



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS ESPECÍFICAS DE  
TELECOMUNICACIÓN

MENCIÓN EN SISTEMAS ELECTRÓNICOS

**Simulación de conducción con sensores  
fisiológicos para monitorización del estado  
del conductor**

Autor:

**D. Rubén García Casado**

Tutor:

**D. David González Ortega**

Valladolid, 6 de septiembre de 2017



---

**TÍTULO:** Simulación de conducción con sensores fisiológicos para la monitorización del estado del conductor

**AUTOR:** Rubén García Casado.

**TUTOR:** David González Ortega

**DEPARTAMENTO:** Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

---

**TRIBUNAL**

---

**PRESIDENTE:** Miriam Antón Rodríguez

**VOCAL:** Mario Martínez Zarzuela

**SECRETARIO:** David González Ortega

**SUPLENTE:** Francisco Javier Díaz Pernas

**SUPLENTE:** José Fernando Díez Higuera

---

---

**FECHA:** 6 de septiembre de 2017

**CALIFICACIÓN:**

---

---



# RESUMEN

---

En los últimos años, se han usado simuladores de conducción en muchas investigaciones. Por ejemplo, se ha probado su idoneidad para evaluar el comportamiento de los conductores y como medio de aprendizaje de hábitos de conducción segura y eficiente. La inclusión en los simuladores de sensores fisiológicos que monitorizan al conductor les confiere de un potencial añadido para un estudio multifactorial de pruebas de conducción.

En este Trabajo Fin de Grado, se ha desarrollado y adaptado una aplicación Android que monitorizará el estado del conductor con los sensores fisiológicos de la plataforma Shimmer, que incluye un sensor ECG (Electrocardiograma), EMG (Electromiograma) y GSR (Respuesta Galvánica de la piel), junto a un acelerómetro y giróscopo. La aplicación realiza procesados en tiempo real para determinar cuándo el estado del conductor no es propio de una conducción segura. Los datos almacenados durante las simulaciones también permiten realizar un análisis *offline* de la conducción.



# ABSTRACT

---

In the last years, driving simulators have been used in many research studies. For example, its functionality has been tested to evaluate driver behavior and as a resource of learning safe and efficient driving habits. The inclusion in the simulators of physiological sensors that monitor the driver give them the added potential to make a multifactorial study of driving test.

In this Final Degree Project, an Android application have been developed and adapted to monitor the driver's state with the physiological sensors of the Shimmer platform, which includes an ECG (Electrocardiogram), EMG (Electromyogram) and GSR (Galvanic Skin Response) sensors, together with an accelerometer and gyroscope. The application makes real-time processing to determine when the driver's condition is not proper for a safe driving. The data stored during the simulations also allow an offline analysis of the driving.





## **Agradecimientos**

A mis padres y mi hermana por todo su apoyo y consideración a lo largo de toda mi formación académica, ya que sin ellos sería imposible haber llegado hasta aquí. La educación y valores que me han inculcado han tenido como resultado que logre metas como esta.

A todas aquellas personas que tan importantes considero que son en mi vida y de las cuales desde que comencé estos estudios hasta el final no han dejado de mostrarme su apoyo incondicional, en especial mencionar a los que han aguantado y sufrido conmigo la realización de este proyecto.



