Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

MENCIÓN EN COMPUTACIÓN

Estudio de la frecuencia de muestreo en el reconocimiento biométrico mediante la forma de andar y el ECG usando smartwatches

Autor:
Pablo Peláez Marín

Tutores:
Carlos Enrique Vivaracho Pascual
María Aránzazu Simón Hurtado

Agradecimientos

En primer lugar a mis tutores de TFG, que me ofrecieron la posibilidad de trabajar con ellos en un proyecto que ya habían comenzado previamente, depositando en mí toda su confianza y ayudándome siempre que fue necesario.

También en el ámbito académico a todos los profesores, que impartieron sus asignaturas con entusiasmo y me permitieron adquirir conocimientos que he puesto en práctica en este trabajo y seguro que me serán esenciales en el futuro.

Un gran reconocimiento a los compañeros de la universidad, que muchos de ellos se convirtieron en grandes amigos.

Por último, un agradecimiento especial a mi familia, amigos y personas que han pasado por mi vida en estos últimos años. Su apoyo incondicional me ha permitido trabajar día tras día.

Resumen

La popularización del uso de vestibles ("wearables") como relojes y pulseras inteligentes ha hecho que el interés en su uso para el reconocimiento biométrico haya aumentado en los últimos años. Son muchas las referencias que se pueden encontrar en este campo.

Uno de los aspectos menos estudiados es la parte del consumo de recursos en el dispositivo, dadas las limitaciones que normalmente estos tienen. Es un aspecto importante en sistemas reales. Uno de los parámetros que más influyen en este consumo es la frecuencia a la que se muestrean las señales de los dispositivos inerciales (acelerómetro y giróscopo), usados para el reconocimiento de la forma de andar. Lo mismo pasa con los sensores de ECG. En este trabajo nos centramos en este aspecto.

El objetivo es analizar el rendimiento de los sistemas de reconocimiento con respecto a la frecuencia a la que se muestrea la señal. Se busca un óptimo equilibrio entre el consumo de recursos usando frecuencias de muestreo bajas y buen rendimiento en el reconocimiento, es decir, estudiar hasta dónde se puede bajar la frecuencia de muestreo, sin que esto afecte grandemente al rendimiento del sistema.

Abstract

The increasing popularity in the use of wearables such as smartwatches or smart bracelets has made the analysis of the use of these tools for biometric recognition far more interesting in recent years. There are many references which can be found in this field.

One of the least studied aspects is the part of resource consumption of the device itself, given the limitations that they usually have. This is an important aspect of the real systems. One of the parameters that most influences this consumption is the frequency to which signals of the inertial devices (accelerometer and gyroscope), used for recognition of the way of walking. The same applies to the electrocardiogram sensors. In this project, we are going to focus on this regard.

The objective is to analyse the performance of the recognition systems with respect to the frequency at which the signal is taken. What is to be found, is the optimum equilibrium between the consumption of resources using low frequency sampling and good performance in the recognition, this means, to study until what point you can lower the frequency of sampling without decreasing greatly the efficiency of the system.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El tema abordado en el presente trabajo forma parte de un proyecto iniciado hace varios años por profesores de la Universidad de Valladolid. Concretamente, este trabajo es una continuación de un TFG y TFM anterior:

- TFM del año 2020 del Máster en Inteligencia de Negocio y Big Data en Entornos Seguros realizado por la alumna Irene Salvador Ortega y tutorizado por Carlos Enrique Vivaracho Pascual y María Aránzazu Simón Hurtado [1]. Se titula "Investigación y Desarrollo de un Sistema de Reconocimiento Biométrico mediante Dispositivos Ponibles (Wearables)" y en este trabajo, además de capturar datos, se creó un sistema de reconocimiento biométrico basado en la forma de andar.

- TFG del año 2022 del Grado en Ingeniería Informática realizado por el alumno Mario Garrido Tapias y tutorizado por Carlos Enrique Vivaracho Pascual y María Aránzazu Simón Hurtado [2]. Se titula "Reconocimiento Biométrico Mediante ECG Usando Dispositivos Ponibles (Wearables)" y en este trabajo se creó un sistema de reconocimiento biométrico basado en ECG. Además, también se capturaron los datos con los que se trabajó.

Como se puede ver, el ámbito del trabajo a realizar se enmarca en el reconocimiento biométrico de personas, es decir, la autenticación del usuario mediante rasgos físicos (huella, iris, etc) o de comportamiento (firma, tecleo o forma de andar).

1.1. Motivación

A la hora de plantear el trabajo, resulta importante comprender que se trata de un proyecto de investigación, en el cual se desarrolla y mantiene un software para realizar experimentos y analizar todos los resultados. Esto tiene sus ventajas e inconvenientes. El principal inconveniente es que el campo de investigación es muy amplio, por lo que normalmente el alcance del trabajo a realizar tiene que ser limitado. Sin embargo, las numerosas ventajas de este campo me animaron a tomar con ganas el proyecto, como por ejemplo el trabajo colaborativo y la oportunidad de descubrir nuevos temas cada día.

Además, el ámbito de investigación resulta innovador y de interés actual. Se trata de los sistemas inteligentes, basados en programas de computación con características que se asemejan a la inteligencia humana. El tema de la biometría va más allá en la seguridad informática, ya que se ofrece como alternativa al uso tradicional basado en contraseñas, que requieren que nos acordemos de ellas y ser lo suficientemente complejas como para no ser descubiertas. En este trabajo el esfuerzo se centra sobre el reconocimiento biométrico mediante la forma de andar y mediante los electrocardiogramas (ECG), ambos capturados mediante vestibles (wearables) como relojes o pulseras inteligentes. Para la captura de la forma de andar se usa un reloj "Motorola Moto 360" ("Moto") y una pulsera de actividad "Microsoft Band 2" ("Micro"). Para la captura del ECG se usa el reloj "Withings Move ECG". En la Sección 2 se pueden observar imágenes de estos dispositivos.

1.2. Objetivos

El presente trabajo se centra en la investigación sobre el consumo de recursos en dispositivos vestibles, ya que estos son limitados. Para ello, el parámetro de la frecuencia a la cual se muestrea la señal resulta crucial. Por lo tanto, el objetivo principal es analizar el rendimiento de los sistemas de reconocimiento con respecto a la frecuencia de muestreo de la señal. Sería deseable un equilibrio entre la optimización del consumo de recursos usando frecuencias de muestreo bajas y buen rendimiento en el reconocimiento.

Los objetivos mostrados aquí son los del trabajo finalmente realizado, ya que los inicialmente planteados, como se verá más adelante (Sección 3), eran diferentes. Como se mostrará, un riesgo que inicialmente se suponía muy bajo, pero que era catastrófico, ocurrió y hubo que replantear el trabajo.

El objetivo general planteado se divide en los siguientes objetivos específicos:

- Entender la biometría basada en la forma de andar y el ECG.
- Analizar, comprender y modificar el software desarrollado en trabajos anteriores para su adaptación a las necesidades de este trabajo.
- Adquirir más muestras de ECG, para tener una base datos más numerosa y, así, obtener resultados más confiables.
- Analizar y comparar el rendimiento de un sistema de reconocimiento tanto de forma de andar como de ECG para distintas frecuencias de muestreo.

1.3. Organización de la memoria

La documentación de este proyecto viene estructurada de la siguiente manera:

- Capítulo 1: **Introducción**. Aquí se presenta el contexto del trabajo y una breve descripción del mismo.
- Capítulo 2: **Marco teórico**. Se expone el fundamento teórico base para la realización del trabajo.
- Capítulo 3: **Planificación y herramientas**. Se comentan todas las tecnologías y herramientas útiles en el trabajo, además de su planificación.
- Capítulo 4: **Desarrollo**. Contiene aspectos relacionados con la mplementación del software necesario, recogida de datos y realización de los experimentos.
- Capítulo 5: **Conclusiones**. En este capítulo se presentan las conclusiones del trabajo, además de las dificultades encontradas. También se incluyen las líneas futuras, con posibles nuevas líneas de investigación y retos a partir de las conclusiones obtenidas.

Capítulo 2

Contexto teórico

2.1. Biometría

La biometría viene definida como el reconocimiento automatizado de individuos basado en sus características biológicas y de comportamiento [3]. Dentro del ámbito informático, se puede denominar autentificación biométrica y consiste en la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo para autentificar su identidad [4]. Este sistema automatizado que realiza tareas de biometría también se denomina "sistema biométrico".

Se comenzó a investigar en este tema a finales del siglo XIX hasta la actualidad, donde la biometría ha llegado a nuestro a día a día con ejemplos tan cotidianos como la huella dactilar y el reconocimiento facial del móvil. En la Figura 2.1 se pueden observar algunos ejemplos de biometría. A continuación, se van a clasificar las distintas características humanas que se estudian en biometría [5]:

- **Características de tipo fisiológico**. Tienen que ver con las medidas y datos que se obtienen a partir de las partes del cuerpo humano. Destacan por su uso más habitual las huellas dactilares, el iris, la retina, la voz, la forma de la mano y el rostro.
- **Características de comportamiento**. En este caso las medidas y datos se obtienen a partir de las acciones de una persona, que en algunos casos, como por ejemplo el reconocimiento mediante la voz, indirectamente también incluyen características físicas (el tracto bucal en el caso de la voz). Las más destacables son el uso de un teclado y la firma.

Figura 2.1: Ejemplos de biometría [6]

Además de estas características biométricas, hay otras como pueden ser el termograma, las venas de las manos, la forma de andar y los electrocardiogramas. En estos dos últimos aspectos se centrará este trabajo.

Para que una característica pueda ser usada en el reconocimiento biométrico, idealmente debe cumplir las siguientes propiedades [7]:

- Universalidad: todas las personas deben tener la característica biométrica en cuestión.
- Unicidad: cualquier par de personas que se elijan al azar deben tener diferencias en esa característica.
- Permanencia: dicha característica debe permanecer lo suficientemente constante en el tiempo para cada persona.
- Recolectable: la característica debe poder medirse de manera cuantitativa.

En la Tabla 2.1, se resumen las ventajas y desventajas de las técnicas biométricas más destacadas [5].

Todas estas técnicas biométricas pueden tener sus problemas asociados de seguridad y privacidad. Para evitarlo, se puede optar por el cifrado de datos biométricos y la seguridad de los sistemas biométricos. Además, se requiere de una colaboración entre investigadores, industria y reguladores para garantizar la seguridad y privacidad de los sistemas biométricos. [7].

Otra posibilidad importante es la de utilizar biometría multimodal, pudiendo combinar sistemas biométricos como, por ejemplo, Iris + Huella dactilar + Rostro, Iris + Huella dactilar + Venas, Huella dactilar + Firma. Estos sistemas multimodales son más fiables ya que aportan evidencias que son idependientes entre sí. Su falsificación es, por lo tanto, mucho más complicada [8].

	Ventajas	**Desventajas**
Reconocimiento facial	Sencillo, rápido, bajo coste	Alterable por la iluminación
Lectura huella digital	Bajo coste, seguridad	Lesiones en el dedo pueden alterar la autenticación
Lectura del iris y retina	Seguridad	Intrusivo
Lectura palma de la mano	Poco almacenamiento necesario	Lento, poca seguridad
Reconocimiento de firma	Bajo coste	Alterable según el momento
Reconocimiento de voz	Bajo coste, útil para accesos remotos	Lento, alterable, reproducible

Tabla 2.1: Ventajas y desventajas de algunas técnicas biométricas

2.2. Sistema biométrico de la forma de andar

La forma de caminar humana es un movimiento periódico, compuesto por un paso de la pierna derecha y un paso de la pierna izquierda. Se trata de un tipo de biometría basada en comportamiento. Cabe destacar que la propiedad de permanencia no se cumple, por eso este tipo de biometría se conoce como "débil" (Soft Biometrics) [1].

Los sistemas iniciales para capturar la forma de andar han evolucionado, pasando de ser cámaras de grabación, sensores de presión y de fuerza a sistemas más complejos y más difíciles de falsificar. Estos sistemas se basan en los sensores que llevan los teléfonos y relojes inteligentes. Dos de los más útiles son el acelerómetro y el giróscopo, que permiten medir la aceleración y el movimiento en tres dimensiones respectivamente del dispositivo que se esté usando [9].

Las señales que se obtienen de los sensores se dividen en ciclos ya que estos son periódicos. Se conoce como ciclo de marcha al transcurso del tiempo desde que se despega un pie del suelo hasta que se va a volver a despegar ese mismo pie del suelo. En la Figura 2.2 se puede observar el proceso del ciclo de marcha.

Este mecanismo de identificación a partir de la forma de andar tiene sus ventajas y desventajas [1]:

Ventajas:

- Es cómodo para el usuario ya que solamente tiene que andar, sin tener que

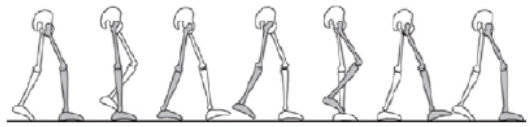

Figura 2.2: Esquema de un ciclo de la marcha [9]

interactuar con el sistema.

- El reconocimiento es continuo, ya que el propietario se mantiene automáticamente autorizado para el acceso al dispositivo.
- Se puede capturar la información a distancia.
- Permite avanzar en el ámbito sanitario, llegando a identificar algunas enfermedades como Parkinson y esclerosis múltiple a través de la forma de andar.
- Resulta complicado de falsificar.

Desventajas:

- La existencia de algunos factores externos que podrían influir en la forma de andar, por ejemplo, el estado de la superficie, las condiciones meteorológicas o la vestimenta.
- También hay otros factores internos que influyen en la forma de andar como por ejemplo el estado físico y mental.
- Un impostor podría observar la forma de caminar de un usuario.
- Al ser una manera de identificar a las personas de manera única, es información sensible que si se roba podría ser muy perjudicial. Esto también podría revelar condiciones médicas de las personas.

Los dispositivos “wearables” utilizados en trabajos anteriores para capturar datos fueron el “Microsoft Band 2” y “Motorola Moto 360”. En las Figuras 2.3 y 2.4 se pueden observar estos dispositivos.

Figura 2.3: Microsoft Band 2

Figura 2.4: Motorola Moto 360

2.3. Sistema biométrico del electrocardiograma

El electrocardiograma es un procedimiento que permite registrar la actividad eléctrica del corazón de una persona. Esto es posible ya que cuando late el corazón, se emite una señal eléctrica que activa las dos aurículas y los dos ventrículos del corazón para que la sangre se bombee de manera correcta por todo el cuerpo.

Los electrocardiogramas se pueden visualizar como un conjunto de líneas onduladas y su interpretación nos da un registro de la velocidad de latido del corazón, la constancia del ritmo de latido, la sincronización de las señales eléctricas del corazón y en ocasiones el tamaño o posición de las cavidades que forman parte del corazón [10]. El ECG tiene dos fases principales: despolarización y repolarización de las fibras musculares que componen el corazón. La fase de despolarización comprende a la onda P (despolarización auricular) y la onda QRS (despolarización ventricular). La fase de repolarización comprende a las ondas T y U (repolarización ventricular) [11]. Todas esas ondas se muestran en la Figura 2.5.

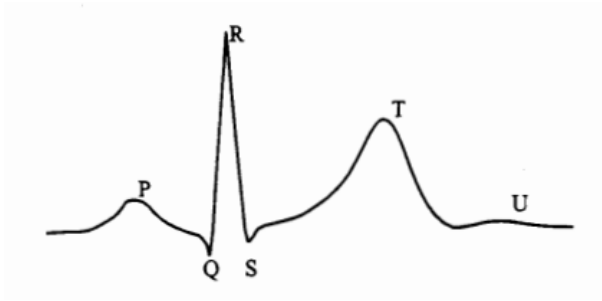

Figura 2.5: Elementos de un ECG [11]

Se trata de una prueba muy extendida en la actualidad y se usa para viligar la actividad del corazón y detectar enfermedades cardiacas. Tradicionalmente se

vienen realizando en hospitales o centros de salud, pero en la actualidad tenemos dispositivos vestibles como el que se utiliza en este trabajo que también permiten registrar electrocardiogramas. Se trata de dispositivos reconocidos por la comunodad médica ya que permiten detectar arritmias y en concreto episodios de fibrilación auricular con una alta precisión [12].

En cuanto a su aplicación en la biometría (tipo fisiológico), todavía estamos en fases iniciales. Hay algunos estudios como el del artículo [13] que muestran tasas de error bajas en la recogida de electrocardiogramas mediante dispositivos vestibles. El inconveniente es que hay una propiedad de la biometría que realmente no se cumple: esta es la permanencia, ya que la biometría de los ECG es cambiante con el tiempo para un mismo individuo. Para contrarrestarlo, se propone realizar ECGs a intervalos regulares y así reducir los cambios naturales. También podría ser interesante combinar el uso de ECG con otros mecanismos biométricos para garantizar la seguridad.

El dispositivo "wearable" utilizado en el trabajo para capturar datos de ECG es el "Withings Move ECG". Su funcionalidad más destacada es la medida de electrocardiogramas con una precisión elevada. Se puede visualizar en la Figura 2.6.

Figura 2.6: Withings Move ECG

2.4. Frecuencia de muestreo

Se conoce como frecuencia de muestreo a la cantidad de muestras que se toman por unidad de tiempo para convertir la señal analógica en señal digital [14].

En los dispositivos vestibles como el que se usa en este trabajo, se necesita digitalizar la señal para trabajar con formas de onda. Para lograr aproximarse a la señal analógica, influye el valor que tengan la frecuencia de muestreo y la profundidad en bits. Se denomina profundidad al número de bits con que se codifica la muestra. En principio, a mayor valor de estos parámetros, mayor será la aproximación a la señal

analógica. En los dispositivos comerciales que vamos a usar aquí, solo es configurable la frecuencia de muestreo a la que se adquiere la señal. Por eso, nos centramos solo en esa parte.

En la Figura 2.7 se muestra de manera gráfica de qué manera influye el valor de la frecuencia de muestreo en la representación de una onda senoidal.

Figura 2.7: Onda senoidal [14]

Cada línea roja vertical representa una muestra de la onda en ese instante de tiempo. En este caso la frecuencia de muestreo es de 6Hz. Combinando todas esas muestras se produce la onda digital que podemos utilizar, visible en la Figura 2.8.

Figura 2.8: Onda digital senoidal con frecuencia de muestreo 6Hz [14]

Utilizando una frecuencia de muestreo algo mayor (10Hz), la aproximación de la onda digital se vuelve más precisa tal y como se aprecia en la Figura 2.9.

