



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

DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES

Juan Luis Landaverde González

Asesorado por el Ing. Gabriel Alejandro Díaz López

Guatemala, octubre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JUAN LUIS LANDAVERDE GONZÁLEZ

ASESORADO POR EL ING. GABRIEL ALEJANDRO DÍAZ LÓPEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIA	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
EXAMINADOR	Ing. Sergio Arnoldo Méndez Aguilar
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 13 de marzo de 2020

Juan Luis Landaverde González

Guatemala, 31 de Mayo de 2021

Ingeniero Carlos Azurdia
Revisor de Trabajo de Graduación
Escuela de Ciencias y Sistemas
Faculta de Ingenieria

Respetable Ing. Azurdia:

Me complace saludarle, haciendo referencia al trabajo de graduación titulado "DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES", desarrollado por el estudiante universitario JUAN LUIS LANDAVERDE GONZÁLEZ, que como asesor apruebo el contenido y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Para su conocimiento y efectos, sin otro particular me suscribo.

Atentamente



Ing. Gabriel Alejandro Díaz López
COL. 11630

Ing. Gabriel Alejandro Díaz López
Ingenieria en Ciencias en Sistemas
Asesor de Trabajo de Graduación



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

Guatemala 8 de julio de 2021

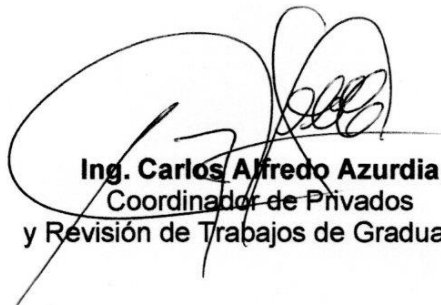
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JUAN LUIS LANDAVERDE GONZÁLEZ** con carné **199811522** y CUI **2228 18026 0101** titulado **“DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES”** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación “**DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES**”, realizado por el estudiante, JUAN LUIS LANDAVERDE GONZÁLEZ aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Msc. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 26 de octubre de 2021



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Decanato
Facultad de Ingeniería
24189101- 24189102
secretariadecanato@ingenieria.usac.edu.gt

DTG. 558.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DESARROLLO DE APLICATIVO MÓVIL PARA MONITOREO DE CORREDORES**, presentado por el estudiante universitario: **Juan Luis Landaverde González**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Inga. Anabela Cordova Estrada
Decana



Guatemala, octubre de 2021

AACE/asga

ACTO QUE DEDICO A:

Dios

Quien me ha dado cada amanecer un nuevo aliento de vida y la sabiduría para dirigir y corregir mi vida de manera correcta.

Mi madre

Gregoria González, quien es el motor de mi vida, siempre impulsándome hacia adelante, de quien he aprendido todo lo que soy, siendo el modelo del que me he formado. Una mente curiosa, inquieta y para la cual nunca será muy tarde para seguir descubriendo los misterios del universo.

Mi hermana

Patricia Landaverde González, la científica, la hermana latosa, mi otra mamá, la sensata, mi ejemplo a seguir.

Mi hermano

Robín Emilio Landaverde González, el futuro ingeniero en sistemas, que espero lo haya inspirado de buena manera.

AGRADECIMIENTOS A:

**Universidad de San
Carlos de Guatemala**

Por haber sido mi hogar por tantos años, y sí que fueron muchos, haberme dado la oportunidad de superarme y aportar a la sociedad de Guatemala.

Facultad de Ingeniería

Por haber reunido tanta gente buena en un solo lugar que me ayudaron y guiaron, no solo en lo académico sino también para la vida en general.

**Mis amigos de la
Facultad**

Los que me incentivaron a seguir adelante cuando todo parecía perdido.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IX
RESUMEN	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. BENEFICIOS DEL DEPORTE EN LA SALUD FÍSICA Y MENTAL.....	1
1.1. Tipos de ejercicios.....	2
1.1.1. Ejercicios aeróbicos.....	2
1.1.2. Ejercicios de flexibilidad.....	2
1.1.3. Ejercicios de fuerza y resistencia.....	3
1.2. Riesgos en la práctica de deportes	3
2. TECNOLOGÍAS DE LA INFORMACIÓN E INDICADORES PARA CORREDORES.....	5
2.1. Indicadores para corredores.....	5
2.1.1. Paso	5
2.1.2. Ritmo cardíaco.....	6
2.1.3. Zonas de ritmo cardíaco	6
3. REDES SOCIALES	9
3.1. Definición de red social.....	9
3.2. Tipos de redes sociales	10
3.2.1. Redes sociales con enfoque a los deportes	10

3.3.	Aplicaciones móviles interconectadas con dispositivos.....	11
3.3.1.	Dispositivos para mediciones de corredores	11
4.	HERRAMIENTA PARA DESARROLLO DE APLICACIÓN	13
4.1.	Tipos de aplicaciones.....	13
4.1.1.	Nativas	13
4.1.2.	Nativas híbridas.....	13
4.1.3.	Generadas.....	14
4.2.	Decisión de herramienta de desarrollo.....	14
4.3.	Base de datos	14
4.3.1.	Modelo de datos.....	15
5.	INTERFAZ DE COMUNICACIÓN	17
5.1.	Interfaz de comunicación API.....	17
5.1.1.	Autenticación.....	17
5.1.2.	Interacción con el dispositivo.....	18
5.2.	Comunicación por <i>Bluetooth</i>	19
6.	METODOLOGÍA DE DESARROLLO	21
6.1.	Metodologías ágiles	21
6.1.1.	Modelo Espiral.....	22
6.1.2.	<i>Test driven developer</i>	22
7.	INTERFAZ DE USUARIO	23
7.1.	<i>Human Computer Interaction</i>	23
8.	DESARROLLO DE APLICATIVO	25
8.1.	Configuración de ambiente Android y Flutter	25
8.2.	Desarrollo inicial de servidor de aplicaciones.....	27

8.2.1.	Prototipo pantalla principal	27
8.2.2.	Modelo de datos	29
8.2.3.	Servidor de aplicaciones.....	29
8.2.3.1.	Modelo de datos y objetos	30
8.2.3.2.	Repositorio de datos.....	32
8.2.3.3.	Controlador	32
8.2.3.4.	Lógica de negocio.....	33
8.2.4.	Pruebas unitarias.....	34
8.3.	Desarrollo de pantallas de aplicación móvil.....	34
8.3.1.	Pantalla Inicial.....	35
8.3.2.	Pantalla de mapa.....	36
8.4.	Agregando funcionalidad de <i>login</i> y posicionamiento	37
8.4.1.	Manejo de usuario	38
8.4.1.1.	Pantalla de <i>login</i> de usuario monitor....	38
8.4.1.2.	Capa de presentación de manejo de usuario.....	40
8.4.1.3.	Capa de lógica de negocios de manejo de usuario	41
8.4.2.	Mapa de recorridos.....	42
8.4.2.1.	Pantalla de mapa de recorrido.....	43
8.4.2.2.	Pantalla de aplicación de lado del corredor.	44
8.5.	Agregando manejo de autenticación por correo	45
8.5.1.	Pasos de autenticación por correo	45
8.5.2.	Implementación de autenticación por correo	46
8.5.2.1.	Servicio de envío de correo desde <i>Spring boot</i>	47
8.5.2.2.	Servicio para envío de correo	47
8.5.2.3.	Recordar usuario	49

8.6. Próximas características a agregar	49
CONCLUSIONES.....	51
RECOMENDACIONES	53
BIBLIOGRAFÍA.....	55

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Modelo entidad relación inicial	16
2.	Estructura de petición Post para autenticación a Huami	18
3.	Instalación de Android SDK	25
4.	Configuración de Android SDK para admitir Flutter	26
5.	Prototipo de pantalla inicial	27
6.	Prototipo de mapa	28
7.	Modelo de datos.....	29
8.	Objetos del servidor de aplicaciones.....	30
9.	Clases del modelo.....	30
10.	Código de test programado.....	34
11.	Clases de dart para interfaz Flutter	35
12.	Pantalla inicial de aplicación desarrollada en Flutter.....	35
13.	Pantalla de persona monitor	36
14.	Pantalla de mapa de recorrido de corredor	37
15.	Pantalla de ingreso de usuario monitor	39
16.	Botón para salir de sesión.....	39
17.	Código reactivo para guardar un usuario	42
18.	Mapa de posicionamiento	43
19.	Agregar posición en <i>widget</i> de mapa	44
20.	Actualización de posición desde aplicación	45
21.	Autenticación por correo, servidores involucrados.....	46
22.	Configuración de <i>application.properties</i>	47
23.	Servicio de envío de correo.....	48

TABLAS

I.	Zonas de ritmo cardíaco.	7
II.	Respuesta de autenticación de servicio Huami	18
III.	Estructura para envío de correo.....	48

LISTA DE SÍMBOLOS

Símbolo	Significado
km	Kilómetro
m	Metro

GLOSARIO

Android	Sistema operativo móvil, desarrollado por Google, basado en el Kernel de Linux u otro software de código abierto.
API	Es un servicio que provee un aplicación para comunicarse de forma fácil.
Avatar	Figura que representa a un usuario en una aplicación o red social.
Background	Tipo de ejecución de aplicaciones en un sistema operativo móvil en el cual el usuario no percibe que se esté ejecutando aplicación alguna.
Big data	Son sistemas de datos capaces de almacenar y analizar grandes cantidades de datos de diferentes fuentes.
Bluetooth	Es una especificación para la conexión entre dispositivos, mediante un enlace de radiofrecuencia.
Dart	Es un lenguaje de programación diseñado para ser ejecutado del lado del cliente con características de lenguaje orientado a objetos.
Dashboard	Es un panel de indicadores clave para una organización.