# ÍNDICE GENERAL

---

|        |   |    |
|--------|---|----|
| 1.     | INTRODUCCIÓN.....   | 15 |
| 1.1.   | Motivación y objetivos planteados .....   | 16 |
| 1.2.   | Fases del proyecto.....   | 18 |
| 1.3.   | Medios y herramientas empleadas.....  | 19 |
| 1.4.   | Organización de la memoria .....  | 20 |
| 2.     | SEGURIDAD EN LA CONDUCCIÓN, RIESGOS Y CAUSAS.....   | 22 |
| 2.1.   | Seguridad vial .....  | 22 |
| 2.2.   | Riesgos por fatiga mental y nivel de estrés.....  | 25 |
| 2.2.1. | Fatiga mental .....   | 25 |
| 2.2.2. | Estrés.....   | 28 |
| 3.     | SENSORES FISIOLÓGICOS SHIMMER .....   | 32 |
| 3.1.   | Sensores de la gama shimmer.....  | 33 |
| 3.1.1. | Equipamiento Hardware.....  | 33 |
| 3.1.2. | Emparejamiento de los sensores.....   | 41 |
| 3.1.3. | Calibración .....   | 42 |
| 4.     | SISTEMA OPERATIVO: ANDROID.....   | 43 |
| 4.1.   | Introducción y Arquitectura .....   | 43 |
| 4.2.   | Versiones de Android .....  | 45 |
| 4.3.   | Herramientas de desarrollo .....  | 46 |
| 5.     | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS.....   | 48 |
| 5.1.   | PROCESADO ONLINE .....  | 48 |
| 5.1.1. | Aplicación de Android base .....  | 48 |
| 5.1.2. | Introducción sobre las nuevas implementaciones .....  | 50 |
| 5.1.3. | Procesado online de la señal ECG .....  | 53 |
| 5.1.4. | Procesado online de la señal EMG.....   | 69 |
| 5.1.5. | Procesado online de la señal GSR .....  | 73 |
| 5.1.6. | Interfaz gráfica y generación de ficheros del procesado en tiempo real de las señales ECG, EMG y GSR..... | 76 |
| 5.2.   | PROCESADO OFFLINE .....   | 81 |
| 5.2.1. | Introducción y herramienta de desarrollo.....   | 81 |
| 5.2.2. | Procesado offline de la señal ECG.....  | 82 |
| 5.2.3. | Procesado offline de las señales EMG y GSR .....  | 89 |
| 6.     | BALANCE ECONÓMICO DEL PROYECTO .....  | 91 |

|                                       |    |
|---------------------------------------|----|
| 7. CONCLUSIONES Y LÍNEAS FUTURAS..... | 92 |
| 7.1. Conclusiones .....               | 92 |
| 7.2. Líneas futuras .....             | 95 |
| BIBLIOGRAFÍA.....                     | 96 |

# ÍNDICE DE FIGURAS

---

|  |    |
|--|----|
| Figura 1. Evolución del número de fallecidos (1960-2016).....                                    | 22 |
| Figura 2. Desplazamientos en carretera (2012-2016).....  | 24 |
| Figura 3. Accidentes y víctimas según tipo de accidente.....                                     | 28 |
| Figura 4. Evolución y fases del estrés en el ser humano.....                                     | 29 |
| Figura 5. Unidad básica de Shimmer .....   | 34 |
| Figura 6. Shimmer Dock 2 .....   | 35 |
| Figura 7. Diagrama de bloques simplificado del sensor ECG .....                                  | 36 |
| Figura 8. Ejemplo del posicionamiento de los electrodos para la medida ECG.....                  | 37 |
| Figura 9. Módulo ECG y colocación de electrodos.....   | 38 |
| Figura 10. Módulo EMG y colocación de electrodos .....   | 39 |
| Figura 11. Módulo GSR y colocación de electrodos.....  | 40 |
| Figura 12. Módulo 9DOF.....  | 41 |
| Figura 13. Aplicación Shimmer 9DOF.....  | 42 |
| Figura 14. Estudio de uso de Smartphone y Tablets en la sociedad.....                            | 44 |
| Figura 15. Logotipo Android 8.0.0 .....  | 46 |
| Figura 16. Logotipo Android SDK .....  | 47 |
| Figura 17. Logotipo Eclipse Java Oxygen.....   | 47 |
| Figura 18. Pantalla principal de la aplicación Shimmer App.....                                  | 48 |
| Figura 19. Pantalla de manejo de los dispositivos en la aplicación .....                         | 49 |
| Figura 20. Pantalla de manejo de ficheros en la aplicación.....                                  | 50 |
| Figura 21. Estructura de datos en librería Shimmer .....   | 52 |
| Figura 22. Forma de onda de la señal ECG.....  | 55 |
| Figura 23. Triángulo formado por el complejo QRS.....  | 58 |
| Figura 24. Forma de onda de la señal ECG obtenida del sensor fisiológico Shimmer .....           | 58 |
| Figura 25. Cálculo y representación de la señal HRV .....  | 62 |
| Figura 26. PSD y bandas de frecuencia de la señal HRV .....                                      | 67 |
| Figura 27. Señal EMG adquirida por medio de electrodos de contacto .....                         | 70 |
| Figura 28. Señal EMG adquirida por medio de electrodos de aguja.....                             | 70 |
| Figura 29. Forma de onda de la señal GSR obtenida del sensor Shimmer .....                       | 74 |
| Figura 30. Layout de la pestaña correspondiente al sensor ECG en la aplicación.....              | 78 |
| Figura 31. Layout de la pestaña correspondiente al sensor ECM en la aplicación .....             | 79 |
| Figura 32. Layout de la pestaña correspondiente al sensor GSR en la aplicación .....             | 80 |
| Figura 33. Comparativa de la señal ECG original y tras pasar por los filtros FIR en Matlab ..... | 83 |
| Figura 34. Detección de los picos R usando la función findpeaks () .....                         | 84 |
| Figura 35. Obtención y normalización de los picos R de forma definitiva.....                     | 85 |
| Figura 36. Representación de la señal HRV .....  | 86 |
| Figura 37. Representación de la densidad espectral de potencia (PSD).....                        | 88 |
| Figura 38. Representación de la señal EMG original .....   | 89 |