Hay un estudio previo sobre la influencia de la frecuencia de muestreo en el rendimiento de dispositivos comerciales que miden la forma de andar [15]. Nos basamos en su metodología y resultados para avanzar en nuestro estudio. En ese artículo se han calculado la estimación del error promedio entre todos los usuarios y algunos índices de rendimiento en el proceso de autenticación.

En cuanto a la metodología, utilizan un reloj inteligente TicWatch con sensor acelerómetro y frecuencias de muestreo de 200Hz, 100Hz, 50Hz, 25Hz y 12.5Hz.

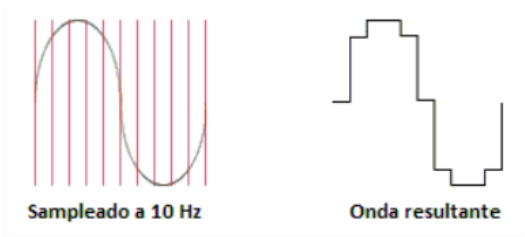

Figura 2.9: Onda digital senoidal con frecuencia de muestreo 10Hz [14]

Centrando la comparación entre la frecuencia de muestreo y el error de autenticación EER, los resultados que se obtienen se muestran en la Figura 2.10. Tomando una frecuencia de 12.5Hz, el error es más alto (en torno al 10 %). Desde 25Hz en adelante el error parece mantenerse constante, aunque con frecuencia de 100Hz el error parece en torno a un 1 % superior respecto a frecuencias de 25Hz o 50Hz [15].

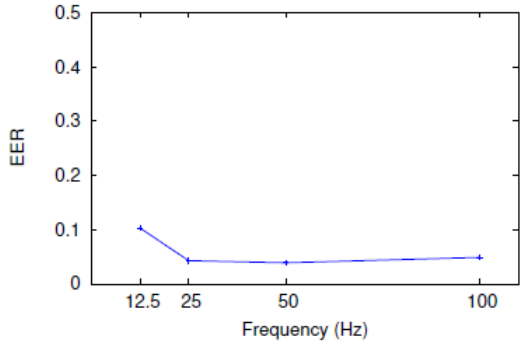

Figura 2.10: Comparación frecuencia de muestreo y EER [15]

Así, parece que el aumento de frecuencia de muestreo proporciona mayor precisión en la autenticación pero hasta cierto punto, en el cual ya no mejora la precisión. En cuanto al consumo de energía, se puede observar en la Figura 2.11 cómo el consumo aumenta a medida que va aumentando la frecuencia de muestreo, por lo que tampoco interesaría incrementar demasiado la frecuencia de muestreo. Aunque está fuera del ámbito de estudio aquí, nos parece interesante comentar que en el trabajo referenciado también se estudia la opción de externalizar parte de la computación a un dispositivo externo, pero como se muestra esta opción no es conveniente ya que la energía gastada en transferir los datos a un dispositivo externo es mayor que la energía gastada realizando el cómputo internamente; aunque esto, es conveniente señalar, va a depender mucho del sistema de clasificación que se use.

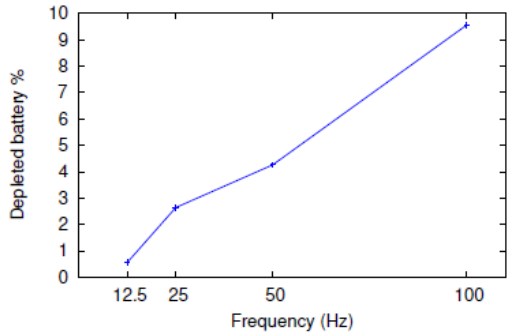

Figura 2.11: Comparación frecuencia de muestreo y gasto energético [15]

En la Tabla 2.2 se pueden apreciar las tres subtareas que más energía consumen en promedio.

La subtarea "muestreo del acelerómetroïncluye las operaciones del muestreo de la señal del sensor acelerómetro, la conversión de los componentes de aceleración y el guardado de estos componentes en un buffer circular. Su consumo es de 3mW, el mayor de todos. Además, este consumo de energía es una línea base que no puede evitarse.

En segundo lugar, se muestra la transmisión a un dispositivo móvil. Se implementa un protocolo para transmitir de manera fiable una ventana de muestras de cinco segundos, se mide el tiempo requerido para transmitir cada ventana (ciclo de trabajo) y se combina la información del ciclo de trabajo con el consumo de energía de transmisión. El resultado da un consumo de 2.9mW.

En último lugar, se muestra la extracción de características con un consumo de 1.1mW.

Subtarea	**Consumo promedio de energía**
Muestreo de aceleración	3.0mW
Transmisión al móvil	2.9mW
Extracción de características	1.1mW

Tabla 2.2: Consumo energético por subtareas [16]

Capítulo 3

Planificación

3.1. Metodología de trabajo

Se ha elegido una metodología ágil para el desarrollo del trabajo. La finalidad de esta metodología es proporcionar pequeños resultados cada poco tiempo, aumentando progresivamente la funcionalidad. Generalmente se da un enfoque flexible y los equipos de trabajo son pequeños, por lo que vendrá bien al proyecto [17].

Las reuniones programadas de manera habitual forman un papel fundamental en esta metodología. Se organizan tanto por videoconferencia como de manera presencial. Para algunos progresos intermedios se utiliza la vía email. También se realizan sprints en algunas etapas del proyecto para lograr los objetivos marcados, mediante entregas iterativas.

3.2. Planificación del proyecto inicial

El proyecto se enmarca dentro del reglamento de la Universidad de Valladolid, y dentro del programa del Grado en Ingeniería Informática consta de 12 créditos ECTS. El número de horas estipulado para la realización del trabajo es de 300, comenzando en octubre de 2023 y llegando hasta junio de 2024.

En este punto es conveniente indicar que el enfoque inicial del proyecto no era el que finalmente se ha realizado y se ha mostrado en los objetivos. La propuesta inicial era continuar con un trabajo anterior para la aplicación de una nueva alternativa en la extracción de características, basada en FMM (Frecuency Modulated Möbius) [9].

El objetivo era ampliar ese estudio ampliándolo a más configuraciones del sistema de reconocimiento basado en la forma de andar y al reconocimiento biométrico mediante ECG.

A continuación, se presentan las distintas fases que se planificaron inicialmente para el proyecto, con sus actividades concretas a realizar:

- Sprint 1: Preparación del proyecto
  - Instalación del software para el desarrollo
  - Investigación sobre los proyectos existentes
- Sprint 2: Software de adquisición (análisis-diseño-pruebas)
  - Configuración manual de Withings
  - Puesta en marcha de los formularios
- Sprint 3: Ampliación del corpus en ECG
  - Gestión del protocolo de adquisición
  - Realización de tomas para nuevos usuarios
  - Almacenamiento en base de datos
- Sprint 4: Investigación sobre paralelización del sistema biométrico en datos de la forma de andar
  - Preparación del sistema biométrico
  - Investigación sobre las alternativas de paralelización

En la Figura 3.1 se puede observar el diagrama de Gantt correspondiente a la planificación inicial, mostrando todas las actividades planificadas en el tiempo.

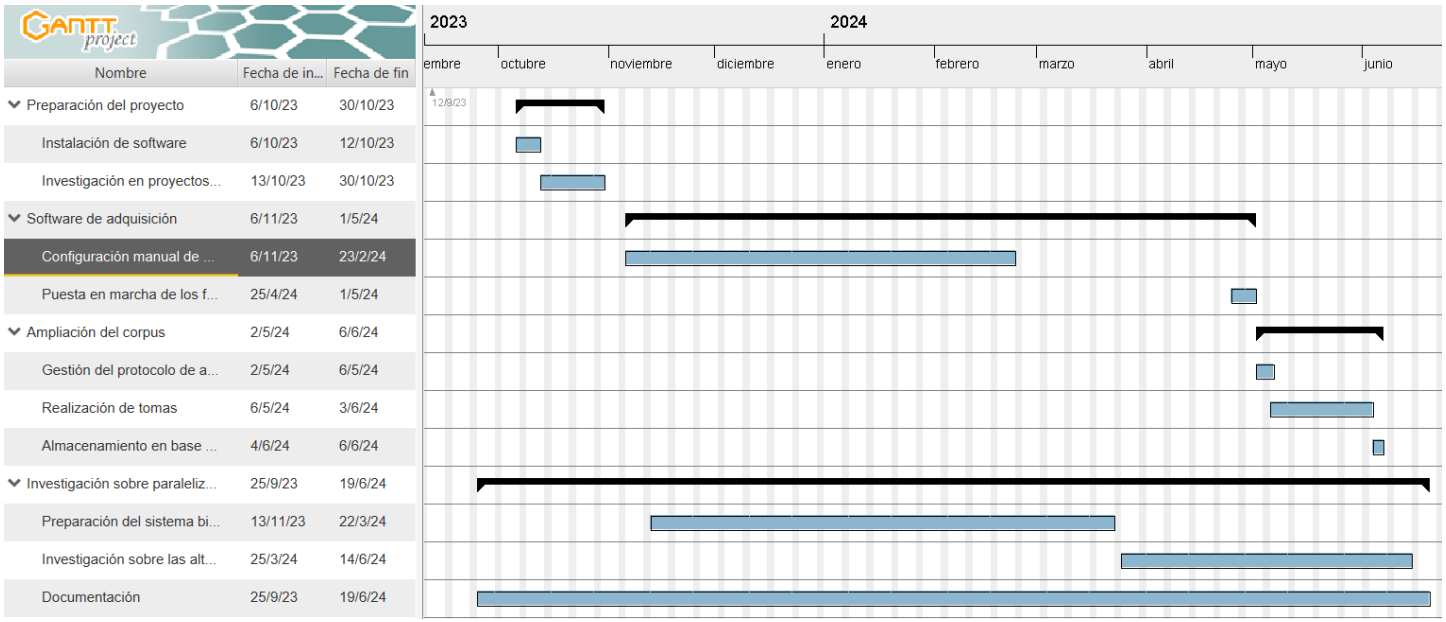

Figura 3.1: Diagrama de Gantt del proyecto inicial

3.3. Riesgos del proyecto inicial

El proyecto en cuestión tenía una serie de riesgos que conviene describir y detallar, con su probabilidad de ocurrencia y su relevancia en caso de que ocurran. Se abarca desde riesgos propios en la elaboración de software hasta riesgos particulares por la propia agenda del alumno que realiza el trabajo. Cada uno de los riesgos tiene asociada una posible respuesta en caso de que ocurran.

En la Tabla 3.1 aparecen enunciados los riesgos junto a su probabilidad e importancia. La probabilidad viene clasificada como "Low - Moderate - Significant - High" de menor a mayor probabilidad de ocurrencia. Por otra parte, el impacto viene clasificado como "Low - Moderate - Significant - High" de menor a mayor impacto. Dentro de esas categorías, también se les da una puntuación numérica para posteriormente calcular la matriz de probabilidad e impacto de manera cuantitativa [18].

Id	**Descripción**	**Probabilidad**	**Impacto**
01	Nuevos requisitos o modificaciones en los mismos. Tiempo y esfuerzo de más	Moderate 0.5	Moderate 0.2
02	A causa de una planificación poco realista, las tareas de desarrollo y documentación duran más de lo esperado	Significant 0.7	Low 0.1
03	Por falta de compatibilidad, posibles interferencias al capturar datos de ECG	Moderate 0.5	High 0.4
04	Por despiste sobre los detalles concretos se producen errores inesperados en la paralelización y pruebas del código	High 0.8	Low 0.1
05	Aparecen tareas complicadas, teniendo alguna sección que no se consigue implementar	Low 0.3	High 0.8
06	Por fallos de la computadora, pérdida de información y de tiempo	Moderate 0.5	High 0.4
07	Interrupción del funcionamiento de la API de Withings	Moderate 0.5	Moderate 0.2
08	Cambios en el orden de tareas	Moderate 0.5	Low 0.1

Tabla 3.1: Descripción de riesgos

En la Tabla 3.2 aparece la matriz de probabilidad e impacto, detallando la relación entre la probabilidad y el impacto de que ocurran los riesgos de un proyecto. La gama de colores representa la gravedad del riesgo en conjunto, señalando en rojo las amenazas más importantes, en amarillo las intermedias y en verde las amenazas más leves.

Probabilidad	Amenazas/Oportunidades				
0.90	0.05	0.09	0.18	0.36	0.72
0.70	0.04	0.07	0.14	0.28	0.56
0.50	0.03	0.05	0.10	0.20	0.40
0.30	0.02	0.03	0.06	0.12	0.24
0.10	0.01	0.01	0.02	0.04	0.08
Impacto	0.05	0.10	0.20	0.40	0.80

Tabla 3.2: Matriz de probabilidad e impacto [19]

A continuación, en la Tabla 3.3, se pueden apreciar los riesgos del proyecto con su calificación según la amenaza que presentan para el desarrollo del proyecto según la cronología esperada.

Riesgo	Probabilidad	Impacto	Calificación
1	0.5	0.2	0.1
2	0.7	0.1	0.07
3	0.5	0.4	0.2
4	0.8	0.1	0.08
5	0.3	0.8	0.24
6	0.5	0.4	0.2
7	0.5	0.2	0.1
8	0.5	0.1	0.05

Tabla 3.3: Puntuación de los riesgos [19]

Respuesta a cada uno de los riesgos:

- Id 01: Tratar de reducir las funcionalidades del código y los experimentos a lo especificado en las reuniones iniciales.
- Id 02: Dedicar un tiempo razonable a la planificación del proyecto para ser lo más preciso posible, además de tener una visión general del trabajo que se debe desarrollar y la disponibilidad a lo largo de los meses.

- Id 03: Localizar bases de datos alternativas o proyectos anteriores para tener datos sobre los que trabajar.

- Id 04: Realizar comprobaciones y depurar el código.

- Id 05: Tratar de simplificar el problema y formarse sobre el tema. En caso de no haber solución, propuesta de alternativas y nuevos medios de investigación.

- Id 06: Realización de copias de seguridad periódicas, trabajar en entornos de texto online y herramientas de control de versiones.

- Id 07: Tener localizadas APIs operativas, así como la forma de ponerse en contacto con los mantenedores del sistema.

- Id 08: Realizar la planificación con detenimiento, priorizando las tareas por importancia y dificultad. Garantizar el desarrollo de tareas e hitos, dejando un seguimiento escrito.

El riesgo que ha tenido unas consecuencias graves ha sido el número 5, ya que a pesar de ser poco probable que ocurriera, su impacto es total ya que ha implicado cambiar el tema de investigación.

De todos estos riesgos ocurrió el número 05. Hubo problemas con el software de paralelización desarrollado en un TFG anterior. La probabilidad de que ocurriera era baja, ya que funcionó sin problema en la realización del TFG indicado [9]. Sin embargo, el error ocurrió y el software dio problemas. Se aplicaron las medidas de contingencia previstas. Primero se analizó el software junto con los tutores y expertos en R para detectar el error. Tras múltiples pruebas, se comprobó que la biblioteca de extracción de las características FMM funcionaba correctamente. Sin embargo, en cuanto se intentaba paralelizar, daba error. Siguiendo las medidas de contingencia, se buscó el asesoramiento de expertos en paralelización con R. Ninguna de estas alternativas logró solucionar el problema. Intentar la ejecución sin paralilizar el problema era inviable por cuestiones de tiempo requerido por la biblioteca de R para la extracción de los parámetros FMM, que tiene un coste computacional muy alto. Los detalles de esta fase del trabajo se exponen en la Sección 4

Como la ocurrencia del riesgo impidió continuar con lo inicialmente planificado, se optó por aplicar una segunda medida de contingencia: replanificar el objetivo del TFG.

3.4. Planificación inicial del proyecto definitivo

Al cambiar el tema del proyecto debido a la ocurrencia de un riesgo catastrófico, se ha realizado una nueva planificación con fecha 9 de abril de 2024. Algunas actividades referentes a la adquisición de datos que se realizaron en el proyecto inicial sí que sirven para el nuevo proyecto.

En la Figura 3.2 se puede observar el diagrama de Gantt correspondiente al proyecto definitivo, mostrando todas las actividades planificadas en el tiempo. La parte de "Proyecto Inicial" en el diagrama de Gantt no es una planificación realmente, ya que pertenece a la asignación del proyecto inicial.

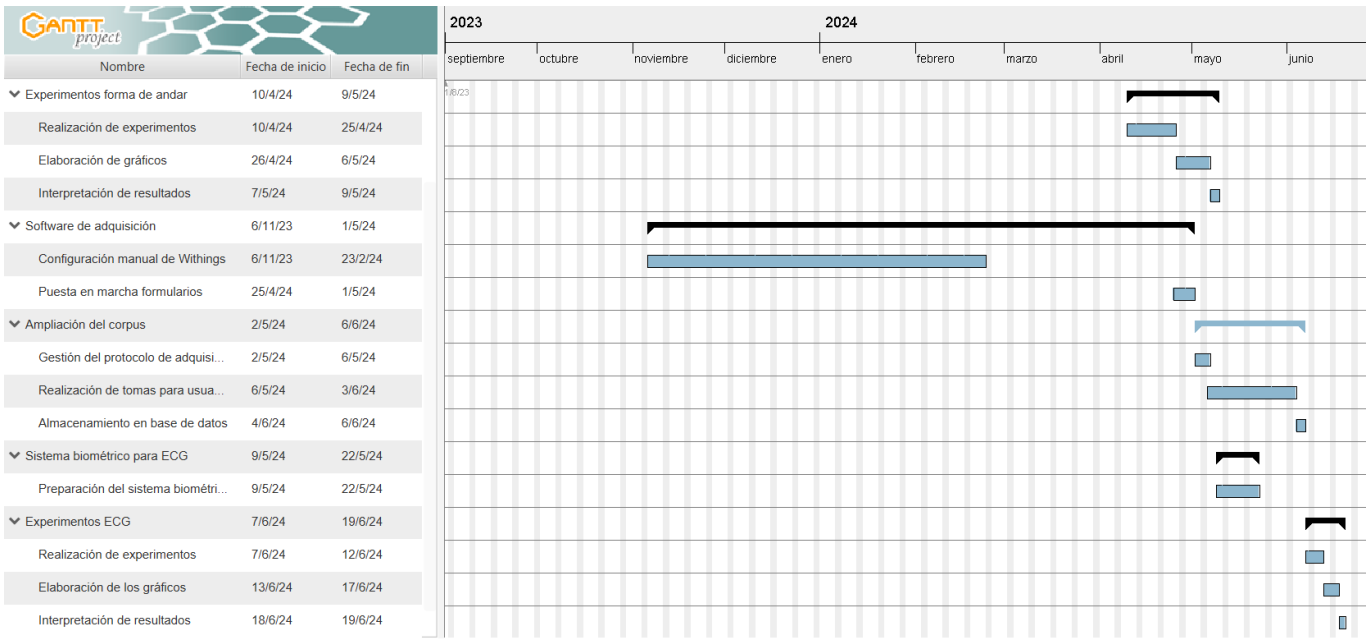

Figura 3.2: Diagrama de Gantt del proyecto definitivo

3.5. Riesgos del proyecto definitivo

El nuevo proyecto también tiene una serie de riesgos que conviene describir y detallar su probabilidad de ocurrencia y su relevancia en caso de que ocurran. Cada uno de los riesgos tiene asociada una posible respuesta. Aunque los riesgos sean similares, las probabilidades de ocurrencia y el impacto cambian ligeramente debido a la experiencia previa con el proyecto inicial.

