



Universidad de Valladolid

Escuela de Ingeniería Informática

Grado en Ingeniería Informática

Mención en Computación

**Reconocimiento Biométrico Mediante ECG
Usando Dispositivos Ponibles (*Wearables*)**

Alumno: Mario Garrido Tapias

Tutor: Carlos Enrique Vivaracho Pascual

Tutora: María Aránzazu Simón Hurtado

Resumen

Cada vez son más los servicios que utilizan la biometría como factor de autenticación usándola como primer factor, o en muchos casos como segundo. A su vez el avance tecnológico en el mundo de los sensores ha ido ampliando las posibilidades de esta tecnología.

Una de las características con poco recorrido es el uso de la actividad eléctrica del corazón, conocido como electrocardiograma. Es por ello que el objetivo principal de este trabajo será el estudio del reconocimiento biométrico mediante capturas de electrocardiogramas (ECG) con dispositivos comerciales.

El presente documento expone un punto de partida para la captura del electrocardiograma desde el estudio de las posibilidades del mercado, pasando por la creación del *software* de adquisición y hasta el análisis del rendimiento de un sistema biométrico creado a partir de otro cuya característica de identificación era la forma de andar.

Abstract

More and more services are using biometrics as a first factor, in many cases, second authentication factor. At the same time, technological advances in the world of sensors have been expanding the possibilities of this technology.

One of the features that is still in its infancy is the use of the electrical activity of the heart, known as electrocardiogram. That is why the main objective of this work will be the study of biometric recognition by capturing electrocardiograms (ECG) with commercial devices.

This paper presents a starting point for ECG capture from the study of the market possibilities, through the creation of the acquisition software, to the analysis of the performance of a biometric system created from another one whose identification feature was the gait.

Agradecimientos

En primer lugar, dar mi gratitud a todos y cada uno de los profesores implicados en mi aprendizaje durante estos cinco años. Y en especial, a los dos tutores de este trabajo, Carlos y Arancha, dejándome formar parte de uno de sus proyectos y dándome la oportunidad de aprender a través de él un mundo complicado pero interesante como es el de la biometría.

Una mención especial a mis compañeros, por hacer más fácil y amena esta formación.

Por último, pero no menos importante, a ese círculo cerrado y familiar que me ha animado incesantemente y ha confiado ciegamente en mí.

Índice general

Resumen	1
Abstract	3
Agradecimientos	5
1. Introducción	13
1.1. Motivación	13
1.2. Objetivos	14
1.3. Objetivos personales	15
1.4. Estructura de la memoria	15
2. Marco teórico	17
2.1. ¿Qué es la biometría?	17
2.2. Electrocardiograma	19
2.3. Sistema biométrico	20
3. Trabajos relacionados	25
4. Planificación y herramientas	29
4.1. Metodología de trabajo	29

4.1.1. Aplicación en el proyecto	30
4.2. Planificación inicial	30
4.3. Riesgos	31
4.4. Planificación final	34
4.5. Dispositivos de prueba	38
4.6. Presupuesto	39
4.7. Tecnologías utilizadas	40
5. Desarrollo	43
5.1. Iteración 0: Proyecto inicial	43
5.2. Iteración 1: El dispositivo	43
5.2.1. Análisis de mercado	43
5.3. Iteración 2: Software de adquisición	45
5.3.1. Análisis	50
5.3.2. Diseño	54
5.3.3. Pruebas	56
5.4. Iteración 3: Obtención del corpus biométrico	58
5.4.1. Protocolo de adquisición	58
5.4.2. Descripción del corpus obtenido	60
5.5. Iteración 4: Creación del sistema biométrico	61
5.6. Iteración 5: Experimentación y resultados	63
5.6.1. Experimentos	63
5.6.2. Resultados	68
5.6.3. Análisis de resultados	77
6. Conclusiones	79
6.1. Conclusiones	79

6.2. Trabajos futuros	80
Bibliografía	83
A. Código utilizado	87
A.1. Estructura	87

Índice de tablas

4.1. Riesgo 001	32
4.2. Riesgo 002	32
4.3. Riesgo 003	32
4.4. Riesgo 004	33
4.5. Riesgo 005	33
4.6. Riesgo 006	33
4.7. Riesgo 007	33
4.8. Riesgo 008	34
4.9. Riesgo 009	34
4.10. Riesgo 010	34
4.11. Inversión del tiempo esquematizada por subtareas	38
4.12. Descripción smartphone de prueba	39
4.13. Descripción smartwatch de prueba	39
4.14. Presupuesto del proyecto	40
5.1. Caso de uso 3 - Crear archivos sobre crudos de una toma	52
5.2. Caso de uso 1 - Registrar un usuario	53
5.3. Caso de uso 2 - Recuperar una toma	54
5.4. Prueba 001	57

5.5. Prueba 002	57
5.6. Prueba 003	57
5.7. Prueba 004	58
5.8. Metadatos de los usuarios	60

Índice de figuras

2.1. Comparativa de los principales sistemas biométricos [8]	18
2.2. Sistema eléctrico del corazón y ECG básico	19
2.3. Etapas de un sistema biométrico	20
2.4. Ejemplo de ventana con solapamiento	21
2.5. Ejemplo de fusión de ventanas en postprocesamiento, para $n =$ 3 y solapamiento igual a 1.	23
4.1. Diagrama de Gantt del proyecto	31
4.2. Segundo diagrama de Gantt del proyecto	35
5.1. Withings Move ECG	45
5.2. Sistema Withings presentado en [22]	45
5.3. Icono ECGetter	46
5.4. Vista del portal de desarrolladores de Withings	47
5.5. Operación POST para obtener un token de acceso	48
5.6. Operación POST para obtener un token de acceso	49
5.7. Formulario para recuperar una toma	50
5.8. Diagrama de casos de uso	52
5.9. Diagrama de entidad-relación la base de datos	55
5.10. Diagrama de interacción caso de uso 1	56

5.11. Diagrama de interacción caso de uso 3	56
5.12. Análisis visual Usuario 007	64
5.13. Análisis visual Usuario 004	64
5.14. Explicación gráfica EER [25]	67
5.15. Múltiples imágenes	67
5.16. Experimento Monosesión-En reposo-Izquierda	69
5.17. Mejores algoritmos para DT-Monosesión-En reposo-Izquierda	69
5.18. Mejores algoritmos para DF-Monosesión-En reposo-Izquierda	70
5.19. Experimento Monosesión-En reposo-Derecha	70
5.20. Mejores algoritmos para DT-Monosesión-En reposo-Derecha	71
5.21. Mejores algoritmos para DF-Monosesión-En reposo-Derecha	71
5.22. Experimento Monosesión-Andando-Izquierda	72
5.23. Mejores algoritmos para DT-Monosesión-Andando-Izquierda	72
5.24. Mejores algoritmos para DF-Monosesión-Andando-Izquierda	73
5.25. Experimento Multisesión-En reposo-Izquierda	74
5.26. Mejores algoritmos para DT-Multisesión-En reposo-Izquierda	74
5.27. Mejores algoritmos para DF-Multisesión-En reposo-Izquierda	74
5.28. Experimento Multisesión-En reposo-Derecha	75
5.29. Mejores algoritmos para DT-Multisesión-En reposo-Derecha	75
5.30. Mejores algoritmos para DF-Multisesión-En reposo-Derecha	75
5.31. Experimento Multisesión-Andando-Izquierda	76
5.32. Mejores algoritmos para DT-Multisesión-Andando-Izquierda	76
5.33. Mejores algoritmos para DF-Multisesión-Andando-Izquierda	76

Capítulo 1

Introducción

1.1. Motivación

Internet ha cambiado el comercio, la educación, las comunicaciones o incluso la forma de relacionarnos. Hoy en día cualquier persona que se dispone a realizar una transacción en la web, necesita en la mayoría de sitios web una cuenta, en cuyo acceso es necesario una autenticación. Los sistemas más extendidos son los pares usuario y contraseña, pero debido a la cantidad de registros a los que pertenecemos, esto nos lleva a la mayoría a hacer un uso poco seguro de ellas, manteniendo la misma contraseña para varios sitios o colocando *passwords* sencillas pero que nos son fáciles de recordar.

Una alternativa bastante interesante es el uso de biometría, que bien aplicada, aporta al usuario comodidad dado que no necesita recordar ninguna contraseña engorrosa, ya que estos sistemas dependen de características intrínsecas del propio individuo, siendo estas intransferibles y difíciles de falsificar. La biometría no fue usada en Occidente hasta finales del siglo XIX, siendo China el precursor en el siglo XIV con el uso de impresiones de palmas de mano, con el objetivo de identificar a niños.

En la actualidad son muchos los procesos cotidianos en los que es necesario el uso de biometría, y esto se debe al auge de hardware que nos permite detectar nuestra huella dactilar, escanear nuestra cara o nuestro iris, y aportar así mayor seguridad a nuestros dispositivos o a nuestras compras online. Estos sensores se han implementado principalmente en los teléfonos móviles, aunque ya existen en algunos ordenadores portátiles. Nos permiten realizar las confirmaciones de nuestras transacciones, implementadas por el PSD2 [1]. Una de sus novedades es la Autenticación Reforzada de Clientes (SCA en inglés) consistente en la autenticación de doble factor en pagos y accesos a cuentas bancarias online.

Aunque extendido, como se ha comentado, el reconocimiento biométrico de personas no es un tema cerrado ni mucho menos. A pesar de que existen rasgos, como los indicados, ya usados comercialmente, todos tienen algún inconveniente: incomodidad para el usuario, uso de sensores dedicados, susceptibilidad o rechazo por parte de los usuarios por estar relacionados con temas penales (como es el caso de la huella dactilar), etc. Por eso, es un tema de continuo interés en nuevas alternativas. Aquí es donde se integra este trabajo, enfocado a la investigación del uso del ECG (Electrocardiograma) para el reconocimiento biométrico de personas. Para que el enfoque sea más realista, además, no vamos a usar dispositivos médicos especializados en su captura, sino que se busca que esta sea realizada mediante dispositivos comerciales de uso general, es decir, dispositivos que cualquier usuario pueda portar, como un reloj inteligente que, eso sí, permita, entre sus funciones, la captura del ECG.

Los sistemas de identificación mediante ECG no se encuentran aún igual de extendidos que los que utilizan otro tipo de características comentadas anteriormente. Sí que existe un producto comercial (Nymi [2]) que integra una pulsera que captura ECG con un sistema que permite el reconocimiento del usuario mediante él. El inconveniente es que es un sistema cerrado y propietario, que usa un dispositivo propio y dedicado y del que se desconoce, por supuesto, sus tasas de acierto. Sin embargo, a nivel de investigación, aunque es más fácil encontrar trabajos, no es una biometría tan ampliamente estudiada como otras, por lo que se puede considerar una biometría emergente, en este sentido; y más si se se captura con dispositivos comerciales, donde el número de trabajos es pequeño. Por tanto, existen algunos aspectos bajo discusión como son la fiabilidad, su comportamiento en situaciones reales, la configuración de los sensores utilizados o los algoritmos a usar a la hora de autenticar. Una ventaja que tiene el electrocardiograma es que no es sencilla de falsificar frente a las otras características más usadas, para las cuales ya se conocen casos de adulteración mediante guantes para la huella dactilar, lentillas para el iris, o maquillaje para el reconocimiento facial.

1.2. Objetivos

La meta principal de este trabajo es analizar el posible uso de dispositivos posibles comerciales para el reconocimiento biométrico mediante el análisis del electrocardiograma. Este objetivo puede dividirse en varios más específicos, que nos permitan la creación de una ruta para determinar si la línea de trabajo es interesante, o si por el contrario, es mejor no seguir indagando en el tema. Estos son:

1. Analizar el mercado de *wearables* con la función del electrocardiograma disponible, junto al estudio de la obtención de los datos crudos para cada uno de ellos.

2. Aprender y entender el funcionamiento de un sistema biométrico.
3. Obtener un corpus biométrico que nos permita ejecutar, valorar y medir la eficacia de nuestro sistema.
4. Crear un sistema de reconocimiento biométrico basado en ECG. Para el cual partiremos de uno creado en un TFM anterior, por Irene Salvador Ortega, en un trabajo similar pero correspondiente a la forma de caminar de un individuo [3]. Por tanto, será necesario su acondicionamiento para el caso particular de los electrocardiogramas.
5. Discutir de los resultados obtenidos para sacar las conclusiones de la eficacia de este tipo de sistema.

1.3. Objetivos personales

Dado el carácter educativo de este trabajo, los fines de este a título personal se centran en mejorar y poner a prueba mis competencias como Ingeniero Informático:

- Ligado al objetivo principal del trabajo, me centraré en el aprendizaje del funcionamiento y creación de un sistema biométrico.
- Aprender las bases de un lenguaje nuevo para mí, como es *PHP*.
- Trabajar con una base de datos propia y reforzar mis conocimientos de *SQL*.
- Conocer los problemas que se pueden sufrir a la hora de realizar un proyecto en solitario.
- Aprender a seguir una planificación lo más parecida a lo que se plantea en el mundo laboral.

1.4. Estructura de la memoria

La organización del presente documento es la siguiente:

Capítulo 1: Introducción. En este capítulo se presenta la causa que promovió la idea del trabajo y qué objetivos se pretenden respecto al proyecto y al estudiante.

Capítulo 2: Conceptos teóricos. En este capítulo se alojan explicaciones breves sobre las ideas que se utilizarán posteriormente.

Capítulo 3: Trabajos relacionados. En este capítulo se presentan algunos de los últimos trabajos cuyo objetivo principal era el reconocimiento mediante el ECG, además se comentarán sus distintos experimentos y resultados obtenidos para cada uno de ellos.

Capítulo 4: Planificación y herramientas. En este capítulo se explicará la metodología elegida, la programación del proyecto y la estimación económica prevista para este.

Capítulo 5: Desarrollo del trabajo. En este capítulo se expondrán las diferentes fases o iteraciones por las que ha pasado el estudio y las diferentes tareas realizadas.

Capítulo 6: Conclusiones y trabajos futuros. En este capítulo se expondrán las conclusiones finales obtenidas respecto a los objetivos fijados y a los resultados conseguidos. Además se propondrán posibles líneas futuras para mejorar y enriquecer el proyecto.

Capítulo 2

Marco teórico

2.1. ¿Qué es la biometría?

Se denomina **biometría** al reconocimiento automatizado de individuos a partir de sus características biológicas o de comportamiento [4].

Las huellas dactilares, la retina, el iris, los patrones faciales, las venas de las manos o la geometría de la palma de la mano, componen ejemplos de características físicas, mientras que algunos ejemplos de características del comportamiento son la firma, el paso o el tecleo. Una comparativa de las principales características usadas en la biometría se encuentra en la figura 2.1.

Estas características son relativamente constantes y diferenciables, por lo que su uso tiene como fin sustituir o, al menos, ampliar los sistemas de contraseñas para ordenadores, *smartphones* y edificios con acceso restringido.

Algunos usos de la biometría son: la protección de documentos oficiales mediante el reconocimiento de voz, por parte de *CitiBank*, el control de acceso a zonas restringidas mediante el reconocimiento de las venas del dedo, por la empresa Hitachi [5], Korean Air, en cambio, usa las venas de la palma para la identificación de los pasajeros a sus distintos vuelos [6], la inclusión de características físicas como el iris, la huella dactilar y una fotografía digital del rostro del portador en los nuevos pasaportes electrónicos de Estados Unidos siendo pionera, u otras utilizaciones que puedes encontrar entrando en el directorio de biometría de [7].

El uso de biometría tiene una serie de ventajas como son: la dificultad de falsificación, la comodidad de uso respecto a las contraseñas o que es intrasferible. Pero esta también tiene sus inconvenientes, entre los que se encuentran:

la complejidad de puesta en marcha de un sistema biométrico, la posibilidad tanto de falsos rechazos como de falsas aceptaciones, la piratería en sus bases de datos, la no alternativa, dependiendo del rasgo, a “cambiar la clave” (ej. si “roban” las huellas dactilares de una persona, la seguridad queda comprometida sin solución), o la desconfianza del usuario.

	Ojo (Iris)	Ojo (Retina)	Huellas dactilares	Geometría de la mano	Escritura y firma	Voz	Cara
Cómo trabaja	Captura y compara los patrones del iris	Captura y compara los patrones de la retina	Captura y compara patrones de la huella digital	Mide y compara dimensiones de la mano y dedos	Captura y compara ritmo, aceleración, y presión de la firma	Captura y compara cadencia, pitch, y tono de la voz	Captura y compara patrones faciales
Fiabilidad	Muy alta	Muy alta	Alta	Alta	Media	Alta	Alta
Facilidad de uso	Media	Baja	Alta	Alta	Alta	Alta	Alta
Prevención de ataques	Muy alta	Muy alta	Alta	Alta	Media	Media	Media
Aceptación	Media	Media	Media	Alta	Muy alta	Alta	Muy alta
Estabilidad	Alta	Alta	Alta	Media	Baja	Media	Media
Posibles Incidencias	Luz	Gafas	Ausencia de miembro	Edad, ausencia de miembro	Edad, cambios, analfabetismo	Ruido, temperatura y meteorología	Edad, cabello, luz
Costo	Muy alto	Alto	Bajo	Bajo	Alto	Alto	Medio

Figura 2.1: Comparativa de los principales sistemas biométricos [8]

Existen 2 conceptos que pueden llevar a confusión y usarlos hablando sobre biometría como si fueran equivalentes, es por ello que antes de adentrarnos en este mundo, vamos a aclarar cada uno de ellos. Estos términos son los siguientes:

- **Autenticación.** Consiste en comprobar que alguien es quien dice ser, es decir, una persona aporta un rasgo biométrico, siendo este comparado con otro, guardado en el momento en el que el individuo se inscribió y estando este en la base de datos.
- **Identificación.** A diferencia del término anterior, en la base de datos no tendremos un único individuo sino varios. Por tanto, contrastaremos con todos hasta identificar a la persona.

2.2. Electrocardiograma

El **electrocardiograma** (ECG) es la señal que permite describir el comportamiento eléctrico del corazón. Esta señal está compuesta por una adición de ondas que conforman el complejo cardíaco: onda P, onda Q, onda R, onda S y onda T [9], por orden de ocurrencia como se muestra en la figura 2.2, donde, además, podemos observar la relación de cada una de estas ondas con los componentes del sistema eléctrico del corazón.

El ECG forma parte de las exploraciones básicas en cardiología. Es usado de manera frecuente en el diagnóstico inicial de muchas cardiopatías, siendo útil para la valoración de más de un 80 % de ellas, permitiendo detectar crecimientos en las distintas cámaras cardíacas (aurículas y ventrículos) o alteraciones en el ritmo del corazón.

La manera más corriente de extraer un electrocardiograma es través de electrodos, los cuales detectan los pequeños cambios eléctricos, consecuencia de la despolarización del músculo cardíaco, seguida de la repolarización durante cada ciclo cardíaco, es decir, cada latido. Se puede conocer más sobre las 5 fases del potencial de acción cardíaco en la revista médica Medigraphic [10].

Hay varias razones que hacen al electrocardiograma una señal interesante para incluir en un sistema biométrico, la principal ventaja es su robustez frente a robo o imitación. Además, cabe mencionar que existe una propiedad que la diferencia entre las demás características usadas, y esta es que el propio ECG es un indicador de vida, lo que garantiza que la aplicación pueda asegurarse de que el individuo es realmente él, no como en el caso de las huellas dactilares o la cara, que ya se ha visto que son replicables, haciendo vulnerables los sistemas que las usan.

