

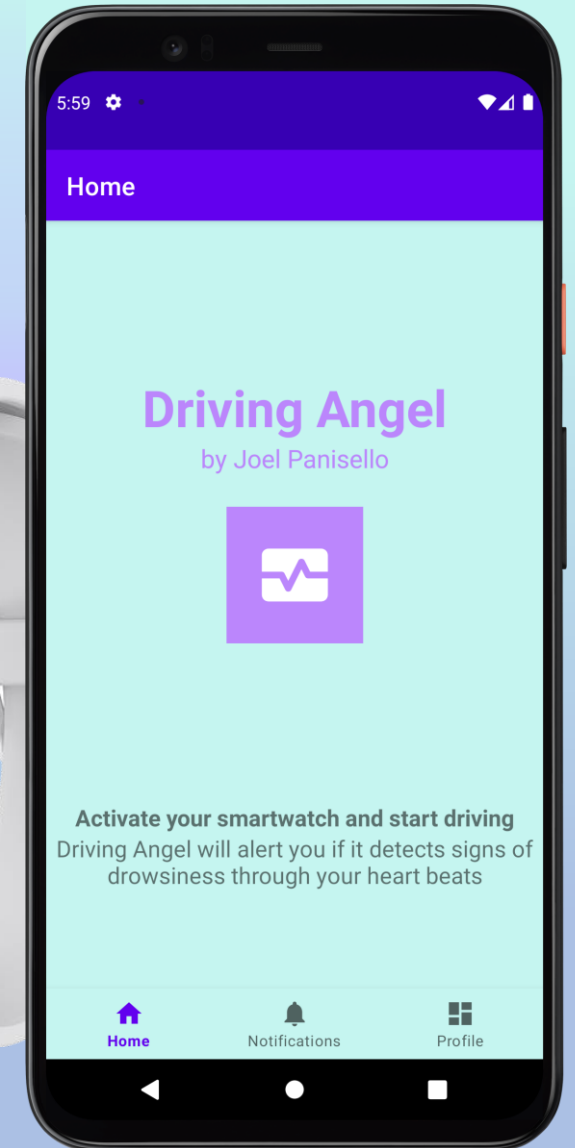
DESARROLLO DE APLICACIÓN “DRIVING ANGEL” PARA LA DETECCIÓN DE SOMNOLENCIA AL VOLANTE

GRADO EN INGENIERÍA INFORMÁTICA
Curso 2022/2023



UNIVERSITAT ROVIRA I VIRGILI

Joel Panisello Lozano



ÍNDICE DE CONTENIDO



1

Introducción

2

Tecnologías Utilizadas

3

Diseño e Implementación

4

Evaluación

5

Conclusiones

1

INTRODUCCIÓN

OBJETIVO GENERAL DEL TRABAJO

Enfoque en la prevención de accidentes causados por la somnolencia en carretera



1

INTRODUCCIÓN

SOLUCIONES FRENTE A LA SOMNOLENCIA AL VOLANTE

Nap Zapper

Basado en la detección de cambios en la inclinación de la cabeza mediante un giroscopio y emisión de sonido

Drowsy Driving Alert

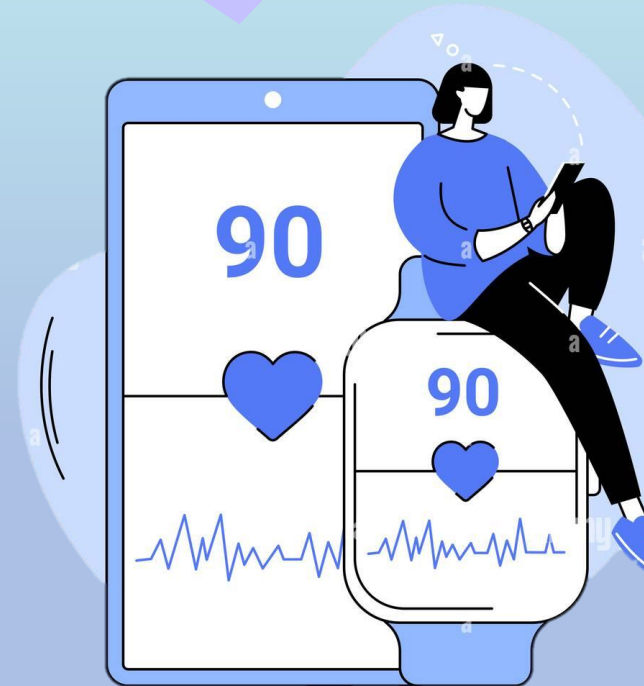
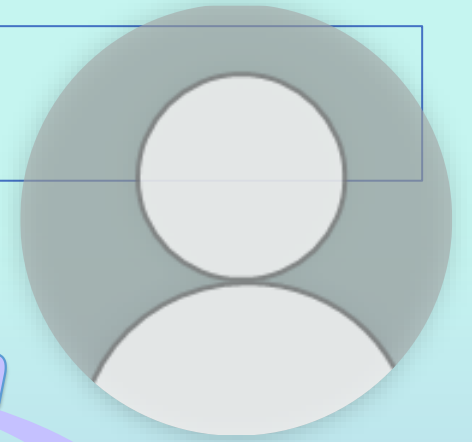
Basado en la detección del rostro. Emite una alarma al detectar el cierre de los ojos