<i>Debug</i>	Modo de ejecución de una aplicación en ambiente de desarrollo para detectar errores. En español es frecuentemente traducido a depuración.
Flutter	Es un conjunto de herramientas de interfaz de usuario, para el desarrollo de aplicaciones con características visuales estandarizadas para Android e iOS.
Fondo	Es un entrenamiento que consiste en correr a ritmo suave una distancia mayor a 5 km hasta unos 42 km según el objetivo del entrenamiento.
<i>Framework</i>	Es un conjunto de herramientas utilizadas para simplificar el desarrollo de aplicaciones.
Google Maps	Es un servicio desarrollado por Google con el cual se puede encontrar la ubicación exacta de ciudades, negocios o atracciones en internet
<i>Hardware</i>	Se le llama así a toda parte tangible de un dispositivo móvil, computadora o cualquier dispositivo electrónico que procesa datos.
<i>Host</i>	Es la dirección de un servidor web, puede ser un ip o un nombre de dominio.
Huami	Compañía China que desarrolla <i>smartwatches</i> para Xiaomi.

IntelliJ	Es una herramienta de desarrollo para varios lenguajes de programación, tales como java, C, Kotlin, entre otros.
iOS	Es un sistema operativo móvil desarrollado por Apple.
Java	Es un lenguaje de programación orientado a objetos, comercializado por primera vez en 1995.
<i>New Age</i>	Es una corriente filosófica que combina tradiciones orientales, principalmente.
<i>Machine learning</i>	Es el estudio de algoritmos computacionales que pueden mejorarse a sí mismo mediante la experiencia y la estadística de mucha información
<i>Open Source</i>	Es un término que es utilizado para identificar que un producto incluye permiso para usar el código fuente, documentos de diseño o contenido para futuros desarrollos.
Pilates	Es un método de entrenamiento físico y mental creado por Joseph Hubertus Pilates, uniendo el dinamismo y la fuerza muscular con el control mental.
REST	Es una forma de describir una interfaz HTTP y así obtener datos o mandar datos, más simplificado que los otros protocolos HTTP.

RFCOMM	Es un protocolo mediante transmisión por frecuencias.
Ritmo basal	Frecuencia cardiaca mínima con la que se puede mantener nuestras constantes vitales activas en estado de reposo.
Running	Es una disciplina en la cual se realiza carrera continua y se ha vuelto muy popular por los beneficios que aporta a la salud.
Sistema Operativo	Es el soporte lógico (<i>software</i>) principal que administra las partes físicas (<i>hardware</i>), proveyendo de servicios a los programas de aplicación que conforman el sistema en su totalidad.
Smartwatch	Viene de la combinación de las palabras en inglés: reloj e inteligente, es un dispositivo de pulsera sobrecargado de funcionalidades que un reloj tradicional no tiene.
Software	Es el soporte lógico para que la parte física (<i>hardware</i>) de un dispositivo haga la tarea para la cual fue diseñado.
Spring boot	Es un <i>framework</i> para el desarrollo de aplicaciones para la plataforma Java.
String	En programación es un tipo de dato en el cual es posible almacenar cualquier carácter o símbolo.

Taichí	Es un arte marcial, basado en meditación y la realización de ejercicios de forma lenta y suave.
VO2Max	Es la mayor cantidad de oxígeno que puede el cuerpo humano aprovechar durante un ejercicio de alta intensidad.
Token	Es un identificador que es devuelto por un servicio de autenticación.
Widget	Es una pequeña aplicación que generalmente provee información visual al usuario y este a la aplicación.
Yoga	Disciplina que busca el bienestar y el equilibrio físico, mental y espiritual.

RESUMEN

Este proyecto está dirigido a las personas que gustan de salir a correr, al igual que de la tecnología y al mismo tiempo a las personas que no les gusta salir a correr pero tienen a un adicto a las carreras en casa.

Lo que principalmente motivó a desarrollar esta aplicación fueron dos percances recientes sucedidos en carreras en Guatemala, uno tuvo lugar en Totonicapán el 1 de octubre de 2019 y el otro en la Ciudad de Guatemala, el 26 de enero de 2021. En ambos casos fueron jóvenes menores de 30 años, que lucían en buen estado de salud. Siempre al correr ese tipo de eventos, maratones o medias maratones, se tiene un sentimiento de angustia, aun sabiendo que se va con cierto nivel de entreno y los familiares se deben de preocupar incluso más.

Entonces se empezó a idear este proyecto, lo primero que se analiza son los parámetros de medición de ritmo cardiaco y la posición en cierto momento dado para calcular la velocidad, después se analiza las tecnologías existentes las cuales ayudan a la recopilación de los datos de entrada, ya teniendo esta información se discute la manera cómo se presentará al usuario, así por último se hace el diseño y desarrollo general de la aplicación.

Se espera que la aplicación sea útil para minimizar los riesgos en las carreras y adicionalmente tenga un crecimiento de componentes para ayudar a los usuarios en sus entrenamientos.

OBJETIVOS

General

Desarrollar una aplicación móvil que monitoree a una persona que realiza ejercicio, la cual puede determinar si esa persona en algún momento tiene problemas de salud o sobreesfuerzo mediante el análisis de los indicadores que son proporcionados por su reloj inteligente o banda de monitor cardiaco a la aplicación de la persona que está haciendo ejercicio, transferida a la aplicación de monitoreo y esta daría la señal de alerta. Además, crear un historial de usuario de forma que esa información sea útil para poder tener mejor precisión al predecir su capacidad y detectar el sobreesfuerzo mencionado al principio.

Específicos

1. Desarrollar una herramienta capaz de leer la interfaz de las aplicaciones más comunes utilizadas.
2. Desarrollar un módulo de autorizaciones de interfaces.
3. Desarrollar una interfaz adecuada que haga fácil para un usuario sin conocimientos previos de la disciplina de correr entender lo que los indicadores significan.
4. Capturar toda la información necesaria para dar un diagnóstico fundamentado.

5. Almacenar la información de los usuarios suscritos de forma adecuada, pudiendo hacer ingeniería de datos posteriormente.
6. Proporcionar una herramienta que pueda hacer estimaciones con la información almacenada.

INTRODUCCIÓN

En las últimas décadas ha aumentado la popularidad de salir a correr, así como de las carreras de media maratón o 10 km; ya que es una actividad excelente para la salud física y mental, ayuda a mejorar la circulación, a quemar grasa, mantener un peso ideal, aclara las ideas, baja el nivel de estrés, entre otros.

En muchas ocasiones las personas que se dedican a practicar deportes al aire libre se enfrentan a dificultades físicas o lesiones, generalmente se dejan llevar por su motivación intrínseca de ser un poco más veloz; en otras situaciones tales como eventos masivos, se tiene un cuadro más complejo, en donde siempre habrán personas alentando a desconocidos, cosa que me parece bien, pero en algunos momentos el corredor realmente no estaría en condiciones de continuar y esto los impulsa hasta arriesgar la vida. Una persona externa con la información adecuada podría evaluar el real estado y poder determinar si está arriesgando su integridad física. Esta persona podría también podría ser un familiar o un amigo además podría tener la tranquilidad de saber la localización.

En la mayoría de aplicaciones, la información de los resultados casi siempre están privados y únicamente accesibles para cada cuenta y si el dispositivo móvil es borrado esta información puede perderse. Las estadísticas previas de un corredor es información valiosa para los organizadores de una carrera, para poder saber el nivel de entrenamiento del corredor. Actualmente existen modelos matemáticos que predicen tiempos para diferentes distancias según el desempeño en entrenamientos o distancias más cortas, podríamos

mejorar estos modelos y con la información ir madurando y obteniendo mejores predicciones, así mismo estos resultados podrían predecir si el esfuerzo realizado es superior de las capacidades de un corredor y así mismo alertar.

Con el desarrollo de las aplicaciones móviles han surgido un sinnúmero de estas para impensables necesidades. Las más exitosas hacen uso de las funcionalidades de los dispositivos, tales como: sistema de posicionamiento global, la cámara, mensajería, entre otros; y casi todas están tendiendo a volverse o nacieron como una red social, donde los usuarios comparten e interactúan entre sí, compartiendo algún objetivo o característica en común.

Con este incremento de aplicaciones móviles han aparecido bastantes de estas con objetivos muy claros, uno de estos, el condicionamiento físico, ya que la mayoría están enfocadas en seguimiento de los entrenamientos, haciendo fácil para el usuario tener una bitácora de su evolución. Bastantes ya están orientadas a crear comunidades y así compartir experiencias y conocimientos.

Como lo comentado al inicio, podría haber personas interesadas en participar en algún tipo de red social que no necesariamente hagan cierta actividad, pero sí en las realizadas por algún amigo o familiar, puesto que no siempre se tiene la disponibilidad de comunicarse con alguien mientras se realiza un entrenamiento, esta participación sería como un observador o apoyo. Podría ser una función no participativa pero importante en el seguimiento, preparación e historial de entrenamiento y así mismo en circunstancias de emergencia poder reaccionar en menor tiempo.

1. BENEFICIOS DEL DEPORTE EN LA SALUD FÍSICA Y MENTAL

Desde que el ser humano alcanzó cierto grado de desarrollo se ha ido agravando el tema del sedentarismo, como muestra se puede mencionar que hace 100 años el hombre más obeso del mundo sería un hombre gordo promedio en la actualidad. Según la OMS (Organización mundial de la Salud), una cuarta parte de la población mundial, está en peligro de padecer enfermedades cardiovasculares; se estima que casi el 30 % de la población adulta no realiza la actividad física mínima para mantenerse saludable, el cual se recomienda según la OMS 150 minutos de ejercicio moderado o 75 minutos de ejercicio vigoroso por semana.

Según una tesis de investigación de un profesional de psicología¹, en la cual se concluyó que la aplicación de un programa de acondicionamiento físico de manera metódica, da como resultados la mejora en la autoestima, sentimiento de bienestar propio, responsabilidad, estabilidad emocional y autocontrol; lo cual desemboca en un incremento en el rendimiento laboral y académico.

¹ Incidencia del ejercicio físico sobre el estado de salud mental. Trabajo de graduación Universidad de San Carlos: 2005. [en línea]. http://biblioteca.usac.edu.gt/tesis/13/13_1307.pdf. p. 90. Consulta: 13 de marzo de 2020.