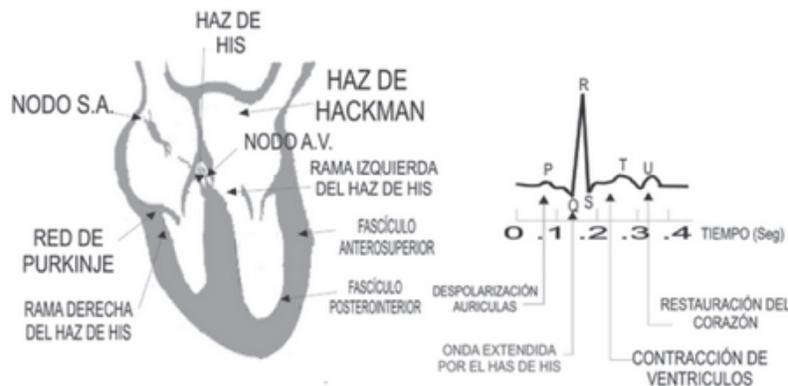


Figura 2.2: Sistema eléctrico del corazón y ECG básico

2.3. Sistema biométrico

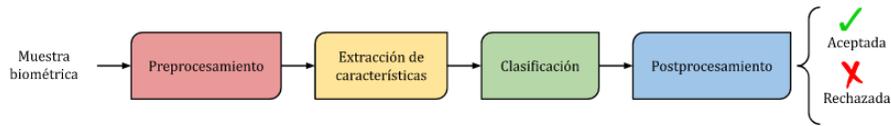


Figura 2.3: Etapas de un sistema biométrico

En esta sección se explicarán los fundamentos de un **sistema biométrico**, entendiendo por este como un procedimiento automatizado que realiza labores de biometría basándose en cualidades físicas o psicológicas. Está formado tanto por componentes hardware como software. La parte del hardware incluye componentes electrónicos y sensores capaces de leer datos de patrones biométricos específicos. En cambio, la parte software hace uso de algoritmos para mejorar el procesamiento de estas características y reconocer estos datos. Siguiendo la figura 2.3 tiene las siguientes etapas:

1. Preprocesamiento

En esta etapa se limpia y prepara la señal cruda para facilitar su tratamiento en fases posteriores. Entre las transformaciones que se aplican en esta fase tenemos:

- **Limpieza de la muestra**

Consiste en eliminar las zonas de ruido de nuestra señal. Estas consistirán en zonas donde la adquisición no respeta la periodicidad del electrocardiograma. En nuestro caso, la detección de este problema se hará mediante un análisis visual de las señales adquiridas.

- **Normalización de la amplitud**

Transformación necesaria en el uso de técnicas de *Machine Learning*, para así lograr una escala común en los datos.

- **Aplicación de filtros**

El filtrado de la señal tiene como finalidad, en nuestro caso, la consecución de un electrocardiograma suavizado. Dependiendo de cómo se devuelvan los datos crudos, será necesario o no.

2. Extracción de características

En esta fase nos encargaremos de pasar la señal a una representación matemática susceptible de ser tratada por los distintos clasificadores que presentaremos en el siguiente apartado. Al trabajar con ECGs, estos son señales temporales, es decir, tienen naturaleza continua, como la voz o la forma de andar. Además, son cuasiperiódicas, esto significa que son

aproximaciones a una señal periódica, las cuales se repiten exactamente iguales en tiempos iguales. Por todo esto la extracción se dividirá en varias etapas.

a) División en ciclos

Tiene como objetivo aislar cada electrocardiograma individualmente.

b) División en ventanas temporales

Cada una de estas ventanas estará formada por un grupo de ciclos anteriores. Un procedimiento normal en este tipo de etapas es usar un solapamiento entre ventanas, esto favorece a una variación más suavizada de las características. Una explicación gráfica se puede observar en la figura 2.4.

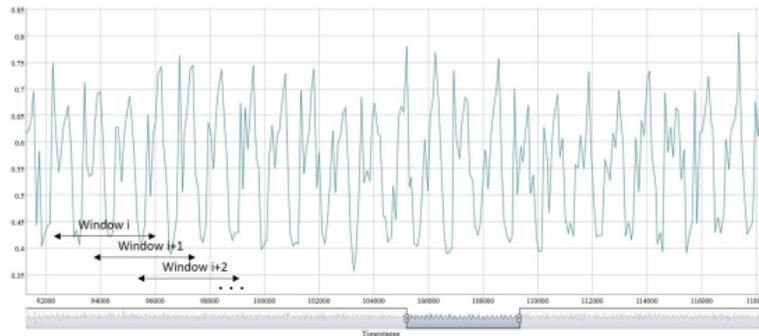


Figura 2.4: Ejemplo de ventana con solapamiento

c) Extracción de características de cada ventana

Al trabajar con señales temporales las características se pueden obtener de dos dominios diferentes: el dominio del tiempo y el dominio de la frecuencia. Las propiedades extraídas para cada dominio son las siguientes:

- **Dominio del tiempo.** Se extraerán las siguientes características: Media, Mediana, Máximo, Mínimo, Desviación típica, Rango máximo, Kurtosis, Percentil 25, Percentil 75, Coeficiente de asimetría, Energía y Máximo valor de autocorrelación
- **Dominio de la frecuencia.** Además de las características extraídas para el dominio anterior, se añaden las siguientes: Amplitud máxima, segunda Amplitud máxima, primera frecuencia dominante y segunda frecuencia dominante.

3. Clasificación

La disyuntiva a la cual se enfrentan los sistemas biométricos es binaria, ya que consiste en aceptar o rechazar la identidad reclamada por el usuario, a través de contrastar una muestra tomada en el momento de acceso con

el modelo del usuario en la base de datos. Los algoritmos que hemos escogido para nuestro proyecto, que se encuentran explicados de forma más extendida en [11], son los siguientes:

- ***K-Nearest Neighbor***

Es un algoritmo de aprendizaje supervisado no paramétrico y basado en instancias. Consiste en el cálculo de la separación entre el punto a clasificar y el resto usando un tipo de distancia determinado. Entre los más comunes se encuentran: euclídea, manhattan, minkowski. Tras dicho cómputo se buscan los K más próximos, para después votar en función de las etiquetas a esos vecinos. En nuestro caso, usaremos 1-vecino más próximo, por lo que le asignaremos la etiqueta de dicha muestra.

- ***Random Forest***

Es un método de aprendizaje supervisado que se fundamenta en la ejecución de varios árboles de decisión, y para el caso de clasificación, ya que puede ser utilizado también en regresión, cada uno de ellos da una clasificación, para finalmente etiquetar la muestra en cuestión a la clase con mayor número de votos.

- ***Naïve Bayes***

Algoritmo fundamentado en el Teorema de Bayes.

- ***Bagging***

Es un método de ensemble de modelos. Se apoya en la técnica de remuestreo *bootstrap*, ya que escoge varios conjuntos aleatorios de muestras del conjunto global, para después entrenar esos modelos por separado. La salida final es la combinación de las salidas de estos submodelos.

- ***Support Vector Machines (SVM)***

Es un algoritmo que se basa en la búsqueda del hiperplano óptimo que realice la separación de los datos en dos espacios dimensionales, en los que cada uno de ellos se encuentran bien diferenciadas dos clases.

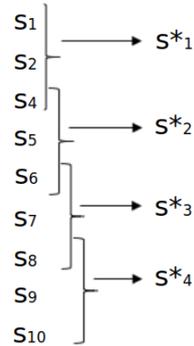


Figura 2.5: Ejemplo de fusión de ventanas en postprocesamiento, para $n = 3$ y solapamiento igual a 1.

4. Postprocesamiento

En esta última etapa se busca solucionar un dilema que se provoca con el uso de los clasificadores, que es la pérdida de la relación temporal entre ventanas. La solución propuesta en nuestro caso se denomina **fusión de ventanas**, y consiste en la asociación de n ventanas consecutivas. Un ejemplo de esta operación se encuentra en la figura 2.5. Este tratamiento se puede ubicar tanto en la etapa de extracción de características como en la de clasificación. En este trabajo se realiza en la clasificación, fusionando mediante la mediana las salidas del clasificador de n ventanas consecutivas. Se usa este tipo de estadístico, ya que fue el que mejor resultados presentó en el trabajo de Irene Salvador [3].

Capítulo 3

Trabajos relacionados

Este capítulo sirve para echar una vista a nuestro alrededor y ver el estado en el que se encuentra la situación del electrocardiograma en el mundo de la biometría. Esto nos permitirá tener una orientación de la forma de trabajar de otros investigadores con dicha característica, así como observar sus resultados y conclusiones.

Existe un número considerable de trabajos relacionados, por tanto vamos a destacar unos cuantos, priorizando los de mayor actualidad encontrados. Destacamos [12], donde los investigadores se centraron en la autenticación mediante ECG teniendo en cuenta las diferencias que pudiera ocasionar a la señal que el individuo estuviera en reposo o se encontrará después de realizar un entrenamiento básico. Para ello elaboraron una base de datos propia, disponible en [13], donde capturaron el electrocardiograma a 45 individuos, 33 hombres y 12 mujeres, cuyo rango de edad se encontraba entre los 18 y 22 años. Se tomaron 5 minutos previos a la actividad física y 150 segundos posteriores a ella. Con esto se plantearon hasta 10 tipos de experimentos diferentes ideando 4 tipos de escenarios:

- Los datos de entrenamiento y test se cogían únicamente de las muestras en reposo.
- Los datos de entrenamiento y test se cogían únicamente de las muestras posteriores a la actividad física, siendo el primer 70 % los que formarían el conjunto de entrenamiento y el 30 % restante el conjunto de prueba.
- Los datos de entrenamiento y test eran escogidos de la forma anterior pero utilizando el 70 % último para entrenamiento y el 30 % recogido primero en prueba.

- Los datos de entrenamiento eran cogidos de las muestras en reposo, mientras que para test se utilizarían los posteriores al ejercicio.

En cuanto a los resultados obtenidos para el primer escenario se tuvo un 90 % de precisión en la mayoría de los experimentos planteados. Mientras que para los escenarios que solo usaban las muestras post-ejercicio esta eficacia se vio reducida en porcentajes variables dependiendo del experimento. Para el último escenario, el cual era el objetivo principal del estudio, los resultados no fueron muy satisfactorios, siendo menores de un 20 % en todos los experimentos excepto para aquel cuya extracción de características estaba basada en Kullback-Leibler, consiguiendo una precisión en el conjunto de test de un 61.4 %.

Un trabajo donde se usan varios conjuntos de datos, aunque esta vez no propios, es [14]. Aplicaron distintos enfoques a 6 bases de datos, categorizadas por “*on-the-person*” y “*off-the-person*”, cuya diferencia se encontraba en la tipología de los dispositivos utilizados para medir las señales de ECG, donde la primera clase usaba sensores corporales en la superficie del cuerpo del individuo siendo necesario un gel conductor, como es el caso de los electrodos convencionales utilizados para el diagnóstico médico. En cambio, la otra categoría hacia uso de dispositivos con electrodos secos, teniendo este tipo de conjunto una mayor cantidad de ruido. Sumando los individuos de cada una de ellas llegaban hasta 1691 sujetos, siendo la situación de estos muy diferente entre ellos, en ámbitos como la salud o la edad.

En cuanto al enfoque de su experimentación podemos diferenciar distintos métodos en cada una de las etapas del sistema biométrico de autenticación que plantean. Las posibilidades que plantearon por etapas son las siguientes:

- Preprocesamiento - Usaron dos tipos de filtros para la separación de la característica biométrica del ruido. Estos fueron el filtro de Kalman y de respuesta de impulso infinito.
- Segmentación - Se realizó con el objetivo de limitar el tamaño de la señal para la siguiente etapa, buscando patrones repetidos en la composición de las ondas P, QRS y T. Estos tipos de segmentación eran FW y RR.
- Extracción de características - Existen dos puntos de vista a la hora de extraer características en un electrocardiograma: el primero se centra en la detección de las principales ondas, denominado “*fiducial*”, mientras que el otro trata la señal como un todo aplicando análisis en el dominio del tiempo o la frecuencia y sacando propiedades estadísticas, designado como “*non-fiducial*”.
- *Matching* - Como sabemos, en esta etapa se mide la coincidencia con la plantilla creada a través de las muestras de inscripción. Los métodos de correspondencia usados en este artículo fueron: la distancia euclídea y la deformación temporal dinámica (DTW).

Aplicando esta experimentación por separado a cada una de las bases de datos, las conclusiones fueron que el algoritmo de coincidencia DTW funciona algo mejor para los diferentes tipos de extracción de características y de segmentación, aunque dicho proceso era más lento que aplicando la distancia euclídea. Además, para las 3 bases de datos del tipo “*off-the-person*” la tasa de falsos positivos era algo peor en el enfoque “*non-fiducial*” debido a que este planteamiento se centra en toda la señal y, como mencionamos anteriormente, este tipo de conjuntos iba a tener mayor cantidad de ruido produciendo este empeoramiento. Respecto a la tasa de equierror, en la cual nosotros centraremos nuestra discusión de resultados, fijandonos en la tabla 4 de [14], la máxima observada es 12 % siendo los valores más usuales los del rango [2.5, 5].

Capítulo 4

Planificación y herramientas

4.1. Metodología de trabajo

La metodología seguida para el desarrollo de este trabajo ha sido de tipo ágil. Por definición, las **metodologías ágiles** son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para adaptar el proyecto y su desarrollo a las circunstancias específicas del entorno. Es por esto por lo que nos hemos decantado por ellas, y más concretamente por *Scrum*, metodología ya conocida por el estudiante, al trabajar con ella durante su periodo de prácticas en empresa.

Scrum es un marco de trabajo contemplado dentro de los métodos ágiles de desarrollo de software. Es un proceso en el que son aplicadas de manera regular una unión de buenas prácticas con el objetivo de optimizar y conseguir mejores resultados en los proyectos. Su objetivo primordial es satisfacer las necesidades del cliente a través de un entorno de transparencia en la comunicación, responsabilidad colectiva y progreso continuo.

En *Scrum* se realizan entregas parciales del producto final, priorizando el valor que puede aportar al beneficiario del proyecto. Se ejecuta en ciclos temporales cortos y con una duración fija (normalmente entre 1 y 4 semanas), intentando conseguir en cada iteración un resultado completo con el objetivo de conseguir una retroalimentación continua del cliente para conseguir un resultado satisfactorio.

Existen 3 papeles principales en lo que configuraría un Equipo *Scrum*, cada uno de ellos tiene diferentes responsabilidades y debe rendir cuentas de distinta

manera, tanto entre ellos como para el resto del grupo. Estos son: Product Owner, que es el encargado de confeccionar los requerimientos del producto, *Scrum Master*, que su labor se centra en la organización del proyecto junto con la solución de problemas que puedan ir surgiendo en este, y el Equipo de trabajo, encargado de desarrollar y sacar adelante las tareas planteadas en cada uno de los sprints.

Además, esta metodología destaca por su alto nivel de comunicación, teniendo hasta 4 tipos de reuniones: diarias, de planificación, de demo y las retrospectivas.

4.1.1. Aplicación en el proyecto

Respecto a los roles, teniendo en cuenta que únicamente tres miembros van a estar involucrados en el proyecto (estudiante y dos tutores), la jerarquía será la siguiente: el estudiante ejercerá simultáneamente de *Scrum Master* principal y equipo de trabajo, y los tutores ejercerán el papel de *Product Owner* y en algunos momentos puntuales de *Scrum Master*. De esta forma el estudiante se encargará de desarrollar el producto y de cumplir con las reglas de Scrum. Mientras que los tutores priorizarán las funcionalidades del producto y ayudarán a resolver distintos obstáculos difícilmente abordables únicamente por el alumno.

En cuanto a las reuniones, se tendrán videollamadas aproximadamente cada 3 o 4 semanas (duración del sprint) cumpliendo así las reuniones de retrospectiva de las semanas trabajadas y la de planificación de la próxima iteración. Las reuniones diarias se sustituirán por *emails* contando el progreso, pero no se harán cada día de trabajo sino únicamente cuando los avances sean importantes. Las reuniones de demo, al ser un proyecto de investigación, no serán muy frecuentes, siendo sobre todo necesarias en las iteraciones 2 y 5 de la siguiente sección.

4.2. Planificación inicial

A continuación se expone brevemente en que consistía el proyecto inicialmente, junto a su diagrama de Gantt con las horas estimadas en cada una de las subtareas planificadas. La suma total estimada es de 266 horas, dejando el resto (34 horas) para la documentación del trabajo. Esto se representa en la figura 4.1.

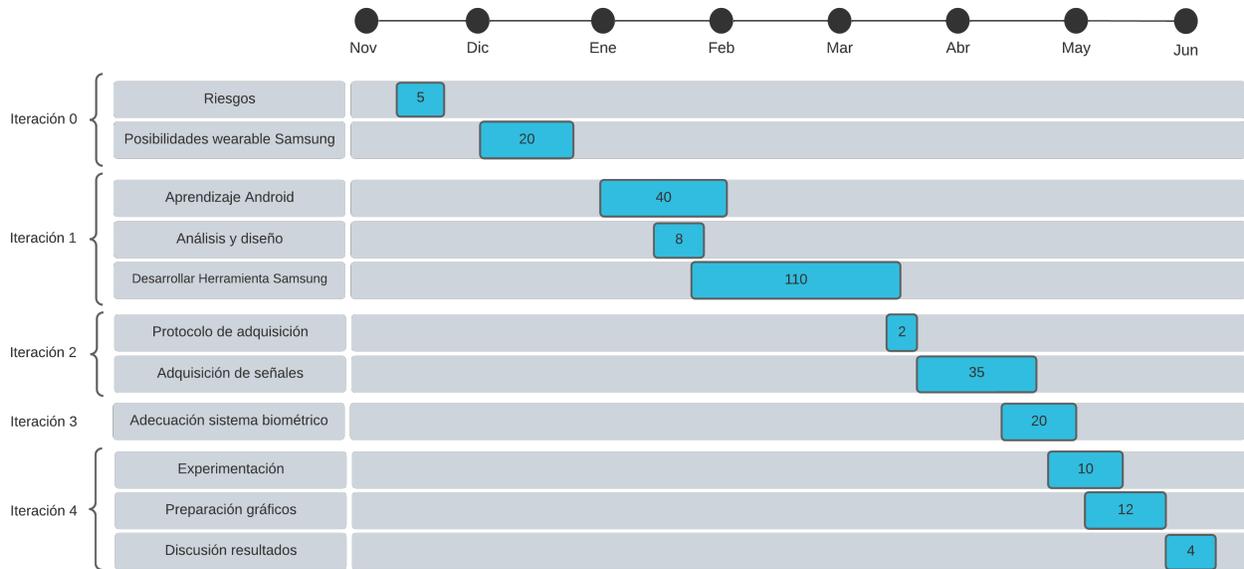


Figura 4.1: Diagrama de Gantt del proyecto

4.3. Riesgos

Antes de mencionar la importancia de la gestión de riesgos en un proyecto, es conveniente conocer que entendemos por **riesgo** (según *PM-BOK*): “Evento o condición incierta que, si se produce, tiene un efecto positivo o negativo en uno o más de los objetivos de un proyecto”.

Al tratarse de un proyecto software es importante detectar los diferentes riesgos que pueden surgir durante el proceso, de esta forma podremos crear un plan de contingencia en el caso de que ocurriera además de realizar una valoración del impacto que podría provocar al proyecto. En nuestro caso encontraremos algún riesgo no característico de un proyecto software dado su enfoque de investigación.

Para cada uno de los riesgos presentados en esta etapa de identificación y análisis de los riesgos se han teniendo en cuenta 2 métricas importantes:

- **Probabilidad:** Se trata de la posibilidad de que el riesgo puede ocurrir. Se asigna uno de los valores de esta escala:
 - Alta: Una probabilidad de ocurrencia mayor que el 50%

- Significativa: Una probabilidad de ocurrencia entre el 30 % y el 50 %.
 - Moderada: Una probabilidad de ocurrencia en el intervalo 10-29 %
 - Baja: Una probabilidad de ocurrencia menor que el 10 %
- **Impacto:** Se trata del grado de efecto que tendría sobre el proyecto, en caso de que el riesgo se materializase. Se asigna uno de los valores de esta escala: {Alto, Significativo, Moderado, Bajo}.

Aparte de las métricas presentadas, en el análisis también podemos encontrar planes de acción, tanto para evitarlos como para tratarlos una vez ocurran.

Identificador	R001
Descripción	Imposibilidad de crear registros del ECG para distintos usuarios mediante un <i>wearable</i> determinado.
Probabilidad	Moderada
Impacto	Alto
Plan de mitigación	Revisar APIs disponibles y proyectos similares.
Plan de contingencia	Plan principal: Buscar alternativas del mercado para detección del ECG mediante relojes inteligentes, incluyendo dispositivos con distintos sistemas operativos. Plan alternativo: Buscar bases de datos biométricas abiertas con datos de ECG.