Samsung Copiloto

Aplicación de smartwatch que emite vibraciones basada en sensores internos del reloj

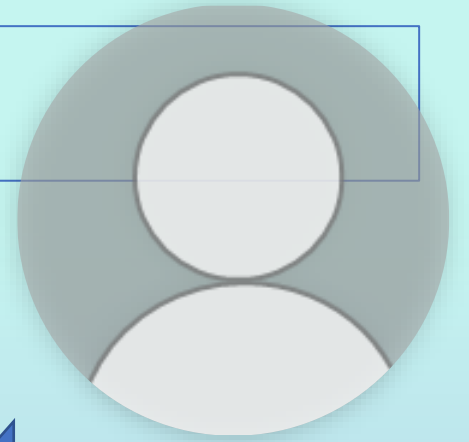
Smartwatch

Solución elegida en este trabajo



1

INTRODUCCIÓN



Desarrollo de una aplicación para teléfonos móviles Android, enfocada a la detección de somnolencia y/o fatiga mediante el uso de dispositivos externos wearables con Wear OS.

OBJETIVOS

APP para smartwatch capaz de leer los latidos para enviarlos a la APP móvil.

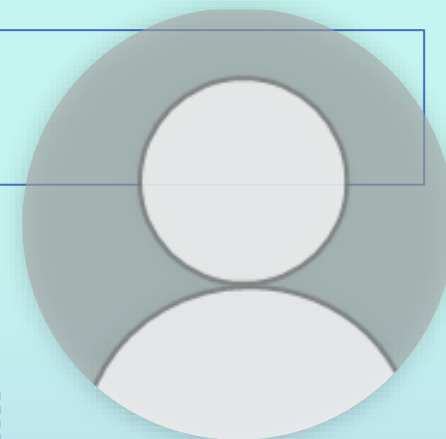
APP para móvil capaz de detectar un smartwatch mediante la tecnología BLE.

Algoritmo de detección de somnolencia que pueda avisarlo con un sonido en su móvil.

Visualización de los latidos obtenidos en pantalla mediante un gráfico de líneas en tiempo real.

2

TECNOLOGÍAS UTILIZADAS



Lenguaje: KOTLIN

El lenguaje Kotlin es el lenguaje oficial para dispositivos Android y ha sido desarrollado por JetBrains.

Mayor cuota de mercado

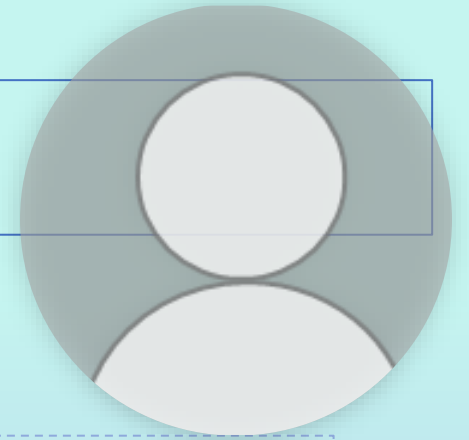
Mayor acceso a APIs de Google

Código Expresivo y Conciso

Código Seguro

2

TECNOLOGÍAS UTILIZADAS



IDE: Android Studio

Android Studio es el IDE oficial para Android. Está especialmente diseñado para el desarrollo de APPs en Android y se mantiene actualizado con las últimas características y mejoras.

