

Grupo 9

Informe Semestral del Proyecto de Detección de Fibrilación Auricular mediante Señales PPG

Nicolás David Sabogal Velasquez

Andres Jose Acevedo Avila

Profesor encargado: Jonathan Roberto Torres Castillo, PhD



Semanas 1 y 2

Fundamentos e Investigación Inicial

- Revisión de bibliografía sobre señales PPG.
- Análisis de artículos científicos y los relojes empleados en estudios previos.
- Exploración de estrategias iniciales para la obtención de PPG desde los relojes disponibles.





Semanas 3–6

Pruebas y Dificultades Técnicas



- Pruebas de conexión Bluetooth con Python y lectura de servicios/UUIDs.
- Intentos de extraer PPG crudo desde relojes genéricos → sin resultados positivos.
- Análisis del repositorio Gadgetbridge, sin utilidad para los dispositivos actuales.

```
UUID.py M X
Antiguo > UUID.py > ...
1 import asyncio
2 import csv
3 from datetime import datetime
4 from bleak import BleakClient, BleakScanner, BleakError
5
6 # Dirección MAC del dispositivo
7 DEVICE_ADDRESS = "E7:CF:A3:30:D8:9E"
8
9 # UUID de la característica a la que deseas suscribirte
10 CHAR_UUID = "00000010-0000-3512-2118-0009af100700"
11
12 # Nombre del archivo CSV de salida
13 CSV_FILENAME = "datos_BLR.csv"
14
15 # Función para manejar notificaciones entrantes y guardar el CS
```

```
bleak_services.py M X
bleak_services.py > ...
1 """Herramientas para conectar a dispositivos BLE, listar servicios/características y más.
2
3 Importaciones necesarias para interactuar con dispositivos BLE.
4
5 Definición de la función connect_to_device que recibe la dirección del dispositivo y el tiempo de conexión.
6
7 La función conecta al dispositivo usando su dirección y retorna un cliente BleakClient.
8
9 Parámetros:
10 - address: dirección del dispositivo (por ejemplo 'AA:BB:CC:DD:EE:FF' o 'E7:CF:A3:30:D8:9E').
11 - timeout: tiempo máximo (segundos) para intentar la conexión.
12
13 Retorna:
14 - Un cliente BleakClient conectado al dispositivo.
15
16 Ejemplo de uso:
17
18 from bleak import BleakClient
19
20 def main():
21     client = connect_to_device("E7:CF:A3:30:D8:9E", timeout=10)
22     if client:
23         print("Conexión establecida con éxito!")
24     else:
25         print("No se pudo conectar al dispositivo.")
```

Semanas 7–8

Nuevos Dispositivos y Reorganización

Fitbit Sense 2, Huawei Watch GT6, Galaxy Watch 7.

Asignación individual:

- Nicolás → Galaxy Watch 7
- Andrés → Huawei GT6
- Configuración de entornos con SDKs y herramientas de desarrollo.
- Fitbit descartado por no permitir acceso a PPG crudo.





Rejected

Rejected

Rejected

Semanas 9-10

Avances Técnicos y Póster

Universidad Francisco José de Caldas

Development of an Application for the Early Detection of Atrial Fibrillation Using Digital Signal Processing Techniques Applied to Photoplethysmography (PPG) Data

Andres Jose Acevedo Avila¹, Nicolás David Sabogal Velasquez¹, Jonathan Roberto Torres Castillo^{1,2,3}, Esperanza Camargo Casallas², Miguel Ángel Padilla Castañeda³, Lely Adriana Luengas Contreras², Nancy Esperanza Olarte López⁴

¹ Virtus - Grupo de Investigación en Ambientes Virtuales de Aprendizaje, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
² DIGITI - Sistemas Digitales Inteligentes, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
³ UIDT - Grupo de Bioinstrumentación y Unidad de Investigación en el Hospital General de México - Instituto de Ciencias Aplicadas y Tecnología (ICAT), Universidad Nacional Autónoma de México, Ciudad de México, México
⁴ GI-ITEC - Grupo de Investigación e Innovación Tecnológica en Electrónica y Comunicaciones, Universidad Militar Nueva Granada, Bogotá, Colombia

1. Introduction

Smartwatches use PPG sensors, but access to raw PPG data is often limited. To validate AF detection algorithms, a complete capture-transmission-storage pipeline is required. This work develops a system that retrieves raw PPG data from a smartwatch, sends it via HTTP, and stores it for analysis.