Tabla 4.1: Riesgo 001

Identificador	R002
Descripción	Fallos técnicos en el equipo de trabajo del estudiante.
Probabilidad	Moderada
Impacto	Significativo
Plan de mitigación	Hacer copias de seguridad de los archivos importantes del proyecto. Trabajar con herramientas de control de versiones.
Plan de contingencia	Recuperar dichas copias para trabajar en otro dispositivo.

Tabla 4.2: Riesgo 002

Identificador	R003
Descripción	Mala estimación de los tiempos
Probabilidad	Significativo
Impacto	Moderado
Plan de mitigación	Reestimación del esfuerzo de las tareas en las reuniones de revisión.
Plan de contingencia	Replanificación de las iteraciones posteriores.

Tabla 4.3: Riesgo 003

Identificador	R004
Descripción	Incorporación de excesivas funcionalidades, es decir, aparición del fenómeno " <i>Gold plating</i> "
Probabilidad	Moderada
Impacto	Moderado
Plan de mitigación	Intentar seguir la planificación con respecto al <i>product backlog</i> .
Plan de contingencia	Reducción de las funcionalidades ajustandose a los requisitos fijados

Tabla 4.4: Riesgo 004

Identificador	R005
Descripción	Interfaces de usuario no amigables o difíciles de utilizar
Probabilidad	Baja
Impacto	Moderado
Plan de mitigación	Respetar los <i>mockups</i> pautados en las tareas de diseño.
Plan de contingencia	Revisar y modificar los diferentes interfaces donde se ha diagnosticado el problema.

Tabla 4.5: Riesgo 005

Identificador	R006
Descripción	Aparición y desconocimiento de una nueva tecnología durante el proyecto.
Probabilidad	Bajo
Impacto	Moderado
Plan de mitigación	Análisis exhaustivo en las diferentes reuniones de las tecnologías a usar.
Plan de contingencia	Búsqueda de material asequible para adquirir las competencias necesarias.

Tabla 4.6: Riesgo 006

Identificador	R007
Descripción	Implicación inferior en alguno de las semanas de trabajo, respecto a lo estimado, por parte del estudiante
Probabilidad	Baja
Impacto	Medio
Plan de mitigación	Análisis de los eventos externos al proyecto en la planificación del trabajo.
Plan de contingencia	Replanificación de las iteraciones posteriores y uso de las semanas de refuerzo.

Tabla 4.7: Riesgo 007

Identificador	R008
Descripción	Modificación de los requisitos del producto.
Probabilidad	Media
Impacto	Bajo
Plan de mitigación	Aclarar los requisitos del producto al comienzo del proyecto. Entrega continua de un producto con valor para obtener aprobación de los tutores.
Plan de contingencia	Modificación de la pila de tareas pendientes intercambiando funcionalidades que sumen la misma o parecida complejidad, en consenso con los tutores.

Tabla 4.8: Riesgo 008

Identificador	R009
Descripción	Corte de funcionamiento de las APIs externas utilizadas
Probabilidad	Media
Impacto	Moderado
Plan de mitigación	Uso de APIs reconocidas y populares.
Plan de contingencia	Ponerse en contacto con los creadores del servicio por todos los medios posibles para gestionar una solución rápida.

Tabla 4.9: Riesgo 009

Identificador	R010
Descripción	Alejamiento de los principios y prácticas de <i>Scrum</i>
Probabilidad	Moderada
Impacto	Bajo
Plan de mitigación	Intentar seguir la planificación en respecto al <i>sprint backlog</i> , completando y priorizando correctamente las tareas. Respetar los eventos e hitos pautados.
Plan de contingencia	Reescritura y rastreo de las tareas planificadas, junto con un seguimiento exhaustivo de las reuniones e hitos fijados

Tabla 4.10: Riesgo 010

4.4. Planificación final

Esta sección está destinada a indicar los posibles riesgos que se han producido, y lo que estos han afectado a la planificación. Dando finalmente una planificación del proyecto real teniendo en cuenta los problemas surgidos.

Comenzando por el riesgo planteado en la tabla 4.1, este era un riesgo que existía en las primeras fases del proyecto, el cual nos hizo cambiar tanto la forma de conseguir las muestras como la planificación como ya se ha comentado, invirtiendo un tiempo en el estudio de la viabilidad del uso de la herramienta de desarrollo de *Samsung* a la cual después no pudimos tener acceso.

Otro riesgo que originó complicaciones en la iteración 3, consistente en la obtención de un corpus, es el presentado en la tabla 4.9. La API que hemos utilizado sufrió unas semanas de mantenimiento, en las cuales actualizaron algunas funcionalidades y durante un tiempo perdimos la capacidad de recuperar correctamente las señales. Tras diferentes mensajes con la plataforma, la recuperación volvió a funcionar con normalidad. Durante estas semanas hubo que pasar a la siguiente iteración preparando el sistema biometrico con las pocas muestras que tenía.

Por último, la implicación del estudiante no pudo ser, en algunas de las semanas de las últimas iteraciones, como se había planificado, riesgo presentado en la tabla 4.7. Es por ello que la entrega del proyecto se alargó hasta la segunda convocatoria. Una planificación más realista y que tiene en cuenta todos estos riesgos, ya que ha sido modificada en base a estos, se presenta en el diagrama de Gantt de la figura 4.2.

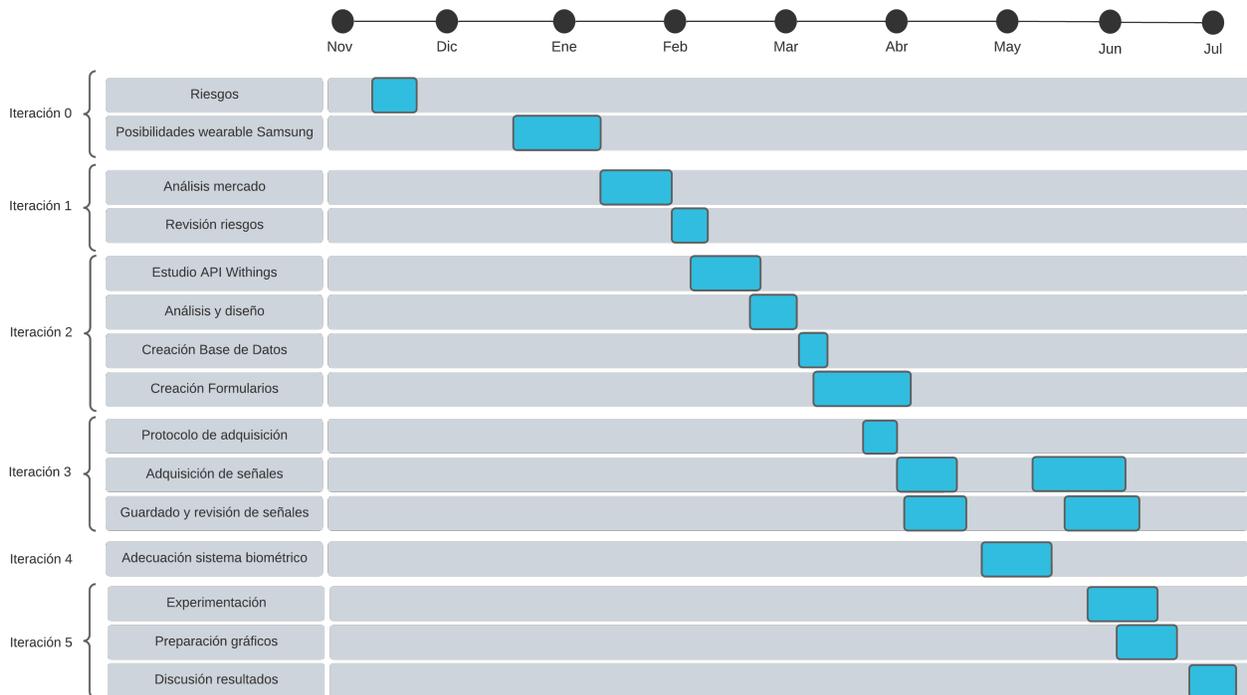


Figura 4.2: Segundo diagrama de Gantt del proyecto

Cabe indicar que se han suprimido las dos subtareas dedicadas a la planificación. Particularmente existen tareas que pueden realizarse al mismo tiempo que otras, como por ejemplo la “Adquisición de señales para los distintos usuarios”,

que debido a que será bastante tediosa se alargará bastante tiempo por la dificultad de búsqueda y reunión con los distintos voluntarios.

Después de indicar los riesgos sufridos, se expondrán detalladamente las 5 grandes etapas o iteraciones que finalmente ha tenido el proyecto, estas se basan en la división de objetivos planteada en la sección 1.2, para facilitar el progreso en el trabajo y atajar mejor los riesgos que se puedan ocasionar, estas etapas se han dividido en subtareas. Cabe mencionar que se va incluir una fase previa en la que se inició el proyecto, que consistía en usar el dispositivo *Samsung* del que ya se disponía.

■ Iteración 0.

Etapla correspondiente al proyecto inicial, consistente en la creación de una aplicación Android que capturará el electrocardiograma a los distintos usuarios a través del sensor del *Samsung*. Esta etapa preliminar consiste en:

- Planificación del proyecto.
- Análisis de riesgos.
- Estudio de las posibilidades de *wearables* de la marca *Samsung*.

■ Iteración 1.

Tras la negativa por parte de *Samsung* para usar su herramienta de desarrollo, esta etapa se centra en el análisis exhaustivo de las distintas alternativas del mercado de dispositivos ponibles en ese momento. Además de listar las distintas opciones deberemos analizar sus distintas formas de acceder a los datos crudos o su sensor de electrocardiograma. Esta primera etapa se compone de:

- Análisis del mercado.
- Replanificación del proyecto.
- Revisión de riesgos.

■ Iteración 2.

Una vez encontrada y seleccionada la opción que más se ajuste a nuestras necesidades, esta segunda etapa se centra en la creación del *software* de adquisición. Esta segunda etapa se compone de:

- Estudio de la API de *Withings*.
- Análisis de software.
- Diseño de software.
- Creación de la base de datos.
- Creación de los distintos formularios.

■ **Iteración 3.**

Después de realizar las pruebas pertinentes al *software* de la iteración anterior podremos comenzar a adquirir nuestro corpus, es decir, conseguir distintas grabaciones de electrocardiogramas de diferentes usuarios. Esta tercera etapa se compone de:

- Preparación protocolo de adquisición.
- Adquisición de señales para distintos usuarios.
- Guardado de las señales en la base de datos.

■ **Iteración 4.**

A partir del momento en el que nuestra base de datos empiece a guardar registros podremos comenzar esta cuarta etapa, que consiste principalmente en la creación de un sistema biométrico. Debido al carácter prospectivo del trabajo, se partirá de otro desarrollado en otros proyectos similares. Esta cuarta etapa se compone de:

- Adecuación del sistema biométrico al ámbito de los electrocardiogramas.

■ **Iteración 5.**

Una vez completadas las dos últimas iteraciones estaremos en disposición de comenzar la experimentación. Esta etapa se centra en reflexionar los diferentes experimentos junto con la preparación de gráficas para su posterior discusión. Esta quinta etapa se compone de:

- Discusión de los parámetros a experimentar.
- Preparación del material gráfico para comparación.
- Discusión de resultados.

En la tabla 4.11 se presentan los tiempos invertidos reales en horas para cada uno de los distintos cometidos divididos en función de la iteración a la que pertenecen.

Tarea	Subtarea	Tiempo invertido (h)
Iteración 0	Planificación del proyecto	4
	Análisis de riesgos	3
	Estudio de las posibilidades de <i>wearables</i> de la marca <i>Samsung</i>	20
Iteración 1	Análisis de mercado	20
	Replanificación del proyecto	3
	Revisión de riesgos	2
Iteración 2	Estudio de la API de <i>Withings</i>	10
	Análisis de software	5
	Diseño de software	5
	Creación de la base de datos	5
	Creación de los distintos formularios	75
Iteración 3	Preparación del protocolo de adquisición	5
	Adquisición de las señales para distintos usuarios	40
	Guardado de las señales en la base de datos	5
Iteración 4	Adecuación del sistema biométrico al ámbito de los electrocardiogramas	25
Iteración 5	Discusión de los parámetros a experimentar	5
	Preparación del material gráfico para comparación	15
	Discusión de resultados	5
Documentación		65
Total		317

Tabla 4.11: Inversión del tiempo esquematizada por subtareas

4.5. Dispositivos de prueba

Para el desarrollo de este proyecto se necesitarán dos dispositivos principalmente, un smartphone y un smartwatch, la descripción de los dispositivos utilizados es la siguiente:

Nombre del dispositivo	OPPO A53s
Sistema operativo	Android 10
RAM	4 GB
Pantalla	6.5"
Procesador	Qualcomm Snapdragon 460
Memoria interna	128GB
Bluetooth	5.0
Batería	5000 mAh

Tabla 4.12: Descripción smartphone de prueba

Nombre del dispositivo	Withings Move ECG
Sistema operativo requerido	Android 6 o superior
Bluetooth	<i>Low Energy</i>
Batería	12 meses

Tabla 4.13: Descripción smartwatch de prueba

Conociendo estos detalles, ya estamos en disposición de pasar a la estimación del coste del proyecto.

4.6. Presupuesto

En un presupuesto lo primero que tendríamos en cuenta es el sueldo de los empleados, en este caso, dado que se trata de un proyecto de Fin de Grado realizado única y exclusivamente por un estudiante, dicho salario podría no tenerse en cuenta. En cambio, para obtener unos costes aproximados del proyecto, un ingeniero informático promedio en España cobra 26500€ al año [15], o lo que es lo mismo 12.74€/hora. Por tanto, dado que la inversión total de horas ronda las 317 horas, esto hace un total de **4038,58€** (sueldo neto). Esos 4038.58€ que se lleva el trabajador no son el gasto real que produce a la empresa, a esto hay que sumarle los gastos para pagar la Seguridad Social y la prestación social, siendo estos de aproximadamente el 30 %, por lo que para que el ingeniero reciba esa cantidad el gasto total que generaría a la empresa sería de 5769.4€ (sueldo bruto), lo que conlleva **1730.82€** (resta entre ambos) a sumar a nuestro presupuesto.

El espacio de trabajo de dicho estudiante será su casa en casi la totalidad del proyecto, exceptuando que alguna de las horas que dedique en el proyecto decida hacerlas en espacios de trabajo comunes como salas de estudio o bibliotecas. Por tanto, teniendo en cuenta que las 300 horas se hagan desde casa y mirando que el consumo medio del portátil del alumno es 73.74 Wh, en 300h será de

22,12 kWh. Teniendo en cuenta que en la mayoría de ocasiones se trabajará en la franja horaria entre las 16h y 21h, el precio medio del kWh es dicha zona (punta), para la compañía Endesa, es de 0.2349 €/kWh [16], resultando un coste total de **5.2€**. Un gasto necesario, es la conexión a internet, que proporcionará al estudiante la búsqueda de información o mantenerse en contacto con sus tutores. Una tarifa de fibra de 300MB con la compañía *Finetwork* [17] cuesta 24.90€/mes, por lo que suponiendo que se comparte la casa con al menos 2 personas, ronda los 8.3€/mes. Por lo que durante los 7 meses de trabajo hace un coste total de **58.1€**. El equipo de trabajo del estudiante es un *HP Notebook* [18] con precio de compra aproximado es de 700 €, si estimamos su vida útil en 4 años, es decir, 48 meses, su coste por mes sería 14.58€. En los 7 meses que se prolongará el proyecto, en condiciones ideales, su coste amortizado es **102.06€**. Los dispositivos de prueba son un *OPPO A53S* [19] y un *Samsung Galaxy Watch Active2* [20], estimando su vida útil de nuevo en 48 meses, y siendo sus precios 219€ y 259,9€ respectivamente, sus costes por mes son de 4.56€ y 5.41€ respectivamente. Siendo sus costes amortizados de **31.92€** y **37.87€**.

Teniendo en cuenta el cambio de dispositivo ponible a un *Withings Move ECG*, disponible por 99.99€ en *Decathlon*. Si seguimos el mismo procedimiento anterior, el coste por mes de este nuevo reloj es inferior, y más concretamente de 2.08€, teniendo un coste amortizado de **14.56€**.

Concepto	Precio unitario	Cantidad	Total
Salario ingeniero	12.74€/hora	317 horas	4038.58€
Coste laboral del trabajador			1730.82€
Electricidad	0.2349 €/kWh	22,12 kWh	5.2€
Conexión a internet	8.3€/mes	7 meses	58.1€
Equipo del estudiante (<i>HP Notebook</i>)	14.58€	7 meses	102.06€
Smartphone de pruebas (<i>OPPO A53S</i>)	4.56€	7 meses	31.92€
Smartwatch de pruebas (<i>Samsung GW Active2</i>)	5.41€	7 meses	37.87€
Smartwatch de pruebas (<i>Withings Move ECG</i>)	2.08€	7 meses	14.56€
TOTAL			5923.14€

Tabla 4.14: Presupuesto del proyecto

4.7. Tecnologías utilizadas

PhpMyAdmin

Herramienta gratuita de software utilizada para poder acceder a la base de datos de manera gráfica a través del navegador.

API Withings

Interfaz para la recuperación de la nube de la empresa *Withings* de las señales grabadas.

Visual Studio Code

Editor de código utilizado para manejar los archivos relacionados con el *software* de recuperación de las señales. Las extensiones más útiles utilizadas fueron:

- Live Server
Servidor web donde ver rápidamente los cambios de nuestra web en nuestro navegador.
- SFTP
Para sincronizar mis archivos locales con los del servidor de manera automática.
- ThunderClient
Para testear el funcionamiento de las llamadas a la API de *Withings*.

RStudio

Entorno de trabajo para programar en el lenguaje *R*, elegido para la implementación del sistema biométrico y la creación de las visualizaciones de los resultados.

Overleaf

Plataforma web usada para la creación online del documento.

Astah

Programa de creación de diagramas UML para el análisis y diseño de *software*.

Canva

Herramienta de diseño gráfico, de carácter sencillo e intuitivo, utilizada para hacer alguna de las representaciones necesarias en la aplicación.

Capítulo 5

Desarrollo

5.1. Iteración 0: Proyecto inicial

En primera instancia, la idea de este proyecto consistía en la elaboración de una aplicación móvil, concretamente para *Android*, que fuera capaz de conectarse con el reloj *Samsung GW Active2* y así ser capaz de acceder a su sensor para capturar el electrocardiograma durante el tiempo que se creyera conveniente. Esto se había hecho para un TFG anterior, pero el sensor que se buscaba era el del acelerómetro, por lo que el inicio de la investigación se hizo a partir de dicho proyecto, que presentaba problemas a la hora de capturar ya que cortaba sin avisar la grabación de la señal.

Tras investigar acerca de los trabajos similares disponibles en internet, se llegó a la conclusión que era necesario utilizar el *SDK* (conjunto de herramientas enfocadas para el desarrollo) de Samsung Health, pero para ello era necesario tener una “membresía” con la empresa, a la cual se podía intentar acceder dando a descargar dicho *SDK* en [21]. Para ello se cumplimentó de manera formal y correcta su formulario de acceso, para después de un par de semanas de espera recibir la negativa.

5.2. Iteración 1: El dispositivo

5.2.1. Análisis de mercado

Debido al rechazo recibido por parte de la compañía *Samsung* para acceder a su sistema de herramientas que nos daría acceso al sensor y por consiguien-

te a los datos crudos del electrocardiograma, se realizó una exploración de las alternativas disponibles en el mercado para proseguir con el trabajo. A continuación, se enumeran las posibilidades encontradas, agrupadas por empresas, junto a las razones que nos hicieron descartarlos o decantarnos en el caso de la opción elegida:

- *Apple*. A esta firma se la considera referencia en el mundo de los electrocardiogramas vinculado a los *smartwatches*. Y es que fue la primera en implementar sensores para dicha funcionalidad a gran escala. Desde su versión del Apple Watch Series 4, y hasta la 7, su última versión en el mercado, posee un sensor para capturar electrocardiogramas.