Integración con herramientas específicas

Crear y organizar elementos para la interfaz sin código

Control de versiones

Emuladores móviles/smartwatch

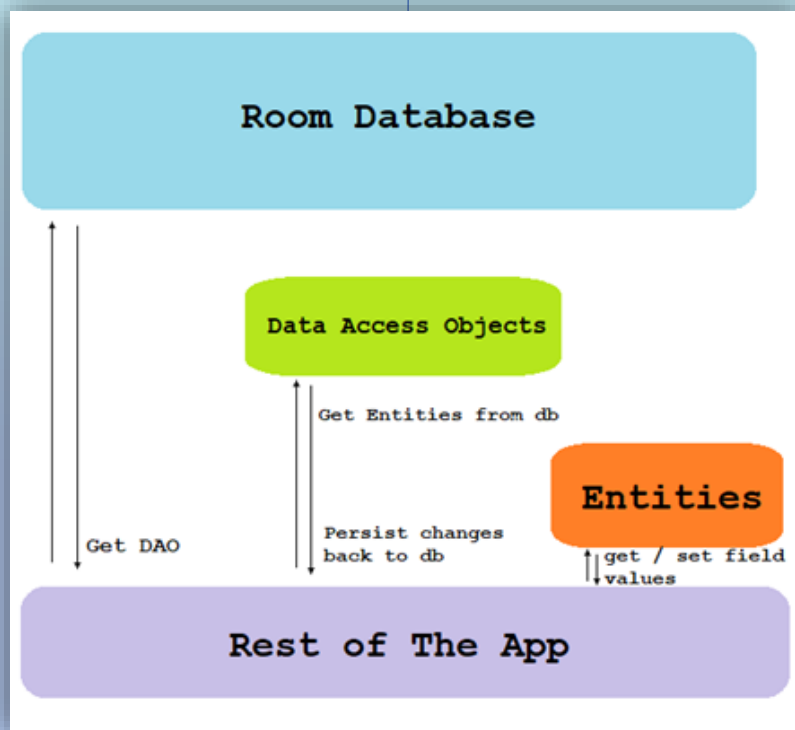


2

TECNOLOGÍAS UTILIZADAS

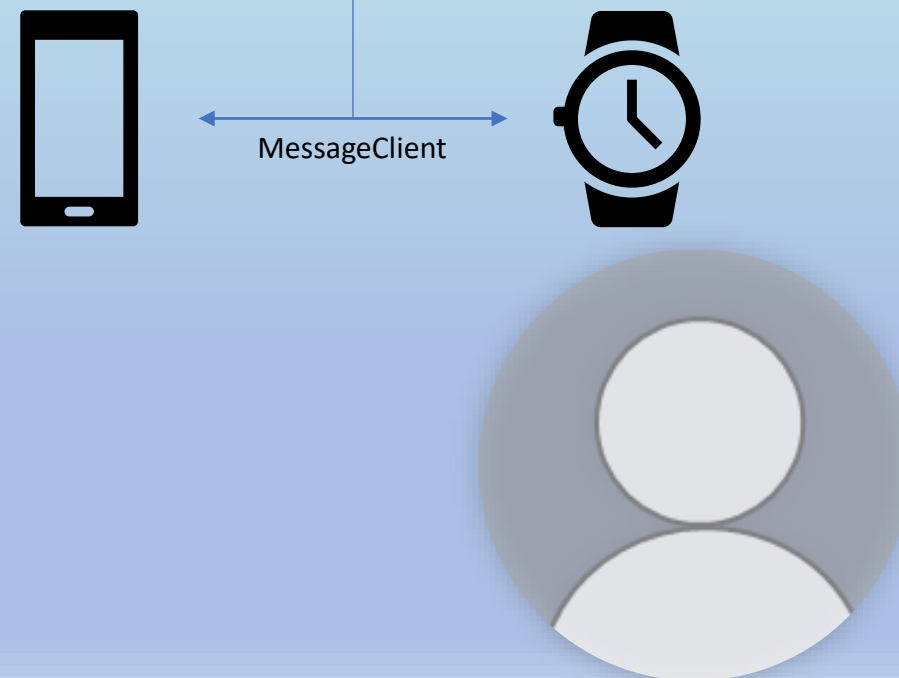
Base de Datos: SQLite

Room es un ORM creada por Android, que simplifica la tarea de trabajar con bases de datos en Android.



Conectividad: Data Layer

Esta API permite enviar mensajes entre dispositivos Android mediante un mecanismo de comunicación unidireccional.