1.1. Tipos de ejercicios

Existen tres tipos de ejercicios: aeróbicos, de flexibilidad y de fuerza y resistencia. Los describiremos brevemente para entender sus principales beneficios.

1.1.1. Ejercicios aeróbicos

Este tipo de ejercicio involucra casi en su totalidad los músculos del cuerpo, generalmente aumenta el ritmo cardíaco, siendo posible realizarlo por períodos prolongados con intensidad baja y dependiendo del nivel de entrenamiento. Con cargas superiores puede causar aumento rápido en el ritmo cardíaco y si se trata mantener ese mismo ritmo puede causar un accidente cardiovascular o hasta muerte súbita.

1.1.2. Ejercicios de flexibilidad

Denominados también como ejercicios para amplitud de movimiento, su función principal es mantener la elasticidad de los músculos y el buen funcionamiento de las articulaciones. Deben ser gratos para los músculos involucrados, no causar dolor sino que únicamente estiramiento. Existen muchas disciplinas que se pueden practicar para obtener buenos resultados tales como: yoga, taichí, pilates, entre otros, muy popular entre adultos mayores, aunque con el auge del *new age* han aumentado la cantidad de jóvenes que buscan este tipo de ejercicios de relajación.

Generalmente no es necesario tener un control de ritmo cardíaco para este tipo de ejercicio; aunque si se usa para relajación después de hacer un

fondo se puede tener una métrica del nivel de relajación que puede aportar contra el ritmo basal.

1.1.3. Ejercicios de fuerza y resistencia

Son ejercicios realizados con barras y pesas, generalmente se realizan para aumentar la masa muscular. Pueden ser combinados para obtener mejores resultados en la práctica del *running*.

1.2. Riesgos en la práctica de deportes

Con el aumento de la práctica del *running* se ha visto la aparición de casos de muertes súbitas en Guatemala, se puede mencionar dos sucedidos recientemente, con similares características, el 26 de enero de 2020 y anteriormente el 1 de octubre de 2019, fueron en carreras de aproximadamente 21 km, en ambos casos fueron jóvenes entre 20 y 30 años y su condición física parecía adecuada para realizar la carrera². En otros países se han tenido muchos más casos de este tipo, por ejemplo, en España según el diario “20 Minutos” en el año 2016 se contabilizaron alrededor de 120 muertes súbitas, otro diario español “La Vanguardia” publica en una edición de abril del 2017 que según el Registro Nacional de Muerte Súbita en el Deporte³, se estima que en promedio mueren 150 a 200 deportistas aficionados o profesionales. Esto levanta las alarmas de lo que está pasando, según este mismo diario la forma como se practica este deporte ha cambiado, hace 15 años las personas corrían

² Joven deportista fallece tras correr la medio maratón Max Tott. [en línea]. Prensa Libre. <https://www.prensalibre.com/guatemala/sucesos/joven-deportista-fallece-tras-correr-la-medio-maraton-max-tott/>. Consulta: 15 de marzo de 2020

³ Dos corredores muertos: ¿Cómo evitar la muerte súbita? ¿Quién puede hacer sobreesfuerzos? [en línea]. Diario 20 Minutos. <https://www.20minutos.es/deportes/noticia/muerte-subita-maraton-reglas-practicar-deporte-2718616/0/>. Consulta: 18 de marzo de 2020

más por diversión o salud, en cambio ahora corren para ganar y bajar sus tiempos, exigiéndose a niveles peligrosos.

2. TECNOLOGÍAS DE LA INFORMACIÓN E INDICADORES PARA CORREDORES

El término más habitual utilizado en la actualidad es tecnología de la información y la comunicación, pero en este desarrollo se está más enfocado en la recolección de información, almacenamiento y análisis de la misma. Se podría mencionar que desde hace mucho tiempo se ha hecho registro del desempeño de algún deportista realizando alguna disciplina, para luego almacenarla, clasificarla y analizarla para hacer predicciones en muchas variables de interés. El ejemplo más claro es el béisbol.

2.1. Indicadores para corredores

En el mundo de los corredores se maneja un lenguaje muy particular, puesto que la mayoría están habituados a utilizar dispositivos de mediciones, para obtener datos tales como distancia, tiempo, paso, desniveles, ritmo cardíaco, algunos más especializados pueden obtener Vo2Max.

Los indicadores clave que son más importantes para determinar el desempeño de un corredor son los siguientes:

2.1.1. Paso

En algunas ocasiones es llamado *Pace* (del inglés que significa ritmo), es el tiempo que se toma para recorrer 1 km. Un paso de profesional sería cercano a los tres minutos, un corredor aficionado tendría con cierto nivel de

entrenamiento tiene un paso de cinco minutos hasta seis minutos y medio, esto para carreras de distancias mayores a 10 km.

2.1.2. Ritmo cardíaco

Es utilizado para determinar el nivel de esfuerzo requerido para realizar un recorrido a cierto paso, la percepción del esfuerzo a veces no es objetiva, pero teniendo mediciones del ritmo cardíaco se tiene una métrica más precisa. En algunas ocasiones el ritmo cardíaco es llamado *heart rate*.

2.1.3. Zonas de ritmo cardíaco

El ritmo cardíaco máximo que un individuo puede alcanzar sin sufrir daños durante el estrés del ejercicio generalmente decrece con los años. Se puede calcular mediante una prueba de esfuerzo, en esta prueba el individuo es sujeto a un estrés fisiológico controlado en una caminadora, mientras es monitoreado por una máquina de electrocardiograma. La intensidad del ejercicio es aumentada gradualmente hasta que ciertas funciones cardiacas son detectadas en el monitor, en ese momento el sujeto de prueba es detenido gradualmente.

Por cuestión de equipo y poco practicismo, el cálculo del ritmo cardíaco máximo es obtenido generalmente con algunas fórmulas. Aunque han sido criticadas porque generalizan y solo se enfocan en la edad. La más común (Haskell y Fox)⁴ dice que el ritmo cardíaco máximo es igual a 220 menos la edad del individuo ($F_{max} = 220 - \text{edad}$).

⁴ *Tools for Echocardiographic Calculations. Saric echonomy.*
<http://saric.us/echonomy/Maximum%20Heart%20Rate.htm>. Consulta: 16 de marzo de 2020.

Las zonas se determinan mediante el porcentaje de intensidad con respecto al ritmo cardíaco máximo, si se toma la descrita anteriormente tendríamos que a un 85 % de intensidad una persona debería tener un ritmo de 153.

Existe otro método en el cual se toma la frecuencia cardíaca basal (en reposo). Es llamada el método *Karvonen*.

$$F = (F_{\max} - F_b) * \% \text{ intensidad} + F_b.$$

Donde F_b = Frecuencia basal.

Las zonas de entrenamiento están divididas en número o colores de la siguiente forma y para esto se incluye otra variable, la intensidad:

Tabla I. **Zonas de ritmo cardíaco.**

Zona o color	Porcentaje Máximo	Tipo de paso
Zona 1, Gris	50 a 60 %	Caminata suave
Zona 2, Azul	60 a 70 %	Trote muy suave
Zona 3, Verde	70 a 80 %	Trote suave
Zona 4, Naranja	80 a 90 %	Trote normal, poco de esfuerzo
Zona 5, Roja	90 a 100 %	Trote fuerte, no se recomienda exceder más de 5 minutos con este paso

Fuente: elaboración propia, empleado LibreOffice 6.4.7.2.

3. REDES SOCIALES

Se tiene que hablar de este concepto por el gran impacto que se tiene en el desarrollo de aplicaciones móviles, pues se busca tener interacción de los usuarios entre sí. Primero definiremos exactamente qué es, y por qué la importancia de mencionarlo.

3.1. Definición de red social

Un servicio de red social (SNS, de sus siglas en inglés, *Social Networking Services*), es una plataforma integrada por usuarios que comparten intereses en común, sin importar su ubicación geográfica mediante el uso de internet. Adicionalmente se puede decir que están en constante cambio y se podría equiparar las sociedades físicas con las redes sociales.

Muchos no han formado parte o no recuerdan la era de los foros, en los cuales alguien creaba un foro con un tema específico, invitaba a sus amigos cercanos y estos a otros; algunos de estos foros se volvieron muy grandes, entonces necesitaron personas encargadas de moderar (administrar) cada sala del foro, otros en administrar la seguridad de los usuarios, otros en supervisar la armonía de los miembros y conforme el foro iba obteniendo más usuarios, hasta se iba diversificando el tema inicial del foro convirtiéndose en toda una sociedad virtual y por tanto era necesario más miembros con roles administrativos; podría mencionar el caso en Guatemala de “velocidad máxima”, este empezó como un foro con temática acerca de las personas aficionadas a los autos y sus modificaciones, el foro fue creciendo, por lo que empezaron a abrir salas para otros temas como computadoras, deportes, noticias, chats de temas libres,

secciones de negocios y hasta secciones para conseguir empleo; y esto conlleva la definición de nuevas reglas más específicas que las iniciales, desencadenando a un meticuloso control de las publicaciones, hasta llegando al punto de tener la capacidad de resolver conflictos entre usuarios; tal como pasa en una sociedad física.

3.2. Tipos de redes sociales

Según un artículo de la Universidad de Bournemouth (Reino Unido) las redes sociales están principalmente orientadas a las siguientes aplicaciones: negocios, aplicaciones médicas, investigaciones, bien social (es aquí donde podríamos ubicar una red orientada al deporte) y por último educación⁵.

3.2.1. Redes sociales con enfoque a los deportes

Su objetivo principal es poner en contacto a personas con los mismos gustos en función a la práctica de un deporte, pudiendo así crear conexiones con nuevas personas. Algunas redes sociales de este tipo se han especializado tanto, que pueden administrar contratación de espacios deportivos como canchas, equipos completos de algún deporte específico, organizar eventos, administración de roles diferentes como jugador, entrenador o dirigente, pudiendo hasta administrar un club profesional. Es un mundo nuevo, haciendo una pequeña investigación pude encontrar aplicaciones para los que gustan del fútbol, basquetbol, tenis, ciclismo, *running*, deportes extremos, entre otros.