Las razones que nos llevaron a descartar este tipo de dispositivos fueron principalmente su precio, sumado a que su sistema operativo no era el que se buscaba en este trabajo.

- *Samsung*. Además del dispositivo con el que se contaba, el Samsung Galaxy Watch Active 2, sus dos gamas más altas: el Samsung Galaxy Watch 3 y 4, también poseían sensores capaces de generar ECGs.

Su descarte es por el mismo argumento del dispositivo de partida, explicado en la iteración 0.

- *Fitbit*. Su reloj Fitbit Sense contiene ECG y medidor de la saturación del oxígeno. Su precio, a pesar de ser algo inferior a los anteriormente comentados seguía siendo alto, además su sistema operativo propio, *Fitbit OS*, no nos daba facilidad para ingresar en su programa de desarrolladores, no permitiendo la captura de los datos del sensor, que es lo que se necesitaba.

- *Xiaomi*. Con su producto Amazfit Verge 2, equipado con ECG para detectar arritmias en la persona que lo utiliza. El sistema operativo propio y la dificultad para adquirirlo en el mercado, nos llevó a rechazarlo.

- *Prixton*. Es una marca dedicada a productos deportivos, y su Prixton ECG, de muy reducido precio, alrededor de 50 euros, era una opción bastante interesante.

La falta de información acerca del método para la recuperación de los datos crudos nos llevó a descartar este *wearable*.

- *Withings*. Empresa dedicada al cuidado de la salud mediante tecnología, con 2 *smartwatches* en el mercado con posibilidad de capturar electrocardiogramas: Withings ScanWatch y Withings Move ECG, con algunas opciones más el primero respecto al segundo, pero con un precio más elevado también. La amplia documentación y sus precios razonables nos hicieron decantarnos finalmente por esta compañía, y más concretamente, por su Withings Move ECG. Podemos ver su apariencia en la figura 5.1.



Figura 5.1: Withings Move ECG

5.3. Iteración 2: Software de adquisición

La alternativa escogida permitía tener acceso a los datos crudos una vez eran recogidos y enviados a la nube de Withings. El proceso de describe en la figura 5.2. El protocolo de recuperación consiste en los siguientes pasos:

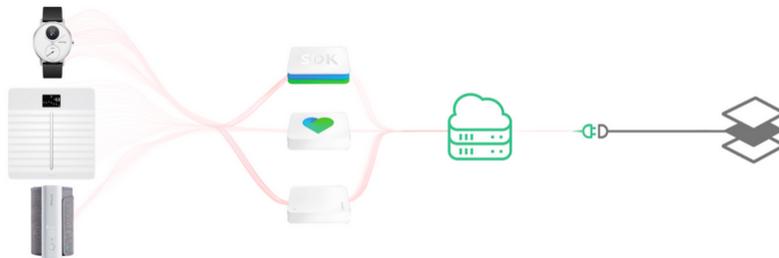


Figura 5.2: Sistema Withings presentado en [22]

1. Darse de alta como desarrollador Withings

Para esto es necesario rellenar un formulario con algunos datos personales, junto con una breve explicación del propósito que tienes al usar sus servicios. Entre los datos a aportar está una *callback URI* donde serán enviados los datos que se soliciten si existe algún retraso por la alta ocupación de

sus servicios. Otro dato que piden es una pequeña imagen del icono de la aplicación que crearás.

La imagen de nuestra aplicación fue creada por la web *Canva* y se muestra en la figura 5.3. En ella se puede ver el nombre de la aplicación junto un reloj en cuyo centro se puede ver un dibujo de un electrocardiograma.



Figura 5.3: Icono ECGetter

Una vez hecha la inscripción se puede acceder al portal de desarrolladores donde se encuentra el *ClientID*, necesario para el siguiente paso.

2. Conseguir el código de autorización

Utilizando el método *authorize*, mediante la siguiente url `https://account.withings.com/oauth2_user/authorize2`, se consigue el código de autorización para acceder a los datos. En ella se deberán indicar los siguientes parámetros de consulta:

- *response_type*. Indicando un *string* con la palabra “code”.
- *client_id*. Tu identificador de cliente disponible en el portal. Una representación de la apariencia de este portal se encuentra en la figura 5.4.
- *state*. Un valor definido por el usuario; es útil para evitar falsificaciones.
- *scope*. Una lista separada por comas de los permisos a los que se quiere acceder, en nuestro caso, “user.metrics”. La lista de los accesos disponibles se encuentra en [23].
- *redirect_uri*. La URI donde se redigirá al usuario después de dar los permisos.

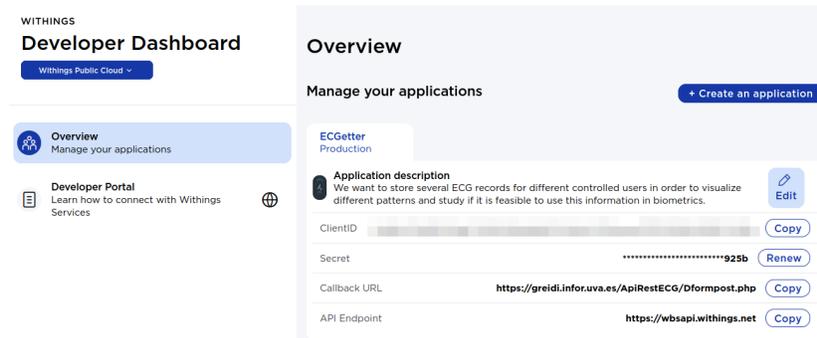


Figura 5.4: Vista del portal de desarrolladores de Withings

3. Conseguir el token de acceso

Este token es necesario para realizar cualquier petición a la API. Una vez los permisos son aceptados, es simple obtenerlos mediante una operación POST a la URL `https://wbsapi.withings.net/v2/oauth2` con los siguientes parámetros de consulta:

- *action*. Con el nombre de la operación a realizar en formato string, en este caso, “requesttoken”.
- *client_id*. El identificador de cliente disponible en el portal. En el paso anterior se ha indicado cómo verlo.
- *client_secret*. Número secreto de cliente, disponible también desde el portal de desarrolladores. De nuevo visible donde localizarlo en la figura 5.4.
- *grant_type*. Indicando un *string* con la palabra “authorization_code”.
- *code*. El código de autorización obtenido en el paso anterior.
- *redirect_uri*. La URI donde se redigirá al usuario después de dar los permisos.

En nuestro caso, nos apoyamos en una extensión de Visual Studio Code llamada Thunder Client para hacer las primeras pruebas. Una operación similar a esta se puede ver en la figura 5.5, donde se ha pixelado los códigos e identificadores personales, y donde se puede observar el formato de la respuesta obtenida. Este token de acceso caduca a las 3h, por lo que será necesario guardar el *refresh_token* para renovarlo. En la figura se muestra cómo refrescarlo.

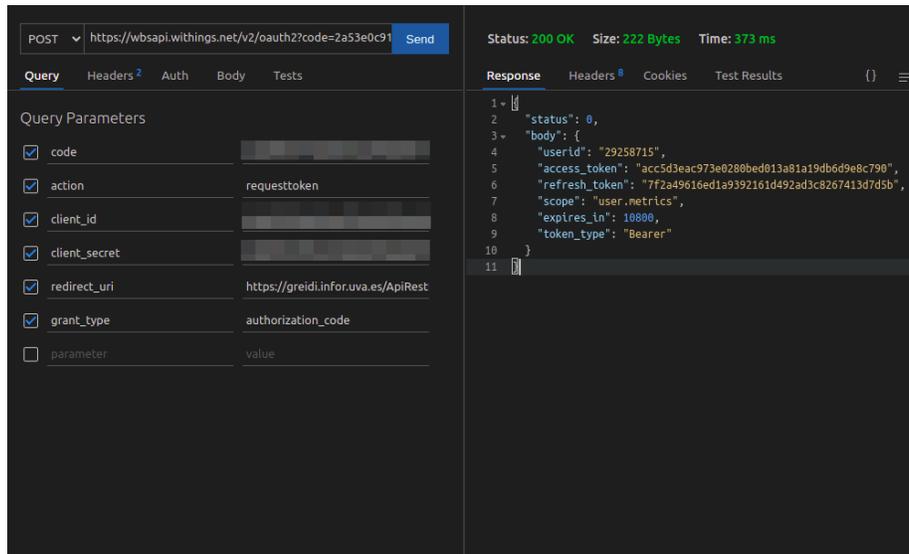


Figura 5.5: Operación POST para obtener un token de acceso

4. Obtener un listado de los ECGs capturados en un intervalo de tiempo

Una vez se tiene un token de acceso disponible, y antes de conseguir los datos crudos, se necesita conocer el identificador de la captura que se quiere recuperar. Para ello la API de Withings tiene una llamada mediante la URL <https://wbsapi.withings.net/v2/heart> y los parámetros de consulta siguientes:

- *action*. Con el nombre de la operación a realizar, en este caso, “list”.
- *startdate*. Fecha de inicio de la medición.
- *enddate*. Fecha de fin de la medición.

Un ejemplo de cómo resultaría la consulta y el formato de salida se encuentran en la figura 5.6. En dicha figura podemos encontrar que el formato de fecha es un número entero, correspondiente a la cantidad de segundos transcurridos desde el 1 de Enero de 1970 a las 00:00:00 UTC.

En la respuesta podemos encontrar información sobre el identificador de dispositivo, el modelo, el instante de captura o si la captura es considerada una fibrilación auricular.

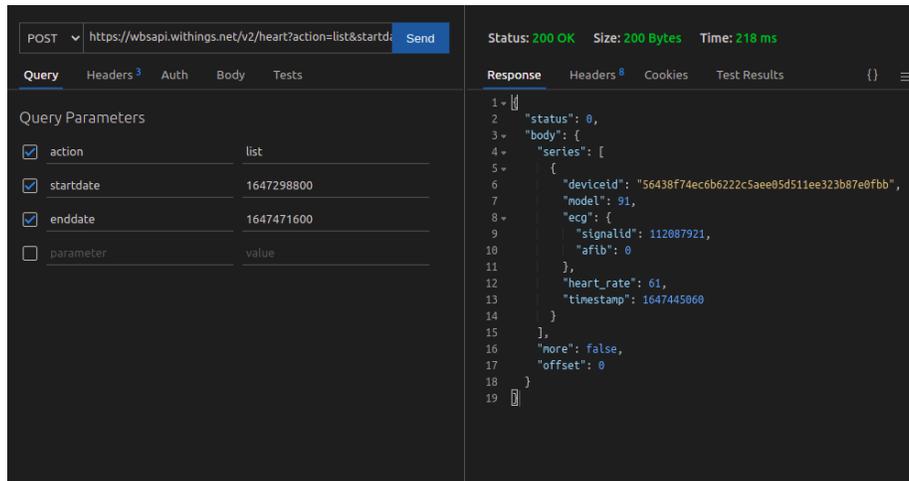


Figura 5.6: Operación POST para obtener un token de acceso

Cabe mencionar que tanto para esta operación como para la del siguiente paso es necesario indicar nuestro token de acceso entre los parámetros de cabecera.

5. Obtención de los datos crudos

Una operación POST similar a la anterior, hacia la misma URL y con dos únicos parámetros de consulta:

- *action*. Con el nombre de la operación a realizar, en este caso, “get”.
- *signalid*. Con el ID de la señal obtenido en el paso anterior.

Los dos últimos pasos serán los que tengamos que repetir para cada recuperación de las capturas obtenidas, salvo que el token haya caducado, y en ese caso el tercer paso se deberá cambiar por la operación de refrescarlo.

Dada la cantidad de capturas que vamos a tener que recuperar, como se verá más adelante, se ha creado un software, que consiste en un formulario con el que recuperar las capturas deseadas de manera sencilla y con el proceso anterior automatizado. Para ello se ha utilizado HTML y CSS para la parte del frontend y PHP para la parte del backend. La forma de dicho formulario es la que se muestra en la figura 5.7.

ECGetter APP

Formulario para datos

Id usuario:	<input type="text"/>
Día:	<input type="text" value="dd/mm/aaaa"/>
Hora inicio:	<input type="text" value="--:--"/>
Hora fin:	<input type="text" value="--:--"/>
Nº muestras (reposo) IZQ:	<input type="text" value="10"/>
Nº muestras (reposo) DRCH:	<input type="text" value="10"/>
Nº muestras (andando):	<input type="text" value="10"/>

Figura 5.7: Formulario para recuperar una toma

5.3.1. Análisis

Requisitos funcionales

- Registrar usuario. El sistema deberá permitir al técnico registrar un usuario con los datos de un voluntario.
- Generar conjuntos de muestras. El sistema deberá generar al técnico el archivo con las muestras para una toma.
- Guardar toma. El sistema deberá permitir al técnico archivar las muestras tomadas en una toma para un usuario concreto.

Requisitos funcionales de información

- Usuario. El sistema deberá guardar los siguientes datos relativos a los usuarios:
 - Sexo
 - Edad
 - Mano dominante
 - Enfermedades
- Datos muestra. El sistema deberá guardar los siguientes datos relativos a las muestras:
 - Identificador de la señal
 - Datos de la señal

- Metadatos muestra. El sistema deberá guardar los siguientes datos relativos a una muestra:
 - Identificador de la señal
 - Fecha de adquisición
 - Hora de adquisición
 - Identificador del usuario
 - Mano portadora
 - Pulso
 - Actividad
 - Sesión
 - Toma
 - Muestra

Requisitos no funcionales

- Base de datos. El sistema deberá utilizar la base de datos que ya está funcionando en la organización, implementada con PHPMyAdmin.
- Formularios. El sistema deberá ser implementado en un servidor Linux/Apache que ya está en servicio.
- Restricciones de uso. El sistema se deberá poder usar tanto desde dispositivos móviles como desde ordenadores.

Casos de uso

En esta sección se describen los tres casos de uso sencillos que presenta nuestra aplicación. El diagrama de casos de uso se encuentra en la figura 5.8; a nuestro sistema únicamente tiene sentido que acceda un técnico biométrico, que conocerá los detalles de las diferentes muestras adquiridas para cada usuario. Para este trabajo en concreto, esa función la desempeñará el alumno, que será el responsable de realizar las adquisiciones vigiladas.

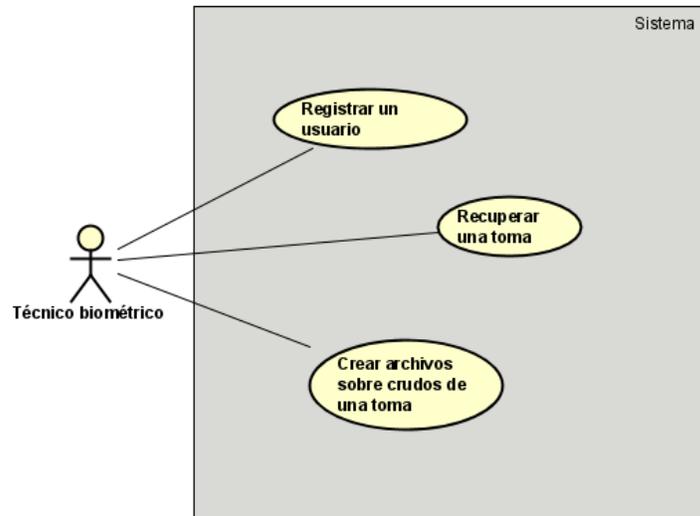


Figura 5.8: Diagrama de casos de uso

Caso de uso	Crear archivos sobre crudos de una toma
Descripción	El sistema permitirá al técnico biométrico indicar los metadatos de un usuario, y con ellos realizar su registro.
Actor	Técnico biométrico
Precondición	El usuario ha sido registrado en la base de datos.
Postcondición	
Secuencia básica	<ol style="list-style-type: none"> 1. El técnico introduce el identificador del usuario. 2. El técnico introduce la sesión. 3. El técnico introduce la toma. 4. El técnico pulsa el botón de Enviar.
Secuencia alternativa	

Tabla 5.1: Caso de uso 3 - Crear archivos sobre crudos de una toma

Caso de uso	Registrar un usuario
Descripción	El sistema permitirá al técnico biométrico indicar los metadatos de un usuario, y con ellos realizar su registro.
Actor	Técnico biométrico
Precondición	El usuario ha indicado correctamente sus metadatos.
Postcondición	
Secuencia básica	<ol style="list-style-type: none"> 1. El técnico introduce el sexo del usuario. 2. El técnico introduce la edad del usuario. 3. El técnico introduce la mano dominante del usuario. 4. El técnico introduce “Ninguna” en el campo enfermedad. 5. El técnico pulsa el botón de Enviar.
Secuencia alternativa	4.1. El usuario padece alguna enfermedad. El técnico introduce el nombre de dicha enfermedad.

Tabla 5.2: Caso de uso 1 - Registrar un usuario

Caso de uso	Recuperar una toma
Descripción	El sistema permitirá al técnico biométrico recuperar y guardar en la base de datos todas las muestras referentes a una toma referenciada con sus tiempos de adquisición.
Actor	Técnico biométrico
Precondición	El usuario ha sido registrado en la base de datos.
Postcondición	
Secuencia básica	<ol style="list-style-type: none"> 1. El técnico introduce el identificador del usuario. 2. El técnico introduce la fecha de la toma. 3. El técnico introduce la hora de comienzo de la toma. 4. El técnico introduce la hora de finalización de la toma. 5. El técnico verifica que se indica el número predefinido de muestras por actividad y mano. 6. El técnico pulsa el botón de Enviar.
Secuencia alternativa	5.1. El usuario ha realizado un número distinto de 10 para alguno de los comportamientos.

Tabla 5.3: Caso de uso 2 - Recuperar una toma

5.3.2. Diseño

Creación base de datos

Como se indicó en la sección de planificación dentro de esta iteración existía la tarea de crear la base de datos. Esto se debe a la necesidad de guardar información tanto para el usuario, ya que se guardarán metadatos que nos puedan ser útiles luego a la hora de comparar el rendimiento del sistema, como para las muestras, que tendrán referencias importantes como el momento de adquisición, el usuario al que pertenece o el ritmo cardíaco presentado.

Tras analizar todos los requisitos que se tiene, el diagrama de la base de datos final se muestra en la figura 5.9. En este diagrama entidad-relación podemos ver

las relaciones existentes entre las distintas tablas. En nuestro caso la base de datos se implementará a través del gestor de bases de datos PhpMyAdmin.

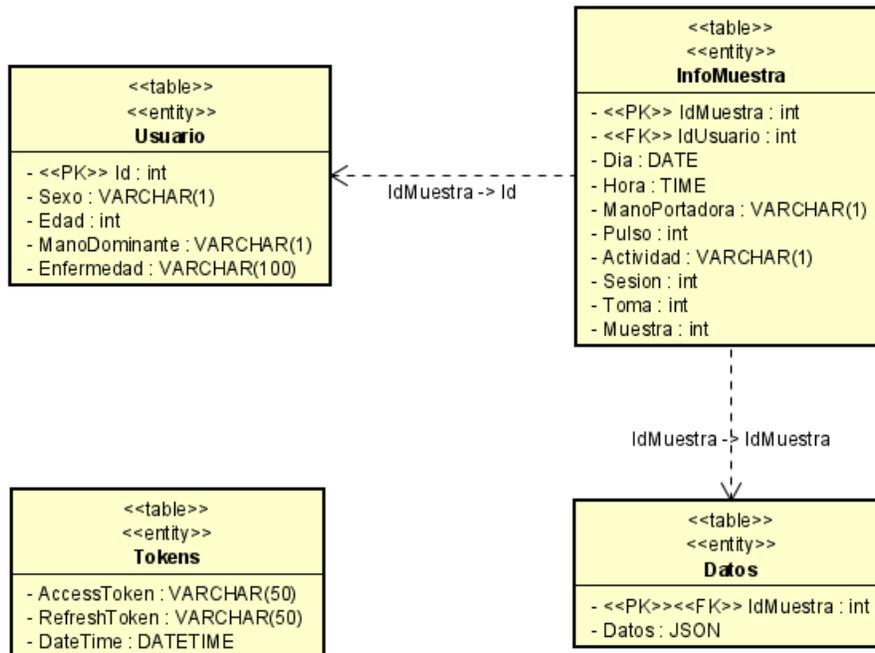


Figura 5.9: Diagrama de entidad-relación la base de datos

La mayoría de atributos de nuestras tablas se explican por sí solos mediante su propio nombre. Quizás la tabla que necesita más explicación es la de “Tokens”, que surge por la obligación ya comentada que una vez se solicita un token de acceso este caduca a las 3 horas, por lo que fue necesario guardar tanto el token de acceso solicitado como el necesario para refrescarlo. Esta tabla únicamente tendrá una instancia, ya que una vez caducada la clave de acceso esta se modifica a la vez que su clave de refresco.