# 1.INTRODUCCIÓN

---

El desarrollo de las nuevas tecnologías y su evolución cada vez más rápida son un aspecto muy importante y a tener en cuenta en la sociedad de hoy en día. Cada vez son más los sistemas de comunicación y sistemas electrónicos con los que convivimos y que nos brindan mayores facilidades a la hora de resolver nuestras tareas cotidianas.

Pero no solamente estos sistemas están destinados a funciones básicas si no que nos ofrecen la posibilidad de desarrollar importantes y complejas aplicaciones que, cada vez a un nivel mayor, mejoran campos muy importantes dentro de la sociedad como por ejemplo pueden ser la seguridad o la medicina. Ahora más que nunca, un individuo puede estar en constante comunicación y monitorización a través de estos sistemas y esto permite reducir riesgos que pongan en peligro la vida del propio sujeto y de quienes lo rodeen.

Esto nos acerca al campo de investigación y aplicación que se estudiará en este proyecto. Uno de los más importantes avances es la interacción entre los sistemas electrónicos y los usuarios, ya que es posible capturar y analizar datos que nos ofrezcan estos equipos y utilizarlos para el desarrollo de aplicaciones que ayuden a mejorar la eficiencia en el ámbito en que se estén dando.

Por ejemplo, la detección y la posterior interpretación de parámetros relacionados con el estado físico o el cuerpo humano en general permiten conocer cómo funciona la mente y el cuerpo de una persona ante una situación determinada y así ayudar al desarrollo de herramientas que mejoren las condiciones en dicha situación.

Esta será la premisa por la cual en este presente proyecto se dan las motivaciones que han llevado a realizarlo, así como los objetivos planteados y dispuestos a ser resueltos.

Para alcanzar dichos objetivos ha sido necesario seguir una serie de fases y aplicar unos métodos a través de los medios adecuados que se detallarán a lo largo de esta memoria.

### **1.1. Motivación y objetivos planteados**

Tras explicar cómo influyen las nuevas tecnologías en los diferentes ámbitos que nos rodean hoy en día vamos a enmarcar y a poner en contexto las motivaciones que nos llevan a realizar este proyecto.

La seguridad es uno de los principales ámbitos en el que las tecnologías con las que trabajamos actúan con una mayor profundidad. En concreto este proyecto se centra en la seguridad durante la conducción, que más adelante desarrollaremos en uno de los apartados.