En la Tabla 3.4 aparecen enunciados los riesgos junto a su probabilidad e importancia. Dentro de las categorías que se utilizaron en los riesgos iniciales, también se les da una puntuación numérica para posteriormente calcular la matriz de probabilidad e impacto de manera cuantitativa. Al haber menos tiempo disponible hasta la fecha de entrega, los retrasos tienen un impacto más elevado.

Id	**Descripción**	**Probabilidad**	**Impacto**
01	Nuevos requisitos o modificaciones en los mismos. Tiempo y esfuerzo de más	Low 0.3	Moderate 0.2
02	A causa de una planificación poco realista, las tareas de desarrollo y documentación duran más de lo esperado	Moderate 0.5	High 0.4
03	Por falta de compatibilidad, posibles interferencias al capturar datos de ECG	Low 0.3	High 0.4
04	Por despiste sobre los detalles concretos se producen errores inesperados en los experimentos y pruebas del código	High 0.8	Low 0.1
05	Aparecen tareas complicadas, teniendo alguna sección que no se consigue implementar	Low 0.3	High 0.8
06	Por fallos de la computadora, pérdida de información y de tiempo	Moderate 0.5	High 0.4
07	Interrupción del funcionamiento de la API de Withings	Moderate 0.5	Moderate 0.2
08	Cambios en el orden de tareas	Low 0.3	Low 0.1

Tabla 3.4: Descripción de riesgos

A continuación, en la Tabla 3.5, se pueden apreciar los riesgos del proyecto definitivo con su calificación dependiendo de la amenaza que presentan para el

desarrollo del proyecto según la cronología esperada.

Riesgo	Probabilidad	Impacto	Calificación
1	0.5	0.2	0.06
2	0.5	0.4	0.2
3	0.3	0.4	0.12
4	0.8	0.1	0.08
5	0.3	0.8	0.24
6	0.5	0.4	0.2
7	0.5	0.2	0.1
8	0.5	0.1	0.03

Tabla 3.5: Puntuación de los riesgos [19]

Respuesta a cada uno de los riesgos:

- Id 01: Tratar de reducir las funcionalidades del código y los experimentos a lo especificado en las reuniones iniciales.
- Id 02: Dedicar un tiempo razonable a la planificación del proyecto para ser lo más preciso posible, además de tener una visión general del trabajo que se debe desarrollar y la disponibilidad a lo largo de los meses.
- Id 03: Localizar bases de datos alternativas o proyectos anteriores para tener datos sobre los que trabajar.
- Id 04: Realizar comprobaciones sobre el caso de prueba y depuración de los experimentos.
- Id 05: Tratar de simplificar el problema y formarse sobre el tema. En caso de no haber solución, propuesta de alternativas y nuevos medios de investigación.
- Id 06: Realización de copias de seguridad periódicas, trabajar en entornos de texto online y herramientas de control de versiones.
- Id 07: Tener localizadas APIs operativas, así como la forma de ponerse en contacto con los mantenedores del sistema.
- Id 08: Realizar la planificación con detenimiento, priorizando las tareas por importancia y dificultad. Garantizar el desarrollo de tareas e hitos, dejando un seguimiento escrito.

3.6. Planificación final del proyecto definitivo

A continuación, se presentan las distintas fases reales del proyecto, con sus actividades concretas a realizar después de haber modificado los objetivos:

- Sprint 0: Proyecto inicial
  - Planificación del proyecto
  - Instalación del software para el desarrollo
  - Puesta en marcha del software
- Sprint 1: Experimentos y resultados con forma de andar ("gait")
  - Realización de experimentos
  - Elaboración de los gráficos de errores
  - Interpretación de resultados
- Sprint 2: Software de adquisición del ECG (análisis-diseño-pruebas)
  - Configuración manual de Withings
  - Puesta en marcha y actualización del software desarrollado en [2]
- Sprint 3: Ampliación del corpus
  - Gestión del protocolo de adquisición
  - Realización de tomas para nuevos usuarios
  - Almacenamiento en base de datos
- Sprint 4: Sistema biométrico para ECG (adaptación a Python) (análisis-diseño-pruebas)
  - Preparación del sistema biométrico para ECG
- Sprint 5: Experimentos y resultados con ECG
  - Realización de experimentos
  - Elaboración de los gráficos de errores
  - Interpretación de resultados

Fase	Actividad	Tiempo dedicado (h)	Tiempo fase (h)
Sprint 0 Proyecto inicial	Planificación del proyecto	15	53
	Instalación del software para el desarrollo	3	
	Puesta en marcha del software	35	
Sprint 1 Experimentos y resultados con forma de andar	Realización de experimentos	20	36
	Elaboración de los gráficos	12	
	Interpretación de resultados	4	
Sprint 2 Software de adquisición	Configuración manual de Withings	25	27
	Puesta en marcha de los formularios	12	
Sprint 3 Ampliación del corpus	Gestión del protocolo de adquisición	4	49
	Realización de tomas para nuevos usuarios	38	
	Almacenamiento en base de datos	7	
Sprint 4 Sistema biométrico para ECG	Preparación del sistema biométrico para ECG	35	35
Sprint 5 Experimentos y resultados con ECG	Realización de experimentos	20	34
	Elaboración de los gráficos	10	
	Interpretación de resultados	4	
Documentación	Elaboración de la memoria	70	70
Total			314

Tabla 3.6: Tiempo invertido por actividades

En la Tabla 3.6 se adjuntan las diferentes actividades del proyecto y su duración en horas. También se incluye la elaboración de la memoria, que se realiza de manera gradual a lo largo del proyecto. La fase correspondiente al proyecto inicial es la que incluye las actividades y el tiempo dedicado al proyecto antes de cambiar el tema.

En la Figura 3.3 se puede observar el diagrama de Gantt correspondiente al proyecto definitivo, mostrando en color rojo las actividades que se han retrasado en el tiempo, mientras que las actividades en color verde se han adelantado. Para representarlo, se ha utilizado la herramienta de líneas de base de GanttProject.

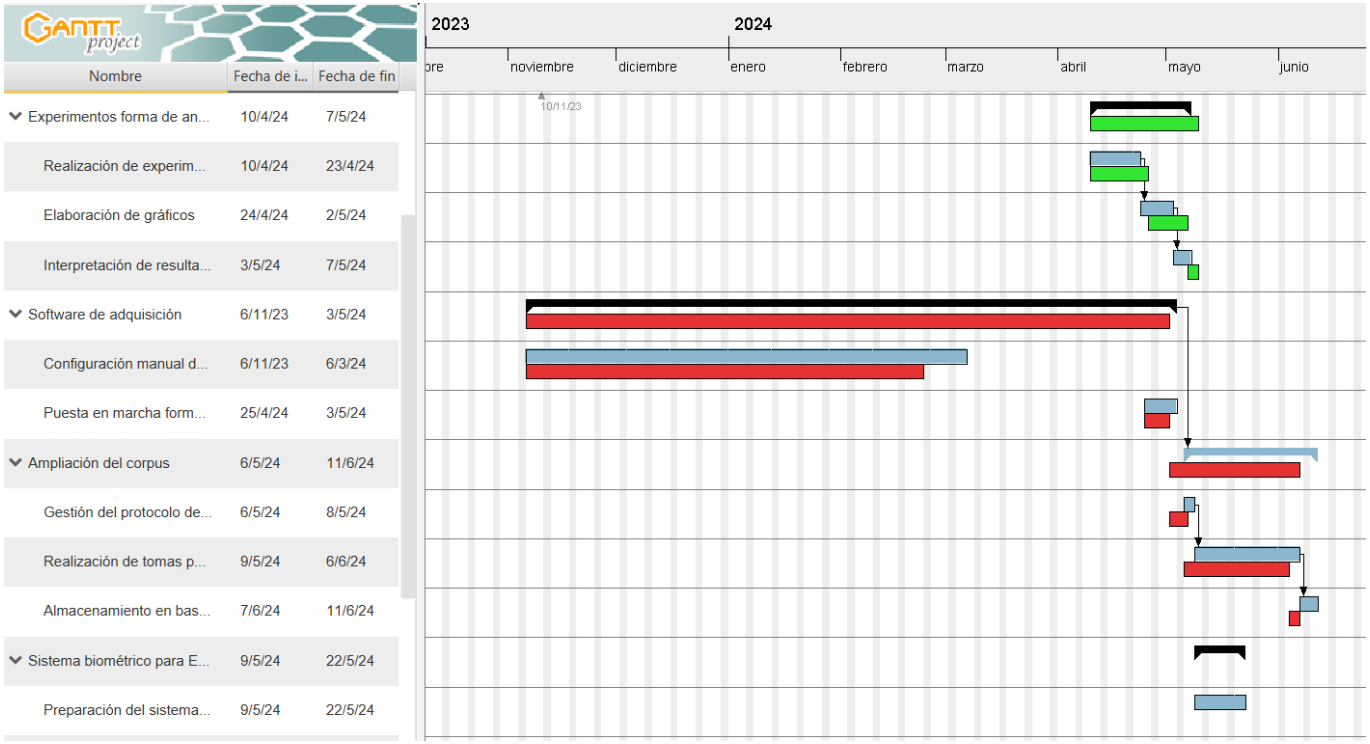

Figura 3.3: Diagrama de Gantt real del proyecto definitivo

3.7. Costes

Al tratarse de un proyecto informático, resulta importante realizar un análisis de los costes del mismo. Se hará de manera ficticia, ya que el estudiante no recibe una dotación económica por la realización del trabajo.

Para ello, se comienza buscando el sueldo medio de un Ingeniero Informático recién titulado en España, que en el año 2024 se sitúa en 28000 euros anuales brutos [20], lo que supone 13.46 euros brutos la hora. Las horas invertidas en el proyecto son 314, por lo que salen 4226.44 euros de coste en cuanto al personal involucrado, con la Seguridad Social incluida.

Para contabilizar los gastos de equipamiento, se tiene en cuenta que el lugar de trabajo es el domicilio del alumno. En este caso la electricidad tiene un coste medio de 0,233291 euros/kWh según la compañía Iberdrola [21]. Los dispositivos de los que se hablará a continuación tienen un consumo conjunto de 90Wh, lo cual para 314 horas se traduce en 28.26 kWh. La conexión a Internet también resulta ser clave para el desarrollo del proyecto, y tiene un coste de 27 euros mensuales con la compañía Vodafone [22].

El ordenador portátil que se utiliza para el trabajo es del alumno, con un coste de 900 euros. Al utilizarse durante 9 meses para este proyecto y suponiendo que la vida útil de un portátil es de aproximadamente 4 años. su coste amortizado es de 168.75 euros. También se usa un monitor de trabajo con coste de 144 euros, vida útil de 6 años y por tanto un coste amortizado de 18.75 euros. Por último, se ha utilizado un dispositivo Withings Move ECG cuyo precio es de 99.99 euros. La vida útil de este dispositivo se valora en 4 años, por lo que su coste amortizado es de 18.72 euros.

En cuanto a las licencias, todo el software utilizado es libre, con la salvedad de Windows 10 (incluido en el precio del ordenador portátil) y Microsoft Office (aportado por la Universidad de Valladolid).

En la Tabla 3.7 se muestra un resumen de todos estos costes, con la suma del presupuesto total.

Recurso	**Tiempo/Cantidad**	**Precio**	**Coste total**
Humano	314 horas	13.46 euros/hora	4226.44 euros
Electricidad	27 kWh	0.233291 euros/kWh	6.59 euros
Internet	9 meses	25 euros al mes	225 euros
Portátil	9 meses	900 euros	168.75 euros
Monitor	9 meses	144 euros	18.75 euros
Reloj Withings	9 meses	99.99 euros	18.72 euros
Total			4664.25 euros

Tabla 3.7: Presupuesto final para el proyecto

3.8. Herramientas

Las herramientas y lenguajes de programación utilizados en este trabajo se muestran a continuación:

- API Withings: se trata de la interfaz que permite obtener todas las tomas realizadas con el dispositivo "Withings". Es necesario crear una cuenta de usuario para acceder a este contenido.

- Visual Studio Code: ha pasado a ser uno de los editores de código más utilizados en la actualidad ya que es gratuito, de código abierto y multiplataforma. Permite crear código para prácticamente cualquier lenguaje de programación. Entre otras muchas opciones, permite depurar aplicaciones, gestionar los proyectos, realizar un control de versiones e instalación de extensiones. Para este trabajo se ha utilizado la extensión "ThunderClient", que permite hacer peticiones a la API de "Withings" [23].

- R: se trata de un lenguaje de programación de código abierto e interpretado. Permite crear programas, aplicaciones, mediciones estadísticas, gran variedad de gráficos, modelos lineales, no lineales, series temporales, problemas de clasificación y muchas más cosas. Aunque R tiene su propio equipo de desarrollo, permite que los usuarios puedan crear librerías que formen parte del proyecto R y puedan ser utilizadas por otros usuarios [24].

- RStudio: es un entorno de desarrollo que está diseñado para trabajar con el lenguaje de programación R. En concreto, se usa para este proycto la última versión estable disponible, correspondiente a la de diciembre del año 2022. Algunas de sus prestaciones son el editor de código fuente, que permite ejecutarlo desde ahí. También posee una herramienta de depuración interactiva, consola, visor de datos y distintos espacios de trabajo, entre otras funcionalidades [25].

- Python: es un lenguaje de programación utilizado en la actualidad principalmente para aplicaciones web, desarrollo de software, ciencia de datos y machine learning. Destaca por su eficiencia, facilidad de uso, por ser multiplataforma, gratuito y fácilmente integrable en otros sistemas. Además, posee un gran número de bibliotecas útiles que se utilizan en el proyecto como "Matplotlib", "Pandas" y "NumPy" [26].

- Anaconda Navigator: interfaz gráfica de usuario que permite trabajar con paquetes y entornos sin tener que escribir comandos conda en una ventana de terminal, tan solo hay que instalarlos. [27].

- Jupyter Notebook: aplicación web que permite crear y compartir documentos computacionales. Es fácil de utilizar de manera optimizada. Soporta muchos lenguajes de programación distintos [28].

- GanttProject: es una herramienta para la gestión de proyectos software. Permite realizar diagramas de Gantt para la gestión del tiempo y actividades del proyecto. Requiere de poca configuración y su uso es muy intuitivo. Además, permite exportar los diagramas a otras plataformas [29].

- Overleaf: se trata de la herramienta que se ha utilizado para la escritura de la memoria. Está pensada para la creación colaborativa de texto principalmente científico y de investigación. En este caso me permite la edición en código LaTeX de manera online con guardado automático, pudiendo añadir todas las fórmulas, imágenes y referencias de manera rápida y sencilla [30].

- Excel: es un programa Microsoft que permite editar hojas de cálculo y está disponible en los sistemas operativos habituales. También se pueden hacer cálculos, gráficas, tablas y posee un lenguaje de programación macro para aplicaciones. Otra particularidad es que se pueden realizar cálculos matemáticos mediante fórmulas y operadores matemáticos [31].

- HTML: siglas de "Lenguaje de Marcado de Hipertexto", es el código empleado para estructurar y desplegar los contenidos de una página web. Este contenido incluye párrafos, lista con viñetas, imágenes, tablas, etc.[32].

- PHP: "Hypertext Preprocessor" es un lenguaje de código abierto útil para el desarrollo web y que permite incluirse en HTML [33].

- PhpMyAdmin: es una herramienta escrita en lenguaje PHP que nos permite manejar una base de datos MySQL a través de una página web [34].

- GitHub: se trata de un servicio basado en la nube que aloja un sistema de control de versiones. Al ser una herramienta colaborativa, se pueden realizar

cambios en proyectos compartidos y mantener un seguimiento detallado del progreso. Presenta una estructura de ramas en repositorios [35].

- WinSCP: aplicación gratuita y de código abierto que se trata de un cliente SFTP gráfico para Windows que emplea SSH. Su utilidad radica en facilitar la transferencia segura de archivos entre un sistema local y uno remoto. WinSCP se basa en la implementación del protocolo SSH de PuTTY y el protocolo FTP de FileZilla [36].

Capítulo 4

Desarrollo

4.1. Sprint 0: Proyecto inicial

Como ya ha sido comentado, el enfoque inicial del trabajo a realizar no era el finalmente planteado, pero por una serie de complicaciones insalvables en la puesta en funcionamiento del software necesario, se decidió cambiar a otro proyecto perteneciente al mismo ámbito de investigación, que es el que se documenta en los demás sprints del proyecto mostrados en este capítulo.

En esta sección se comentan las tareas desarrolladas para esa asignación de trabajo, que aunque no llegó a dar frutos sí que ocupó un periodo de tiempo relevante y proporcionó aprendizaje para el alumno.

El trabajo se centraba en el ámbito del reconocimiento biométrico, con el objetivo de realizar y estudiar una extracción de características novedosa en el ámbito del reconocimiento mediante la forma de andar y el ECG. Dado el coste computacional de la biblioteca a usar, era necesario paralelizar su ejecución. Paralelización que ya se había iniciado en un TFG anterior. Esto permitiría comparar el rendimiento de las opciones paralelizadas y sin paralelizar. También existía la posibilidad de realizar procedimientos de minería de datos y aprendizaje automático orientados a datos de ECG.

Otro objetivo planteado era añadir más usuarios a una base de datos previa de ECG. Esta fase de recogida de datos ECG con el dispositivo Withings se ralentizó ya que el servidor de Withings dio fallos durante unos días al registrar los electrocardiogramas.

4.2. Sprint 1: Experimentos y resultados con forma de andar

En esta parte se tiene como objetivo investigar en la hipótesis inicial, consistente en que para frecuencias de muestreo bajas los resultados son tan buenos como para frecuencias más altas. Al conseguir una frecuencia de muestreo menor, el coste energético y de almacenamiento disminuirá y de esta forma la batería del dispositivo tendrá mayor duración. Para tratar de demostrarlo, se analizará la dependencia entre el rendimiento y la tasa de muestreo.