Diagramas de secuencia

A continuación se presentarán los distintos diagramas de secuencia del primer y tercer caso de uso, expuestos en la sección de análisis. Estos son útiles para representar las distintas interacciones entre los componentes de un sistema. Hemos prescindido del caso de uso dos ya que se ha descrito detalladamente en la sección anterior.

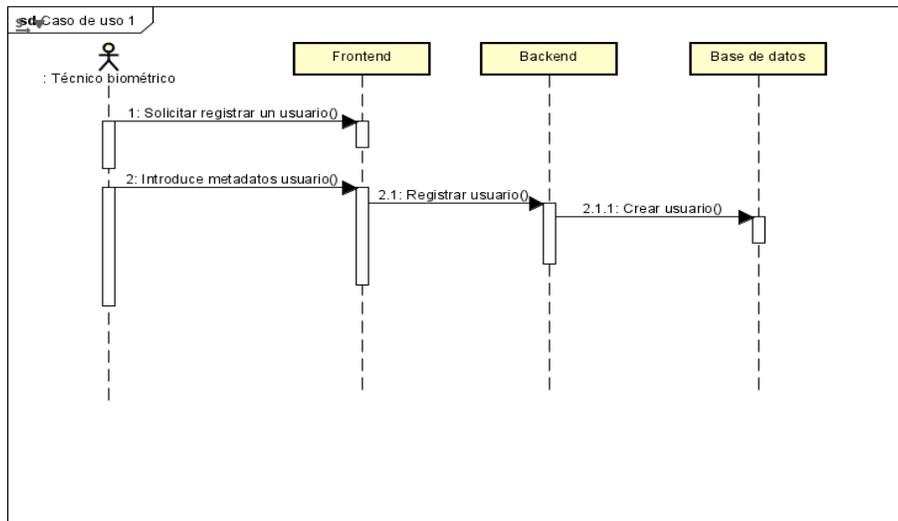


Figura 5.10: Diagrama de interacción caso de uso 1

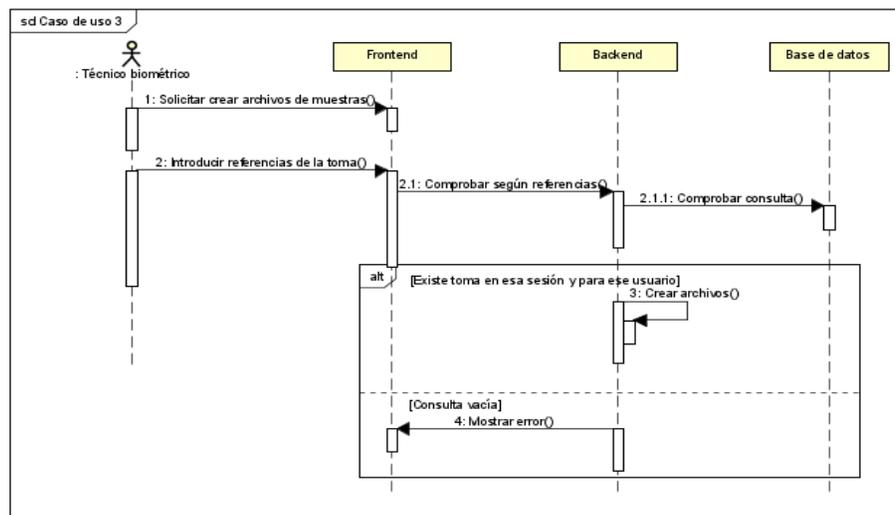


Figura 5.11: Diagrama de interacción caso de uso 3

5.3.3. Pruebas

En esta sección se detallarán los diferentes ensayos a los que se sometió a la aplicación durante su fase de desarrollo a fin de comprobar el correcto

funcionamiento de esta.

Identificador	001
Descripción	Existen 2 usuarios ya creados en la base de datos y se registra uno nuevo
Salida esperada	El identificador del nuevo usuario será igual a 3
Salida obtenida	Id = 3

Tabla 5.4: Prueba 001

Identificador	002
Descripción	El técnico olvida rellenar alguno de los campos de alguno de los tres formularios.
Salida esperada	Mensaje de error “Algún campo se encuentra vacío”.
Salida obtenida	Mensaje en rojo debajo de la caja que conforma el formulario con dicho mensaje.

Tabla 5.5: Prueba 002

Identificador	003
Descripción	La nube que tiene las muestras ha perdido alguna de ellas y no concuerda con la cantidad para alguno de los escenarios.
Salida esperada	Error con mensaje “El número de muestras indicado no concuerda”.
Salida obtenida	Guardado en la base de datos de todas las muestras con su atributo muestra mal numerado.

Tabla 5.6: Prueba 003

Identificador	004
Descripción	Se introduce un parámetro para crear los archivos de datos con las muestras de una toma que no existe en la base de datos.
Salida esperada	Error con mensaje de “Toma no existe” y no se crea ningún archivo.
Salida obtenida	No se crea ningún archivo.

Tabla 5.7: Prueba 004

5.4. Iteración 3: Obtención del corpus biométrico

De cara a aclarar la descripción de corpus biométrico, concepto sobre el que trata esta fase del proceso, se va a indicar la definición técnica de este. La asociación ISO, define en su última publicación sobre estándares en terminología biométrica (ISO/IEC 2382-37:2022) una **base de datos biométrica** como una base de datos compuesta por datos biométricos, siendo estos la representación analógica o digital de una característica biométrica; en nuestro caso, el ECG.

La descripción y el contenido de este corpus biométrico dependerá del objetivo con que se capture. En nuestro caso va a ser, siguiendo los objetivos planteados, la investigación en el uso del ECG obtenido mediante dispositivos comerciales, en el reconocimiento biométrico de individuos.

Antes de pasar a adquirir estos datos, hay que planificar detenidamente su captura, que será lo que se muestre en la siguiente sección.

5.4.1. Protocolo de adquisición

Una vez probado el funcionamiento correcto del *software* de adquisición y antes de buscar voluntarios para crear el conjunto de datos con el que trabajar, es necesario conformar un protocolo de adquisición. Esta parte de planificar la captura de la base de datos biométrica es fundamental, ya que va a definir qué y cómo tomar datos. Estos datos serán usados, posteriormente, para las pruebas. Por lo tanto, el alcance de estas estará condicionado por cómo definamos esta parte. Un mal planteamiento de la captura, puede hacer que, después del esfuerzo realizado en esa, los datos no sean útiles para el trabajo a realizar. A continuación se exponen las normas seguidas para todas las capturas realizadas.

Para los experimentos que se iban a plantear eran necesarias al menos 2 **sesiones**, espaciadas en el tiempo mínimo de 2 o 3 semanas. Esto va a permitir realizar pruebas que tengan en cuenta la variación del rasgo biométrico con el tiempo. Esto es fundamental para poder garantizar la validez del rasgo (ECG en este caso). Normalmente, el clasificador se entrena con las muestras del usuario obtenidas un determinado día; con eso se crea el patrón del usuario, que se usará para su posterior identificación. Generalmente, el rasgo biométrico varía con el tiempo, y si no el rasgo, el comportamiento del usuario al usar el sensor que lo captura. Por eso es fundamental analizar cómo esta variación afectará al rendimiento de nuestro sistema.

Antes de cada una de las sesiones fue necesario establecer los **metadatos** a capturar para cada individuo. En nuestro caso, fijamos:

- Sexo
- Edad
- Mano dominante
- Existencia de alguna enfermedad, siendo más relevantes las cardíacas.

El objetivo de estos metadatos es el de poder realizar en el futuro pruebas teniendo en cuenta sus valores, como por ejemplo, comparar el rendimiento por edad o sexo. En este trabajo, al ser un estudio prospectivo, no se ha tenido en cuenta. Estos datos, además de los de la captura de ECG que se mostrarán un poco más adelante, serán asociados a un identificador del usuario. Este será único y completamente anonimizado, es decir, a partir de ese identificador es imposible saber la identidad real del usuario. En otras palabras, al acabar la captura del usuario no habrá en la base de datos ningún dato personal de él. Esto es fundamental para cumplir con la ley de protección de datos. El identificador que usaremos será un número, asignado de manera incremental a cada voluntario, tomando como número de partida el 1.

Cada una de las sesiones va a estar constituida por dos **tomas**, con una separación de al menos 2 o 3 horas entre toma y toma. En cada una de esas tomas se van a adquirir los siguientes datos de ECG:

1. 10 muestras (5 minutos) en reposo, con el reloj en la muñeca izquierda.

El reloj solo permite adquirir de manera continuada 30 segundos de señal de electrocardiograma. Como no conocemos a priori cuánta “señal” necesitaremos para crear el patrón del usuario, decidimos adquirir un tiempo “suficientemente” largo. Por la experiencia con otros rasgos biométricos, para los que la cantidad suficiente eran 5 minutos, en nuestro caso, determinamos realizar 10 tomas seguidas del ECG con el reloj. La misma dinámica se usa en el resto de adquisiciones.

2. 10 muestras (5 minutos) en reposo, con el reloj en la muñeca derecha.
3. 10 muestras (5 minutos) andando, con el reloj en la muñeca izquierda.
4. Fin de la toma. En caso de ser la primera toma de la sesión respetar el horario de parada para después repetir la toma desde el primer paso.

No existe un descanso estipulado entre cada una de las tres primeras fases. Aun así, no será inmediato debido al cambio de muñeca entre cada una de las etapas. La idea de la variedad de escenarios es poder comparar el rendimiento del sistema con respecto a la mano portadora o la actividad del usuario (en reposo/andando).

Cabe mencionar que para todas las adquisiciones el reloj debe estar vinculado vía *bluetooth* con un *smartphone*, y este tener descargada la aplicación *Health Mate* con una cuenta vinculada. Esto se debe a que el *wearable* tiene memoria interna capaz de almacenar solo 12 capturas de ECG sin estar vinculado a un dispositivo, pero, si la asociación existe, la sincronización es automática y todas las capturas se van guardando al instante. El número de capturas de cada toma es 30, ya que cada captura dura 30 segundos. La adquisición de cada individuo ha sido vigilado por el responsable del trabajo. Esta toma de datos controlada permite la corrección de errores al momento.

5.4.2. Descripción del corpus obtenido

Únicamente se fijó como requisito para la creación de dicho conjunto de muestras, que hubiese equilibrio en la distribución por sexo. En nuestro conjunto de 10 personas, tenemos 6 hombres y 4 mujeres. Una breve descripción de nuestro corpus se presenta en la tabla 5.8.

Nº Usuario	Edad	Sexo	Mano dominante	Muestras sesión 1	Muestras sesión 2
001	22	Hombre	Derecha	61	66
002	22	Mujer	Derecha	60	61
003	32	Mujer	Derecha	61	60
004	59	Mujer	Derecha	60	61
005	60	Hombre	Derecha	59	60
006	23	Mujer	Derecha	63	61
007	22	Hombre	Derecha	66	
008	23	Hombre	Derecha	59	60
009	55	Hombre	Derecha	63	62
010	23	Hombre	Derecha	62	60

Tabla 5.8: Metadatos de los usuarios

Se ha omitido la columna “Enfermedad”, dado que ninguno de los individuos que formaron parte del estudio tenían una enfermedad.

La información completa con las muestras repartidas por sesiones, tomas, horarios y cantidad en cada una de las muñecas se puede encontrar en este enlace: <https://docs.google.com/spreadsheets/d/1uJcbnGtfNae14v1rCa3Zc0DGBzJZsoLgZ2rWw9NdIyU/edit?usp=sharing>.

5.5. Iteración 4: Creación del sistema biométrico

Como ya hemos mencionado, para esta etapa no se comenzaba de cero, ya que se disponía de un sistema de referencia creado en un Trabajo de Fin de Master [3]. El sistema contaba con 7 archivos diferentes, creados también durante dos TFGs anteriores (realizados de forma paralela). En nuestro caso, debido a la limpieza del electrocardiograma que se devolvía y al carácter de dicho rasgo solo necesitaremos 3 de ellos: el que extrae las características y los que aplican los algoritmos.

Para ello se contó con la ayuda de los tutores (con mayor conocimiento de los programas) y, tras varias sesiones de reunión conjunta más cierto trabajo individual del alumno, se adecuaron y optimizaron los programas para el problema que se debía abordar en este trabajo. Además, cabe mencionar que se redujo el número de archivos a 2, dejando uno para la extracción y otro para la clasificación.

Ahora se van a exponer las funciones principales de los dos programas. Para el programa encargado de la extracción de características la función tiene la siguiente estructura:

```
ejecutaTablasExtensionesVentana <- function(usuarios_completos ,
                                             actividad ,
                                             mano ,
                                             extension_ventana ,
                                             solapamiento ,
                                             interpolaLinealmente ,
                                             filtroMediaMovil_Orden5 ,
                                             filtroMediaMovil_Orden3)
```

Como se puede apreciar, es necesario indicar en un vector los usuarios que tiene el sistema, de los cuales extraer las características de la actividad, mano, extensión y solapamiento indicados. El resto de parámetros son *boolean*, y en nuestro caso se mantuvieron a *FALSE*.

Para el programa encargado de la etapa de clasificación la estructura de su función principal es la siguiente:

```

aplicarClasificadores <- function(ruta_salida,
                                   dispositivo,
                                   ruta_inicio_auxiliar,
                                   preprocesamiento,
                                   k,
                                   experimental,
                                   dominio,
                                   numVentanas,
                                   solapamientoVent,
                                   extension_ventana,
                                   solapamiento,
                                   usuarios_completos,
                                   clasificador,
                                   guardar,
                                   actividad,
                                   mano,
                                   balanceo = "ninguno")

```

Los diferentes parámetros explicados son:

- ruta_salida. Directorio de guardado de las salidas.
- dispositivo. Reloj utilizado, en nuestro caso “ECG”.
- ruta_inicio_auxiliar. Directorio donde se encuentran los ficheros con las características extraídas.
- preprocesamiento. En caso de que se haya aplicado algún filtro, en nuestro caso “ninguno”.
- k. El valor de k para el algoritmo k-Vecinos.
- experimental. Vector con los experimentos a probar.
- dominio. Vector con los dominios a probar.
- numVentanas. Tamaño de la ventana en la etapa de postprocesamiento.
- solapamientoVent. Solapamiento incluido en la etapa de extracción.
- extension_ventana. Tamaño de la ventana utilizado en la etapa de extracción.
- solapamiento. Solapamiento a utilizar en la etapa de postprocesamiento.
- usuarios_completos. Vector con los usuarios que tiene el sistema.
- clasificador. Vector con los clasificadores a utilizar.
- guardar. En caso de que se quiera guardar las muestras de entrenamiento, principalmente para hacer algún tipo de comprobación.

- actividad. Actividad que hacía el usuario (enReposo/andando).
- mano. Mano donde tenía ubicado el reloj.
- balanceo. Balanceo a aplicar.

5.6. Iteración 5: Experimentación y resultados

5.6.1. Experimentos

En esta sección se van a comentar los parámetros del sistema biométrico que se han decidido probar y la lista de valores posibles que podrán tomar. Para mejorar la interpretación se exponen en función de la etapa del sistema biométrico donde se pueden modificar, en referencia a las fases explicadas en la sección 2.3.

1. **Preprocesamiento.** El único tratamiento que quizás sería interesante probar sería la aplicación de filtros. Ahora bien, visualizando tanto los ECGs descargados como los mostrados en la aplicación propia de *Withings* se observó que los electrocardiogramas ya estaban suavizados, por lo que no se vio la necesidad de aplicar esta posibilidad. Por tanto, en esta etapa no existirá ningún parámetro del sistema a evaluar.

Como explicamos en la sección 2.3., la parte de limpieza de la muestra de la etapa de preprocesamiento, consistiría en un análisis visual de las señales recogidas. En nuestro caso, preparamos un programa en R, que representará 3 señales aleatorias (es un parámetro de la propia función, se ha fijado en 3 por ser una cantidad que visualmente se veía de forma nítida) de 30 segundos adquiridas para un usuario elegido al azar de nuestra base de datos. Algunos de los resultados se muestran en las figuras 5.12 y 5.13. En ellas podemos ver distintas zonas de ruido, señaladas en rojo, que presentan algunas de las adquisiciones de los usuarios 7 y 4 respectivamente. Para el primero vemos cómo además de ruido en mitad de la señal, también podemos apreciarlo al comienzo de la grabación, un problema común en este tipo de adquisiciones para este tipo de sensores. Para el segundo usuario podemos ver cómo sus 2 primeras señales son bastante regulares, pero la última, adquirida mientras andaba durante la toma 1 de la sesión 1, se muestra bastante caótica, algo que ha ocurrido para la mayoría de usuarios en este tipo de adquisiciones, ya que el propio reloj nos indicaba que el usuario se debía encontrar en reposo (sentado o incluso tumbado) para grabar la señal.

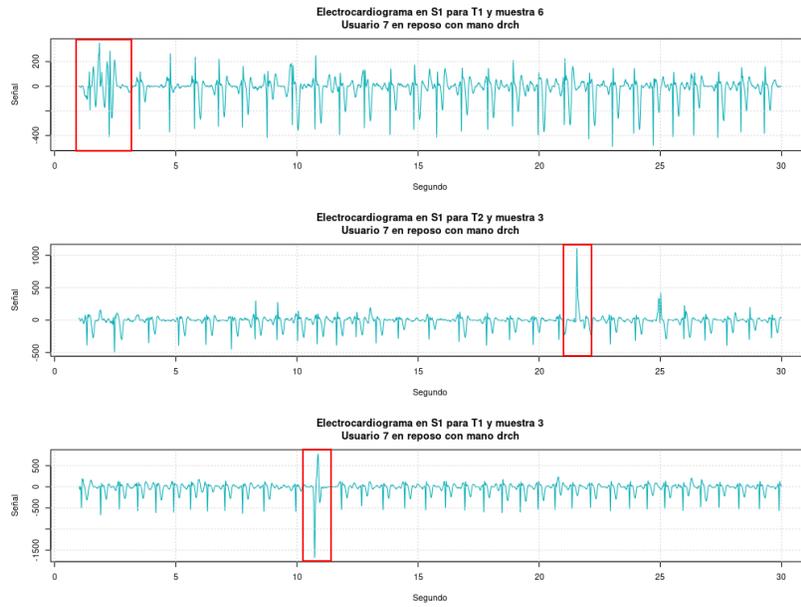


Figura 5.12: Análisis visual Usuario 007

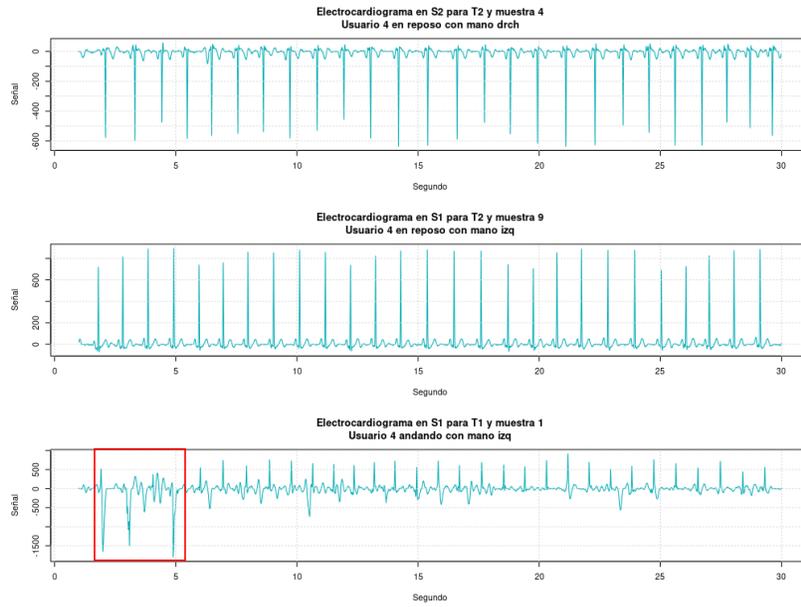


Figura 5.13: Análisis visual Usuario 004

2. **Extracción de características.** Del parámetro que analizaremos su comportamiento será el número de ciclos de los que estará formada cada ventana. Los valores que tomaremos serán 6, 8, 10 y 12. El solapamiento será fijo, y tendrá un valor habitual en este tipo de sistemas, del 20 %, un ejemplo donde se uso ese porcentaje es la publicación [24].
3. **Clasificadores.** Aquí pondremos a prueba 2 criterios: el clasificador y las técnicas de balanceo del conjunto de entrenamiento. En cuanto al primero, probaremos los siguientes algoritmos: 1-NN, Decision tree, Random forest, Naïve Bayes, Bagging, SVM con kernel lineal, SVM con kernel radial, SVM polinomial de grado 4, SVM con kernel sigmoide. Para el balanceo tomaremos 2 valores: sin balanceo y *over sampling*.

