014, doi: 10.1186/1471-2458-14-950.
- [42] I. de la Torre-Díez, B. Martínez-Pérez, M. López-Coronado, J. J. P. C. Rodrigues, and J. Arambarri, “Development and validation of a mobile health app for the self-management and education of cardiac patients,” in *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 2016, pp. 1–5, doi: 10.1109/CISTI.2016.7521516.
- [43] Y. Ranjan *et al.*, “RADAR-Base: Open Source Mobile Health Platform for Collecting, Monitoring, and Analyzing Data Using Sensors, Wearables, and Mobile Devices,” *JMIR Mhealth Uhealth*, vol. 7, no. 8, p. e11734, 2019, doi: 10.2196/11734.

- [44] L. Neubeck, N. Lowres, E. J. Benjamin, S. Ben Freedman, G. Coorey, and J. Redfern, “The mobile revolution—using smartphone apps to prevent cardiovascular disease,” *Nat. Rev. Cardiol.*, vol. 12, no. 6, p. 350, 2015.
- [45] P. Abrahamsson *et al.*, “Mobile-D: An agile approach for mobile application development,” 2004, doi: 10.1145/1028664.1028736.
- [46] Visual Studio Code, “Documentation for Visual Studio Code,” *Visual Studio Code Documentation*, 2015. .
- [47] S. Benedetto, C. Caldato, E. Bazzan, D. C. Greenwood, V. Pensabene, and P. Actis, “Assessment of the Fitbit Charge 2 for monitoring heart rate,” *PLoS One*, vol. 13, no. 2, p. e0192691, 2018.
- [48] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
- [49] W. S. Noble, “What is a support vector machine?,” *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.
- [50] V. J. Pandya, “Comparing handwritten character recognition by AdaBoostClassifier and KNeighborsClassifier,” in *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2016, pp. 271–274.
- [51] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Trans. Syst. Man. Cybern.*, vol. 21, no. 3, pp. 660–674, 1991.
- [52] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [53] A. Natekin and A. Knoll, “Gradient boosting machines, a tutorial,” *Front. Neurobot.*, vol. 7, p. 21, 2013.
- [54] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: unbiased boosting with categorical features,” *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
- [55] C. Zhang, X. Zhang, and R. Halstead-Nussloch, “ASSESSMENT METRICS, CHALLENGES AND STRATEGIES FOR MOBILE HEALTH APPS.,” *Issues Inf. Syst.*, vol. 15, no. 2, 2014.
- [56] T. Lehto and H. Oinas-Kukkonen, “Persuasive features in web-based alcohol and smoking interventions: a systematic review of the literature,” *J. Med. Internet Res.*, vol. 13, no. 3, p. e46, 2011.
- [57] C. Free *et al.*, “The effectiveness of mobile-health technology-based health behaviour change or disease management interventions for health care consumers: a systematic review,” *PLoS med*, vol. 10, no. 1, p. e1001362, 2013.
- [58] L. Dennison, L. Morrison, G. Conway, and L. Yardley, “Opportunities and challenges for smartphone applications in supporting health behavior change: qualitative study,” *J. Med. Internet Res.*, vol. 15, no. 4, p. e86, 2013.
- [59] S. Michie, M. M. Van Stralen, and R. West, “The behaviour change wheel: a new

method for characterising and designing behaviour change interventions,” *Implement. Sci.*, vol. 6, no. 1, pp. 1–12, 2011.

- [60] G. Zichermann and C. Cunningham, *Gamification by design: Implementing game mechanics in web and mobile apps*. “O’Reilly Media, Inc.,” 2011.
- [61] J. King, A. Lampinen, and A. Smolen, “Privacy: Is there an app for that?,” in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, 2011, pp. 1–20.
- [62] B. Ploderer, W. Reitberger, H. Oinas-Kukkonen, and J. van Gemert-Pijnen, “Social interaction and reflection for behaviour change.” Springer, 2014.
- [63] A. Sunyaev, T. Dehling, P. L. Taylor, and K. D. Mandl, “Availability and quality of mobile health app privacy policies,” *J. Am. Med. Informatics Assoc.*, vol. 22, no. e1, pp. e28–e33, 2015.
- [64] W. L. Haskell *et al.*, “Effects of intensive multiple risk factor reduction on coronary atherosclerosis and clinical cardiac events in men and women with coronary artery disease. The Stanford Coronary Risk Intervention Project (SCRIP).,” *Circulation*, vol. 89, no. 3, pp. 975–990, 1994.