Esta sección se basa en datos relacionados con la forma de andar. Estos datos se tomaron para la realización del Proyecto de Fin de Máster de Irene Salvador Ortega [1]. Para ello, se usaron dos tipos de dispositivos: "Micro" y "Moto". El primero se refiere a un dispositivo Microsoft, mientras que el segundo es un dispositivo Motorola. Los sensores empleados son acelerómetro ("ACC") y giróscopo ("GYR"). Las bases de datos y código están contenidos en el Gitlab de la Escuela de Ingeniería Informática de Valladolid [38].

El problema a tratar es la clasificación de un usuario respecto a su autenticidad, siendo un problema de clasificación binaria ya que tiene dos clases respuesta (auténtico o impostor). La tasa de error que se utiliza para comparar los resultados es la tasa de equierror (EER). Se trata del punto de una curva ROC en el cual la Tasa de Falsos Positivos es igual a la Tasa de Verdaderos Positivos. La Tasa de Falsos Positivos se conoce habitualmente como "Especificidad", mientras que la Tasa de Verdaderos Positivos es conocida como "Sensibilidad". Por su parte, la curva ROC representa la sensibilidad frente a la especificidad de una prueba de clasificación y permite evaluar la precisión de las predicciones de modelo [39].

- Especificidad: es la proporción de usuarios autentificados cuando no debían respecto al total de usuarios impostores [40].

$$\text{Especificidad} = \frac{\text{Falso Positivo}}{\text{Falso Positivo} + \text{Verdadero Negativo}}$$

- Sensibilidad: es la proporción de usuarios autentificados de manera correcta respecto al total de usuarios auténticos [40].

$$\text{Sensibilidad} = \frac{\text{Verdadero Positivo}}{\text{Falso Negativo} + \text{Verdadero Positivo}}$$

Se puede observar en la Figura 4.1 una representación gráfica del EER a partir de la curva ROC. En el eje de abscisas se representa la Tasa de Falsos Positivos

(especificidad), mientras que en el eje de ordenadas se representa la Tasa de Verdaderos Positivos (sensibilidad). El EER es mejor cuanto menor es su valor.

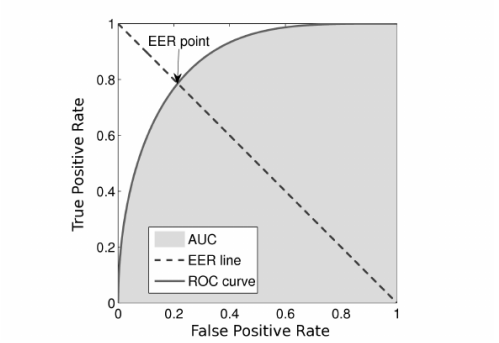

Figura 4.1: Representación del EER a partir de la curva ROC [41]

Las técnicas de clasificación ue se van a usar en el estudio por ser las que mejores resultados han dado en trabajos previos son:

- Random Forest: Traducido como bosque aleatorio, es una técnica de aprendizaje supervisado aplicada en problemas de regresión y clasificación. En esencia es una combinación de árboles de decisión que mediante técnicas de remuestreo y aleatorización en las variables evita el sobreajuste.

  El método Random Forest es un caso particular del método de bagging, generándose muestras patrón bootstrap y desarrollándose al máximo los árboles asociados para unas pocas variables elegidas aleatoriamente. En cada nodo se utilizan $\sqrt{p}$ variables de las p variables originales. Se guardan los árboles resultantes y las nuevas observaciones se asignan a clases usando "votación de la mayoría" [1].

- Support Vector Machines: Este método de aprendizaje supervisado se centra en las observaciones fronterizas entre grupos para realizar una clasificación [1]. En el caso separable, esa frontera es un hiperplano separante que maximiza la distancia entre las observaciones más cercanas de las clases. Cuando las observaciones no son separables linealmente, se pueden usar kernels (núcleos) para llevar los datos a un espacio de dimensión mayor donde sí sean separables. Los dos tipos de núcleos que más se usan son el polinómico y el de bases radiales, cuyos parámetros se definen por validación cruzada. En este caso se utiliza SVM radial.

4.2.1. Análisis

Se presenta una serie de requisitos funcionales y requisitos no funcionales.

Requisitos funcionales

- Clasificar datos. El sistema debe realizar una clasificación de los datos utilizando los métodos Support Vector Machines y Random Forest.
- Almacenar datos. El sistema debe almacenar los resultados en el directorio correspondiente a su frecuencia de muestreo.
- Mostrar resultados. El sistema debe mostrar unos gráficos de barras que comparen las tasas EER con las distintas frecuencias de muestreo.

Requisitos no funcionales

- El sistema debe partir de los programas desarrollados en el TFM de Irene Salvador Ortega [1].
- El sistema debe utilizar las bases de datos UVA y ZJU.

En la Figura 4.2 se muestra el sistema biométrico. Una vez capturados los datos crudos, estos son preprocesados. En esta etapa, la señal es limpiada, eliminando las partes con ruido [42].

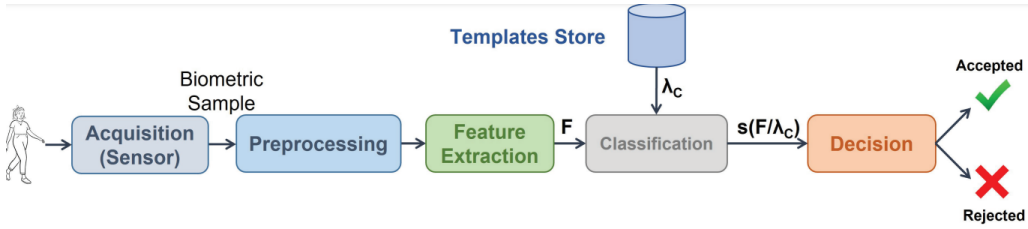

Figura 4.2: Sistema biométrico de la forma de andar

En la señal original, el tiempo entre muestras no es constante, ya que ha sido capturada con un dispositivo real, no dedicado exclusivamente a la tarea objetivo de este TFG. Por eso, tras la limpieza de la señal, esta se re-muestrea a un periodo constante. Es conveniente recordar en este punto que el periodo de muestreo es el inverso de la frecuencia de muestreo. Como queremos comprobar la relación entre

frecuencia de muestreo y rendimiento del sistema, el valor del periodo de muestreo tendrá distintos valores, como se verá más adelante.

Con la señal limpia y muestreada con un periodo entre muestras constante, se divide en ciclos, y estos se agrupan en ventanas. Una ventana es un conjunto de ciclos consecutivos. La ventana es la mínima cantidad de información que usaremos para el reconocimiento del usuario, ya que un ciclo no contiene la cantidad de información necesaria para ello; por eso se usa un conjunto de ellos.

De cada ventana se obtiene su vector de características. Las características extraídas son las siguientes: media, mediana, máximo valor, mínimo valor, desviación estándar, máximo rango (máximo-mínimo), kurtosis, percentil 25, percentil 75, coeficiente de asimetría y máximo valor de autocorrelación.

Los sensores obtienen datos en tres dimensiones: X, Y y Z. La extracción de características mencionada se realiza para cada coordenada. En la bibliografía no se suelen usar las coordenadas de los sensores para el reconocimiento, si no que se suelen fusionar. Una de las alternativas es hacerlo a nivel de señal, es decir, se fusionan previamente las tres mediante su módulo ($\text{Mod} = \sqrt{X^2 + Y^2 + Z^2}$), y se extraen las características de esta señal fusionada. La otra alternativa es hacerlo a nivel de características, es decir, crear un único vector juntando las características extraídas de X, Y y Z. Para simplificar terminología, vamos a llamar a la primera "módulo" y a la segunda "XYZ".

La siguiente y última etapa es la de clasificación. El resultado en nuestros experimentos será un puntaje o valor de salida del clasificador (score), que de alguna forma representa la probabilidad de pertenencia de la ventana al usuario. Será el valor usado para calcular las medidas de error mencionadas.

Se entrena un clasificador para cada usuario de la base de datos. Para ello se necesitan muestras auténticas (pertenecientes al usuario a reconocer) y muestras de impostores (cualquier individuo que no sea el usuario). Para las primeras se usa, siguiendo lo típico en biometría, la muestra uno de la sesión uno de la base de datos. Para las segundas, el resto de usuarios de la base de datos se dividen en dos grupos; el primero se usará como ejemplos de "impostores" en entrenamiento, mientras las muestras del segundo se usarán para las pruebas. Es muy importante que las muestras que se usen para entrenar, sean completamente diferentes de las usadas para probar, por eso los conjuntos en que se dividen el "resto" de individuos de la base de datos son disjuntos.

Para pruebas auténticas (pruebas del usuario no vistas en entrenamiento) se usan las dos muestras de las sesión 2 del usuario.

Con las pruebas de impostores se obtiene la tasa de falsas aceptaciones, mientras

que con las pruebas auténticas se obtiene la tasa de falsos rechazos.

A continuación, se obtienen los EER para las distintas frecuencias de muestreo, permitiendo así implementar los gráficos de barras del EER para cada frecuencia de muestreo.

4.2.2. Diseño

Los datos con los que se trabaja tienen una serie de características. La primera es que la captura de datos andando tiene una duración de 5 minutos. Se anota la mano en la cual se pone el dispositivo y su orientación. Otra característica fundamental es que las capturas se dividen en dos sesiones separadas entre sí un mínimo de dos semanas. Esto permite estudiar el patrón biométrico a lo largo del tiempo. Además, cada sesión se divide a su vez en 6 tomas (3 con cada dispositivo). De esas 3 por dispositivo, 2 son con la mano portadora y la tercera con la mano no portadora [1].

Otro tipo de información que se almacena para cada usuario que participa en el estudio son los metadatos. Concretamente, se determina el identificador de usuario, el sexo, la edad, la mano dominante, la mano portadora y la cantidad de tomas.

Los demás datos que se obtienen son los que realmente se utilizarán para realizar los cálculos del sistema biométrico. Uno es el denominado "Timestamp", que almacena una referencia del tiempo que ha pasado entre una medición y la inmediatamente anterior. Los otros datos son las 3 coordenadas X, Y, Z, que registran la aceleración del sensor hacia izquierda/derecha, adelante/atrás y arriba/abajo respectivamente [1].

Dentro del proyecto GitLab, cuya estructura se puede observar en la Figura 4.3, está el directorio "data", que almacena las bases de datos separadas en tres subdirectorios: UVA, WISDM y ZJU. La primera se corresponde con los datos propios, mientras que las otras dos son bases de datos públicas. De estas dos últimas solamente se trabajará con ZJU, ya que es la única que tiene dos sesiones, y los experimentos se realizan con pruebas "cross-session". La base de datos de UVA es la que se ha descrito al inicio de esta sección, con 38 usuarios. Los datos de ZJU se midieron con un único dispositivo (Microsoft) y sensor de aceleración. Los datos de ZJU pertenecen a 153 usuarios con mediciones realizadas en 2 sesiones diferentes.

Name	Last commit
Analisis1_Graficos	Analysis 1 based on UVA-SmartGait database
Analisis1_Resultados	Analysis 1 based on UVA-SmartGait database
Analisis2_Graficos	change of text position in analysis 2 graphs
Analisis2_Resultados	add ZJU DB with same device scenery to 2nd analysis
code	change of text position in analysis 2 graphs
data	extraction of features from ZJU database
tablas_caract/datos_interpolados	extraction of features from ZJU database
.gitignore	extraction of features from ZJU database
README.md	Update README.md
requirements.txt	Fichero de dependencias (Python 3.10.12)

Figura 4.3: Estructura del proyecto en Git

En la sección del código se encuentran separados los archivos Python dependiendo de si trabajan con la base de datos UVA (Figura 4.4) o ZJU (Figura 4.5). Dentro de cada una hay cinco archivos, cada uno de ellos realiza una tarea diferente pero es lo mismo para las dos bases de datos. Concretamente, estos archivos dividen la señal en ventanas, extraen características de las coordenadas (X, Y, Z y módulo) en los dispositivos (Micro y Moto) y en los sensores (ACC y GYR), para finalmente limpiar el ruido en la señal. También en esta sección aparece el directorio "Analisis2_ALT0", dentro del cual hay tres archivos Python 4.6. De ellos tan solo se usará el primero y el tercero; el primero es para datos de UVA y el otro para ZJU. Este último programa utiliza modelos Support Vector Machines y Random Forest para clasificar los datos.

1_0_ExtraccionCaract_DT_Modulo_UVA.ipynb
1_0_ExtraccionCaract_DT_XYZ_UVA.ipynb
1_1_ExtraccionCaract_DT_AllUsers_UVA.ipynb
1_2_ExtraccionTramosRuido_UVA.ipynb
1_3_TablasCaract_IndicandoRuido_UVA.ipynb

Figura 4.4: Archivos Python UVA

1_0_ExtraccionCaract_DT_Modulo_ZJU.ipynb
1_0_ExtraccionCaract_DT_XYZ_ZJU.ipynb
1_1_ExtraccionCaract_DT_AllUsers_ZJU.ipynb
1_2_ExtraccionTramosRuido_ZJU.ipynb
1_3_TablasCaract_IndicandoRuido_ZJU.ipynb

Figura 4.5: Archivos Python ZJU

Figura 4.6: Archivos Python Analisis2_ALT0

En el interior del proyecto también aparece el directorio "tablas_caract/datos_interpolados", ahí será donde se almacenen los resultados iniciales que se vayan obteniendo. El directorio "Analisis2_Resultados" contiene los resultados referentes a las tasas de error.

Los resultados que da la ejecución de los archivos Python "1_0_ExtraccionCaract_DT_Modulo", etc., se ha tenido que almacenar en subdirectorios dentro del directorio UVA o ZJU respectivamente de "tablas_caract". Estos subdirectorios se llaman F8, F10, etc., en referencia a las frecuencias de muestreo probadas.

Durante la ejecución del archivo Python "2_ALT0_Modelo_EstComparativo" se ha tenido que instalar el paquete "bob.measure" en el entorno Conda introduciendo el comando "conda install bob.measure" en el terminal de Anaconda. El modelo se ejecutaba utilizando solamente el acelerómetro como sensor, por lo que se ha añadido la ejecución del modelo con el giroscopio, cambiando el parámetro sensor a "GYR".

Para almacenar los resultados del código de "2_ALT0_Modelo_EstComparativo" y "2_ALT0_ZJU_Modelo_EstComparativo", se crean subdirectorios dentro de "Analisis2_Resultados/EstComparativo". Estos se llaman "ALT0_F8", "ALT0_F10", etc. Ahí se almacenan los resultados tanto para la base de datos UVA como ZJU. En cuanto al código "2_ALT0_ZJU_Modelo_EstComparativo", se realiza lo mismo para la base de datos ZJU, pero sin añadir el análisis para giroscopio.

A continuación, el propósito es crear gráficos de barras a partir de la información obtenida tras ejecutar los archivos correspondientes. Estos gráficos comparan las tasas de error para distintas frecuencias de muestreo.

Para organizar la información, se ha creado un "csv" que almacena los errores medios. En las columnas están las frecuencias de muestreo y en las filas el dispositivo, sensor y características. El procesamiento para obtener los gráficos deseados se realiza en Python. En primer lugar, se convierte el error en %, es decir, se multiplica por 100. Después, se separan los datos según la base de datos a la que pertenecen utilizando la librería "Pandas". El siguiente paso es crear una nueva fila con el error medio para

cada frecuencia de muestreo y base de datos. El último paso es asegurarse de que los valores son numéricos y realizar los gráficos utilizando la librería "matplotlib.pyplot".

En cada uno de los programas Python para cada base de datos hay que modificar el parámetro "frecuencia_muestreo" dentro de las funciones correspondientes. Tan solo aparece en los programas "1_0_ExtraccionCaract_DT_Modulo", "1_0_ExtraccionCaract_DT_XYZ" y "1_2_ExtraccionTramosRuido", concretamente en las funciones "calcula_desfase", las de ejecución "feature_extraction" y "detectar_ruido". La frecuencia de muestreo original de la base de datos propia es de 12Hz, y las pruebas inicialmente van a incluir valores de 8Hz, 10Hz, 14Hz, 16Hz y 18Hz. La frecuencia de muestreo es igual a la inversa del periodo de muestreo, por lo que la frecuencia de muestreo por defecto de 12Hz es igual a un periodo de muestreo de 0.08333 segundos, lo que a su vez son 83,33 milisegundos. Para la base de datos ZJU también se probarán frecuencias de muestreo de 50Hz y 100Hz porque la frecuencia original de esta base de datos es 100 Hz, lo que nos permite comparar con estas frecuencias de muestreo altas.

Después, en la sección del código y dentro de "Analisis2_ALT0", se utiliza el archivo Python "2_ALT0_Modelo_EstComparativo" para obtener las tasas de error.

Arquitectura del Sistema

1. Captura de Datos
   - Dispositivos: Microsoft y Motorola.
   - Sensores: Acelerómetro y Giroscopio.
   - Sesiones: Dos sesiones con varias tomas por dispositivo, sensor y base de datos.
2. Almacenamiento
   - Directorios estructurados por base de datos (UVA y ZJU).
   - Subdirectorios para almacenar resultados iniciales y tasas de error EER.
3. Procesamiento
   - Extracción de Características:
     - Limpieza del ruido.
     - División en ventanas temporales.
     - Extracción de características (coordenadas X, Y, Z, módulo).
   - Clasificación:
     - Modelos SVM base radial y Random Forest.

- Comparación de tasas de error para diferentes frecuencias de muestreo.

4. Generación de Resultados
   - Almacenamiento de resultados en subdirectorios específicos.
   - Creación de gráficos comparativos utilizando Python y librerías como Pandas y Matplotlib.

4.2.3. Pruebas

Con el fin de verificar el correcto funcionamiento del software, se realiza una serie de pruebas:

1. Prueba de extracción de características

   Objetivo: Verificar que la tabla de características se extrae correctamente.

   Procedimiento:

   - Ejecutar el script "1_0_ExtraccionCaract_DT_Modulo_UVA" y "1_0_ExtraccionCaract_DT_Modulo_ZJU" con los datos de la muestra.
   - Ejecutar el script "1_0_ExtraccionCaract_DT_XYZ_UVA" y "1_0_ExtraccionCaract_DT_XYZ_ZJU" con los datos de la muestra.
   - Revisar los archivos generados para comprobar que contienen las características extraídas de manera correcta.

   Salida esperada y obtenida: Los archivos de características contienen datos consistentes y correctos según los parámetros de dispositivo, sensor, frecuencia de muestreo, máxima autocorrelación, extensión de la ventana y solapamiento.