2

TECNOLOGÍAS UTILIZADAS

Smartwatch: Samsung Galaxy Watch 5

El reloj escogido es el último modelo de la marca Samsung, con los sensores más punteros del sector tecnológico



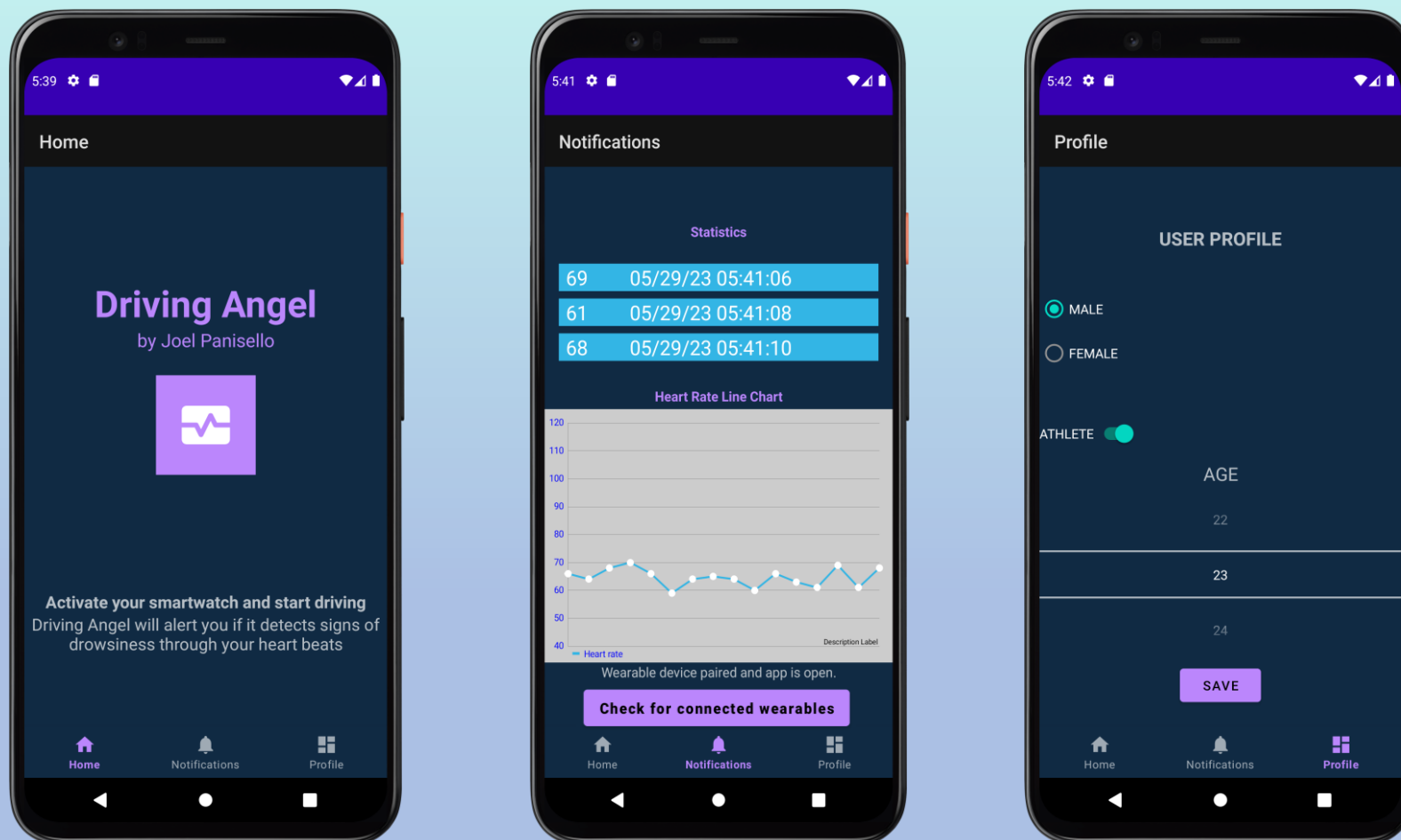
watch
OS



3

DISEÑO E IMPLEMENTACIÓN

Driving Angel Mobile



3

DISEÑO E IMPLEMENTACIÓN

1. Handshake de
comprobación inicial

2. Inicio de la recogida de
datos del reloj

3. Actualización de la lista y de
puntos en la gráfica



5:12



Home

Driving Angel

by Joel Panisello



Activate your smartwatch and start driving
Driving Angel will alert you if it detects signs of
drowsiness through your heart beats