⁵ *The influence of social media on health, business, education and society*. Department of Information Technology, Bournemouth University Bournemouth, United Kingdom. https://www.researchgate.net/publication/347442928_the_influence_of_social_media_on_health_business_education_and_society. Consulta: 19 de Octubre de 2021.

3.3. Aplicaciones móviles interconectadas con dispositivos

En todos los sistemas operativos para móviles existe un estándar de conexión llamado *Bluetooth*. Por medio del cual los desarrolladores de aplicaciones especializadas han podido extender las funcionalidades comunes de los dispositivos.

Las primeras versiones de *Bluetooth* fueron lanzadas a principios de los 2000 mucho antes que las de Android y iOS, aunque ambos sistemas operativos han incluido soporte para la interconexión de dispositivos *Bluetooth*. Sencillamente porque si no lo hicieran estarían dándole ventajas a su competidor, puesto que en la actualidad este tipo de conexiones se han vuelto un estándar.

Explicado de una manera más simple, un fabricante de algún dispositivo que realiza una tarea simple como un interruptor, por ejemplo. Siguiendo las especificaciones de la estandarización de *Bluetooth* podría hacer una aplicación móvil que podría permitir activar o desactivar dicho interruptor, desde la aplicación. Tal vez la necesidad de las conexiones de este tipo no parezca muy importante a simple vista, pero esto ha desencadenado en una revolución de la tecnología y en una tendencia que se ha denominado: *Internet of things*, la cual busca poder interconectar cualquier “cosa” cotidiana a la red para poder tener una automatización de todo, para ello en la mayoría de los casos se utilizan el protocolo de comunicación *Bluetooth*.

3.3.1. Dispositivos para mediciones de corredores

En la actualidad existen una gran cantidad de dispositivos para obtener los indicadores necesarios para ponerse los tenis y salir a correr al buen estilo del

siglo XXI, esto hablando de *smartwatches* o bandas pectorales, todos tienen en común que se comunican con el celular mediante el protocolo *Bluetooth*. Los principales fabricantes son: Garmin, Samsung y Huawei.

4. HERRAMIENTA PARA DESARROLLO DE APLICACIÓN

El mercado actual de dispositivos móviles está dividido en dos, podríamos mencionar que por un lado se tiene a Android y por otro a iOS, por cuestiones de recursos se limitará las pruebas de desarrollo al sistema operativo Android, aunque se desea hacer un desarrollo en alguna herramienta capaz de poder ser ejecutada en cualquier plataforma.

4.1. Tipos de aplicaciones

Primero se hizo un análisis de los tipos de aplicaciones, según lenguaje de desarrollo, el cual lo podemos dividir en tres grandes grupos.

4.1.1. Nativas

Significa que para el desarrollo de estas aplicaciones se utilizó el lenguaje nativo del sistema operativo, esto hace que la aplicación tenga un mejor rendimiento y su interfaz de usuario es óptima para el funcionamiento del sistema operativo, pero presenta la dificultad que la aplicación no es portable. Si se desea crear una aplicación para otro sistema operativo se tendría que desarrollar todo de nuevo. Para Android el desarrollo se haría en Java y Swift para iOS.

4.1.2. Nativas híbridas

Son aplicaciones desarrolladas en lenguajes interpretados por un contenedor nativo como Phonegap/Cordova, Ionic, jQuery mobile, React Native

y Flutter, estos brindan acceso a las funcionalidades del dispositivo, de una forma independiente del sistema operativo. Pudiendo ser esta opción la más apropiada para el objetivo planteado de tener una aplicación que pueda ser desplegada en múltiples plataformas, también se tiene la capacidad de poder crear una aplicación web. De estas se selecciona Flutter por ser el lenguaje con mayor popularidad del momento. Aunque no se descarta la utilización de algún lenguaje nativo por su facilidad de conectividad al estar más dependiente del sistema operativo.

4.1.3. Generadas

En este tipo de herramientas donde se usa un lenguaje específico, luego en el momento de generación o instalación se genera el código en la plataforma destino, para posteriormente ser compilado con las herramientas del sistema operativo destino. También denominadas, “falsas nativas”.

4.2. Decisión de herramienta de desarrollo

Como ya se expuso, se desea desarrollar en una herramienta que sea capaz de instalarse en múltiples plataformas y al mismo tiempo debe ser una herramienta actualizada, para facilitar mantenimientos posteriores, dadas estas características la primera opción a tomar en cuenta es Flutter, aunque no se descarta la utilización de Android SDK, por la facilidad de acceso de recursos del sistema operativo.

4.3. Base de datos

Los sistemas en la actualidad requieren mayor interacción de datos, capacidad que las bases de datos relacionales no poseen, por otro lado existe

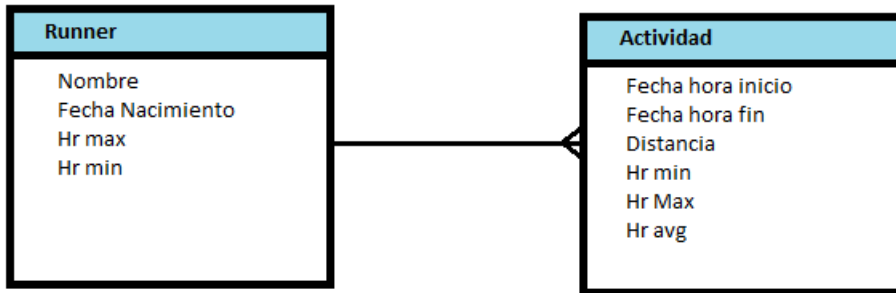
una alta exigencia con los tiempos de respuesta al momento de insertar información y obtener. Actualmente existe una tendencia sesgada hacia las bases de datos orientadas a documento, lo que significa que en lugar de guardar estructuras complejas en base de datos relacionales, los datos se guardan en una estructura de texto denominada BSON (arreglo dinámico de datos).

Por familiaridad de uso, se utilizará base de datos MongoDB, aunque no se descarta la utilización de otras bases de datos NoSQL en el mercado como Cassandra. Esta como todas las bases de datos basada en documentos tienen algunas limitaciones tales como: bloqueo a nivel de filas, si la información supera los 100 gb (según su documentación) tiene problemas de rendimiento, pero por la etapa inicial del proyecto no se considera que se necesite de ambas características.

4.3.1. Modelo de datos

Teniendo como punto de partida los datos que se desean almacenar, se puede tener una primera aproximación del modelo de datos para la aplicación. Se desea tener la información de un corredor, tales como un nombre, fecha nacimiento, frecuencia cardíaca mínima y máxima, esto para poder determinar de manera eficaz su zona de entrenamiento, adicionalmente se debe guardar información de la conexión a la interfaz de su dispositivo, esto se discutirá posteriormente; adicionalmente se quiere tener la disposición de los entrenamientos o carreras realizadas por el corredor, a cierto grado de detalle; y como otro dato que no se debe de escapar es la figura del usuario del monitor, esto nos lleva a tener también un control de que corredor puede monitorear.

Figura 1. **Modelo entidad relación inicial**



Fuente: elaboración Propia, empleado draw.io 14.9.6.

5. INTERFAZ DE COMUNICACIÓN

Las aplicaciones que se instalan con cada fabricante de dispositivo *smartwatch* o banda de monitoreo proveen una aplicación. En su mayoría por no decir su totalidad se comunican por *Bluetooth*. Adicionalmente proveen un API (Interfaz de programación de aplicaciones), la cual facilita la tarea de desarrollo de aplicaciones que deseen acceso a la información del dispositivo.

5.1. Interfaz de comunicación API

En esta primera fase de análisis teórico del proyecto, se hicieron pruebas con un dispositivo Huami, para la comunicación mediante servicios web de tipo REST.

Para tener acceso a estos servicios REST, las aplicaciones que hacen uso tienen que hacer un paso previo

5.1.1. Autenticación

Huami hace uso del protocolo de autenticación OAuth2.0, en la cual la aplicación necesita obtener un *token* de acceso cuando un usuario de Huami autoriza una aplicación a tener acceso a sus datos.

Para realizarlo la aplicación debe de hacer una petición Post al host: `auth.huami.com`, con la siguiente estructura que se muestra en la figura 2. También se debe tomar en cuenta que se debe incluir datos propios de la aplicación que se le va a dar acceso.

Figura 2. **Estructura de petición Post para autenticación a Huami**

```
POST /oauth2/access_token HTTP/1.1
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Host: auth.huami.com

client_id=clientId&client_secret=clientSecret&grant_type=authorization_code&redirect_uri=http%3
```

Fuente: Huamitech. *Petición Post para autenticación*, <https://github.com/huamitech/rest-api/wiki>.
Consulta: 21 de marzo de 2020.

Si la autenticación es exitosa el servicio devolverá un JSON (tipo de formato de archivo de uso libre) con la siguiente información.

Tabla II. **Respuesta de autenticación de servicio Huami**

Campo	Tipo	Descripción
Access_token	String	Token acceso
Token_type	String	Tipo de token
Expires_in	Number	Tiempo de vida en segundos
Refresh_token	String	Usado para renovar el token de acceso

Fuente: Huamitech. *Respuesta de autenticación*, <https://github.com/huamitech/rest-api/wiki>.
Consulta: 21 de marzo de 2020.

5.1.2. Interacción con el dispositivo

Ahora que ya se está autenticada la aplicación, se puede ir a traer información del usuario de Huami, haciendo uso del servicio REST de la siguiente forma <https://api-open.huami.com/users/#id/data>, donde *#id* es el

número de usuario del que se quiere obtener información y data es la información que se quiere obtener. Ejemplo: <https://api-open.huami.com/users/3037112216/profile>. Los datos que se pueden obtener son: *profile*, *activity_daily*, *activity_hourly*, *sleep_daily*, *sleep_hourly*, *motion* y *heartrate*.