2. Prueba de unión de características y extracción del ruido.

   Objetivo: Verificar que las características de diferentes archivos se unen correctamente en un único fichero "csv", además del ruido.

   Procedimiento:

   - Ejecutar el script "1_1_ExtraccionCaract_DT_AllUsers_UVA" y "1_1_ExtraccionCaract_DT_AllUsers_ZJU".
   - Ejecutar los archivos Python "1_2_ExtraccionTramosRuido_UVA" y "1_2_ExtraccionTramosRuido_ZJU",

- Ejecutar los archivos Python “1_3_TablasCaract_IndicandoRuido_UVA” y “1_3_TablasCaract_IndicandoRuido_ZJU”.
- Revisar el fichero csv resultante para asegurarse de que contiene todas las características unidas correctamente con su ruido correspondiente.

Salida esperada y obtenida: El fichero “csv” contiene todas las características unidas de manera correcta y sin pérdidas de datos.

3. Prueba de clasificación y comparación de tasas de error

   Objetivo: Verificar que el sistema de clasificación y la comparación de tasas de error funcionan correctamente.

   Procedimiento:

   - Ejecutar el script “2_ALT0_Modelo_EstComparativo”.
   - Revisar los resultados generados para asegurarse de que las tasas de error EER se calculan y comparan correctamente.

   Salida esperada y obtenida: Las tasas de error EER están correctamente calculadas y los resultados deben permitir una comparación precisa entre diferentes frecuencias de muestreo.

4. Prueba de conversión y separación de datos

   Objetivo: Verificar que los errores EER se convierten a porcentaje y se separan correctamente según la base de datos.

   Procedimiento:

   - Realizar el procesamiento de datos y separación de directorios.
   - Revisar los datos para asegurarse de que los errores están convertidos y separados correctamente.

   Salida esperada y obtenida: Los datos están convertidos a porcentaje y separados de manera interpretable.

5. Prueba de creación de gráficos

   Objetivo: Verificar que los gráficos comparativos se generan correctamente.

   Procedimiento:

   - Ejecutar el código de generación de gráficos en Python.
   - Revisar los gráficos generados para asegurarse de que comparan correctamente las tasas de error EER para diferentes frecuencias de muestreo.

   Salida esperada y obtenida: Los gráficos muestran comparaciones claras y correctas de las tasas de error EER para diferentes frecuencias de muestreo.

4.2.4. Resultados

Para una mejor visualización de los resultados se ha optado por mostrarlos en gráficos de barras. Se han elegido porque el objetivo principal es comparar resultados, no mostrar valores exactos, y estos gráficos permiten hacerlo de manera rápida. Conviene recordar que la hipótesis que teníamos inicialmente era que con frecuencias de muestreo bajas los resultados son tan buenos como para frecuencias más altas.

En estas primeras Figuras 4.7, 4.8, 4.9, 4.10 se pueden observar los diagramas de barras que representan la frecuencia de muestreo en el eje de abscisas y la tasa de equierror en el eje de ordenadas para el dispositivo de la marca Microsoft y el clasificador Random Forest. Con el sensor giróscopo se aprecia de manera más visible que a frecuencias de muestreo bajas el error es ligeramente menor.

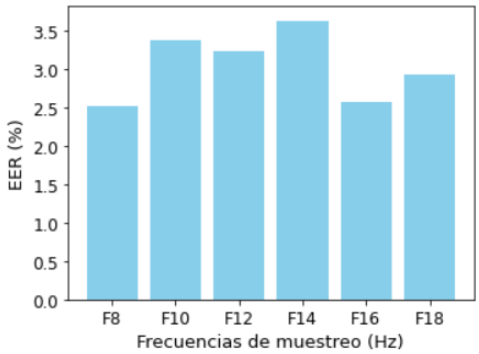

Figura 4.7: Micro ACC Modulo RF

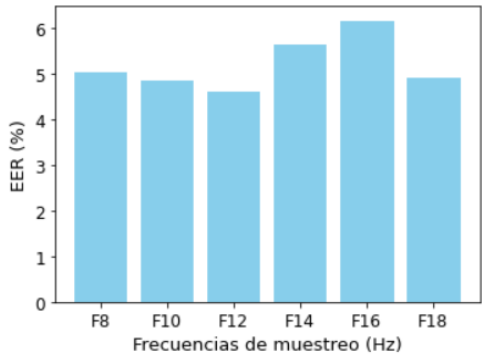

Figura 4.8: Micro ACC XYZ RF

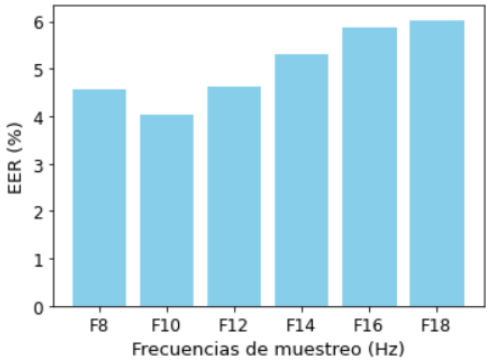

Figura 4.9: Micro GYR Modulo RF

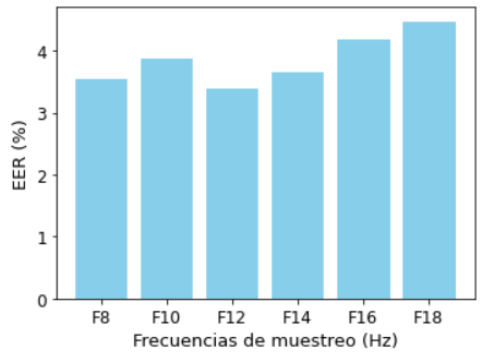

Figura 4.10: Micro GYR XYZ RF

En las siguientes Figuras 4.11, 4.12, 4.13, 4.14 se pueden observar los diagramas de barras que representan la frecuencia de muestreo en el eje X y la tasa de equierror en el eje Y para el dispositivo de la marca Motorola y el método Random Forest. En este caso, con las coordenadas XYZ parece que el error es ligeramente menor a medida que aumenta la frecuencia de muestreo. En cambio, con un sensor giróscopo y medida del módulo es claro que el error es menor con frecuencias de muestreo bajas.

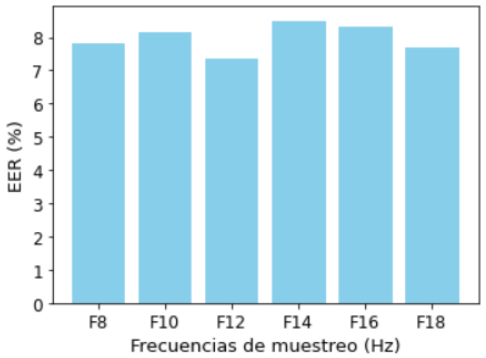

Figura 4.11: Moto ACC Modulo RF

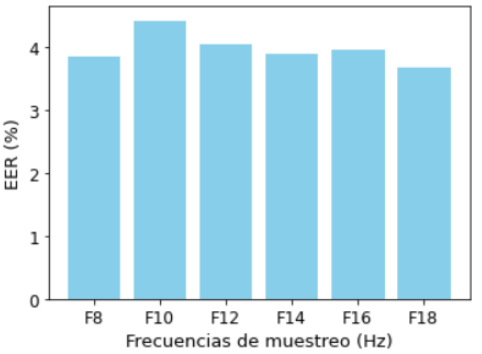

Figura 4.12: Moto ACC XYZ RF

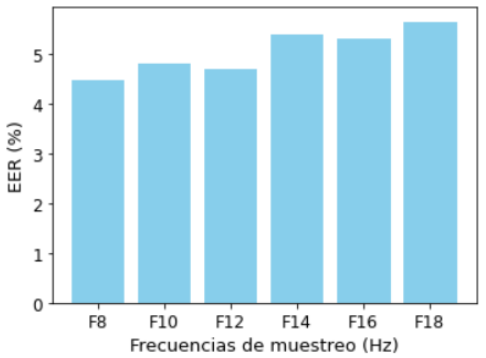

Figura 4.13: Moto GYR Modulo RF

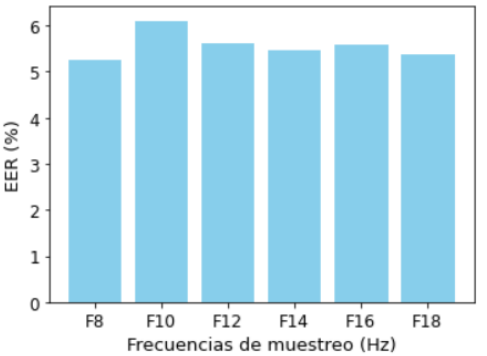

Figura 4.14: Moto GYR XYZ RF

En las siguientes Figuras 4.15, 4.16, 4.17, 4.18 se pueden observar los diagramas de barras que representan la frecuencia de muestreo en el eje X y la tasa de equierror en el eje Y para el dispositivo de la marca Microsoft y el método Support Vector Machines. En este caso parece cumplirse que el error es menor con frecuencias de muestreo bajas.

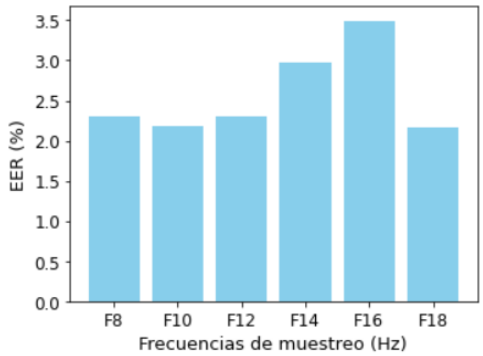

Figura 4.15: Micro ACC Modulo SVM

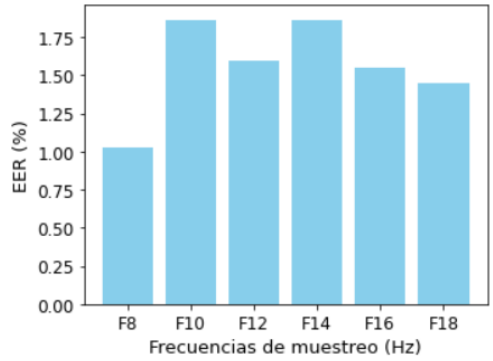

Figura 4.16: Micro ACC XYZ SVM

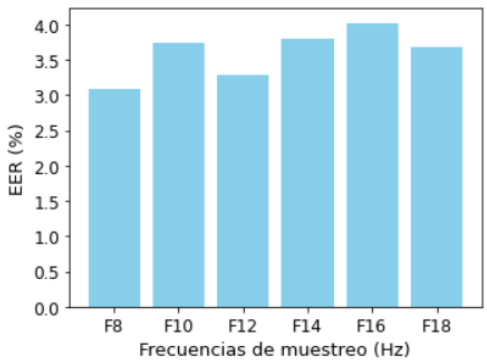

Figura 4.17: Micro GYR Modulo SVM

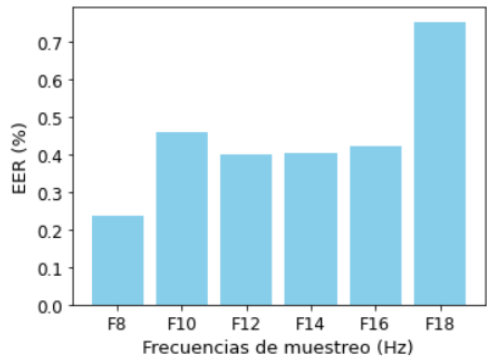

Figura 4.18: Micro GYR XYZ SVM

En las siguientes Figuras 4.19, 4.20, 4.21, 4.22 se pueden observar los diagramas de barras que representan la frecuencia de muestreo en el eje X y la tasa de equierror en el eje Y para el dispositivo de la marca Motorola y el método Support Vector Machines. En este caso parece cumplirse que el error es menor con frecuencias de muestreo bajas cuando se utiliza el sensor giróscopo.

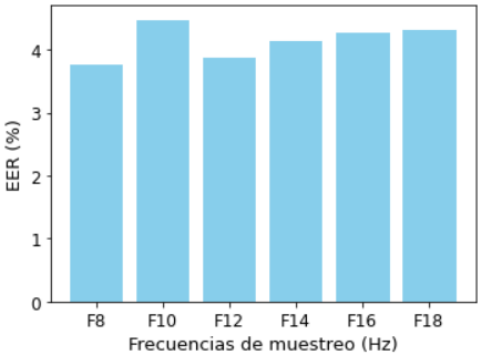

Figura 4.19: Moto ACC Modulo SVM

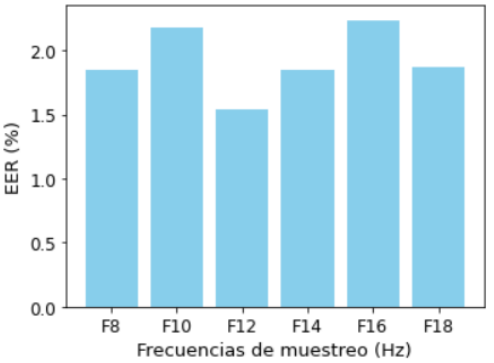

Figura 4.20: Moto ACC XYZ SVM

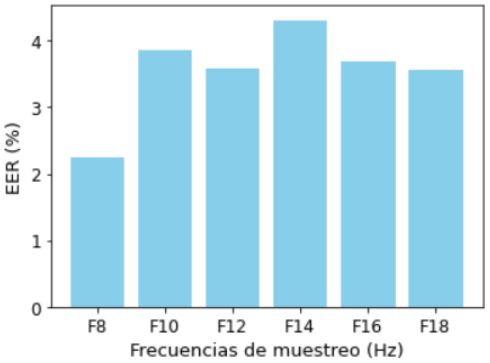

Figura 4.21: Moto GYR Modulo SVM

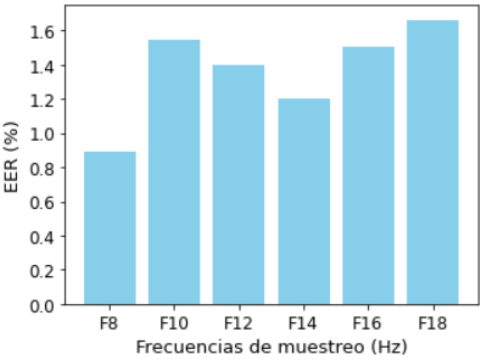

Figura 4.22: Moto GYR XYZ SVM

En la media de todos los métodos que muestra la Figura 4.23 se aprecia que la frecuencia de muestreo que menor tasa de equierror obtiene es la de 8Hz. Parece cumplirse la hipótesis de que con frecuencias de muestreo bajas los resultados son precisos, especialmente con las frecuencias de 8Hz y 12Hz. Al conseguir buen rendimiento con una frecuencia de muestreo menor, el coste energético y de almacenamiento disminuye y por tanto se pueden mejorar las prestaciones en batería del dispositivo.

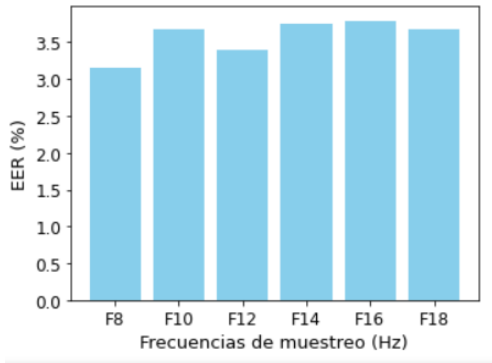

Figura 4.23: Media UVA

En las Figuras 4.24, 4.25, 4.26, 4.27 se pueden observar los diagramas de barras que representan la frecuencia de muestreo en el eje X y la tasa de equierror en el eje Y para la base de datos ZJU. En este caso se cumple que el EER es menor con frecuencias de muestreo bajas para cada uno de los métodos probados.

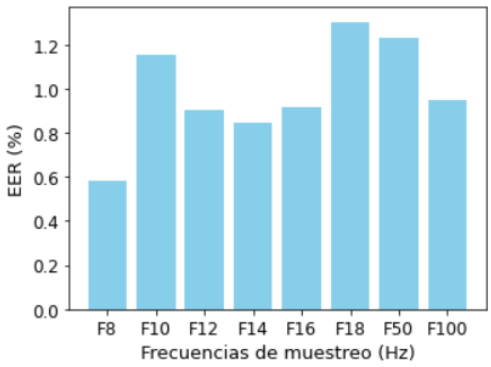

Figura 4.24: ZJU ACC Modulo RF

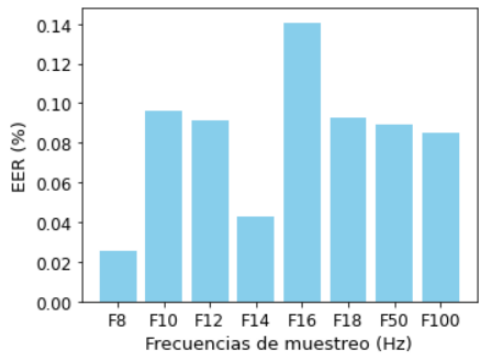

Figura 4.25: ZJU ACC XYZ RF

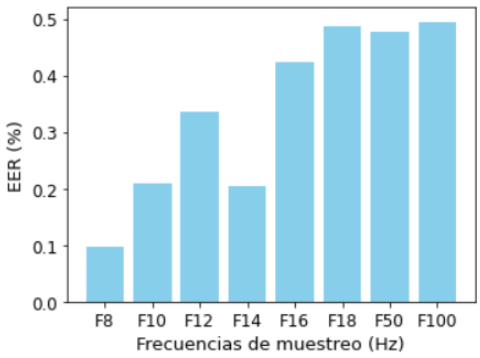

Figura 4.26: ZJU ACC Modulo SVM

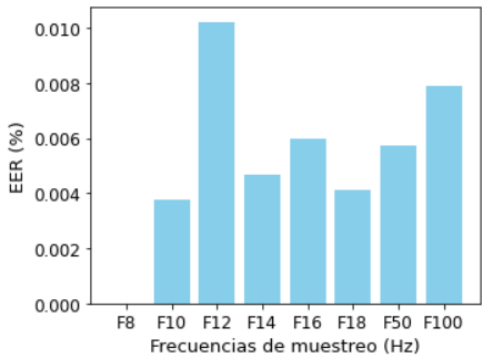

Figura 4.27: ZJU ACC XYZ SVM

En media se llega a la misma conclusión según se aprecia en la Figura 4.28.