2. Methodology

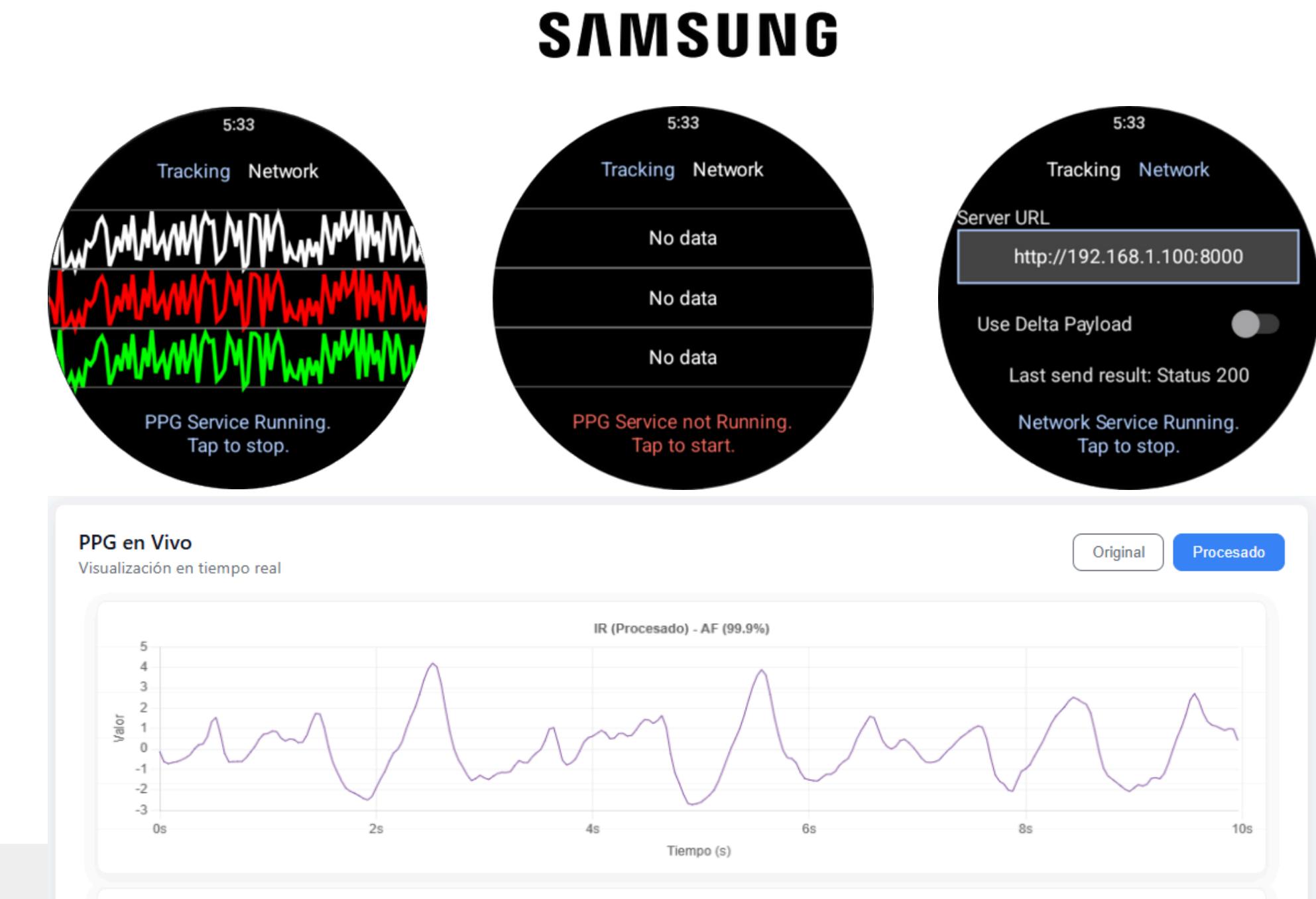
Instruments:
BLE with Python + Bleak, references from Gadgetbridge, Android Studio for the Galaxy Watch 7, and a receiving server developed in FastAPI.
Procedure:

4. Results

It was not possible to capture raw data on devices without an SDK, so the solution focused on the Galaxy Watch 7, which does provide sensor access through Health Connect.