5.2. Comunicación por *Bluetooth*

Otra posibilidad es hacer una conexión directa al dispositivo mediante *Bluetooth*. La plataforma Android de java incluye un API que realiza las tareas de gestión con la pila de Bluetooth del dispositivo.

El API de *Bluetooth* de Android puede hacer lo siguiente:

- Buscar dispositivos Bluetooth.
- Consultar la pila de *Bluetooth* del sistema operativo para buscar dispositivos sincronizados.
- Realizar comunicación mediante canales RFCOMM.
- Realizar apareamiento de dispositivos.
- Administrar varias conexiones.

Esta librería tiene configurada perfiles de uso para su implementación, dicho de otra forma, tiene plantillas de desarrollo para algunos usos comunes del Bluetooth, tales como: auriculares, perfiles de altavoz, dispositivos de salud.

Si se toma esta opción se tendría que evaluar hacer un desarrollo en la plataforma de Android y descartar el desarrollo en Flutter.

6. METODOLOGÍA DE DESARROLLO

Cuando hablamos de metodologías de desarrollo se refiere a una serie de reglas y patrones de trabajo para todas las fases del ciclo de vida del software, tales como: la toma de requerimientos, el desarrollo, implementación, pruebas o verificación e inclusive el mantenimiento, el cual puede ser iterativo, con todos los puntos anteriores puede ser tratado como nuevo requerimiento. Se pueden dividir en dos grandes categorías, metodologías ágiles y tradicionales. Las metodologías tradicionales hacen énfasis en la planificación de cada detalle que se debe tomar en cuenta en la implementación del requerimiento, con un especial cuidado en la documentación, centrado y dirigido en el control del proceso. Este tipo de metodología no se adapta fácilmente a los cambios y la prueba continua, aunque es muy útil en desarrollos de aplicaciones de gran tamaño o en administración de proyectos con equipos de trabajo numerosos. Por otro lado, tenemos las metodologías ágiles, las cuales están orientadas y enfocadas al desarrollo iterativo e incremental, pudiendo agregar una etapa de prueba en cada iteración.

6.1. Metodologías ágiles

Inicialmente se habló a grandes rasgos de los dos grandes grupos de metodologías de desarrollo. Tomando en cuenta que el desarrollo de este proyecto debe de ser un desarrollo dinámico e incremental, se toma la decisión de tomar una metodología ágil y que sea especialmente basada en pruebas.

6.1.1. Modelo Espiral

Este modelo de desarrollo es un poco, aunque por eso no deja de ser efectivo para lograr los objetivos planteados al momento de afrontar el desarrollo de una aplicación.

Las principales directivas de esta metodología es dividir el desarrollo en las secciones más pequeñas, para minimizar el riesgo en cada ciclo de desarrollo. Las fases de cada iteración son: determinar objetivos, identificar y resolver riesgos, desarrollar los objetivos y por último probar y planear la siguiente iteración. Cada iteración se inicia con la identificación de los requerimientos y establecer claramente que se desea hacer y termina con la verificación de lo desarrollado y la adición de las nuevas características.

6.1.2. *Test driven developer*

Para poder tener una metodología actualizada, se puede hacer una combinación de otra metodología de desarrollo bastante más actualizada. La cual está basada en los *frameworks* de java modernos. Se puede integrar en la primera fase del modelo espiral una fase de diseño de prueba programada, para hacer de forma programática la última fase del ciclo y no tener la complejidad de pruebas por usuario. Esto todavía está en una fase de descubrimiento, se puede tener cambios en la fase de desarrollo.

7. INTERFAZ DE USUARIO

Ya habiendo definido lo que en el capítulo dos se detalló acerca de zonas de entrenamiento, se puede establecer una forma de presentar la información, al usuario, necesaria para entender el estado del corredor. Independientemente del conocimiento del lenguaje, el de los parámetros normales de los indicadores típicos de un corredor o de la utilización de aplicaciones móviles; se desea diseñar una configuración de presentación de la información entendible (definición de interfaz de usuario). Existen algunas sugerencias de los desarrolladores de *dashboards* para la interfaz de usuario, que podríamos tomar en cuenta.

Leyendo un artículo que habla acerca de la presentación de las métricas, se puede establecer algunas premisas: no se puede mejorar lo que no se puede medir, la información debe ocupar poco espacio, pero esto no significa comprimirlo todo, acceso instantáneo en los datos importantes, un dato debe de estar siempre acompañado de la información histórica con una correcta técnica de segmentación y lo más importante entender la diferencia entre datos cualitativos y los cuantitativos.

7.1. *Human Computer Interaction*

Es una disciplina de la computación que se podría clasificar como metodología, relacionada con el diseño, evaluación, desarrollo y estudio de los fenómenos de los sistemas informáticos para uso humano. El objetivo de esta disciplina es crear sistemas: usables seguros y funcionales.

Dadas estas premisas se tendría que hacer un estudio de los usuarios de aplicaciones orientadas al mundo del *running* para poder diseñar la interfaz de usuario de manera acertada.

8. DESARROLLO DE APLICATIVO

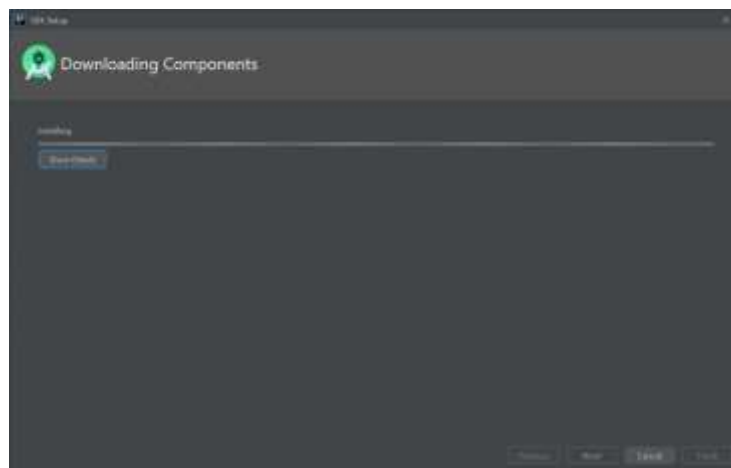
Antes de entrar en materia con la aplicación, se debe instalar el ambiente de desarrollo adecuado con todas las herramientas a utilizar.

8.1. Configuración de ambiente Android y Flutter

Para poder utilizar dichas herramientas de desarrollo móvil, se debe configurar IntelliJ para que sea capaz de utilizar el SDK de Android, Flutter y Dart.

Android, es una plataforma de desarrollo de aplicaciones móviles, desarrollada por Google, y en la cual se basa Flutter para funcionar.

Figura 3. **Instalación de Android SDK**

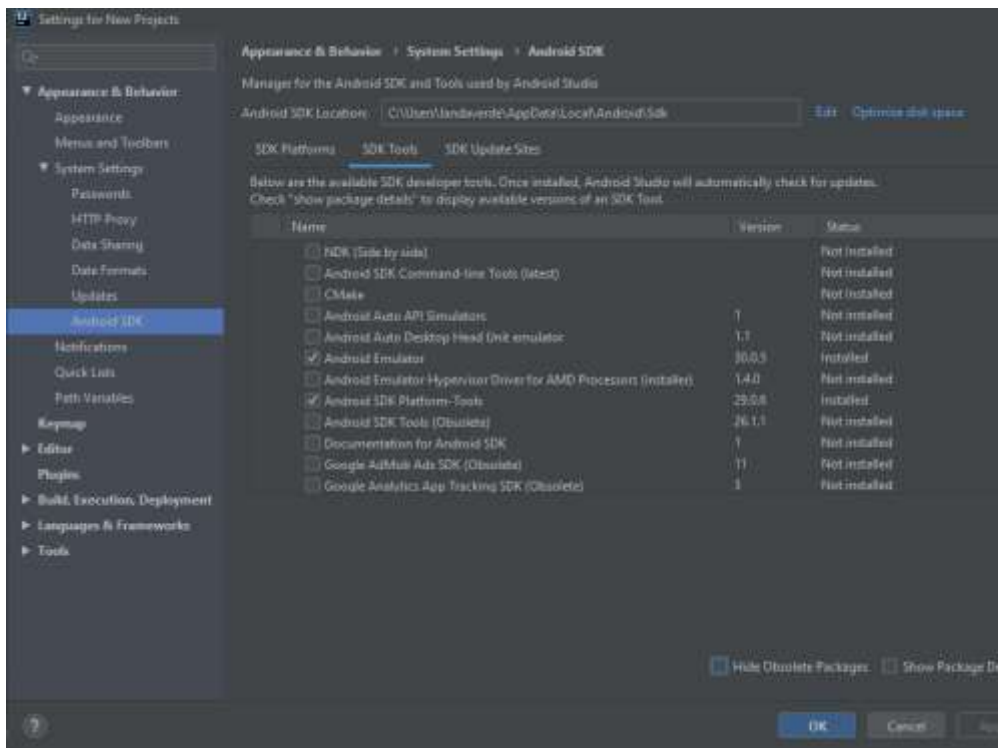


Fuente: elaboración propia, empleado IntelliJ 2020.3.

En el IDE de desarrollo IntelliJ se debe buscar en las herramientas la palabra Android y así se empieza a descargar el SDK.

Como siguiente paso se debe de configurar Android para que admita Flutter, para esto se debe de habilitar las herramientas obsoletas de Android, como se muestra en la siguiente figura.

Figura 4. Configuración de Android SDK para admitir Flutter



Fuente: elaboración propia, empleado IntelliJ 2020.3.

Como último paso se debe descargar el Flutter desde la página oficial y configurar las variables de ambiente, esto puede variar según el sistema operativo.

Como paso adicional se puede agregar un dispositivo virtual en el IDE de desarrollo o configurar un móvil con la opción de desarrollo y habilitar el *debug* y la instalación por usb.