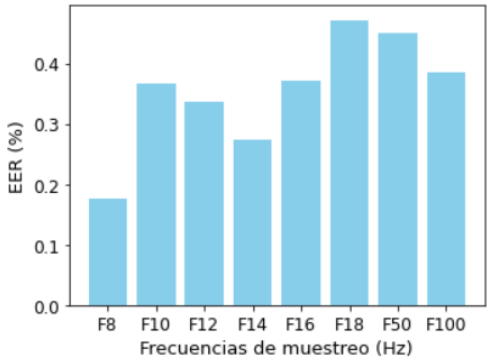

Figura 4.28: Media ZJU

A la vista de estos gráficos, se puede confirmar la hipótesis de que con frecuencias de muestreo bajas los resultados son precisos, teniendo unos EER menores que 0.4.

4.3. Sprint 2: Software de adquisición ECG

En esta fase se tiene como objetivo poner en marcha el software para capturar datos de ECG, realizando las modificaciones que sean necesarias. Una vez funcione este software, se podrá proceder a capturar nuevos datos, tal y como se muestra en la fase posterior.

El dispositivo Withings Move ECG es el elegido para la toma de datos, ya que ofrece una funcionalidad amplia a precio razonable. Además, fue utilizado en un TFG

anterior del alumno Mario Garrido Tapias [2] y su documentación en la web es clara [43]. Una vez que los datos se recogen y envían a la nube de Withings, pasan a estar disponibles para su uso. Para ello, hay que seguir una serie de pasos:

1. **Registro en el portal de desarrolladores Withings**

   Para ello hay que completar un registro habitual en una plataforma, indicando los datos personales. Otro dato menos habitual es la "callback URI", que permite recibir en ella la información en caso de que el portal de Withings no se encuentre operativo en ese momento. Además, hay que explicar la finalidad del uso que se dará a este portal de desarrolladores. Al finalizar el registro se obtiene un identificador de cliente y un número secreto que conviene guardar.

2. **Obtención del código de autorización**

   En este paso y los siguientes se utilizará el entorno de desarrollo Visual Studio Code con la extensión Thunder Client. Utilizando la siguiente URL: https://account.withings.com/oauth2 user/authorize2 y su método "authorize" se obtiene el código de autorización, que solamente tiene una validez de 30 segundos para usarlo en el siguiente paso. Hay unos parámetros de consulta que añadir:

   - response_type: cadena con el valor "code".
   - client_id: identificador del cliente. Se puede encontrar en el panel "overview" del portal de desarrolladores una vez que nos hemos registrado [44].
   - state: valor que define el usuario y permite evitar falsificaciones de nuestro entorno.
   - scope: alcance de los permisos solicitados al usuario. En la referncia [45] se pueden consultar los tipos de alcance ("user.metrics" y "user.activity") con los servicios que aportan, pero en este trabajo se selecciona solamente "user.metrics".
   - redirect_uri: la uri creada para la redirección en caso de que sea necesario. Su dirección es: https://greidi.infor.uva.es/ApiRestECG/Dformpost.php.

3. **Obtención del token de acceso**

   Una vez se dispone del código de autorización, estamos preparados para solicitar el token de acceso y el token de actualización. Resulta importante guardar los ID de usuario y tokens que se van obteniendo. El token de acceso caduca a las 3 horas de obtenerse. En este caso se realiza una operación POST a la URL https://wbsapi.withings.net/v2/oauth2 con parámetros de consulta:

   - action: cadena con el valor "requesttoken".

- client_id: identificador del cliente, es el mismo que se utilizó en el paso anterior.
- client_secret: el número secreto que aparece en el panel "overview" del portal de desarrolladores, justo debajo del "client_id".
- grant_type: cadena con valores "authorization_code" o "refresh_token", en función de si se quiere obtener o refrescar el token.
- code: código de autorización obtenido en el paso 2.
- redirect_uri: la uri creada para la redirección en caso de que sea necesario, la misma que en el paso anterior.

4. **Lista de ECGs capturados en el tiempo**

   En esta sección se obtiene una lista de registros de ECG durante un período de tiempo determinado. Es necesario añadir el token de acceso como parámetro de cabecera en "Authorization". La URL es https://wbsapi.withings.net/v2/heart y los parámetros de consulta:

   - action: cadena con el valor"list".
   - startdate (opcional): entero que muestra la fecha de inicio de los datos.
   - enddate (opcional): entero que muestra la fecha de finalización de los datos.

   Para el tema de las fechas, se utiliza la marca de tiempo Unix, que comienza el 1 de enero de 1970 a las 00:00. El entero que se pasa a las fechas son los segundos transcurridos desde esa fecha de partida hasta el momento de la toma. Para calcularlo en cada caso se utiliza un conversor de tiempo Unix [46].

5. **Obtención de datos crudos**

   En este paso también es necesario añadir el parámetro de cabecera con el token de acceso. La URL para la operación POST es la misma del paso 4, pero cambiando los parámetros de consulta:

   - action: cadena con el valor "get".
   - signalid: identificador de la señal, que se ha obtenido en el paso 4.

Para ir recopilando todas las mediciones, hay que repetir los pasos 4 y 5 sucesivamente una vez hayamos completado los pasos anteriores una vez. Si pasan más de 3 horas, habrá que recargar el token de acceso mediante el paso 3 con el parámetro de consulta "grant_type" establecido en "refresh_token".

A continuación, se detallan las fases de análisis, diseño y pruebas propias de la Ingeniería de Software aplicadas a la parte de adquisición de los datos de ECG. [2]

4.3.1. Análisis

Como el proceso de recuperar las tomas sería demasiado largo, se utiliza el software que se creó en un TFG del año 2022 [2]. Para desarrollarlo se empleó HTML, CSS y PHP. Nuestro objetivo es familiarizarnos con el código para volver a ponerlo en marcha. El software muestra una serie de formularios que permiten realizar la captura de datos de manera más eficiente y automatizada. Los formularios son los siguientes:

- Formulario de usuarios que sirve para registrar un usuario, para la elaboración de este trabajo se otorga al primer usuario el número 11. En la Figura 4.29 se puede observar la información necesaria para registrar un usuario. Se debe seleccionar el sexo, mano dominante, añadir la edad y si se tiene alguna enfermedad. En todos los casos del estudio la respuesta es "Ninguna".

  Enlace: https://greidi.infor.uva.es/ApiRestECG/UserForm.html

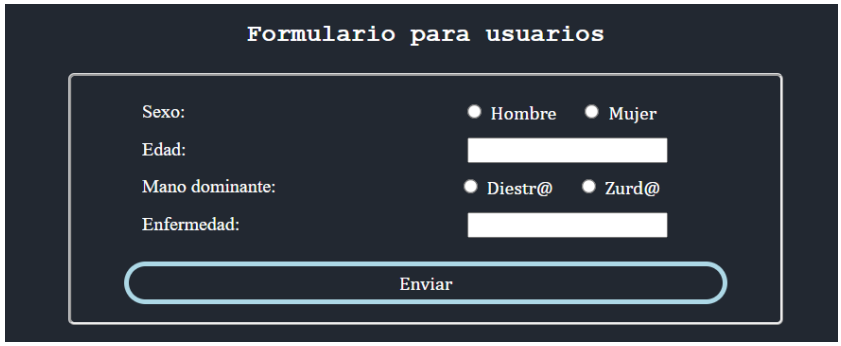

Figura 4.29: Formulario de usuarios

- Formulario de datos con la finalidad de registrar tomas. Estas tomas registradas se almacenan en una base de datos "phpMyAdmin". En la Figura 4.30 se puede ver que la información que se solicita es el identificador de usaurio, el día en el que se tomó la muestra, la hora de inicio, hora de fin y el número de muestras que por defecto será 10.

  Enlace: https://greidi.infor.uva.es/ApiRestECG/DataForm.php

- Formulario de conjunto de datos que permite pasar la información de la base de datos a formato "csv". En la Figura 4.31 se puede observar la información requerida, que es el mismo identificador de usuario que se usó en el formulario anterior, el número de sesión y el número de toma. Las sesiones y tomas podrán ser 1 o 2, ya que no se recogen más muestras por usuario.

  Enlace: https://greidi.infor.uva.es/ApiRestECG/FileForm.html

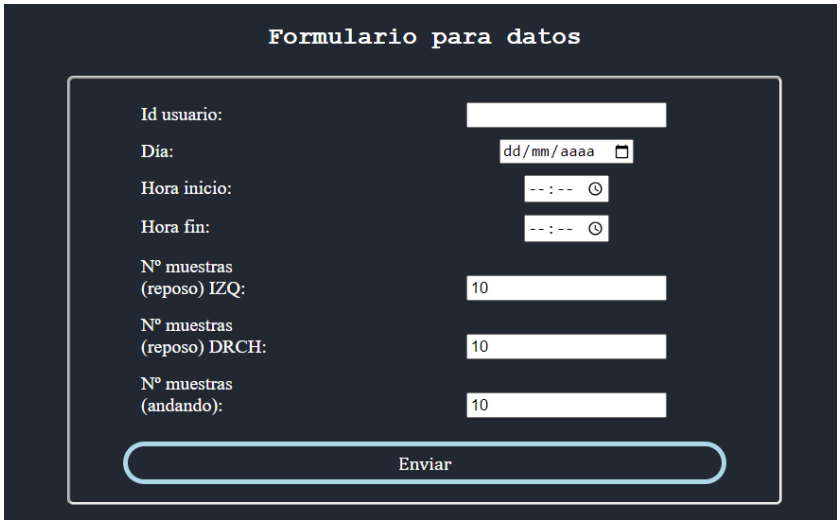

Figura 4.30: Formulario de datos

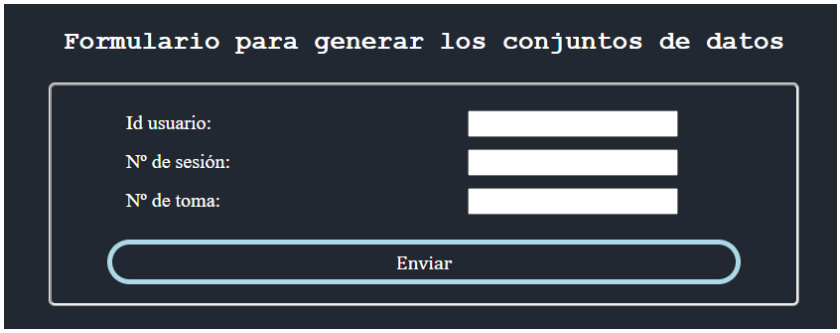

Figura 4.31: Formulario de archivos

Se revisaron todos los archivos “html” y “php” que dan lugar a los formularios. Resultó especialmente destacable un espacio adicional en la URL de la base de datos, por lo que se obtenía un “Malformed input to a URL”.

Existe un contratiempo con las fechas de las tomas a la hora de pasarlas a la base de datos utilizando el formulario de datos. El problema que ha surgido era que desde “php” se trabaja con la hora UTC, pero en España tenemos UTC +2, por lo que las tomas seleccionadas no eran las deseadas, concretamente eran las correspondientes a dos horas después. Para solucionar este inconveniente, se modifica el archivo “Dformpost.php”, indicando que estamos trabajando con la zona horaria UTC +2.

El otro contratiempo importante en este aspecto es el de las muestras con errores en la adquisición, ya que la recogida de datos se queda estancada cuando se llega a una muestra de este tipo. Para solucionarlo, se añade en el mismo archivo "Dformpost.php" un control de las muestras que van pasando, para que cuando una muestra sea incorrecta se muestre el mensaje "es errónea", no se almacene en la base de datos y se pase a la siguiente.

En cuanto al formulario de archivos, al rellenarlo se pasan los datos directamente al servidor "greidi.infor.uva.es". En ese momento se descubrió que todas las capturas correspondientes a la sesión 2 se almacenaron como sesión 1 y toma 2 en la base de datos. Al estar completa la sesión 1, no pasaba a la sesión 2 automáticamente, por lo que se cambió manualmente en el servidor "phpMyAdmin".

A continuación, se utiliza WinSCP para copiar los archivos "csv" desde el servidor al dispositivo local del alumno. Quedan por tanto añadidas las nuevas tomas de usuarios a los 10 que ya existían previamente.

Se presenta una serie de requisitos funcionales (RF), requisitos funcionales de información (RFI) y requisitos no funcionales (RNF).

Requisitos funcionales

- Registrar nuevo usuario. El sistema debe permitir realizar el registro de nuevos usuarios a partir de información básica.
- Obtener nuevas muestras. El sistema debe permitir almacenar en la base de datos las nuevas muestras que se tomen en reposo y andando.
- Generar conjuntos de datos con las tomas. El sistema debe permitir la creación de archivos de datos para cada usuario y sus muestras.

Requisitos funcionales de información

- Usuario. El sistema debe permitir el registro de cada usuario almacenando una serie de información característica de cada uno: sexo, edad, mano dominante y enfermedad.
- Datos muestras. El sistema debe permitir el guardado de las muestras, caracterizadas por el identificador de la señal y datos de la señal.
- Información de la muestra. El sistema debe permitir el almacenamiento de los datos de cada muestra: identificador de la señal, día de adquisición, hora de adquisición, identificador de usuario, mano portadora, pulso, actividad, sesión, toma y muestra.

Requisitos no funcionales

- El sistema debe usar la base de datos "phpMyAdmin" sostenida por el grupo "Greidi" de la Universidad de Valladolid. En ella se irán añadiendo las tomas.

- El sistema debe ser accesible desde dispositivos móviles y desde ordenadores, con distintos sistemas operativos.

4.3.2. Diseño

En cuanto a la base de datos, como lo que se pretende es añadir nuevos usuarios con sus respectivas tomas de la misma manera que se tenía en el proyecto anterior, se toma el mismo diagrama entidad-relación que estaba diseñado anteriormente. Este diagrama se muestra en la Figura 4.32.

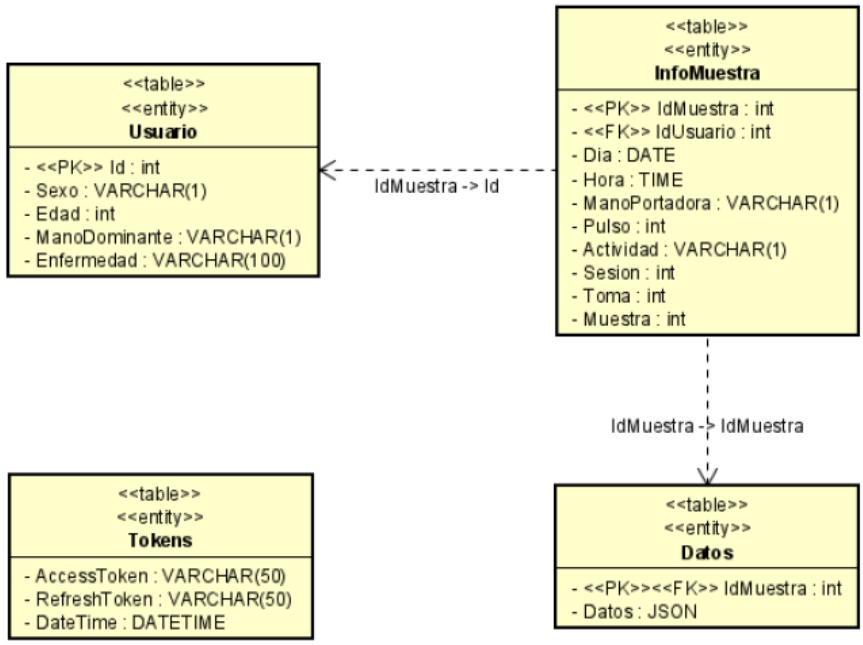

Figura 4.32: Diagrama entidad-relación de la BD [2]

4.3.3. Pruebas

Con el fin de verificar el correcto funcionamiento del software, se realiza una serie de pruebas:

1. Registrar un usuario cuando la base de datos está vacía.

   Entrada: Sexo hombre, edad 23, mano dominante diestro, enfermedad ninguna. No hay ningún usuario creado previamente.

   Salida esperada: El identificador del nuevo usuario es 1.

   Salida obtenida: El identificador del nuevo usuario es 1.

2. Registrar un usuario cuando la base de datos contiene 4 usuarios.

   Entrada: Sexo hombre, edad 23, mano dominante diestro, enfermedad ninguna. Hay 4 usuarios creados previamente.

   Salida esperada: El identificador del nuevo usuario es 1.

   Salida obtenida: El identificador del nuevo usuario es 1.

3. Dejar algún campo sin rellenar.

   Entrada: Sexo hombre, mano dominante diestro, enfermedad ninguna. No se añade la edad.

   Salida esperada: Mensaje de error, "Algún campo se encuentra vacío".

   Salida obtenida: Mensaje de error, "Algún campo se encuentra vacío".

4. Pérdida de alguna muestra en la nube.

   Entrada: Id usuario 11, día 23 de abril de 2024, hora de inicio 13:00, hora de fin 15:00, número de muestras (reposo) izq 9.

   Salida esperada: Mensaje de error, "El número de muestras indicado no concuerda".

   Salida obtenida: Mensaje de error, "El número de muestras indicado no concuerda".

5. Creación de archivos con tomas inexistentes.

   Entrada: Id usuario 11, número de sesión 1, número de toma 3.

   Salida esperada: Mensaje de error, "Toma no existe". No se crean archivos nuevos.

   Salida obtenida: Mensaje de error, "Toma no existe". No se crean archivos nuevos.

4.4. Sprint 3: Ampliación del corpus

En esta fase se pretende ampliar la base de datos de ECG, añadiendo nuevos usuarios con sus respectivas características.

El objetivo de los datos existentes y los nuevos es investigar acerca de la influencia de la frecuencia de muestreo en los electrocardiogramas que se obtienen a partir de dispositivos "wearables" especializados en el reconocimiento biométrico. Además, los datos añadidos sirven para aumentar la base de datos, usándola en este trabajo y ayudar a futuras investigaciones.

Se dispone inicialmente de una base de datos con datos biométricos correspondientes a ECGs obtenidos mediante un dispositivo"wearable". Esta base de datos está gestionada en "phpMyAdmin", una herramienta útil para administrar MySQL a través de una página web, en este caso https://greidi.infor.uva.es/phpmyadmin/.

4.4.1. Protocolo de adquisición

Todas las capturas se van a planificar de igual manera para todos los usuarios que participen en el estudio, para así evitar confusiones y datos erróneos. Todos los datos estarán anonimizados para cumplir así la Ley de Protección de Datos, identificando a cada usuario con números desde el 1 hasta el número total de participantes en la recogida de datos. Desde el 1 hasta el 10 ya estaban recogidos previamente, por lo que nuestro primer usuario a añadir es el 11. Todos los individuos son informados de los datos a tomar y para qué se usarán los datos a tomar, dando su consentimiento.