Muchas son las campañas, medios y herramientas creadas para mejorar y concienciar a las personas sobre la seguridad y las responsabilidades que se deben cumplir durante la conducción. Esto es así ya que se demuestra que el factor humano es el que más influencia tiene cuando se producen siniestros o accidentes en las carreteras. Por lo tanto, conocer el estado físico y mental del conductor y cómo evoluciona dependiendo de en qué condiciones se esté conduciendo es de vital importancia para reducir riesgos que pongan en peligro al propio conductor y los que lo rodean.

Desde hace unos años hacia aquí, han mejorado de forma notable las cifras y la tasa de accidentes y siniestralidad en las carreteras. Pero es necesario seguir ayudándonos de los nuevos avances y herramientas que van surgiendo para continuar con esta mejora y reducir la tasa de accidentes y siniestralidad en carretera. Este es uno de los objetivos más prioritarios y al que se destinan bastantes recursos.

Como ya se ha mencionado anteriormente, con la ayuda de sistemas de adquisición de datos y procesamiento de estos, y también de herramientas de simulación, podemos realizar diferentes estudios sobre el comportamiento de un conductor ante diferentes situaciones de conducción y, como resultado de



## 1 | INTRODUCCIÓN

esta investigación, localizar los principales riesgos que se pueden llegar a dar. Uno de los más importantes y el principal objetivo de este proyecto es evaluar y alertar de un posible estado de somnolencia o fatiga mental. Esto se explica a lo largo de la memoria.

Para resumir, la realización de este proyecto se centra en el ámbito de investigación dentro de la seguridad vial, más en concreto en la seguridad durante la conducción a través de la monitorización del conductor.

Los objetivos que se han planteado para desarrollar en este proyecto están relacionados con esta monitorización del estado del conductor trabajando con los dispositivos electrónicos como son los sensores fisiológicos de la plataforma Shimmer, los cuales se describen en un apartado posterior, que incluyen un sensor ECG (Electrocardiograma), EMG (Electromiograma) y GSR (Respuesta Galvánica de la piel):

- Trabajando desde la aplicación base desarrollada en la plataforma Android por Khadmaoui, Amine (2014) [1] para extraer los datos de los sensores fisiológicos, se implementará el procesado en tiempo real de las señales de ECG, EMG y GSR para almacenar y mostrar a través de la interfaz al usuario los parámetros que se obtienen y que permiten determinar cuándo el estado del conductor es propio de una conducción segura y cuándo existe riesgo.
- Una vez que se implementa el almacenamiento de las señales y los parámetros correspondientes, se desarrollarán, con ayuda de herramientas de procesado de señales en la plataforma *Matlab*, los *scripts* necesarios para el procesamiento y obtención de parámetros de las señales ya mencionadas anteriormente y el posterior análisis de datos, pero esta vez de manera *offline*, una vez adquiridas y almacenadas las señales.

## 1.2. Fases del proyecto

Para cumplir con los objetivos que se buscan es necesario plantear las pautas o fases a seguir para organizar el proyecto:

1. Estudio sobre la seguridad vial y seguridad en la conducción, las principales causas de accidentes mostrando la evolución de tasas y cifras de mortalidad en las carreteras, así como de qué manera el factor humano y su condición física y mental influyen en dichos datos.
2. Análisis de las funcionalidades y el procedimiento de adquisición de las señales a través de los sensores de la plataforma Shimmer siguiendo sus hojas de especificaciones.
3. Análisis y estudio del sistema operativo Android, utilizado en la mayoría de dispositivos móviles actualmente.
4. Estudio de las características de la señal ECG, EMG y GSR. Se estudiarán los distintos procesados de estas señales para obtener los parámetros correspondientes para el análisis que nos hace cumplir los objetivos.
5. Estudio y entendimiento del proyecto base desarrollado en la plataforma Android con el cual adquirimos las señales provenientes de los sensores.
6. Desarrollo del procesado en la aplicación Android para las señales ECG, EMG y GSR con su correspondiente almacenamiento y visualización de parámetros en tiempo real.
7. Desarrollo de *scripts* con la herramienta Matlab para el procesado y la obtención de parámetros de las señales una vez que estas han sido adquiridas y almacenadas.
8. Realización de pruebas de conducción real con diferentes sujetos definiendo distintos escenarios en un entorno real. En estas pruebas se monitoriza al usuario para el posterior análisis de los datos.
9. Conclusiones y líneas futuras.