8.2. Desarrollo inicial de servidor de aplicaciones

Para empezar con la metodología de desarrollo en espiral, se inicia con un prototipo de pantallas, en las cuales se describe la funcionalidad general de la aplicación. Se evitará inicialmente la autenticación del usuario de monitoreo y solo se tendrá los datos generales de un corredor.

8.2.1. Prototipo pantalla principal

Como ya se expuso en el desarrollo de la usabilidad del tablero de indicadores (*dashboard*), se busca no sobrecargar el campo visual del usuario con exceso de información, y sin embargo poder darle la información necesaria.

Figura 5. Prototipo de pantalla inicial



Fuente: elaboración propia, empleado captura de pantalla Android 9.

Como se puede observar en la figura, se tiene la información principal necesaria, añadiendo algunos datos que parecen escaparse a primera vista: el nombre del corredor, la imagen avatar del corredor, el ritmo cardiaco en tiempo real, con un color diferente al del nombre, ritmo cardiaco mínimo o en reposo y el ritmo cardiaco máximo.

Con el ritmo cardiaco en tiempo real se muestra un color diferente pues es el color que se definió cuando se conversó acerca de las zonas de ejercicio, zona 1, gris; zona 2, azul; zona 3, verde; zona 4, naranja y zona 5 rojo, con una separación de 10 % entre cada zona. En la parte de abajo se muestra el ritmo cardiaco mínimo y el máximo, puesto que es el marco de referencia para hacer los cálculos de las zonas de entrenamiento (porcentajes)

Y por último se tiene un ícono que nos lleva al mapa de ubicación para poder saber la ubicación de la persona, como se muestra en la figura seis.

Figura 6. **Prototipo de mapa**

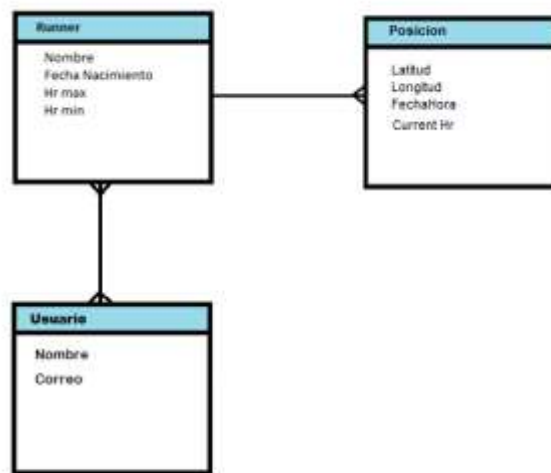


Fuente: elaboración propia, empleado captura de pantalla Android 9.

8.2.2. Modelo de datos

Para poder tener esta información disponible se debe contar con cierta estructura de datos, la cual ya se discutió que se utilizará MongoDB como base de datos.

Figura 7. Modelo de datos

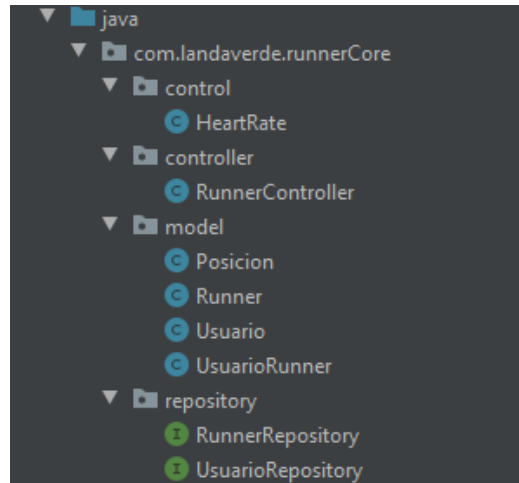


Fuente: elaboración propia, empleado draw.io 14.9.6.

8.2.3. Servidor de aplicaciones

Inicialmente no se había discutido este tema, aunque en el desarrollo de esta primera fase se encontró la necesidad de incluir un servidor de aplicación, puesto que los patrones de diseño utilizados por lenguajes de programación para dispositivos móviles están orientados a separar la capa de presentación, de la capa de datos y lógica de negocio Para eso se utilizó, *Spring Boot Framework*.

Figura 8. **Objetos del servidor de aplicaciones**



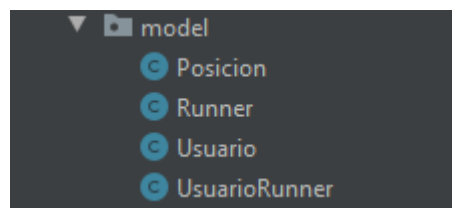
Fuente: elaboración propia, empleado IntelliJ 2020.3.

Spring, es un *framework* de desarrollo liviano y versátil, con posibilidad de conexión a una gran variedad de tecnologías.

8.2.3.1. **Modelo de datos y objetos**

Como buena práctica se recomienda crear un objeto entidad o modelo por cada tabla de la base de datos,

Figura 9. **Clases del modelo**



Fuente: elaboración propia, empleado IntelliJ 2020.3.

Para demostrar el desarrollo de las clases de entidad, se tomará de ejemplo el modelo de *runner*.

Los campos del modelo son: *id*, *nombre*, *minHr*, *maxHr*, *currHr*, *zone*, *currLong*, *currLat*.

- *id*: es el identificador único que la base de datos tipo documento le asigna a la entidad.
- *nombre*: es el nombre del corredor, o el cual se le quiere asignar para la aplicación.
- *minHr*: es el ritmo cardíaco registrado del corredor, al momento de crearlo este valor debe de ser igual a un valor configurado, que podría ser 60.
- *maxHr*: es el ritmo cardíaco máximo del corredor, al momento de crearlo puede ser calculado según su edad, esta función podría configurarse para poder ser modificado dinámicamente.
- *currHr*: este valor es obtenido por el API de comunicación y actualizado en tiempo real en esta tabla para poder ser transmitido a la pantalla principal de la aplicación.
- *zone*: es la ubicación del porcentaje de carga de ejercicio según el ritmo cardíaco, esto es calculado en tiempo real al obtener una lectura desde el API.
- *currLong*: es la longitud de la localización global del corredor obtenida desde el API, y actualizada en esta tabla para poder ser transmitida a la aplicación.
- *currLat*: es la latitud de la localización global del corredor.

8.2.3.2. Repositorio de datos

Para el servidor de aplicaciones existe un grupo de componentes encargados de manejar el acceso, modificación y eliminación de los datos. Para *Spring Boot Framework* existe una librería encargada de realizar esto, su nombre es *Reactive Repository*.

Reactive repository, es un herramienta que es capaz de hacer consultas desde una base de datos y devolver los datos de forma asíncrona, para poder cumplir con los requerimientos de respuesta de las herramientas actuales de desarrollo.

Las respuestas de los datos son empaquetados en objetos especiales de tipo flujo (*stream*) de datos, que como se ha mencionado es capaz de manejar datos asíncronos y proporcionar la característica de publicar datos conforme estos sean modificados.

8.2.3.3. Controlador

El servidor de aplicaciones debe de tener la forma de publicar externamente o permitir modificaciones, este modelo de arquitectura también es conocido como “microservicio”, donde cada acción es publicada mediante un servicio web.

Método *get*, este se utiliza para publicar los datos del servidor de aplicaciones, en el caso específico de la aplicación, existen dos servicios que utilizan el método *get* para el modelo de la tabla corredor.

runner: obtiene todos los corredores almacenados en la base de datos, en posteriores iteraciones se debe filtrar por el usuario de la aplicación.

runners/{id}: obtiene un corredor específico, adicionalmente cuando está obteniendo el usuario, hace el llamado de un método de la lógica de negocio en el objeto *HeartRate.zone*, que describiremos posteriormente.

Método *put*, este método es llamado después de obtener los datos desde el API de comunicación de Huami, pues actualiza los datos de modelo corredor para que este sea leído por la aplicación móvil.

Método *post*, este es llamado para insertar un nuevo registro en la base de datos, en el caso específico del corredor es llamado en el mantenimiento de usuario cuando agrega un nuevo corredor al que desea monitorear.

8.2.3.4. Lógica de negocio

En la lógica de negocio del servidor de aplicación por lo general se coloca aquellas funciones que transforman o validan información importante para la lógica de la aplicación.

Cálculo de zona de ejercicio según el ritmo cardíaco actual y los ritmos mínimos y máximos son realizados en este paquete, en la clase *HeratRateZone*.

$$\text{Porcentaje} = (\text{ritmo actual} - \text{ritmo min}) / (\text{ritmo max} - \text{ritmo min})$$

8.2.4. Pruebas unitarias

Como parte del primer *test* programado del servidor de aplicaciones, se realizó sobre el método para calcular la zona, dado un ritmo cardíaco actual, uno mínimo y máximo.

Figura 10. Código de *test* programado

```
@Test
public void testHeartRateZone() {
    Integer zone =HeartRate.zone( hrMin: 50, hrMax: 200, hr: 187);
    Integer value=5;
    Assertions.assertEquals(zone, value);
}
```

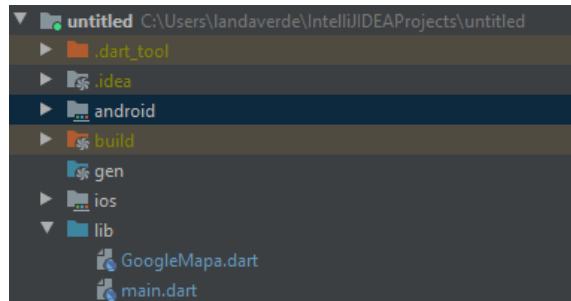
Fuente: elaboración propia, empleado IntelliJ 2020.3.

8.3. Desarrollo de pantallas de aplicación móvil

Como se ha estado mencionando la herramienta para el desarrollo de la aplicación móvil, es Flutter, la cual da la facilidad de poder desplegar en los dos principales sistemas operativos sin tener que hacer ninguna modificación, esto se logra mediante el SDK de Dart.