En cuanto a la hora de ejecutar la autenticación, primero vamos a contextualizar el problema.

La tarea abordada aquí es verificación o autenticación del usuario. Para ello, para cada uno de ellos se entrena un clasificador o modelo diferente. En la etapa de verificación se tendrá la identidad reclamada (lo que en el sistema de verificación tradicional es el “login”) y la muestra biométrica aportada por el usuario (que hará las veces de “password”). Se irá al almacén de modelos en el sistema y se extraerá el correspondiente al usuario, obteniendo su salida a partir del vector de características extraído de la muestra. Con esta salida o “score” se realizará la decisión de aceptar/rechazar la identidad reclamada.

Desde un punto de vista del reconocimiento de patrones, estamos abordando un problema de clasificación binario: el clasificador debe catalogar la entrada como perteneciente o no perteneciente a un determinado usuario.

Para poder realizar esa clasificación, todos los clasificadores anteriores, salvo el 1-NN, deben ser entrenados con muestras de las clases a reconocer: la auténtica (muestra/s del usuario) y la impostora (muestras de cualquier individuo que no sea el usuario).

Para la primera, de cara a comprobar el efecto de la variable tiempo sobre nuestras muestras y medir la robustez respecto a esta, se presentarán 2 escenarios, usando muestras de entrenamiento idénticas pero siendo sus muestras de prueba las que les hagan distintas.

- Entrenamiento:
 - Muestras auténticas: Las correspondientes a la primera toma de la primera sesión del usuario a autenticar.
 - Muestras impostoras: Las correspondientes a la primera toma de la primera sesión de los usuarios cuyo identificador es par si el usuario a autenticar es impar o de los impares si el usuario es par.
- Prueba:
 - Experimento Monosesión:

- Muestras auténticas: Las correspondientes a la segunda toma de la primera sesión del usuario a autenticar.
- Muestras impostoras: Las correspondientes a la primera muestra de la primera sesión de los usuarios pares si el usuario a autenticar es par y de las impares si este es impar.
- Experimento Multisesión:
 - Muestras auténticas: Las correspondientes a la primera y segunda toma de la primera sesión del usuario a autenticar.
 - Muestras impostoras: Las mismas que para el otro escenario, es decir, las correspondientes a la primera muestra de la primera sesión de los usuarios pares si el usuario a autenticar es par y de las impares si este es impar.

Ahora si estamos en condiciones de explicar en que se diferencian los dos tipos de balanceo utilizados:

- **Sin balanceo**, es decir, sin tocar la distribución ni el orden de muestras.
 - **Over sampling**, equiparando las muestras auténticas a las impostoras, es decir, repitiendo dichas muestras hasta que ambas clases tengan el mismo número.
4. **Postprocesamiento.** Aquí únicamente podemos probar el parámetro que indica el número de *scores* (salida del clasificador para el vector de características extraído de una ventana de la muestra) consecutivos fusionados. Esta variable tomara los valores 1 (no se aplica postprocesamiento), 2, 4, 8 y 16. El solapamiento para esta etapa será de 2 ventanas.

De cara a medir el rendimiento de los algoritmos de clasificación utilizados junto con las diferentes variantes explicadas, analizaremos la **tasa de equierror** (EER - Equal Error Rate), medida comúnmente utilizada en biometría, la cual se define como el punto donde se igualan la tasa de falsos positivos (FAR) y la tasa de falsos negativos (FRR). Podemos encontrar una explicación gráfica en la figura 5.14.

Esta tasa de error se puede medir de dos maneras diferentes:

1. Con umbral individual. En este caso, se calculan las salidas del sistema para las muestras de prueba auténticas e impostoras para cada usuario. A partir de estas salidas (scores) se obtiene el EER (tasa de error) para cada usuario de manera individual. La tasa de error final (EER final) será la media de las tasas obtenidas para cada usuario.
2. Con umbral general. Con todas las salidas (scores) para muestras de prueba auténticas de todos los usuarios se crea un conjunto único. Se hace lo mismo con las salidas para muestras de prueba impostoras. A partir de esos dos conjuntos se calcula el EER (tasa de error) final del sistema.

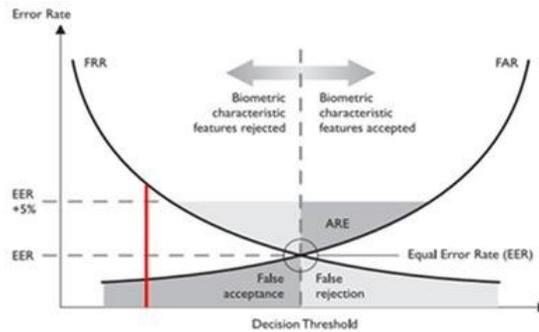


Figura 5.14: Explicación gráfica EER [25]

La diferencia entre ambas radica en que habitualmente las salidas (scores) de un clasificador entrenado para reconocer a un usuario, tienen una distribución diferente que las salidas del clasificador entrenado para reconocer a otro usuario (Fig. 5.15 (a)). Esto hace que la tasa de error con umbral general sea generalmente menor que la obtenida con umbral individual. A esta última la podemos considerar como un cálculo optimista del rendimiento del sistema, en el caso de que fuéramos capaces de poder definir de manera óptima el umbral de decisión para cada usuario, problema no trivial [26].

Para mejorar el EER con umbral general, se suelen aplicar técnicas de normalización del score [26] (Fig. 5.15 (b)).

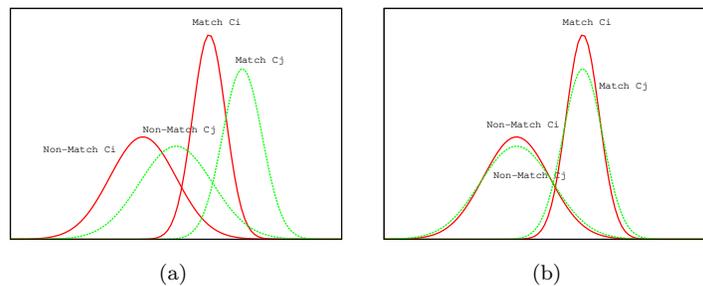


Figura 5.15: Múltiples imágenes

Aquí vamos a mostrar los resultados con ambas formas de calcular el rendimiento del sistema, ya que ambas son muy utilizadas en biometría. Esto servirá, además, para ver si nuestro caso tiene o no el problema descrito de no uniformidad en las distribuciones de salidas del clasificador entrenado para diferentes usuarios: cuanto más cercanos estén los resultados con ambas formas de calcular el EER menor será este problema y a la inversa.

5.6.2. Resultados

Tras esta introducción a la experimentación planteada, se presentarán primero los resultados para los 3 escenarios (reposo en muñeca izquierda, reposo en muñeca derecha y andando en muñeca izquierda) del experimento que confronta solamente las muestras correspondientes a la primera sesión.

Experimentos Monosesión

En ambos dominios, vemos cómo en el caso sin balanceo, SVM con kernel sigmoide no tiene apenas error. En cambio, para el caso de balancear, se convierte en el peor de los algoritmos utilizados, siendo la máquina de vectores soporte con kernel radial la que presenta menor tasa de error. Con respecto al tamaño de la ventana, en general no hay una gran variación, salvo para casos concretos, como por ejemplo *Bagging* en el dominio del tiempo (DT).

En las figuras 5.17 y 5.18, se muestran los mejores algoritmos de cada uno de los 4 gráficos de la figura 5.16, eligiendo siempre el tamaño de ventana que menor tasa de error presentaba. Además de comparar la influencia del parámetro de fusión de *scores* se puede confrontar el umbral individual obtenido con el general.

En ambos casos, el gráfico sin balanceo no presenta cambios, ya que todos son 0. Sin embargo, para el balanceo Over (*oversampling*), en el dominio del tiempo fusionando dos *scores* ya llegamos a una tasa nula, siendo 8 los *scores* fusionados para llegar a cero en el dominio de la frecuencia (DF).

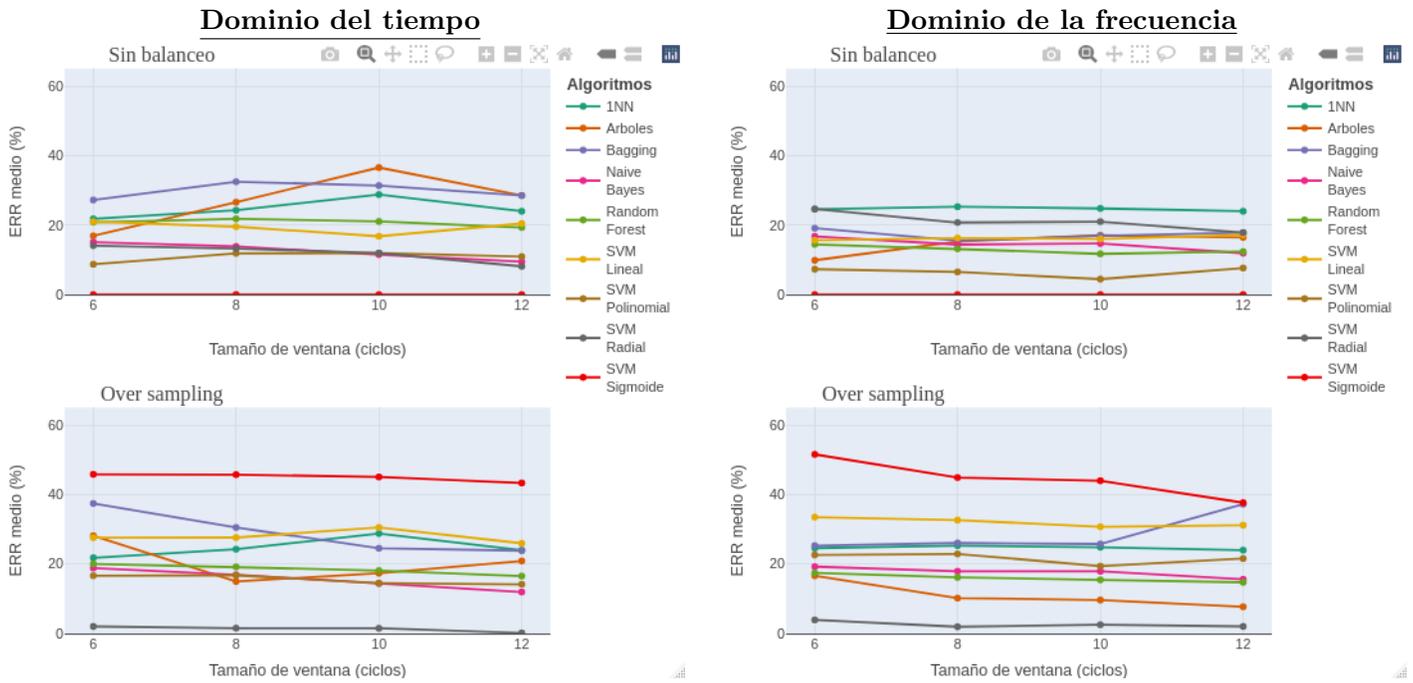


Figura 5.16: Experimento Monosesión-En reposo-Izquierda

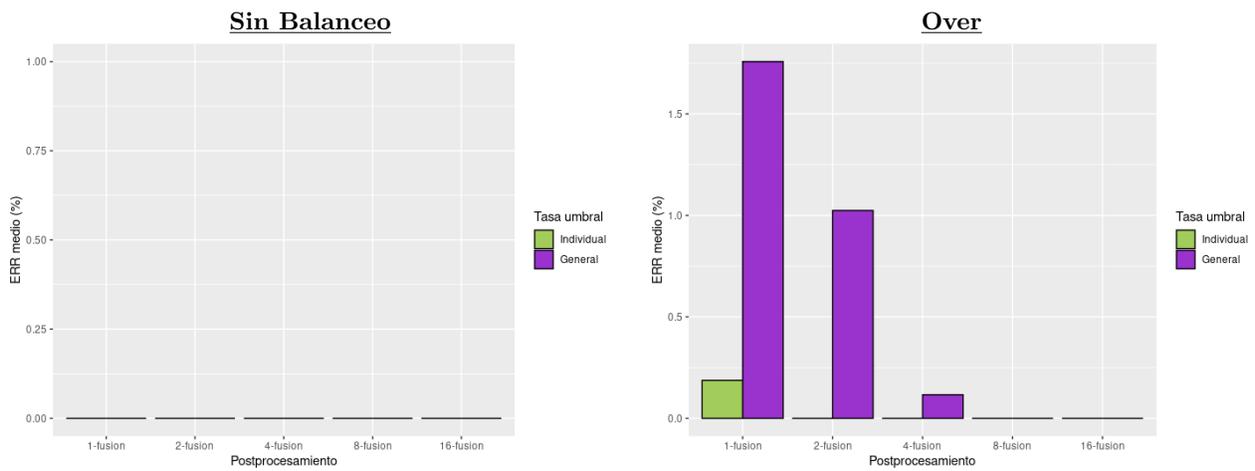


Figura 5.17: Mejores algoritmos para DT-Monosesión-En reposo-Izquierda

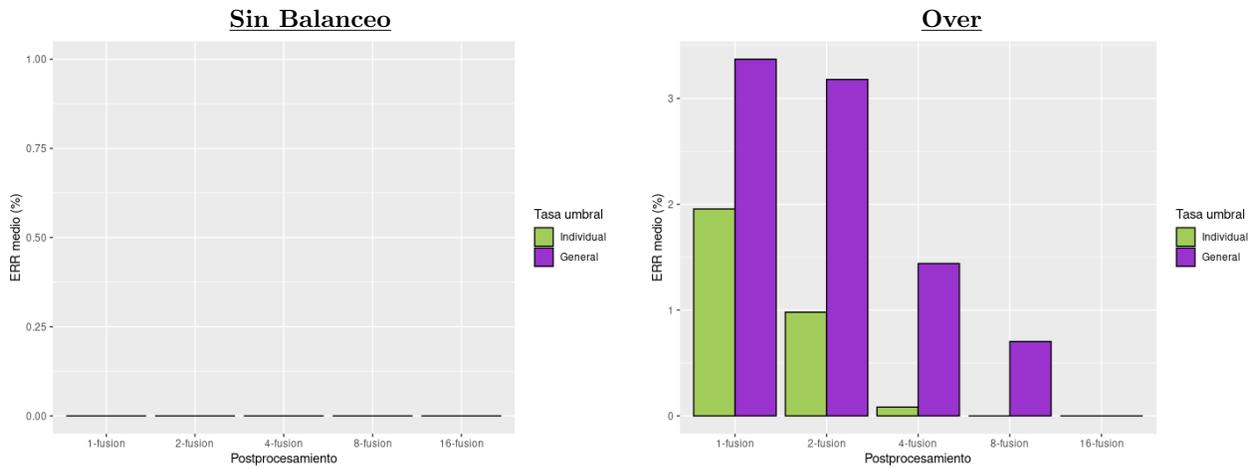


Figura 5.18: Mejores algoritmos para DF-Monosesión-En reposo-Izquierda

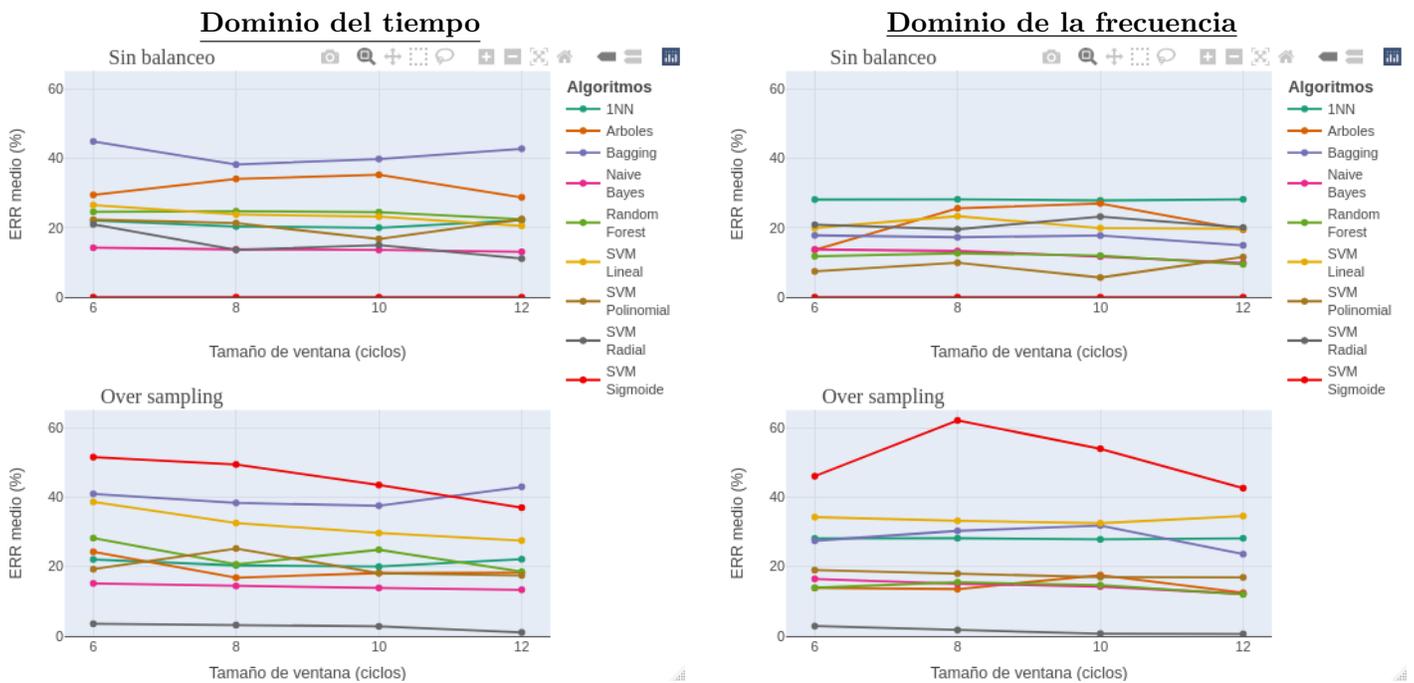


Figura 5.19: Experimento Monosesión-En reposo-Derecha

No existen grandes diferencias con respecto al escenario anterior, donde se colocaba el reloj en la muñeca izquierda. Estas diferencias son prácticamente

inexistentes si nos fijamos en los mejores sistemas.

Como en el anterior escenario, en las figuras 5.20 y 5.21, vemos los mejores sistemas para cada dominio en función de si hemos realizado balanceo o no. Además, podemos ver claramente el efecto del incremento en la fusión de *scores* en la disminución del error, sendas figuras, ambas en el caso con balanceo Over. En el caso del dominio de la frecuencia se alcanza el 0 antes, con 4 *scores* fusionados.