### 1.3. Medios y herramientas empleadas

Durante la realización del proyecto se han utilizado los siguientes medios y herramientas:

- Ordenador portátil ACER Aspire E15:
  - Procesador Intel Core i7-6500U a 2.5GB
  - Memoria RAM DDR4 de 8GB
  - Sistema operativo Windows 10
  - Memoria interna de 500GB
  - Tarjeta gráfica Intel HD graphics 520
  
- Tablet Android Wolder miTab ONE
  - 10,1 " HD IPS 1024x600
  - CPU QUAD – CORE 1.3 GHz GPU Mali 400 MP
  - 16 GB Flash
  - 1 GB RAM
  - CAMERA 2 Mpx / 5 Mpx
  - Micro USB / mini Jack 3,5 mm / microSD (hasta 64 GB)
  - WIFI 802.11 b/g/n
  - BATERÍA: 5000 nAh (5,5 g video)
  - ANDROID 5.1
  
- Sensores de la plataforma Shimmer V2.
  
- Entornos de desarrollo de software para Android
  - Eclipse Java Oxygen (Junio 2017)
  - Android SDK tools
  
- Entorno de procesamiento digital de señal: Matlab R2015b

- Smartphone BQ X5
  - Procesador Qualcomm Snapdragon 412 QuadCore MSM8916T de hasta 1,4 GHz
  - Sistema operativo Android 6.0.1 Marshmallow
  - Memoria interna de 16 GB y RAM de 2 GB
  
- Distintos medios pertenecientes y aportados por el Grupo de Telemática e Imagen dentro de la ETSI de Telecomunicación de la Universidad de Valladolid:
  - Dispositivo OBD Link LX Bluetooth

### **1.4. Organización de la memoria**

A continuación, se detalla la organización que seguirá la presente memoria, que se compone de distintos bloques o apartados en los que se detallan la información, el desarrollo, la implementación y el análisis respecto a los objetivos que se han propuesto y cómo se han llevado a cabo.

En el capítulo de introducción se pondrán en contexto las motivaciones y los objetivos que llevan a la realización del proyecto.

En el segundo capítulo se detalla información general acerca de la seguridad vial, detallando datos relevantes y que influyen en el ámbito en el que estamos trabajando. Se analiza cómo afecta el factor humano en la conducción y qué riesgos existen.

En el tercer capítulo se redacta la información técnica acerca de los sensores de la plataforma Shimmer con los que se están trabajando.

En el cuarto capítulo se especifica información sobre las tecnologías y entornos de desarrollo utilizados en el proyecto, en concreto el sistema operativo Android.

## 1 | INTRODUCCIÓN

En el quinto capítulo se detalla el análisis de las señales provenientes de los sensores y la implementación del procesado, almacenamiento y visualización de los parámetros extraídos. Se realiza en dos subapartados, en los que se diferencia la implementación en tiempo real y de manera *offline*.

En el sexto capítulo se muestran las diferentes pruebas realizadas, así como el análisis de datos estadísticos.

Por último, se redactan las conclusiones correspondientes y se expondrán las líneas futuras del proyecto.

## 2.SEGURIDAD EN LA CONDUCCIÓN, RIESGOS Y CAUSAS

En este capítulo se estudia de una forma general el tema de la seguridad vial, analizando con más énfasis cómo influye el factor humano en este ámbito y cómo afectan síntomas como la fatiga mental o el estrés durante la conducción.

### 2.1. Seguridad vial

Durante los últimos años, reforzar la seguridad en las carreteras ha sido un objetivo muy importante para la DGT (Dirección General de Tráfico). Varias son las campañas de concienciación y las recomendaciones que en ellas se aconsejan seguir a los conductores para prevenir accidentes. Esto ha mejorado. En el gráfico que se presenta a continuación, extraído del Balance de Seguridad Vial en 2016, se observa la evolución del número de fallecidos en vías interurbanas desde 1960 hasta 2016 [2].