Dart es un lenguaje para aplicaciones web y móviles, que es utilizado por Flutter. Inicialmente fue desarrollado únicamente para web y como un complemento de Google Chrome y otros navegadores desarrollados por Google.

Figura 11. **Clases de Dart para interfaz Flutter**



Fuente: elaboración propia, empleado IntelliJ 2020.3.

8.3.1. **Pantalla inicial**

La pantalla inicial en esta iteración de desarrollo ya tiene bastante funcionalidad como se describió en los prototipos.

Figura 12. **Pantalla inicial de aplicación desarrollada en Flutter**



Fuente: elaboración propia, empleado captura de pantalla Android 9

El API de acceso a dispositivo Huami en el servidor de aplicación lee la información proveniente del reloj con el monitor cardiaco.

Figura 13. **Pantalla de persona monitor**



Fuente: elaboración propia, empleado captura de pantalla Android 9.

El proyecto de Flutter consta de dos archivos: *main.dart* y *googleMap.dart*, en el primero se encuentra la pantalla principal de la aplicación, la cual solo recibe la información desde el servidor de aplicaciones mediante API web, por peticiones *get*.

8.3.2. Pantalla de mapa

Al presionar el botón del mapa la aplicación navega hacia la otra ventana y ahí se mostrará el mapa, y mediante el API de información del corredor se obtiene su posición en longitud y latitud; dibujándola en el mapa y

actualizándose automáticamente con los datos obtenidos del servidor de aplicaciones y los cuales son leídos desde el *Smartwatch* en tiempo real.

Figura 14. **Pantalla de mapa de recorrido de corredor**



Fuente: elaboración propia, empleado captura de pantalla Android 9.

8.4. Agregando funcionalidad de *login* y posicionamiento

En esta iteración se busca agregar funcionalidad ya existe a la aplicación, una de las principales es agregarle una capa de usuario, no tan enfocado a seguridad sino que más hacia usabilidad, puesto que la información que se comparte no es tan sensible como las que maneja otro tipo aplicaciones; Otra

también es mostrar recorridos del corredor a través de un mapa y como última funcionalidad la posibilidad de poder manejar varios corredores a la vez en la pantalla de monitor.

8.4.1. Manejo de usuario

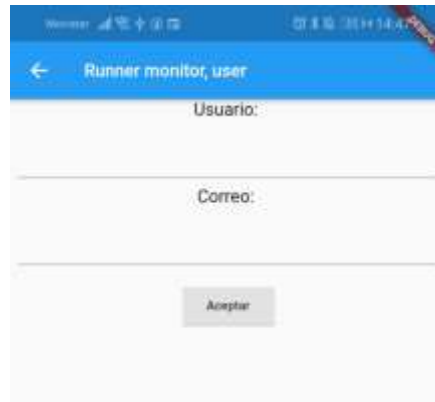
En las primeras fases de desarrollo se utilizó manejo simple de usuario sin ser necesaria la autenticación. La cual se agrega en fases posteriores.

8.4.1.1. Pantalla de *login* de usuario monitor

Como ya se dijo anteriormente en esta fase no se enfocó en la seguridad típica de una aplicación móvil, sino facilidad de uso. Se está considerando no utilizar contraseña para la autenticación, para hacerla simple para el usuario.

Lo más fácil es dejar registrarse con un nombre y un correo electrónico, el cual solo es utilizado para no duplicar persona en la base de datos. Y una vez registrada la persona en el dispositivo donde se encuentra instalada la aplicación yo no será necesario volver a ingresar esos datos, a menos que la persona decida salirse.

Figura 15. **Pantalla de ingreso de usuario monitor**



Fuente: elaboración propia, empleado captura de pantalla Android 9.

Estando autenticado el usuario puede en cualquier momento salirse de la aplicación, puesto se agregó un botón para realizar esa acción en la pantalla principal.

Figura 16. **Botón para salir de sesión**



Fuente: elaboración propia, empleado captura de pantalla Android 9.

8.4.1.2. Capa de presentación de manejo de usuario

Para realizar esta funcionalidad se requirió agregar una pantalla en Flutter, así como lógica de negocio que se discutirá posteriormente.

En Flutter cada nueva pantalla debe de ser representada por un nuevo *widget*, en este caso fue nombrado como “*RunnerUser*”, el cual es encargado de presentar al usuario una pantalla donde puede ingresar sus datos, para que posteriormente la capa de lógica de negocio lo valide y por último estos sean almacenados en la base de datos.

En esta pantalla se puede observar dos campos y un botón, los campos son de tipo “*TextField*” y el botón es de tipo “*RaisedButton*”. Cuando el botón es presionado ejecuta un método de nombre `_save` (como dato importante del desarrollo en Flutter es que los componentes con nombres que se les antepone un guión bajo “_” son de tipo privados, esto quiere decir que no pueden ser accesibles externamente por otro *widget* o clase).

La funcionalidad del método de guardar es hacer uso de una librería propia de Android de nombre *SharedPreferences*, en la cual el usuario puede almacenar una gran cantidad de información en el dispositivo móvil para luego ser consulta sin importar que se haya salido de la aplicación y haber vuelto a ingresar de forma persistente.

8.4.1.3. Capa de lógica de negocios de manejo de usuario

En la capa de lógica de negocios se tomó la consideración de no permitir ingresar dos usuarios con el mismo correo, a pesar que la llave primaria de la tabla usuario es "id" esto para que una persona si en algún momento borra su información de la aplicación pueda volver a ingresar y poder ver la información que ya se encontraba ingresada.

El modelo de datos para los usuarios, ya se había definido en la figura 7, en la iteración anterior. En la cual se puede observar que para un usuario se va a guardar, nombre y correo.

En la lógica encargada de decidir si actualizar un registro existente o crear uno nuevo, se tiene el siguiente algoritmo:

- Paso 1: se busca en la base de datos con el correo ingresado.
- Paso 2: Si se encuentra algún registro se actualiza el nombre de dicho registro con el nombre ingresado. Si no se encuentra algún registro se marca como un registro nuevo.
- Paso 3: Se guarda el registro sin importar si es uno nuevo o una modificación.

Esto se hace mediante programación reactiva, se incluye el código de esta acción para ilustrar cómo esto se realiza.

Figura 17. **Código reactivo para guardar un usuario**

```
64 @PostMapping("@users")
65 @
66 public Mono<User> create(@RequestBody User user){
67     Mono<User> usuarioMono= userRepository.findByUser(user.getUser());
68     usuarioMono =usuarioMono.flatMap(
69         u ->{
70             User u1=new User();
71             u.setUser(null);
72             u.setUser(u.getUser()+" already exists");
73             return Mono.just(u1);
74         }
75     ).switchIfEmpty( userRepository.save(user) );
76     return usuarioMono;
77 }
```

Fuente: elaboración propia, empleado IntelliJ 2020.3.

8.4.2. Mapa de recorridos

Para ayudar al usuario de monitor se agregó una marca con la cual se puede determinar la posición del corredor.

Figura 18. **Mapa de posicionamiento**



Fuente: elaboración propia, empleado captura de pantalla Android 9.

8.4.2.1. Pantalla de mapa de recorrido

Para realizar esto en el código de Flutter se debe agregar una marca en el componente (*widget*) de Google Maps, y esta marca a la vez contiene la ubicación en coordenadas, latitud y longitud de la posición que se quiere marcar.

Figura 19. **Agregar posición en *widget* de mapa**

```
    SizedBox(  
      width: MediaQuery.of(context).size.width,  
      height: (MediaQuery.of(context).size.height - 140) /  
        (_showpanel ? 2 : 1),  
      child: GoogleMap(  
        mapType: MapType.normal,  
        tiltGesturesEnabled: false,  
        initialCameraPosition: _initialPosition,  
        polylines: _polyLines,  
        markers: _markers,  
        onMapCreated: onMapCreated)), // GoogleMap, SizedBox
```

Fuente: elaboración propia, empleado IntelliJ 2020.3.

Ya teniendo esta variable de nombre `_markers`, se puede asignar y actualizar periódicamente con las coordenadas que se desean. Esto se logra mediante un servicio REST el cual se suscribe la aplicación y obtiene la posición, siendo la misma agregada a la variable `_markers`.

8.4.2.2. Pantalla de aplicación de lado del corredor

Leyendo desde el api que provee Huami se tuvo ciertos problemas para obtener dicha información por eso se optó por hacer una pequeña aplicación que debe de estar ejecutándose en *background*. Esto se podrá mejorar en posteriores versiones de la aplicación.

Flutter posee una gran cantidad de librerías que se pueden agregar para darle mayor funcionalidad, en este caso se utilizó una de nombre *geolocator*.

Figura 20. **Actualización de posición desde aplicación**

```
position = await _geolocator.getCurrentPosition(  
    desiredAccuracy: LocationAccuracy.bestForNavigation);
```

Fuente: elaboración propia, empleado IntelliJ 202.3.

8.5. Agregando manejo de autenticación por correo

Consiste en un conjunto de acciones que se toman para poder garantizar la identidad del usuario, para ello, las aplicaciones por lo general le solicitan al usuario un correo electrónico, al cual se le mandará un link de validación de cuenta que solo puede ser abierto desde la cuenta de correo.