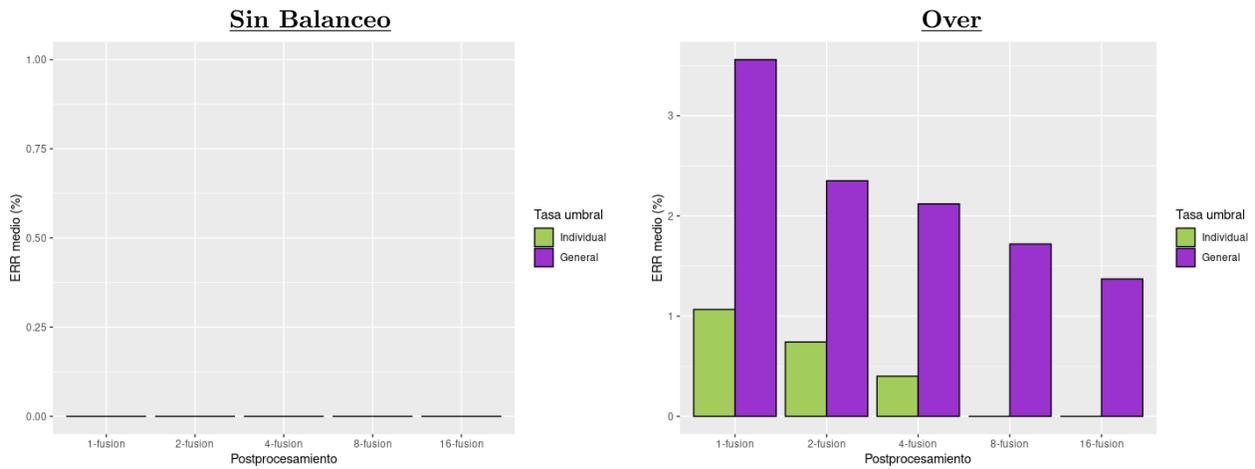


Figura 5.20: Mejores algoritmos para DT-Monosesión-En reposo-Derecha

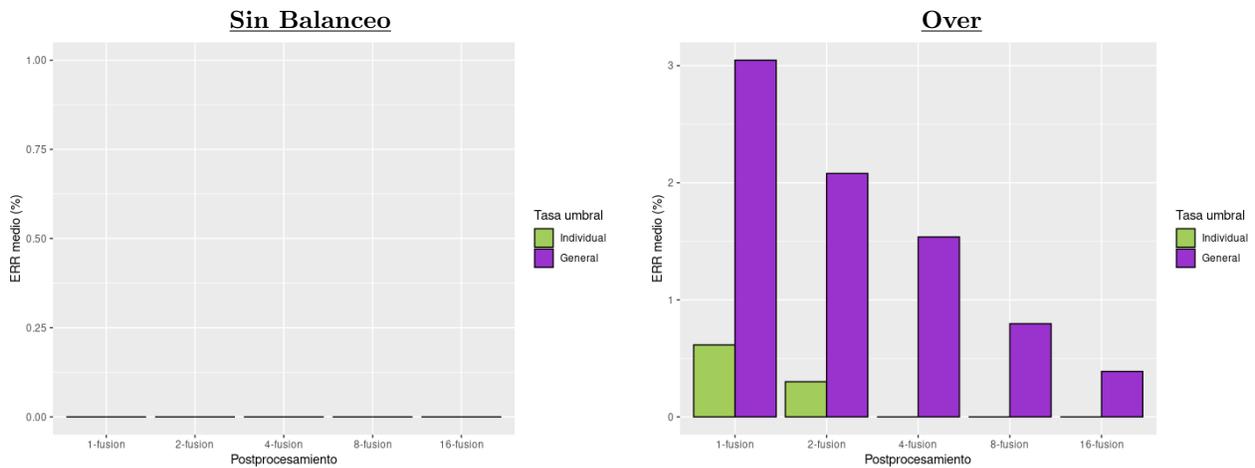


Figura 5.21: Mejores algoritmos para DF-Monosesión-En reposo-Derecha

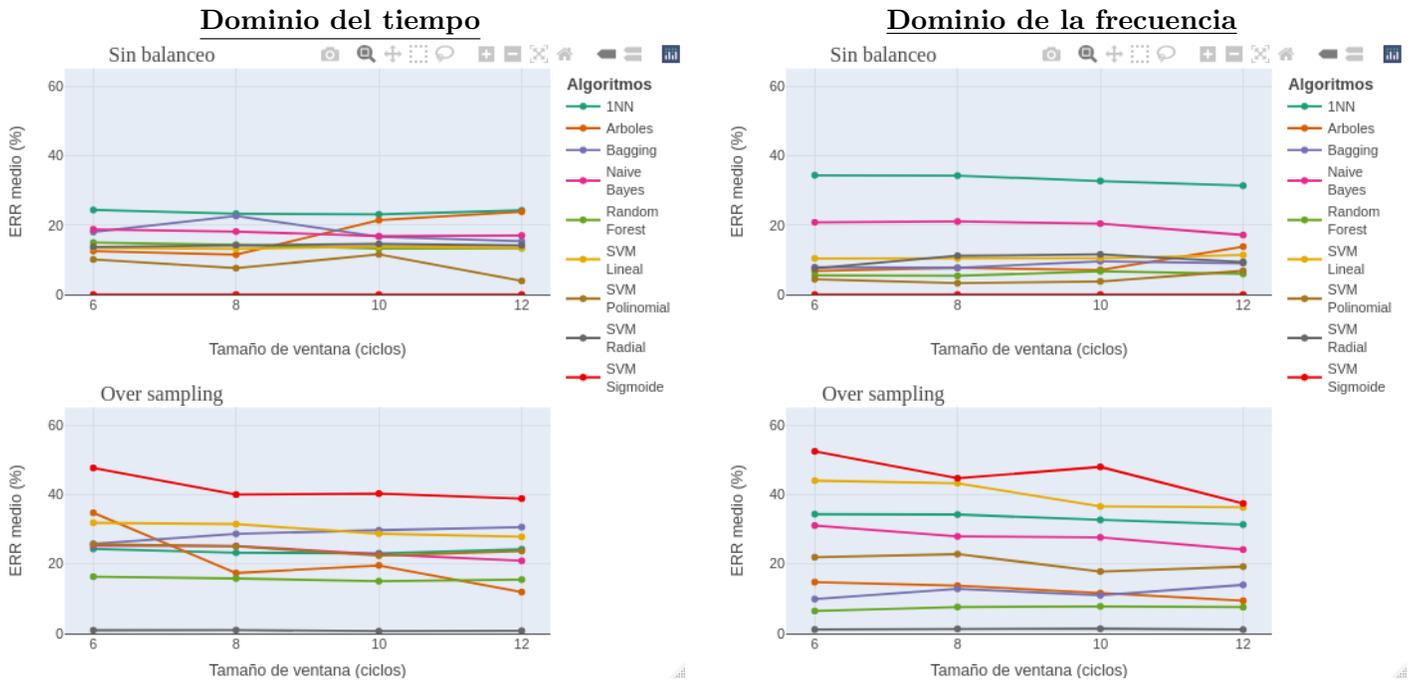


Figura 5.22: Experimento Monosesión-Andando-Izquierda

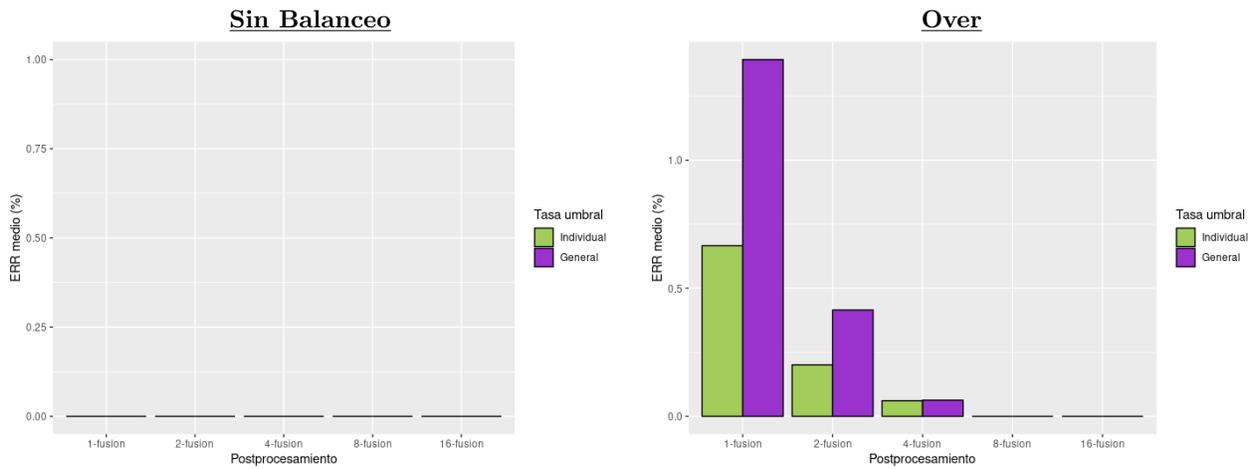


Figura 5.23: Mejores algoritmos para DT-Monosesión-Andando-Izquierda

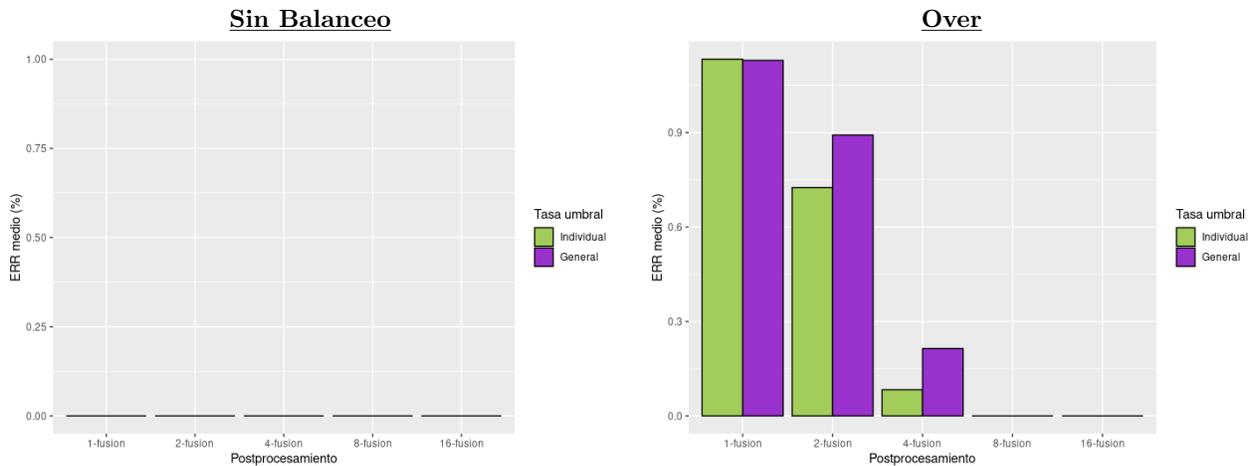


Figura 5.24: Mejores algoritmos para DF-Monosesión-Andando-Izquierda

Para los casos en los que los individuos de nuestro conjunto andaron mientras tenían el reloj en la mano izquierda, se presentan tasas algo menores para los 2 mismos algoritmos que venimos destacando. De nuevo en los casos sin balanceo en las figuras 5.23 y 5.24 no podemos apreciar cambios entre la forma de medir el EER. Pero sí con el balanceo *over sampling*, que, aunque son valores bastante parecidos, son algo menores en el dominio de la frecuencia, y por tanto, se llega antes a tener una tasa de error de 0.

Experimentos Multisesión

En el siguiente apartado se mostrarán los resultados para los mismos tres escenarios. Estos rendimientos se ordenarán de la misma forma anterior: primero los resultados para el sistema base, es decir, sin pasar por la etapa de postprocesamiento, para todos los algoritmos elegidos en función del dominio, para después ver el efecto de la etapa de procesamiento en los dos modelos más destacados, uno para cada dominio.

Después, dado que el objetivo primordial es la comparativa global del experimento con el anterior, se presentarán primero todos los gráficos para posteriormente concluir con el análisis de estos.