Los experimentos constan de dos sesiones separadas 2 semanas aproximadamente entre sí. Cada sesión tiene a su vez dos tomas, separadas entre sí un mínimo de 2 horas. De esta manera se continúa con la metodología que se llevó a cabo en el anterior trabajo [2], teniendo en cuenta la variabilidad de los datos biométricos con el tiempo.

Los datos de ECG obtenidos en cada toma se dividen en tres fases, entre las cuales no es necesario realizar descansos. La primera consiste en recoger 10 muestras en reposo de 30 segundos cada una con el dispositivo abrochado en la muñeca izquierda. La segunda se repite igual pero en la muñeca derecha. La tercera se realiza también en 10 muestras de 30 segundos pero esta vez el usuario tiene que estar andando y colocarse el dispositivo de nuevo en la mano izquierda. La explicación a esto es porque se necesitan 5 minutos de mediciones para cada una de estas fases dentro de una sesión y toma determinada. Al cambiar la muñeca y el estado de reposo o andando nos permitirá extraer conclusiones comparando distintos escenarios.

Para realizar estas fases, el alumno estará presente con el usuario para ayudar y supervisar el correcto desarrollo del proceso, ya que sino podrían tomarse datos erróneos y unas conclusiones equivocadas. El reloj Withings puede almacenar hasta 12 tomas, pero nosotros por cada sesión vamos a medir 30 tomas. Así, es necesrio conectar vía Bluetooth el reloj a la aplicación Health Mate instalada en el Smartphone del alumno para guardar las tomas al instante.

4.4.2. Descripción de los nuevos datos

A continuación, se muestra la información de todas las tomas realizadas. Se añaden metadatos como el sexo, edad, mano dominante y enfermedades. En este proyecto no se usarán ya que es un estudio prospectivo, es decir, se van recopilando datos y a partir de ellos se obtienen unos resultados, pero estos datos concretos no son los que interesan. Se mantienen ya que podrían ser útiles en trabajos futuros que utilicen esta información. Además de los metadatos, también aparece un identificador de usuario anonimizado. Por último, se incluye la cantidad de muestras en las distintas sesiones y tomas, además de las horas de inicio y fin de cada medición.

Cabe destacar que en el momento de capturar los datos, el usuario 18 presentó errores en muchas tomas tanto en reposo como andando. El dispositivo vestible reflejaba que las muestras tenían demasiado ruido, y por lo tanto no era posible capturar el ECG. En la sesión 1 se pudieron obtener 53 muestras correctas para este usuario, pero no fue posible obtener más ya que andando salían casi todas las muestras incorrectas. Fue un problema único para un usuario, por lo que quizá se deba a sus características biométricas concretas. Esto es algo típico en biometría, donde siempre existen errores en la captura del rasgo, por diversas razones.

En la Tabla 4.1 aparecen los metadatos de cada usuario y las muestras tomadas por sesión.

En la Tabla 4.2 se puede observar la información para los primeros cinco nuevos usuarios.

Id	**Sexo**	**Edad**	**Mano hábil**	**Enfermedad**	**Mues s1**	**Mues s2**
011	Mujer	23	Derecha	Ninguna	60	60
012	Hombre	62	Derecha	Ninguna	60	60
013	Mujer	60	Derecha	Ninguna	60	60
014	Hombre	24	Derecha	Ninguna	60	60
015	Hombre	21	Derecha	Ninguna	60	60
016	Hombre	24	Derecha	Ninguna	60	60
017	Mujer	59	Izquierda	Ninguna	60	60
018	Mujer	23	Derecha	Ninguna	53	0
019	Hombre	61	Derecha	Ninguna	60	60
020	Hombre	24	Derecha	Ninguna	60	60

Tabla 4.1: Metadatos de los usuarios y muestras tomadas

User11					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	06/05/2024	06/05/2024		31/05/2024	31/05/2024
	Inicio 19:35	Inicio 22:40		Inicio 12:53	Inicio 15:22
	Fin 20:38	Fin 23:07		Fin 13:20	Fin 15:45
User12					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	07/05/2024	07/05/2024		29/05/2024	29/05/2024
	Inicio 18:54	Inicio 21:35		Inicio 16:18	Inicio 22:06
	Fin 19:16	Fin 22:02		Fin 16:38	Fin 22:30
User13					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	07/05/2024	07/05/2024		29/05/2024	29/05/2024
	Inicio 19:24	Inicio 22:04		Inicio 17:00	Inicio 22:32
	Fin 19:51	Fin 22:39		Fin 17:24	Fin 22:57
User14					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	13/05/2024	13/05/2024		02/06/2024	03/06/2024
	Inicio 13:14	Inicio 18:44		Inicio 11:18	Inicio 14:41
	Fin 13:35	Fin 19:09		Fin 11:39	Fin 15:04
User15					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	15/05/2024	15/05/2024		30/05/2024	30/05/2024
	Inicio 15:35	Inicio 18:28		Inicio 11:17	Inicio 13:44
	Fin 15:58	Fin 18:51		Fin 11:39	Fin 14:06

Tabla 4.2: Información adicional sobre las tomas usuarios 11-15

En la Tabla 4.3 se puede observar la información para los siguientes cinco nuevos usuarios.

User16					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	16/05/2024	16/05/2024		05/06/2024	05/06/2024
	Inicio 10:38	Inicio 20:04		Inicio 14:31	Inicio 19:15
	Fin 11:07	Fin 20:33		Fin 15:01	Fin 20:46
User17					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	17/05/2024	17/05/2024		31/05/2024	31/05/2024
	Inicio 13:08	Inicio 16:03		Inicio 13:21	Inicio 15:52
	Fin 13:34	Fin 16:21		Fin 13:43	Fin 16:13
User18					
Sesion1	Toma1	Toma2			
	17/05/2024	17/05/2024			
	Inicio 13:49	Inicio 16:46			
	Fin 14:41	Fin 17:23			
User19					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	17/05/2024	17/05/2024		31/05/2024	31/05/2024
	Inicio 15:33	Inicio 18:33		Inicio 13:51	Inicio 16:20
	Fin 15:56	Fin 18:56		Fin 14:13	Fin 16:46
User20					
Sesion1	Toma1	Toma2	**Sesion2**	Toma1	Toma2
	17/05/2024	17/05/2024		03/06/2024	03/06/2024
	Inicio 19:57	Inicio 22:33		Inicio 20:09	Inicio 22:26
	Fin 20:29	Fin 23:02		Fin 20:37	Fin 22:51

Tabla 4.3: Información adicional sobre las tomas usuarios 16-20

4.5. Sprint 4: Sistema biométrico para ECG

A continuación, se detallan las fases de análisis, diseño y pruebas propias de la Ingeniería de Software aplicadas a la parte de experimentos con electrocardiogramas.

El objetivo será investigar si para frecuencias de muestreo bajas los resultados son tan buenos como para frecuencias más altas.

4.5.1. Análisis

Se presenta una serie de requisitos funcionales (RF) y requisitos no funcionales (RNF).

Requisitos funcionales

- Clasificar datos. El sistema debe realizar una clasificación de los datos utilizando los métodos Support Vector Machines y Random Forest.
- Almacenar datos. El sistema debe almacenar los resultados en el directorio correspondiente a su frecuencia de muestreo.
- Mostrar resultados. El sistema debe mostrar unos gráficos de barras que comparen las tasas EER con las distintas frecuencias de muestreo.

Requisitos no funcionales

- El sistema debe partir de los programas desarrollados en el TFM de Irene Salvador Ortega [1].
- El sistema debe utilizar la base de datos ECG creada en el TFG de Mario Garrido Tapias [2] y los datos añadidos en este mismo trabajo.

En esta sección se modifican una serie de archivos Python para que se adapten al sistema biométrico para ECG. Estos archivos son los mismos que se utilizaron para el sistema biométrico de la forma de andar. A continuación, se explican las modificaciones realizadas a esos programas.

En primer lugar, "1_0_ExtraccionCaract_DT_XYZ_UVA" se encarga de leer uno a uno los ficheros de datos disponibles sobre ECG para cada usuario, mano, reposo/andando, sesión y toma. A partir de ello, extrae las características y las escribe en un fichero de características para cada fichero de datos. En este caso solamente se utiliza una componente, por lo que se eliminan las operaciones referentes a las coordenadas Y, Z y módulo. La componente X pasa a llamarse "Signal". Para ECG no existen los parámetros dispositivo ni sensor. Además, se han añadido los argumentos "way" y "hand" que indican la manera de capturar los datos (reposo o andando) y la muñeca en la cual se coloca el dispositivo.

Después, "1_1_ExtraccionCaract_DT_AllUsers_UVA" lee cada fichero de características creado por el programa anterior y las junta todas en un fichero csv único. El nombre de este fichero indica los valores de algunos parámetros de la extracción de características. Por ejemplo, "8veces_20 %solap_modulo.csv" significa que este fichero contiene las características usando ventanas de 8 ciclos, con un 20 % de solapamiento entre ellas, es decir, la configuración "estándar".

El último paso se realiza en "2_ALT0_Modelo_EstComparativo" y consiste en utilizar un clasificador y realizar el estudio de rendimiento por cada frecuencia de muestreo. También resulta necesario modificar su funcionamiento ya que solamente hay una coordenada en este caso. Se utiliza el valor de "gamma_real" igual a 0.2. Para organizar el entrenamiento y las pruebas, lo que se hace en primer lugar es seleccionar las muestras de entrenamiento del usuario actual en la sesión S1 y muestra M1. Estas muestras se etiquetan como clase 1 (auténtico). Las muestras impostoras se dividen según par o impar, seleccionando usuarios impares para los usuarios que realmente son pares y viceversa. Estas muestras tienen clase 0 (impostor). Todo ello combinado forma el conjunto de entrenamiento. Para el conjunto de prueba se selccionan las muestras del usuario actual en la sesión S2 y muestras M1 o M2. Estos son datos auténticos del usuario. Los usuarios que no son el actual serán los impostores.

Para este sistema biométrico no influye el ruido, ya que las muestras se obtienen ya limpias. Inicialmente, habría que calcular las propiedades estadísticas básicas. Después, se selecciona una frecuencia de muestreo, para finalmente ejecutar un método de clasificación para cada combinación de datos tomados con mano izquierda/derecha y estado reposo/andando.

A continuación, se crea un conjunto de datos con los EER para las distintas frecuencias de muestreo, permitiendo así implementar los gráficos de barras del EER para cada frecuencia de muestreo.

4.5.2. Diseño

Los datos con los que se trabaja tienen una serie de características. La primera es que cada captura de datos en reposo o andando tiene una duración de 5 minutos. El dispositivo vestible recoge datos en intervalos de 30 segundos, por lo que es necesario realizar 10 muestras de cada tipo. Se anota la mano en la cual se pone el dispositivo. Otra característica fundamental es que las capturas se dividen en dos sesiones separadas entre sí un mínimo de dos semanas. Esto permite estudiar el patrón biométrico del ECG a lo largo del tiempo. Además, cada sesión se divide a su vez en 3 tomas (2 en reposo y una andando). De esas 3 tomas, 2 son con el dispositivo colocado en la mano izquierda y una con el dispositivo en mano derecha [1].

Otro tipo de información que se almacena para cada usuario que participa en el estudio son los metadatos. Concretamente, se determina el identificador de usuario, el sexo, la edad, la mano dominante, la mano portadora y la cantidad de tomas.

También resulta importante la división de los datos en conjunto de entrenamiento y conjunto de prueba, seleccionando las muestras de entrenamiento del usuario actual en la sesión S1 y muestra M1. Estas muestras se etiquetan como clase 1 (auténtico). Las muestras impostoras se dividen según par o impar:

- Usuario Par: Se seleccionan las muestras de usuarios impares. Se usarán para entrenar el modelo.
- Usuario Impar: Se seleccionan las muestras de usuarios pares. Se usarán para probar el modelo en la parte de pruebas de impostor.

Estas muestras tienen clase 0 (impostor). Para el conjunto de pruebas auténticas, se seleccionan las muestras del usuario actual en la sesión S2 y muestras M1 o M2.

Los demás datos que se obtienen son los que realmente se utilizarán para realizar los cálculos del sistema biométrico. Uno es el denominado "Timestamp", que almacena una referencia del tiempo que ha pasado entre una medición y la inmediatamente anterior. El otro dato es la señal que se obtiene del ECG en cada instante de tiempo.

A continuación, el propósito es crear gráficos de barras a partir de la información obtenida tras ejecutar los archivos correspondientes. Estos gráficos comparan las tasas de error para distintas frecuencias de muestreo.

Para organizar la información, se ha creado un fichero "csv" que almacena los errores medios. En las columnas están las frecuencias de muestreo y en las filas el estado, la mano utilizada y el método de clasificación. El procesamiento para obtener los gráficos deseados se realiza en Python. En primer lugar, se convierte el error en %, es decir, se multiplica por 100. Después, se separan los datos según la base de datos a la que pertenecen utilizando la librería "Pandas". El siguiente paso es crear una nueva fila con el error medio para cada frecuencia de muestreo y base de datos. El último paso es asegurarse de que los valores son numéricos y realizar los gráficos utilizando la librería "matplotlib.pyplot".

En cada uno de los programas Python hay que modificar el parámetro "frecuencia_muestreo" dentro de las funciones correspondientes. La frecuencia de muestreo por defecto es de 12Hz, y las pruebas inicialmente van a incluir valores de 8Hz, 10Hz, 14Hz, 16Hz y 18Hz. La frecuencia de muestreo es igual a la inversa del periodo de muestreo, por lo que la frecuencia de muestreo por defecto de 12Hz es igual a un periodo de muestreo de 0.08333 segundos, lo que a su vez son 83,33 milisegundos.

Adicionalmente, se probarán frecuencias de muestreo de 50Hz y 100Hz para ver el efecto de una frecuencia de muestreo elevada.

Arquitectura del Sistema

1. Captura de Datos
   - Dispositivo: Withings Move ECG.
   - Sesiones: Dos sesiones con varias tomas por estado y mano.
2. Almacenamiento
   - Directorios estructurados en base de datos ECG.
   - Subdirectorios para almacenar resultados iniciales y tasas de error EER.
3. Procesamiento
   - Extracción de Características:
     - División en ventanas temporales.
     - Extracción de características.
   - Clasificación:
     - Modelos SVM base radial y Random Forest.
     - Comparación de tasas de error para diferentes frecuencias de muestreo.
4. Generación de Resultados
   - Almacenamiento de resultados en subdirectorios específicos.
   - Creación de gráficos comparativos utilizando Python y librerías como Pandas y Matplotlib.

4.5.3. Pruebas

Con el fin de verificar el correcto funcionamiento del software, se realiza una serie de pruebas:

1. Prueba de extracción de características

   Objetivo: Verificar que la tabla de características se extrae correctamente.

   Procedimiento:

- Ejecutar el script "1_0_ExtraccionCaract_DT_ECG" con los datos de la muestra.
- Revisar los archivos generados para comprobar que contienen las características extraídas de manera correcta.

Salida esperada y obtenida: Los archivos de características deben contener datos consistentes y correctos según los parámetros de mano, estado, frecuencia de muestreo, máxima autocorrelación, extensión de la ventana y solapamiento.

2. Prueba de unión de características

   Objetivo: Verificar que las características de diferentes archivos se unen correctamente en un único fichero "csv".

   Procedimiento:

   - Ejecutar el script "1_1_ExtraccionCaract_DT_AllUsers_ECG".
   - Revisar el fichero csv resultante para asegurarse de que contiene todas las características unidas correctamente.

   Salida esperada y obtenida: El fichero "csv" contiene todas las características unidas de manera correcta y sin pérdidas de datos.

3. Prueba de clasificación y comparación de tasas de error

   Objetivo: Verificar que el sistema de clasificación y la comparación de tasas de error funcionan correctamente.

   Procedimiento:

   - Ejecutar el script "2_ALT0_Modelo_EstComparativo".
   - Revisar los resultados generados para asegurarse de que las tasas de error EER se calculan y comparan correctamente.

   Salida esperada y obtenida: Las tasas de error EER están correctamente calculadas y los resultados deben permitir una comparación precisa entre diferentes frecuencias de muestreo.

4. Prueba de conversión y separación de datos

   Objetivo: Verificar que los errores EER se convierten a porcentaje y se separan correctamente según la base de datos.

   Procedimiento:

   - Realizar el procesamiento de datos y separación de directorios.

- Revisar los datos para asegurarse de que los errores están convertidos y separados correctamente.

Salida esperada y obtenida: Los datos están convertidos a porcentaje y separados de manera interpretable.

5. Prueba de creación de gráficos

   Objetivo: Verificar que los gráficos comparativos se generan correctamente.

   Procedimiento:

   - Ejecutar el código de generación de gráficos en Python.
   - Revisar los gráficos generados para asegurarse de que comparan correctamente las tasas de error EER para diferentes frecuencias de muestreo.

   Salida esperada y obtenida: Los gráficos muestran comparaciones claras y correctas de las tasas de error EER para diferentes frecuencias de muestreo.

4.6. Sprint 5: Experimentos y resultados para ECG

En esta sección se exponen una serie de gráficos de barras que comparan la frecuencia de muestreo con el EER para cada combinación de estado (reposo/andando) y muñeca utilizada (derecha/izquierda). Todo ello se realiza mediante datos de ECG. La hipótesis que se tiene inicialmente es similar a la que se presentó en los experimentos con la forma de andar, es decir, que con frecuencias de muestreo bajas los resultados son tan buenos como para frecuencias más altas.

En las Figuras 4.33, 4.34, 4.35, 4.36, 4.37, 4.38 se pueden observar los diagramas de barras que representan la frecuencia de muestreo en el eje X y la tasa de equierror en el eje Y.