Home



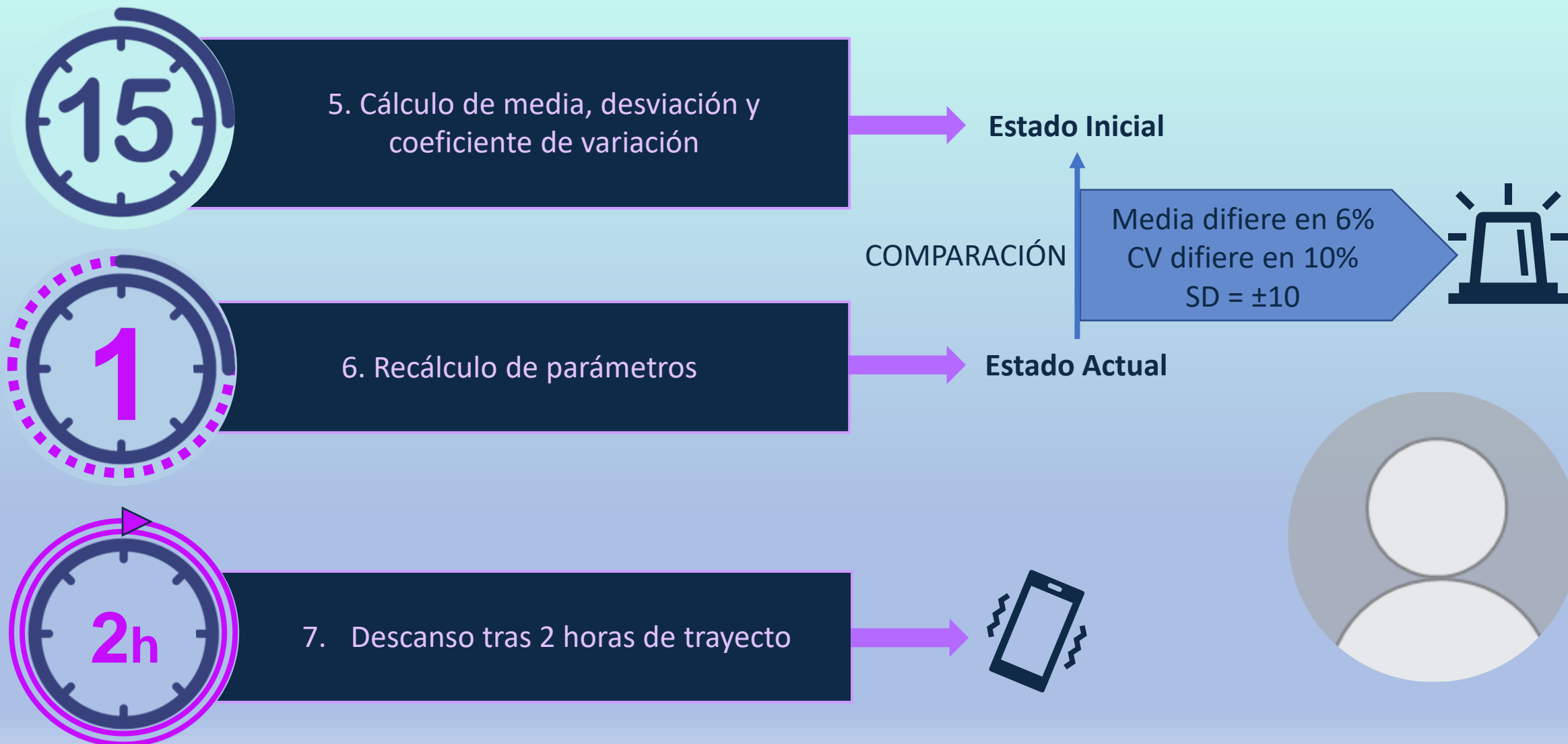
Notifications



Profile