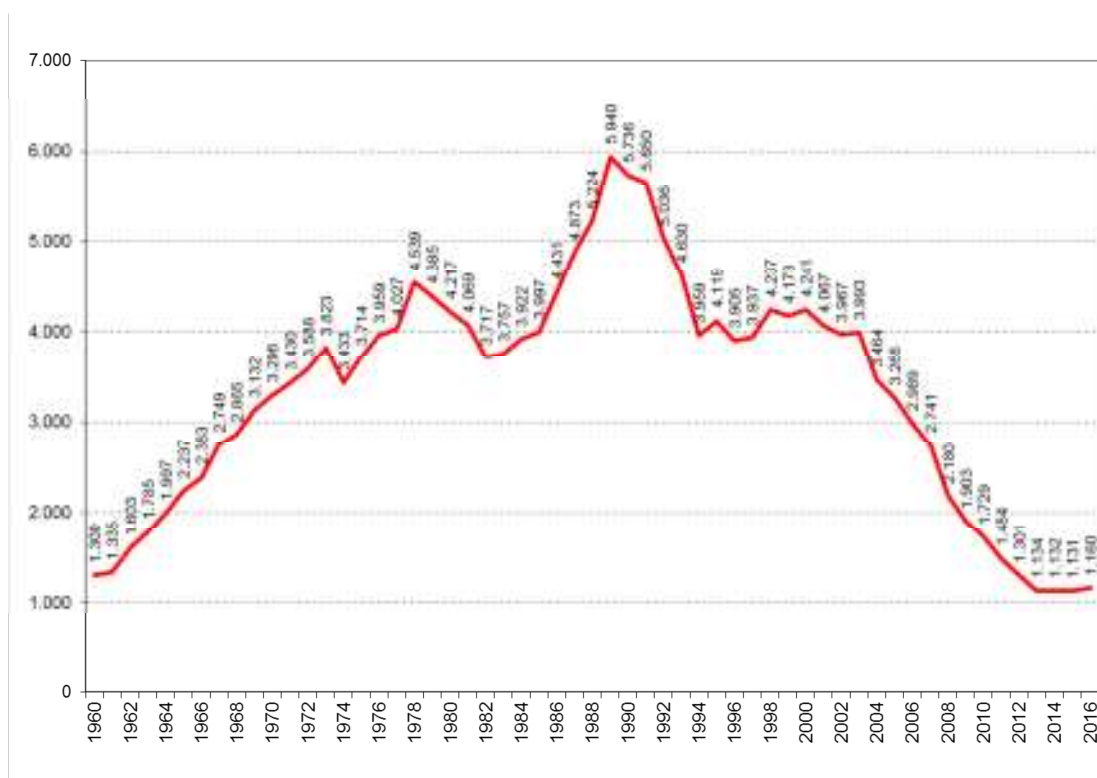


Figura 1. Evolución del número de fallecidos en España en carretera (1960-2016)

Actualmente, la cifra de fallecidos sigue por debajo de la registrada en el primer año en el que se tienen estadísticas. Durante el año 2016 se han registrado 1038 accidentes en vías interurbanas, 15 más que en 2015. Esto va ligado a que se han registrado un total de 392 millones de desplazamientos de largo recorrido, 18,5 millones más, por lo que supone un aumento de la movilidad de un 5% respecto al año anterior.

Las distracciones, la velocidad inadecuada, no respetar las prioridades de paso y el cansancio o sueño (la principal causa en la que se centrará este proyecto), son los principales factores que aparecen en los accidentes mortales o graves.

Por lo tanto, con los datos obtenidos durante el año 2016, la tasa de accidentes en carretera ha pasado de los 11,6 muertos diarios en el año 2000 a los 3,2 fallecidos diarios en 2016.