Display of data received from the Galaxy Watch 7 in the API

Original Procesado

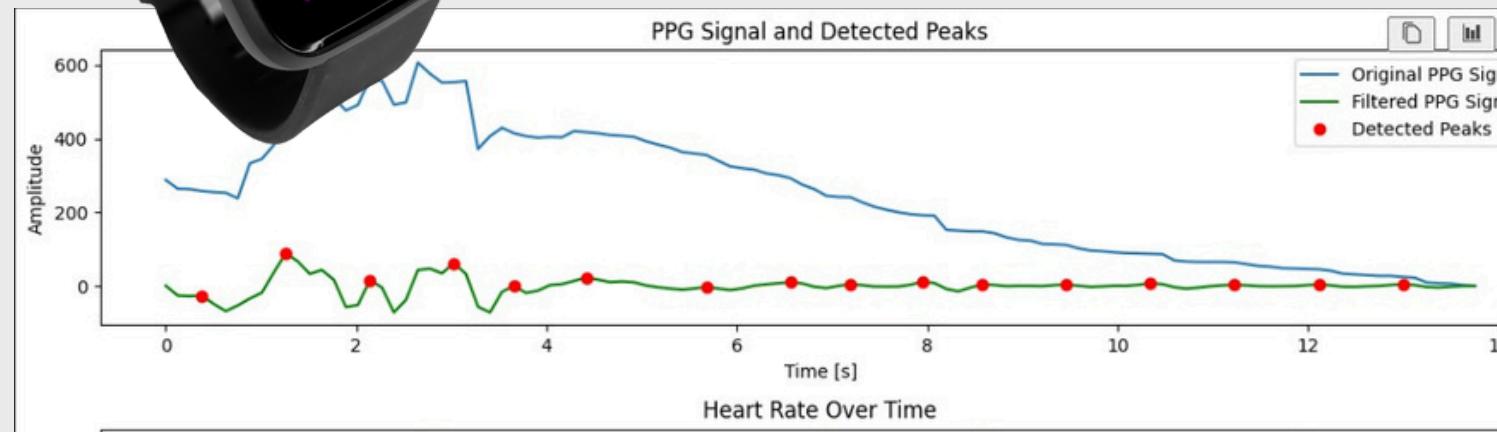


Semanas 11-12

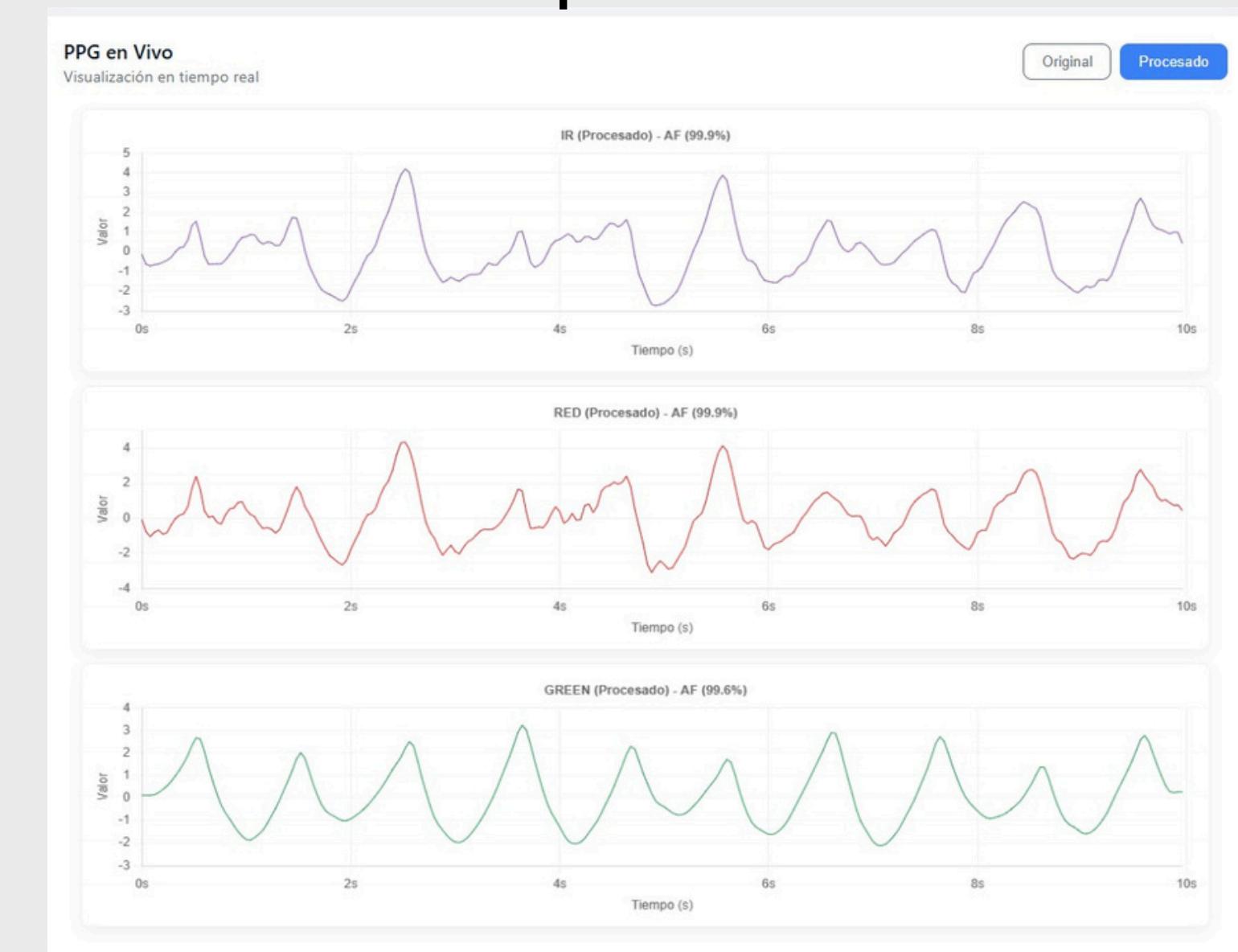
Pinetime y Modelo en la API

PINETIME

- Modificar Firmware
- Capturar PPG



Actualizar la API para consumir el modelo





Próximos pasos

- **Evaluar dispositivos restantes (Xiaomi y Withings)**
- **Desarrollar los endpoints necesarios de la API para que el equipo de frontend pueda implementar la aplicación Android para la detección de fibrilación auricular (FA).**
- **Enviar el resumen a Huawei para posible colaboración.**