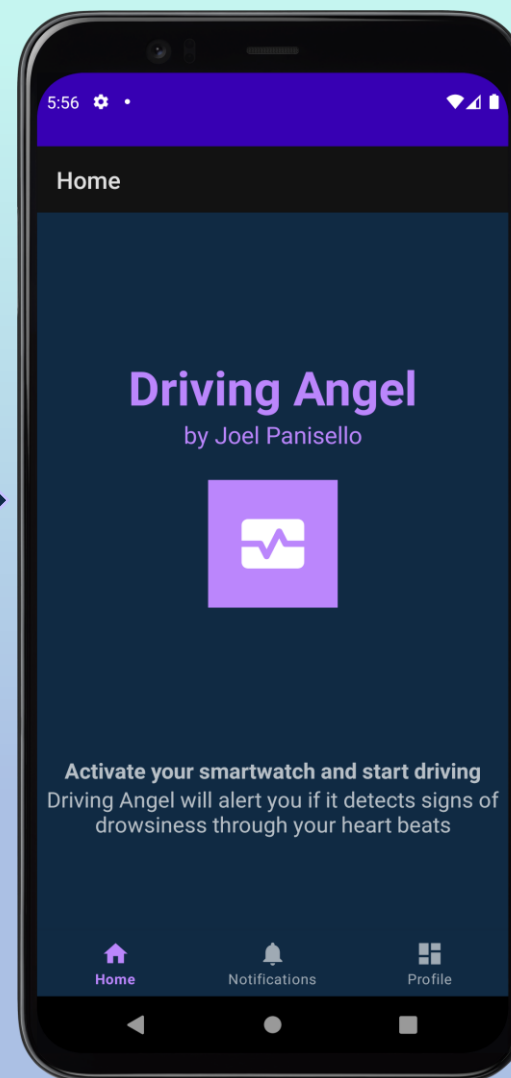
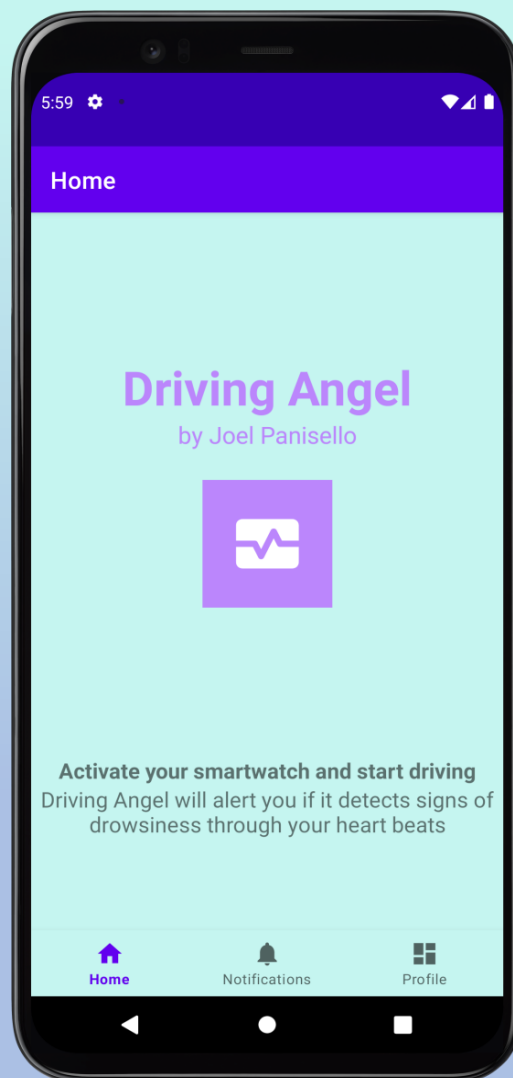
3