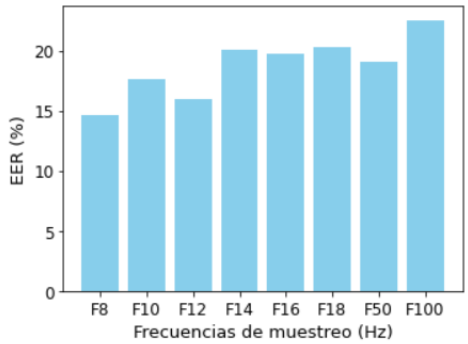

Figura 4.33: Reposo Izq RF

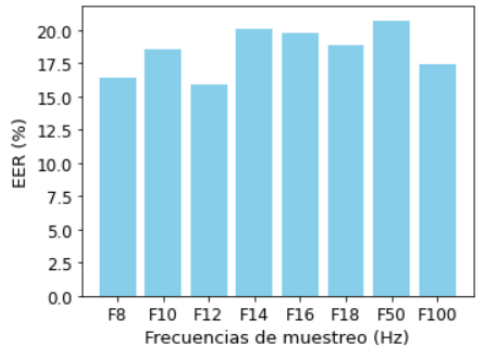

Figura 4.34: Reposo Der RF

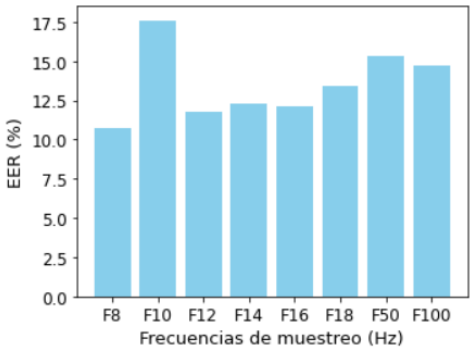

Figura 4.35: Andando Izq RF

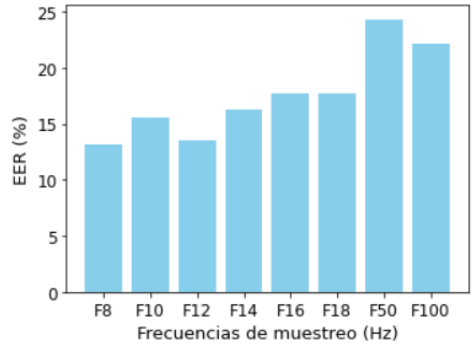

Figura 4.36: Reposo Izq SVM

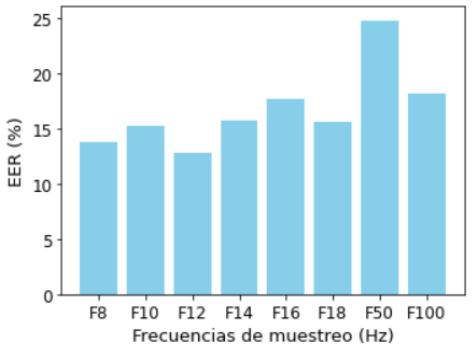

Figura 4.37: Reposo Der SVM

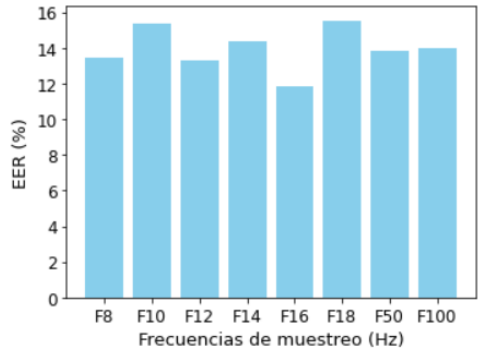

Figura 4.38: Andando Izq SVM

En media parece que las frecuencias de muestreo bajas tienen un EER ligeramente menor, según se aprecia en la Figura 4.39. Concretamente las tasas de equierror menores se dan en las frecuencias de 8Hz y 12Hz.

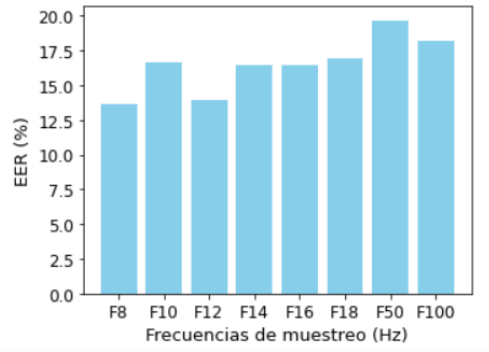

Figura 4.39: Media ECG

A la vista de estos gráficos podemos confirmar de nuevo que la hipótesis de que para frecuencias de muestreo bajas los resultados son tan buenos como para frecuencias más altas.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

Una vez finalizado el proyecto, nos encontramos con unos resultados satisfactorios a pesar de los contratiempos iniciales que lo atrasaron y provocaron cambios significativos. Esto desencadenó en un cambio de tema del TFG, por lo que ha servido para reconocer la gran importancia del análisis de riesgos y sus posibles soluciones en caso de que ocurran.

5.1.1. Conclusiones con respecto a los objetivos planteados

En este trabajo se han cubierto los objetivos que se plantearon inicialmente:

- Se ha logrado entender el funcionamiento de la biometría basada en la forma de andar y el ECG.
- Se ha analizado, comprendendido y modificado el software disponible de trabajos anteriores para su correcta adaptación a las necesidades de este trabajo.
- Se han adquirido más muestras de ECG, para tener una base datos más numerosa y, así, se han podido obtener resultados más confiables.
- Se ha analizado y comparado el rendimiento de un sistema de reconocimiento tanto de forma de andar como de ECG para distintas frecuencias de muestreo.

5.1.2. Conclusiones con respecto a los resultados obtenidos

Se ha conseguido ampliar la investigación disponible sobre la frecuencia de muestreo en smartwatches, concretando sobre la forma de andar y ECG. Los estudios existentes sobre la materia no son muchos, por lo que se han propuesto resultados novedosos y prometedores.

En cuanto a la forma de andar, la tasa de equierror para frecuencias de muestreo comprendidas entre 8Hz y 100Hz es menor utilizando el método SVM.

Se ha logrado capturar datos de ECG para 9 usuarios completos mediante un dispositivo Withings, por lo que se dispone de un total de 19 usuarios en la base de datos ECG.

Aplicando los experimentos al ECG, la tasa de equierror para frecuencias de muestreo comprendidas entre 8Hz y 100Hz es menor utilizando el método SVM. Cabe destacar que la EER es mayor en estos experimentos de ECG que en los de la forma de andar.

Se ha conseguido demostrar la hipótesis principal del trabajo, que establecía que usando frecuencias de muestreo de la señal bajas, el rendimiento en el reconocimiento humano podía ser bueno. De esta manera se logra un equilibrio entre el consumo de recursos y el rendimiento del dispositivo.

Centrándonos en las figuras con los resultados medios: 4.23 y 4.28 para la forma de andar y 4.39 para ECG, los resultados para frecuencias de muestreo bajas, son incluso mejores que para frecuencias más altas. Por ejemplo, el EER medio para la forma de andar y base de datos UVA con frecuencias de muestreo de 8Hz y 12Hz es inferior al 3,5 %, mientras que para todas las frecuencias de muestreo mayores el EER también es mayor. En cuanto a ECG, el EER medio con frecuencias de muestreo de 8Hz y 12Hz es inferior al 15 %, y para frecuencias de muestreo mayores que 12Hz el EER es siempre mayor al 15 %.

Además, resulta interesante apreciar como frecuencias tan bajas como 8Hz, muestran rendimientos muy buenos, normalmente mejores que frecuencias más altas. En definitiva, las frecuencias de muestreo de 8Hz suelen dar como resultado las tasas de error más bajas.

5.2. Aprendizaje obtenido

Durante la realización de este trabajo han ido surgiendo imprevistos y retos a los que nunca me había enfrentado, por lo que he mejorado mi capacidad para aprender

y manejarme con herramientas en las cuales no tenía experiencia previa.

Concretamente, he podido ir más allá en la programación en Python, desarrollando métodos de aprendizaje automático. En cuanto a R, me ha permitido adentrarme en el interesante mundo de la paralelización. Además, he aprendido a manejarme con HTML y PHP, ya que previamente tenía nociones más básicas.

Como el proyecto era continuación de otros anteriores, he aprendido a extraer la información que resulta útil y también a comprender códigos realizados por otras personas en diferentes momentos del tiempo. Cabe destacar la utilidad de documentar siempre el código que se desarrolla, es algo que he aprendido durante el Grado y resulta fundamental para un Ingeniero Informático.

Al desarrollar tareas dentro del ámbito de la biometría, me he dado cuenta de su importancia y de todas las aplicaciones que tiene en la actualidad mas las que se irán descubriendo en los próximos años.

5.3. Líneas futuras

Todos los programas Python modificados para experimentar se encuentran preparados para posibles nuevos avances. Las pruebas se podrían ir ejecutando realizando un cambio en el valor del parámetro que sea de interés.

Una posible mejora podría incluir nuevos métodos de clasificación que quizá mejorasen las tasas de equierror obtenidas en este trabajo.

Además, las bases de datos para la forma de andar y ECG podrían aumentarse respectivamente con los dispositivos vestibles correspondientes en cada caso. Al tener más usuarios en la muestra, podrían incluirse nuevos requisitos, como por ejemplo equilibrio entre sexos, rangos de edad y otros parámetros sanitarios como el estilo de vida.

Otra prueba interesante podría venir relacionada con el tiempo, es decir, probar a dejar más tiempo de margen entre las diferentes sesiones, pudiendo ser de varios meses si los plazos lo permiten. Esto permitiría estudiar el efecto del tiempo sobre el rasgo biométrico.

Finalmente, se podría plantear la recuperación del tema de la paralelización que quedó pendiente, poniéndose en contacto con las personas que desarrollaron los programas existentes y ver el problema de la puesta en funcionamiento para seguir investigando sobre el tema.

Bibliografía

[1] Irene Salvador Ortega *Investigación y desarrollo de un sistema de reconocimiento biométrico mediante dispositivos ponibles (Wearables)*. URL: https://uvadoc.uva.es/handle/10324/44947?locale-attribute=fr Fecha de acceso: 17-04-2024.

[2] Mario Garrido Tapias *Reconocimiento Biométrico Mediante ECG Usando Dispositivos Ponibles (Wearables)*. URL: https://uvadoc.uva.es/handle/10324/57308 Fecha de acceso: 23-04-2024.

[3] Biometrics Institute *'What is biometrics?'*. URL: https://www.biometricsinstitute.org/what-is-biometrics/ Fecha de acceso: 10-05-2024.

[4] Wikipedia *'Biometría'*. URL: https://es.wikipedia.org/wiki/Biometra Fecha de acceso: 10-05-2024.

[5] Elvira Misfud *'Sistemas físicos y biométricos de seguridad'*. URL: http://recursostic.educacion.es/observatorio/web/ca/cajon-de-sastre/38-cajon-de-sastre/1045-sistemas-fisicos-y-biometricos-de-seguridad%7D%7B http://recursostic.educacion.es/observatorio/web/ca/cajon-de-sastre/38-cajon-de-sastre/1045-sistemas-fisicos-y-biometr Fecha de acceso: 10-05-2024.

[6] Academia Pragma *Importancia de la biometría en la era digital*. URL: https://www.pragma.com.co/academia/conceptos/importancia-de-la-biometria-en-la-era-digital Fecha de acceso: 12-05-2024.

[7] Sharath Pankanti, Salil Prabhakar, Anil K. Jain *'Biometric Recognition: Security and Privacy Concerns'*. URL: https://www.researchgate.net/publication/3437477_Biometric_Recognition_Security_and_Privacy_Concerns Fecha de acceso: 10-05-2024.

[8] Arun Ross, Anil K. Jain *'Multibiometric Systems'*. URL: https://dl.acm.org/doi/pdf/10.1145/962081.962102 Fecha de acceso: 10-05-2024.

[9] Guillermo Jáñez Lagüéns, *Reconocimiento biométrico de la forma de andar mediante ponibles inteligentes usando el Modelo FMM (Frecuency Modulated Möbius)". Trabajo de Fin de Grado en Universidad de Valladolid, año 2022.* URL: https://uvadoc.uva.es/handle/10324/57318 Fecha de acceso: 21-02-2024.

[10] MedicinePlus *Electrocardiograma.* URL: https://medlineplus.gov/spanish/pruebas-de-laboratorio/electrocardiograma/ Fecha de acceso: 12-05-2024.

[11] Lena Biel, Ola Pettersson, Lennart Philipson, Peter Wide *ECG Analysis: A New Approach in Human Identification.* URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2d8d20d5cdea1f6dd728f9b7d522807a7be42e81 Fecha de acceso: 12-05-2024.

[12] Arrow *Sensores de ECG para la muñeca: cómo la tecnología portátil cambia las reglas del juego en la atención cardiaca.* URL: https://www.arrow.com/es-mx/research-and-events/articles/wrist-worn-ecg-sensors-how-wearable-tech-is-changing-the-game-for-cardiac-care Fecha de acceso: 12-05-2024.

[13] Nikita Samarin, Donald Sannella *A Key to Your Heart: Biometric Authentication Based on ECG Signals.* URL: https://arxiv.org/pdf/1906.09181 Fecha de acceso: 12-05-2024.

[14] Multison Online *Frecuencia De Muestreo: Qué Es Y Cómo Calcularla.* URL: https://multisononline.com/blog/de-analogico-a-digital-frecuencia-de-muestreo-y-tasa-de-bits-n267 Fecha de acceso: 13-05-2024.

[15] Guglielmo Cola, Alessio Vecchio, Raffaele Nocerino *Gait-based Authentication: Evaluation of Energy Consumption on Commercial Devices.* URL: https://ieeexplore.ieee.org/document/9767367 Fecha de acceso: 17-04-2024.

[16] Guglielmo Cola, Alessio Vecchio, Marco Avvenuti *Continuous authentication through gait analysis on a wrist-worn device.* URL: https://www.sciencedirect.com/science/article/abs/pii/S1574119221001139#preview-section-abstract Fecha de acceso: 17-04-2024.

[17] *¿Qué es la metodología ágil?.* URL: https://www.redhat.com/es/topics/devops/what-is-agile-methodology Fecha de acceso: 22-04-2024.

[18] Project Management Institute *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®) Tercera Edición.* URL: https://topodata.com/wp-content/uploads/2019/10/GUIA_PMBok.pdf Fecha de acceso: 15-05-2024.

[19] Dharma Consulting *Entendiendo los riesgos del proyecto: La Matriz de Probabilidad e Impacto.* URL: https://dharmacon.net/2023/07/26/entendiendo-l

os-riesgos-del-proyecto-la-matriz-de-probabilidad-e-impacto/ Fecha de acceso: 15-05-2024.

[20] *¿Cuánto gana un ingeniero informático en España?*. URL: https://4geeksacademy.com/es/cuanto-gana-un-ingeniero-informatico/cuanto-gana-un-ingeniero-informatico-en-espana Fecha de acceso: 22-04-2024.

[21] *Plan Familias Iberdrola*. URL: https://www.iberdrola.es/luz/plan-familias Fecha de acceso: 22-04-2024.

[22] *Tarifas de fibra y móvil Vodafone*. URL: https://www.vodafone.es/c/particulares/es/productos-y-servicios/fibra-optica-adsl/ Fecha de acceso: 22-04-2024.

[23] Fernán García de Zúñiga, Arsys *¿Qué es Visual Studio Code y cuáles son sus ventajas?*. URL: https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas Fecha de acceso: 11-05-2024.

[24] Jesús Santaella, Talenty Tech *¿Qué es la programación en R?*. URL: https://talently.tech/blog/programacion-en-r/ Fecha de acceso: 11-05-2024.

[25] Wikipedia *RStudio*. URL: https://es.wikipedia.org/wiki/RStudio Fecha de acceso: 11-05-2024.

[26] AWS *¿Qué es Python?*. URL: https://aws.amazon.com/es/what-is/python/ Fecha de acceso: 11-05-2024.

[27] anaconda-navigator *Launch data science applications from your desktop with Anaconda Navigator*. URL: https://www.anaconda.com/anaconda-navigator Fecha de acceso: 11-05-2024.

[28] Jupyter *Free software, open standards, and web services for interactive computing across all programming languages*. URL: https://jupyter.org/ Fecha de acceso: 11-05-2024.

[29] GanttProject *Free desktop project management software*. URL: https://www.ganttproject.biz/#is-easy Fecha de acceso: 11-05-2024.

[30] Overleaf *'About Us'*. URL: https://es.overleaf.com/about Fecha de acceso: 17-03-2024.

[31] Wikipedia *Microsoft Excel*. URL: https://es.wikipedia.org/wiki/Microsoft_Excel Fecha de acceso: 11-05-2024.

[32] MDN web docs *Conceptos básicos de HTML*. URL: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics Fecha de acceso: 11-05-2024.

[33] PHP *¿Qué es PHP?*. URL: https://www.php.net/manual/es/intro-whatis.php Fecha de acceso: 11-05-2024.

[34] phpMyAdmin, About *phpMyAdmin Bringing MySQL to the web*. URL: https://www.phpmyadmin.net/ Fecha de acceso: 11-05-2024.

[35] Hostinger tutoriales *Qué es GitHub y cómo empezar a usarlo*. URL: https://www.hostinger.es/tutoriales/que-es-github Fecha de acceso: 11-05-2024.

[36] Wikipedia *WinSCP*. URL: https://es.wikipedia.org/wiki/WinSCP Fecha de acceso: 07-06-2024.

[37] Itziar Fernandez, Alejandro Rodriguez-Collado, Yolanda Larriba, Adrian Lamela, Christian Canedo, Cristina Rueda *FMM: Rhythmic Patterns Modeling by FMM Models*. URL: https://cran.r-project.org/web/packages/FMM/index.html Fecha de acceso: 13-05-2024.

[38] *Gait_CrossStudy_Python*. URL: https://gitlab.inf.uva.es/ponibles/gait_crossstudy_python Fecha de acceso: 11-04-2024.

[39] IBM *Análisis ROC*. URL: https://www.ibm.com/docs/es/spss-statistics/beta?topic=features-roc-analysis Fecha de acceso: 27-05-2024.

[40] DATAtab *Curva ROC, Análisis ROC*. URL: https://datatab.es/tutorial/roc-curve Fecha de acceso: 27-05-2024.

[41] Roberto Tronci, Giorgio Giacinto, Fabio Roli *Dynamic Score Combination: A Supervised and Unsupervised Score Combination Method*. URL: https://www.researchgate.net/figure/An-example-of-a-ROC-curve-its-AUC-and-its-EER_fig1_225180361 Fecha de acceso: 27-05-2024.

[42] Irene Salvador Ortega, Carlos Vivaracho Pascual, Arancha Simón Hurtado. *A New Post-Processing Proposal for Improving Biometric Gait Recognition Using Wearable Devices*. URL: https://www.mdpi.com/1424-8220/23/3/1054 Fecha de acceso: 03-07-2024.

[43] *Withings developer documentation (2.0)*. URL: https://developer.withings.com/api-reference/ Fecha de acceso: 23-04-2024.

[44] *Withings Developer Dashboard, Overview*. URL: https://developer.withings.com/dashboard/ Fecha de acceso: 23-04-2024.

[45] *All available health data*. URL: https://developer.withings.com/developer-guide/v3/data-api/all-available-health-data/ Fecha de acceso: 23-04-2024.

[46] *Convertir tiempo Unix (Últimos segundos para 1970)*. URL: https://www.tops ter.es/calendario/unixzeit.php?tag=24&monat=10&jahr=2023&stunde=23 &minute=22&sekunde=0 Fecha de acceso: 23-04-2024.