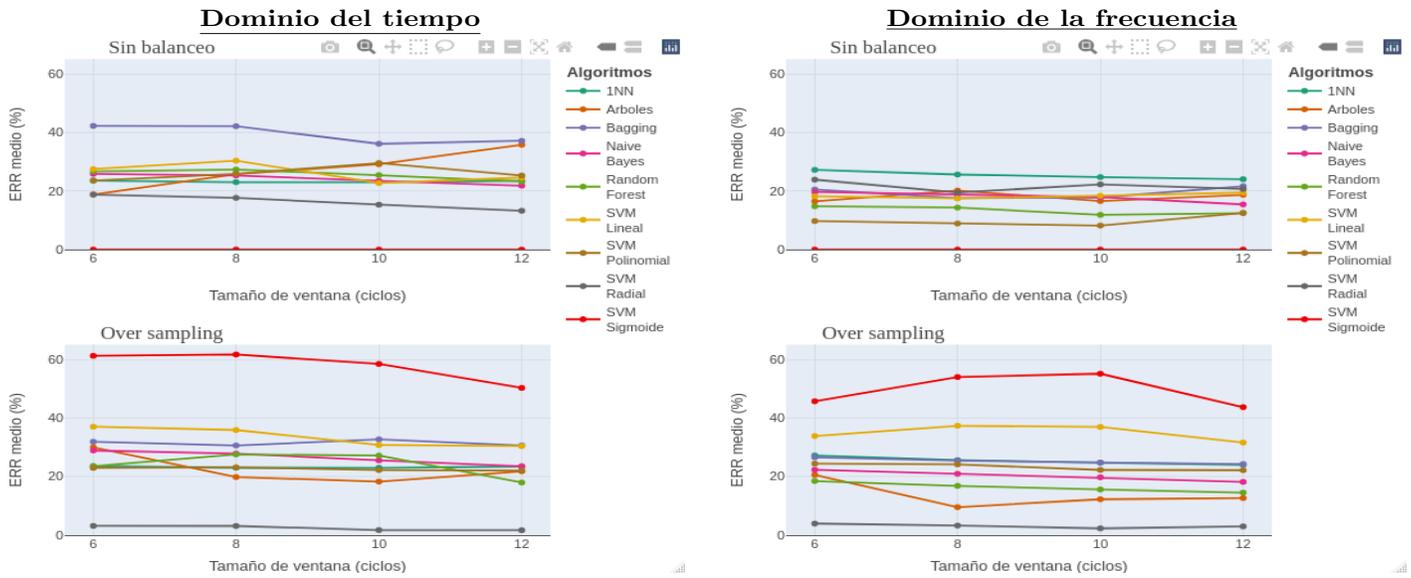


Figura 5.25: Experimento Multisesión-En reposo-Izquierda

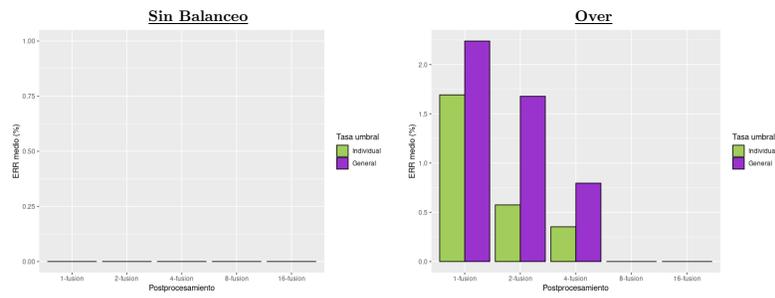


Figura 5.26: Mejores algoritmos para DT-Multisesión-En reposo-Izquierda

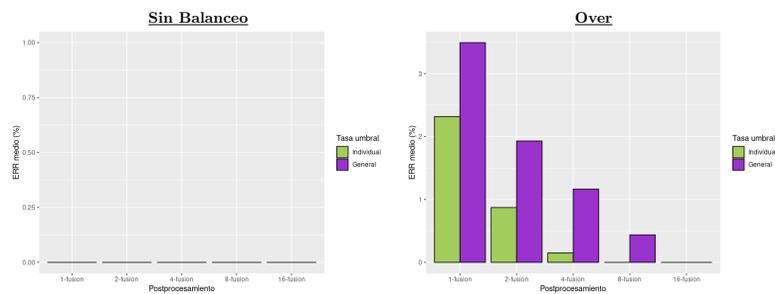


Figura 5.27: Mejores algoritmos para DF-Multisesión-En reposo-Izquierda

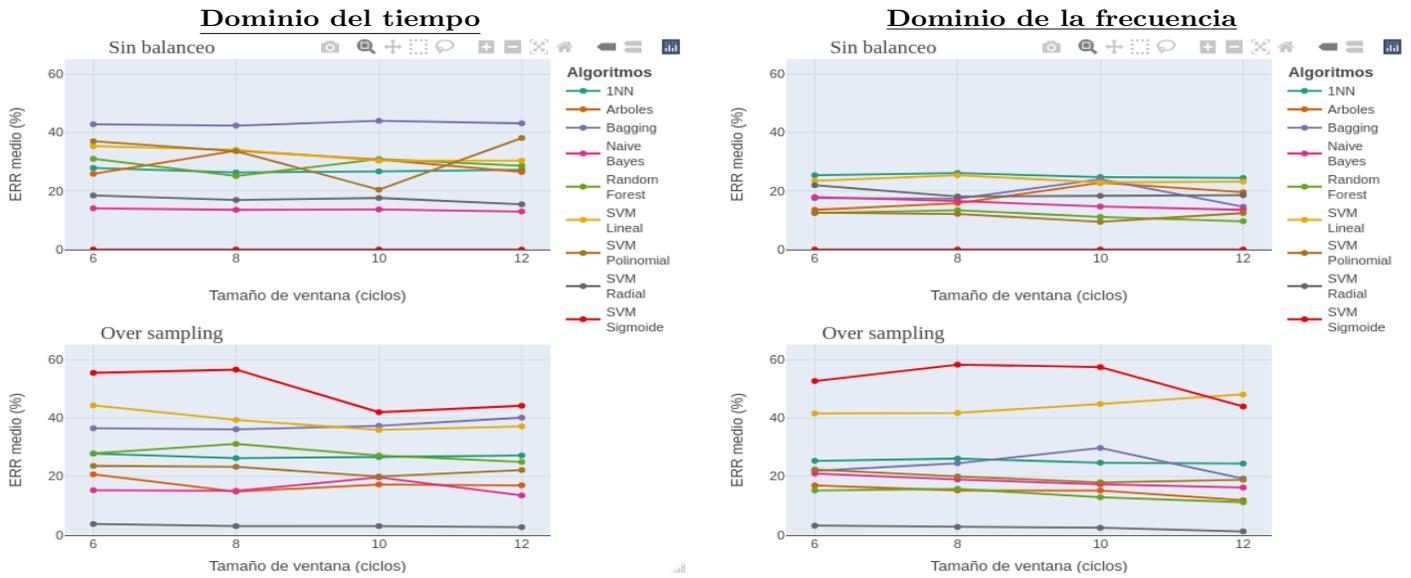


Figura 5.28: Experimento Multisección-En reposo-Derecha

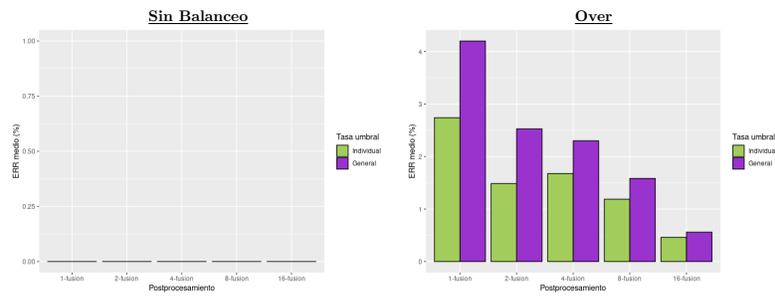


Figura 5.29: Mejores algoritmos para DT-Multisección-En reposo-Derecha

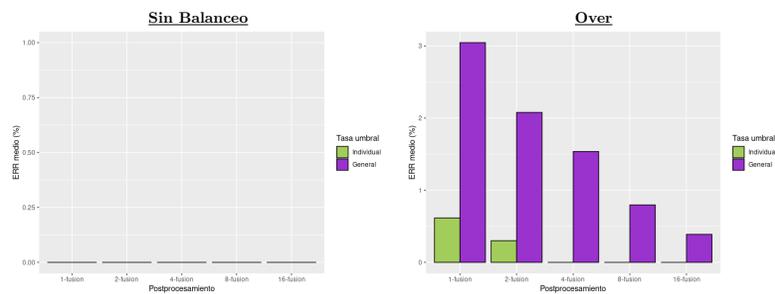


Figura 5.30: Mejores algoritmos para DF-Multisección-En reposo-Derecha

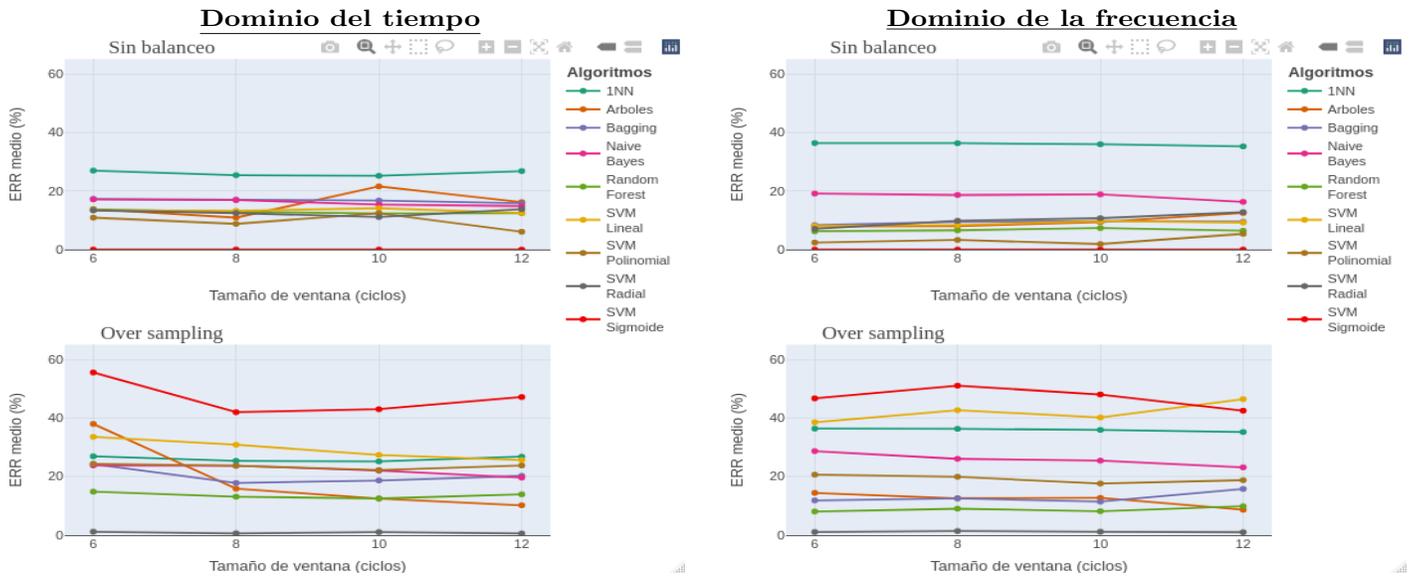


Figura 5.31: Experimento Multisesión-Andando-Izquierda

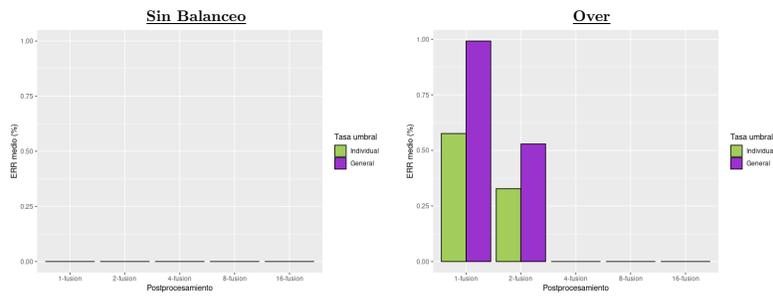


Figura 5.32: Mejores algoritmos para DT-Multisesión-Andando-Izquierda

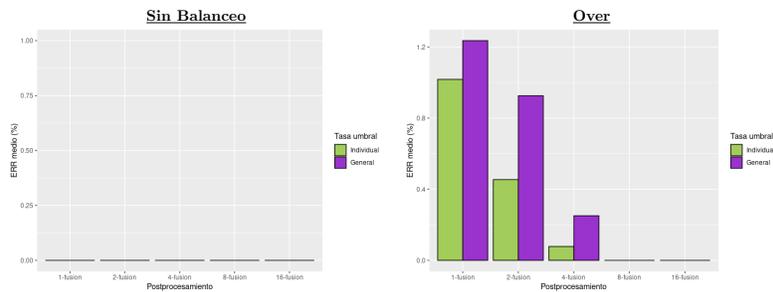


Figura 5.33: Mejores algoritmos para DF-Multisesión-Andando-Izquierda

Las diferencias respecto a los casos del experimento Monosesión son pequeñas. De hecho, es más fácil mencionar las similitudes que encontramos entre los 2 experimentos. Comenzando por que los algoritmos de referencia vuelven a ser SVM con kernel sigmoide para el dominio del tiempo y con kernel radial para el dominio de la frecuencia. Después, vemos que para el escenario en el que el usuario se encuentra en reposo, en el dominio de la frecuencia, todos los algoritmos (menos SVM radial), presentan un rendimiento parecido o al menos existen menos diferencias entre ellos. Pasa un fenómeno parecido cuando el usuario se encuentra en movimiento y no se ha realizado balanceo, pero esta vez más marcado en el dominio del tiempo.

5.6.3. Análisis de resultados

Lo primero a nivel global es destacar el algoritmo *Support Vector Machines* (SVM), como el algoritmo por excelencia, siendo el mejor para ambos dominios, aunque con distintos kernels (sigmoide en el dominio del tiempo y radial en el dominio de la frecuencia).

Respecto a los diferentes escenarios planteados no se han encontrado grandes diferencias, ya que si nos fijamos únicamente en los algoritmos de referencia, indicados en el párrafo anterior, los rendimientos para los casos en los que el usuario tenía el reloj ubicado en la muñeca izquierda tiene resultados parecidos a cuando lo tenía en la derecha. Lo mismo pasa en las distintas actividades (en reposo/andando). Y como se mencionó al final de la sección anterior, tampoco hay diferencias significativas respecto a los dos experimentos planteados, es decir, en cuanto al cambio en las muestras de prueba. En relación con el tamaño de ventana las diferencias no son significativas en la mayoría de algoritmos, aunque sí que existen ciertos casos en los que se reduce o aumenta el rendimiento a la vez que aumentamos el tamaño de ventana.

El tratamiento realizado en la etapa de postprocesamiento se puede concluir que ha sido muy beneficioso ya que mejora los resultados en todos los casos planteados, en algunos llegando incluso a 0.

Como algoritmos a observar en trabajos futuros por su cercano rendimiento en muchos de los experimentos presentados, destacamos de nuevo a SVM pero esta vez con kernel polinomial de grado 4, en los casos en los que las muestras no se balancean, y *Random Forest* en experimentos que sí se realizó el único tipo de balanceo planteado.

Capítulo 6

Conclusiones

6.1. Conclusiones

Con la finalización de este trabajo se confirma la consecución de los objetivos definidos al comienzo de este.

De entrada, se confirma la posibilidad del reconocimiento biométrico mediante *wearables* de tipo comercial. Después de un intento fallido con un dispositivo Samsung, utilizado en anteriores trabajos de esta índole, el estudio del mercado nos dio una opción viable en cuanto a precio, de muy bajo coste en comparación con la mayoría de sus competidores, cuyos importes eran superiores al doble de nuestro dispositivo, y posibilidad de acceso a los datos crudos.

Después de adquirir el reloj de la empresa Withings, se adquirieron señales para 10 usuarios siguiendo un protocolo estricto y almacenando las muestras en una base de datos propia.

Tras las 300 horas de duración del proyecto, el alumno ha realizado una introducción a la comprensión de las distintas etapas de un sistema biométrico junto con algunas de las operaciones más comunes en cada una de ellas. Certificación de esto es, la adecuación correcta del sistema de reconocimiento mediante la forma de andar del que se partía.

Respecto a la discusión de los resultados, tras la aplicación en la etapa de clasificación de hasta 9 algoritmos, contando las máquinas de vectores soporte con distintas funciones núcleo como diferentes, se han generado resultados prometedores. En concreto, han sido 2 SVM los más destacados, con kernel sigmoide por su espectacular rendimiento en las ocasiones que no se realizaba balanceo, pero su inconsistencia en el caso del *over sampling*, siendo el peor

con diferencia llegando a tasas de equierror de hasta casi un 65 %, y con kernel radial destacado en los casos que se equiparaban las muestras auténticas a las impostoras mediante repetición (*over sampling*) y estando por debajo del 20 % en el otro caso.

Comparando los rendimientos obtenidos con los mostrados en otros trabajos, podemos decir que estos son competitivos, teniendo en cuenta que nos encontramos ante una primera aproximación al tema, y usando un dispositivo comercial, el cual no es común en la literatura, donde predomina el uso de instrumentos médicos.

Por último, se han cumplido las metas personales de manera satisfactoria, sacando adelante un proyecto en solitario, adquiriendo conocimientos sobre el mundo de la biometría, automatizando con *PHP* y *SQL* la recuperación de las adquisiciones del reloj.

6.2. Trabajos futuros

A través de este trabajo se ha visto la posibilidad de usar dispositivos comerciales para el reconocimiento biométrico con la ayuda del electrocardiograma, partiendo de un sistema preparado para la identificación mediante la forma de andar. Los resultados obtenidos son buenos y optimistas, pero la base de datos obtenida es bastante reducida, 10 usuarios. Es por esto que el siguiente paso para continuar este trabajo es la obtención de un corpus de mayor tamaño con mayor variabilidad de perfiles. Este proceso se podría describir de la siguiente forma:

- Fijar un tamaño de usuarios mayor a 40 individuos, a los cuales realizar el protocolo, descrito en la sección 7.3.1, de manera estricta.
- Incluir en los requisitos del corpus, además del equilibrio en la distribución por sexos, la existencia de varios individuos para cada uno de los perfiles de estilo de vida que puedan afectar a la periodicidad de la señal, como pueden ser personas sedentarias o deportistas. Otro factor que puede afectar es la edad, por lo que otro requerimiento podría ser concretar un mínimo de personas en rangos de edad de 10 en 10 años.
- En nuestro caso, medimos el efecto de la variable tiempo sobre el rasgo biométrico utilizado, el ECG, únicamente en una segunda sesión espaciada mínimo de 2 a 3 semanas. En un estudio sin fecha límite de entrega o, con mayor cantidad de tiempo a dedicar a las adquisiciones, podría aumentarse dicho examen a una nueva sesión dilatada de 2 a 3 meses con la primera o incluso la segunda sesión.

Como se pudo ver en el análisis del mercado de *wearables*, la empresa estadounidense *Apple*, es una alternativa potente que podría probarse y, si los resultados son buenos, ser un próximo trabajo, abarcando así un nuevo sistema operativo como es *watchOS* y promoviendo el aprendizaje de un alumno en un entorno cerrado como es el de *Apple*, con su lenguaje *Swift*.

Indudablemente, este trabajo prospectivo cumple con su cometido a la vista del rendimiento obtenido, siendo un buen punto de partida para un estudio más exhaustivo teniendo ya las herramientas preparadas para ampliar la cantidad de individuos.

Bibliografía

- [1] *Payment services (PSD 2)*. URL: https://ec.europa.eu/info/law/payment-services-psd-2-directive-eu-2015-2366_en. Accedido el 07/07/2022.
- [2] *Nymi band*. URL: <https://www.nymi.com/>. Accedido el 10/07/2022.
- [3] Irene Salvador Ortega. *Investigación y desarrollo de un sistema de reconocimiento biométrico mediante dispositivos ponibles (Wearables)*. 2020. URL: <https://uvadoc.uva.es/handle/10324/44947?locale-attribute=fr>. Accedido el 13/07/2022.
- [4] Biometrics Institute. *What is biometrics?* URL: <https://www.biometricinstitute.org/what-is-biometrics/>. Accedido el 19/06/2022.
- [5] *Reconocimiento de venas del dedo - Hitachi*. URL: <https://www.hitachi-helc.com/en/solution/shouzhijingmai.html>. Accedido el 05/07/2022.
- [6] *Reconocimiento de venas de la palma - Korean Air*. URL: <https://findbiometrics.com/korean-air-introduces-palm-boarding-domestic-flights-020905/>. Accedido el 05/07/2022.
- [7] *Directorio de usos de biometría para seguridad*. URL: <https://findbiometrics.com/applications/biometrics-security/>. Accedido el 05/07/2022.
- [8] Mario Adolfo Suarez. «Fundamentos de Seguridad Física». En: (). URL: <https://pt.slideshare.net/adolfinho1988/seguridad-fsica-13068797>. Accedido el 28/02/2022.
- [9] Javier. González Barajas. *Electrocardiograma desde una visión digital / Javier González Barajas*. spa. Bogotá, 2016.
- [10] Francisco Jaffet Ramírez-Ramírez. *Fisiología Cardíaca*. URL: <https://www.medigraphic.com/pdfs/revmed/md-2009/md093d.pdf>. Accedido el 19/06/2022.
- [11] Ian H. Witten, Eibe Frank y Mark A. Hall. *Data Mining: practical machine learning tools and techniques (third Edition)*. Ed. Morgan Kaufmann, 2011.

- [12] Wei Cui, Zihan Wang, Yaoguang Li. *ECG-based biometric recognition under exercise and rest situations*. Accedido el 06/07/2022.
- [13] South China University of Technology. *Exercise ECGID database*. URL: <https://github.com/feimadada/exercise-ECGID-database->. Accedido el 06/07/2022.
- [14] Mohit Ingale, Renato Cordeiro y 3 autores más. *ECG Biometric Authentication: A Comparative Analysis*. Accedido el 07/07/2022.
- [15] *Salario medio de un ingeniero informático en España - talent.com*. URL: <https://es.talent.com/convert?salary=2208.33&start=month&end=hour>. Accedido el 02/03/2022.
- [16] Selectra. *Precio de la luz con discriminación horaria*. URL: <https://tarifasgasluz.com/comparador/precio-kwh#kwh-precios-fijos>. Accedido el 01/03/2022.
- [17] Finetwork. *Fibra 300Mb*. URL: <https://www.finetwork.com/tarifas-fibra/fibra-300mb>. Accedido el 02/03/2022.
- [18] Support HP. *HP notebook 15-ay154ns*. URL: <https://support.hp.com/sg-en/document/c05348793>. Accedido el 02/03/2022.
- [19] OPPO Store. *OPPO A53S*. URL: https://store.oppomobile.es/smartphones/72-207-oppo-a53s.html#/52-color-electric_black. Accedido el 01/03/2022.
- [20] Samsung. *Samsung Galaxy Watch Active2*. URL: <https://www.samsung.com/es/wearables/galaxy-watch-active2/buy/>. Accedido el 01/03/2022.
- [21] *Samsung Health SDK*. URL: <https://developer.samsung.com/health/android/overview.html>. Accedido el 20/02/2022.
- [22] *Portal de Desarrolladores Withings*. URL: <https://developer.withings.com/#operation/signaturev2-getnonce>. Accedido el 05/07/2022.
- [23] *Toda la información de salud disponible con Withings*. URL: <https://developer.withings.com/developer-guide/v3/data-api/all-available-health-data/>. Accedido el 05/07/2022.
- [24] Guannan Wu. *A continuous identity authentication scheme based on physiological and behavioral characteristics*. 2018. Accedido el 14/07/2022.
- [25] *Can cyber criminals “compromise speech recognition systems with ease?”* URL: <https://whatsnext.nuance.com/enterprise/can-cyber-criminals-compromise-speech-recognition-systems/>. Accedido el 11/07/2022.

- [26] Carlos Vivaracho-Pascual, Arancha Simon-Hurtado y Esperanza Manso-Martinez. «Improving biometric recognition by means of score ratio, the likelihood ratio for non-probabilistic classifiers. A benchmarking study». En: *IET Biometrics* 10.2 (2021), págs. 127-141. DOI: <https://doi.org/10.1049/bme2.12011>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/bme2.12011>. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/bme2.12011>.
- [27] Mario Garrido Tapias. *Código creado para el trabajo*. Accedido el 12/07/2022. URL: https://gitlab.inf.uva.es/ponibles/mariogarridotapias_ecg.

Apéndice A

Código utilizado

A.1. Estructura

El conjunto de archivos creados para el desarrollo del trabajo se reparte en dos carpetas: *Forms*, en la que encontraremos todos los códigos relacionados con el *software* de adquisición, y *SistBio*, donde se encuentran los programas que conforman el sistema biométrico formado. A continuación, se citan estos archivos creados junto con una breve explicación de su cometido dentro del trabajo:

- **users.css**
Hoja de estilo en cascada para los 3 formularios distintos que conforman el *software*.
- **UserForm.html**
Esqueleto del formulario destinado a la creación de usuarios en la base de datos.
- **Uformpost.php**
Backend para la creación de instancias de usuarios en la base de datos.
- **DataForm.php**
Esqueleto del formulario destinado a la recuperación de una toma de un usuario en concreto y en un intervalo de tiempo concreto.
- **Dformpost.php**
Backend para el guardado de las distintas señales de una toma especificada.

- **FileForm.html**

Esqueleto del formulario destinado a la creación de los 3 archivos, para cada uno de los escenarios planteados (reposo en muñeca izquierda, reposo en muñeca derecha y andando en muñeca izquierda).

- **Fformpost.php**

Backend para la recuperación de la toma de un usuario y sesión indicadas.

- **connect.php**

Backend para conexión con la base de datos.

- **refreshToken.php**

Backend para el refresco del token en caso de que el último token de acceso haya caducado.

- **select.php**

Backend destinado al filtrado necesario de los datos según distintos parámetros indicados.

- **AnalisisVisual.R**

Programa que de manera totalmente aleatoria enseña las muestras solicitadas para un número de usuarios indicado. Su función principal tiene 3 argumentos: un vector con los ids de los usuarios, la cantidad de muestras a pintar por usuario y la cantidad de usuarios. El cambio entre las visualizaciones de los distintos usuarios se hace pulsando la tecla *ENTER*.

```
analisisVisual <- function(usuarios, nUsuariosAPintar, nECGAPintar)
```

- **9_3_TablasCaract_modulo_V2_CicloPorUsuario_Muestra.R**

Programa destinado a la etapa de extracción de características, con el objetivo de crear los archivos de entrada para el siguiente programa.

- **9_5_NF_RendimientoFinal_Clasificadores_VariasVentanasScores.R**

Programa enfocado en la aplicación de la etapa de clasificación para los distintos algoritmos seleccionados, los balanceos elegidos y el experimento indicado.

- **AnalisisResultados.R**

Programa para la creación de los distintos gráficos útiles para la discusión de resultados. Se usan las bibliotecas *ggplot2* y *plotly*, para darles interacción.

Cuenta con 2 funciones principales una para la representación interactiva de los sistemas base y otra para la comparación del procesamiento mediante gráficos de barras de los mejores algoritmos.

- **Salidas**

Carpeta con las salidas de los distintos escenarios planteados. Sigue una jerarquía de carpetas en función de los parámetros de la experimentación.

- **tablas_caract**

Carpeta con las salidas para el archivo 9_3_TablasCaract_modulo_V2_CicloPorUsuario_Muestra.R.

- **usuario**

Una carpeta con los datos crudos para cada uno de los usuarios. Existe un fichero por cada sesión, cada toma y cada uno de los 3 escenarios.

Estos programas son accesibles en el siguiente repositorio público del GitLab de la Escuela. [27].

El esquema del repositorio se encuentra en la siguiente página.