DISEÑO E IMPLEMENTACIÓN



3

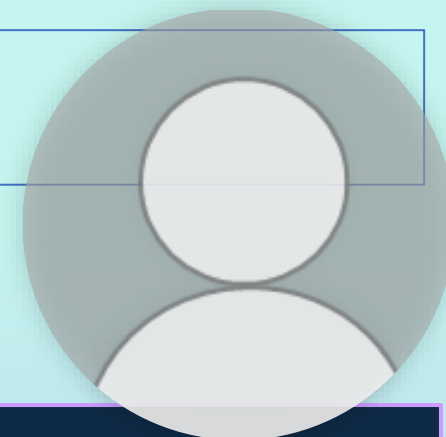
DISEÑO E IMPLEMENTACIÓN



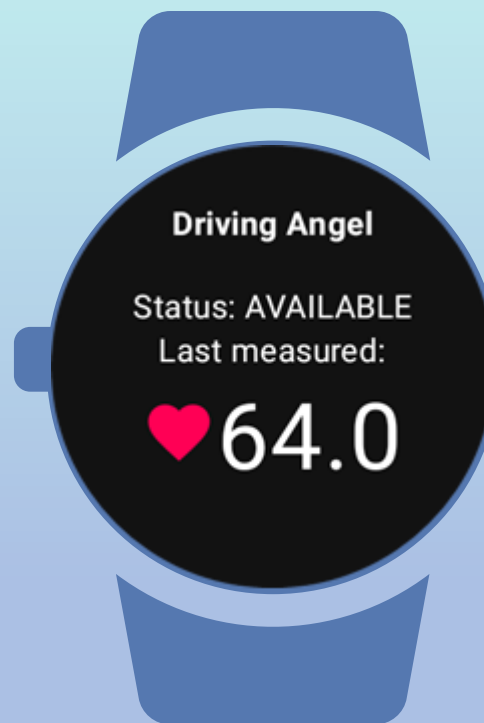
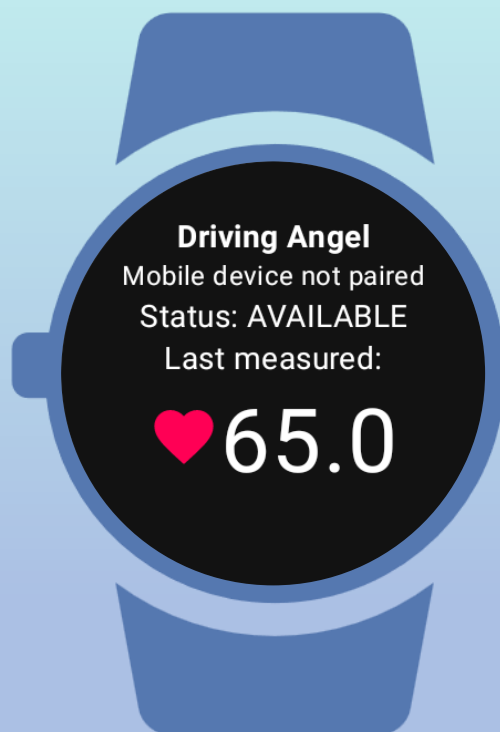
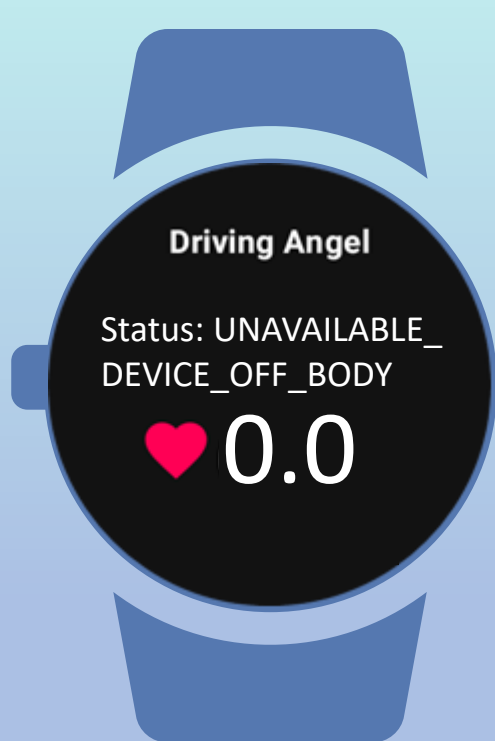
1. Reducción de fatiga visual