8.5.1. Pasos de autenticación por correo

La creación del usuario con autenticación por correo está regida por una sucesión de pasos que se describen a continuación:

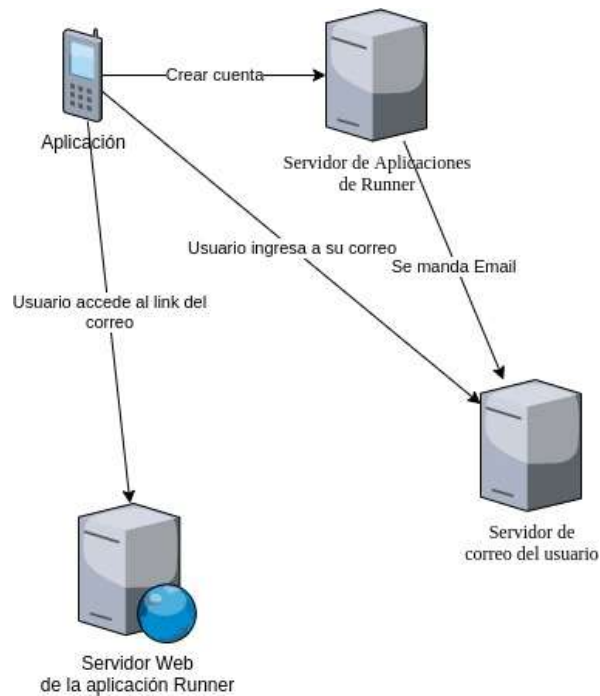
- Creación de cuenta: en este paso el usuario ingresa un nombre de usuario y una cuenta de correo electrónico.
- La aplicación manda un correo electrónico de bienvenida y de verificación de cuenta.
- El usuario ingresa a su correo y accede mediante un enlace a una página web que registra que el correo fue validado para la cuenta asociada.
- El usuario puede ingresar a la aplicación ya que su cuenta fue verificada. Si la acción anterior no es realizada el usuario no puede ingresar a la

aplicación y esta deberá de mandar un mensaje de error donde explique que su cuenta no ha sido verificada.

8.5.2. Implementación de autenticación por correo

La primera acción es configurar el envío de correo mediante *Spring boot*. En la siguiente imagen se muestra la interacción de la aplicación en Android con el servidor de aplicaciones.

Figura 21. **Autenticación por correo, servidores involucrados**



Fuente: elaboración propia, empleado draw.io 14.9.6.

8.5.2.1. Servicio de envío de correo desde *Spring boot*

Lo primero se debe configurar el envío de correo desde el servidor de aplicaciones, para esto se agregó al proyecto la *librería spring-boot-mail-starter*, posteriormente se debe configurar la cuenta remitente, para esto se creó una cuenta empresarial en Gmail, con nombre runner.core.gt@gmail.com, se agrega esta configuración al archivo de configuración de *Spring boot*.

Figura 22. Configuración de *application.properties*

```
3 spring.mail.host=smtp.gmail.com
4 spring.mail.port=587
5 spring.mail.username=runner.core.gt@gmail.com
6 spring.mail.password=
7 spring.mail.properties.mail.smtp.auth=true
8 spring.mail.properties.mail.smtp.starttls.enable=true
```

Fuente: elaboración propia, empleado IntelliJ 2020.3.

Se crea una clase para el envío de correo y se anota como componente, esta clase lee la configuración descrita en la imagen para crear un servicio de envío de correo. Teniendo dicho servicio ya se puede mandar tener un servicio REST para mandar correos.

8.5.2.2. Servicio para envío de correo

Para poder enviar correos se creó un servicio REST adicional a los de manejo de corredor o monitor el cual se le agrega una estructura para poder adjuntar los datos para realizar el envío de correo.

Figura 23. Servicio de envío de correo

```

@PostMapping("/sendVerification")
public void sendVerification(@RequestBody UserEmailData verification) {
    String host = "\r\n";
    String text=String.format(verification.getTemplate(), verification.getUser());
    if(verification.getSrvHost() != null)
        host = "\r\nhttp://"+verification.getSrvHost()+":4268/verification?id="+verification.getUserId();
    text += host;
    emailService.sendSimpleMessage(verification.getTo(), verification.getSubject(), text);
}

```

Fuente: elaboración propia, empleado IntelliJ 2020.3.

Como se puede ver en la imagen el método que realiza el envío de correo espera una estructura que es puesta en el cuerpo de la petición. Dicha estructura es de la siguiente forma:

Tabla III. Estructura para envío de correo

Campo	Tipo	Observación
To	String	Contiene el email a quien se dirige el correo
Subject	String	El asunto del correo
Template	String	Posee un texto base con variables que son sustituidas por valores específicos
User	String	En el nombre del usuario y siendo uno de los campos cambiados en la plantilla (template)
UserId	String	El id interno de la base mongoDB del usuario
srvHost	String	El ip de donde se encuentra desplegado el servidor de aplicaciones

Fuente: elaboración propia, empleado LibreOffice 6.4.7.2.

En la parte *template* (plantilla) se ingresa un texto base en el cual se indican variables que serán sustituidas por valores específicos, tales como nombre y usuarios, en el caso del correo que se está enviando al usuario.

Esta plantilla es guardada en una entidad de la aplicación de nombre *EmailMessage*, que contiene un id, código de evento, lenguaje y mensaje, por el momento se utiliza idioma español como predeterminado.

8.5.2.3. Recordar usuario

Para realizar la acción de recordar usuario se procede a preguntar el correo y rol desde la aplicación, para luego mandar un correo con el nombre del usuario asociado. Para esto la aplicación manda el correo y de forma predeterminada el idioma ES (español) con el cual se va a traer el mensaje predeterminado en la tabla, para finalmente mandar un correo con el mismo servicio descrito en el capítulo anterior.

8.6. Próximas características a agregar

El desarrollo de aplicaciones siempre requiere una constante actualización, por tal motivo se hace un análisis de los siguientes pasos que podrían ser tomados para las siguientes actualizaciones:

- Agregar autenticación con contraseña cifrada.
- Agregar seguridad a los servicios API del servidor de aplicaciones.
- Agregar interfaces de conexión a dispositivos como *Smartwatches* o bandas.
- Agregar un servicio de mensajería entre el usuario y el monitor.

Especialmente con el caso de los dispositivos, no se puede hacer pruebas con todos los dispositivos existentes, pues el mercado es muy grande, la solución a este problema puede ser solventado al dar el código y así definir este proyecto como uno de licencia de código abierto donde cualquier desarrollador puede colaborar en agregar funcionalidades a este proyecto.

CONCLUSIONES

1. El desarrollo de aplicaciones móviles con comunicación mediante interfaces de comunicación o protocolos de comunicación, necesitan cierto grado de granularidad para su diseño e implementación, para determinar eficazmente el esfuerzo real de desarrollo.
2. Las interfaces de comunicaciones generalmente están orientados a servicios de comunicaciones tales como servicios REST.
3. Los fabricantes en la actualidad se encuentran bastante anuentes en el desarrollo código abierto y entienden la importancia de la cooperación.
4. La utilización de patrones de desarrollo de interfaces ayuda al usuario final a entender lo que la aplicación quiere expresar.
5. Inicialmente para este desarrollo solo será necesario determinar la edad del usuario corredor y la información adicional será determinada por la aplicación.
6. En la actualidad existe una gran tendencia hacia las bases de datos no relacionales para el almacenamiento rápido y dinámicamente estructurado

RECOMENDACIONES

1. Evaluar en la implementación de este tipo de aplicaciones las diferentes opciones en la herramienta a utilizar, para la lectura de indicadores básicos de estado de salud, pues se debe de considerar las limitaciones que se tienen con respecto a la accesibilidad de la información.
2. Construir mediante el uso de *machine learning* y *big data* un modelo matemático predictivo para anticiparse a cualquier eventualidad en la práctica del deporte.
3. Desarrollar una alternativa con *hardware* para realizar las lecturas de indicadores básicos de salud, para no depender de terceros.
4. Diseñar aplicaciones amigables al usuario y con estándares orientados al usuario objetivo.

BIBLIOGRAFÍA

1. Castelló, Roberto. *Claves para diseñar un buen Dashboard*. [en línea]. <<https://www.legridd.com/claves-para-disenar-un-buen-dashboard/>>. [Consulta: 21 de marzo de 2020].
2. Developers Android. *Introducción general a Bluetooth*. [en línea]. <<https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es-419>>. [Consulta: 25 de marzo de 2020].
3. Department of Information Technology, Bournemouth University Bournemouth, United Kingdom. The influence of social media on health, business, education and society. [en línea]. https://www.researchgate.net/publication/347442928_the_influence_of_social_media_on_health_business_education_and_society. [Consulta: 19 de octubre de 2021]
4. HuamiTech. *Huami Web Api Documents for Developers*. [en línea]. <<https://github.com/huamitech/rest-api/wiki>>. [Consulta: 21 de marzo de 2020].
5. InnovaAge. *Apps híbridas vs Generadas. ¿Qué decisión tomar?* [en línea]. <<https://www.innovaportal.com/innovaportal/v/696/1/innova.front/apps-hibridas-vs-nativas-vs-generadas-que-decision-tomar>>. [Consulta: 20 de marzo de 2020].

6. Mateo, Daniel. *Dos corredores muertos: ¿Cómo evitar la muerte súbita? ¿Quién puede hacer sobreesfuerzos?* [en línea]. <<https://www.20minutos.es/deportes/noticia/muerte-subita-maraton-reglas-practicar-deporte-2718616/0/>>. [Consulta: 18 de marzo de 2020].
7. Molins Renter, Albert. El aviso de un cardiólogo sobre el riesgo de correr sin control. [en línea]. <<https://www.lavanguardia.com/vida/20170418/421775173679/running-riesgos-cardiologo-muertes-subitas.html>>. [Consulta: 14 de marzo de 2020].
8. Oliva, William; López, Jennifer. *Joven deportista muere tras correr la medio maratón Max Tott.* [en línea]. <<https://www.prensalibre.com/guatemala/sucesos/joven-deportista-fallece-tras-correr-la-medio-maraton-max-tott/>>. [Consulta: 15 de marzo de 2020].
9. Tzita Jocol, Sergio Patricio. *Incidencia del ejercicio físico sobre el estado de salud mental.* Trabajo de graduación de Licenciatura en Psicología. Escuela de Ciencias Psicológicas, Universidad San Carlos de Guatemala, 2005. 130 p. [Consulta: 13 de marzo de 2020]
10. Zayas, Diego J. *Las 7 Mejores redes sociales para deportistas.* [en línea]. <<https://diegojzayas.wordpress.com/2018/05/04/redes-sociales-deportistas/>>. [Consulta: 19 de marzo 2020].