2. Mejora de legibilidad

3. Ahorro de energía



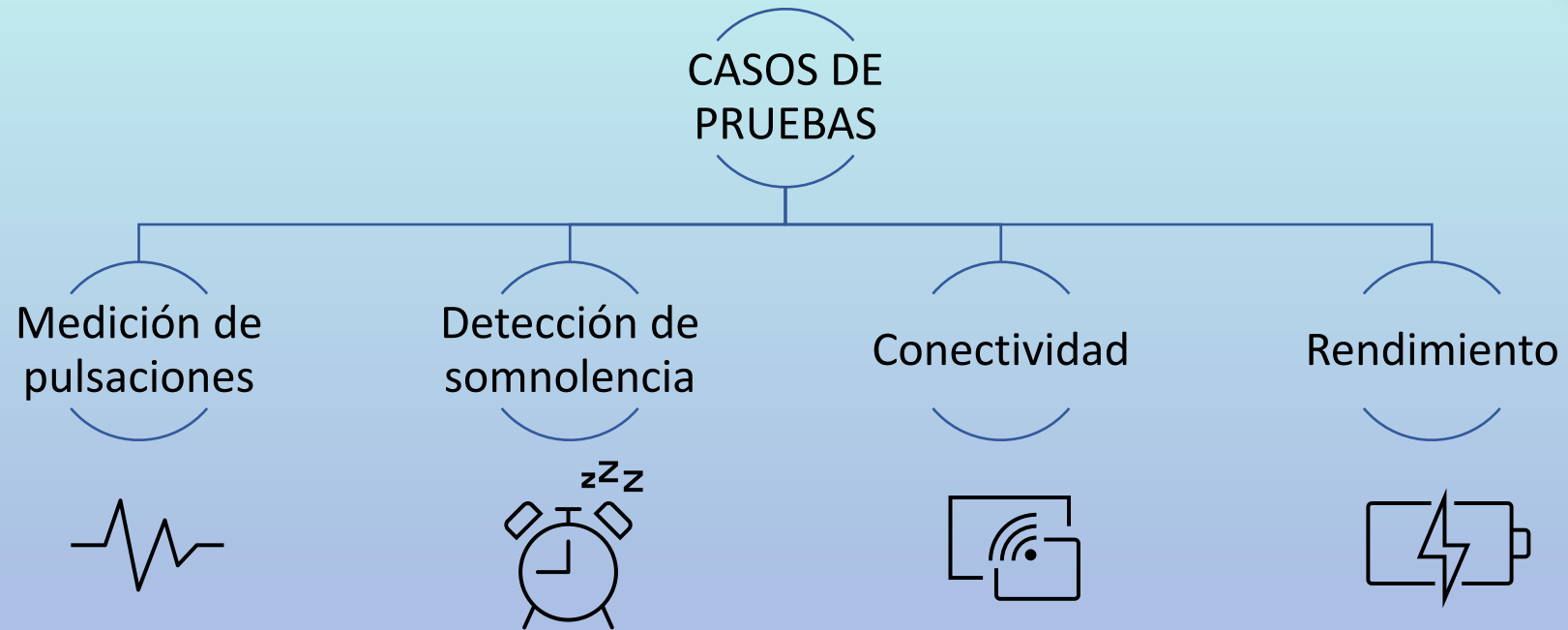
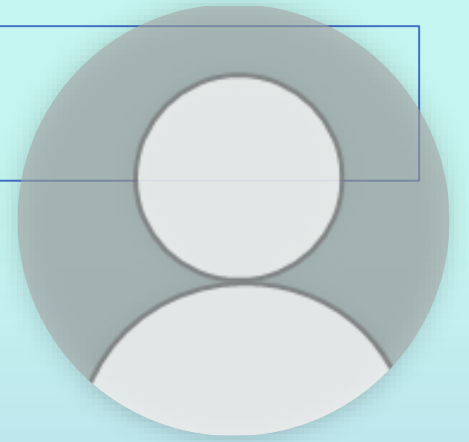
Driving Angel Wear



1. Después del handshake obtiene el ID del móvil

2. Convierte la frecuencia cardíaca a un ByteArray

3. Envía el mensaje y añade un listener para verificar



4

EVALUACIÓN



Uso de más sensores



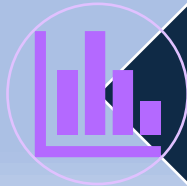
Información de perfil personalizada



Cámara frontal para detección de pestaño

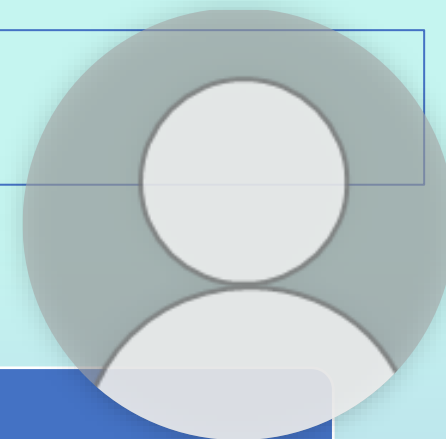


GPS para indicar área de servicio más cercana



Apartado de estadísticas y resumen





Desarrollo de APP funcional para la detección de somnolencia

- Medición de latidos
- Conexión entre móvil y reloj
- Procesamiento de los datos
- Desarrollo de algoritmo

Aplicación de los conocimientos en el área de desarrollo de APPs móviles

- Kotlin
- Corrutinas

Necesidad de mejoras para aumentar la precisión