Dentro del ámbito europeo, España presenta una tasa de 36 muertos por millón de habitantes, muy por debajo de la tasa de mortalidad media de la Unión Europea que se encuentra en 52, según los últimos datos disponibles. Se puede decir que España es uno de los países del mundo con mejores niveles de seguridad vial.

Con respecto a la siniestralidad en 2016, se describen las siguientes causas y circunstancias:

- Como ya se ha mencionado, se registra un aumento de 18,6 millones de viaje de larga distancia por carretera. Esto representa un 5% más que en el año anterior y un incremento acumulado del 10% en los tres últimos años. Desde 2014, han aumentado los movimientos en 37,8 millones.

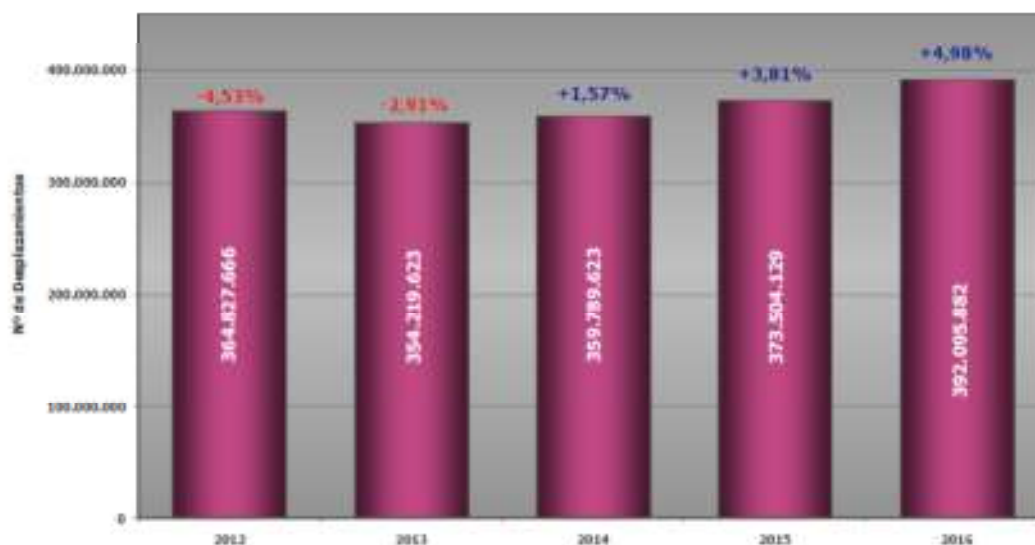


Figura 2. Desplazamientos en carretera en España (2012-2016)

- En 2016 se ha producido un aumento de la antigüedad media de los vehículos implicados en accidentes mortales, lo que incrementa el riesgo de accidentes. Los turismos en los que viajaban los fallecidos tienen una edad media de 13,6 años.
- Pero, sin lugar a dudas, el factor humano es la principal causa de accidentes y mortalidad en carretera con diferencia.

Aumentan las infracciones por consumo de drogas ilegales ya que, de todas las pruebas realizadas, en concreto 60.942, el 39% han resultado positivas, es decir una cifra de 23.822. Más en concreto, fueron realizadas 3.026 pruebas a conductores involucrados en accidentes obteniendo un resultado positivo en 829 pruebas, el 27%. Otras 3.350 pruebas fueron realizadas sobre conductores infractores con 2.114 positivos, el 63%.

Respecto del alcohol, se realizaron un total de 4,6 millones de pruebas, resultando positivas 68.852, el 1,5%.

Existe aún un reducido número de usuarios que continúa sin utilizar los elementos de seguridad. En 2016, 161 fallecidos no hacían uso de los



dispositivos de seguridad como son cinturón, casco, etc., en el momento del accidente.

Algunas características generales de la siniestralidad agrupando los datos por diferentes motivos se muestran a continuación.

Se tiene un mayor porcentaje de fallecidos del sexo masculino. La proporción es del 79%. El rango de edades con mayor porcentaje de fallecidos se sitúa en el grupo de edad de 45 a 54 años con 225 muertos, un 19% del total.

En las vías de gran capacidad, el 47% de los fallecidos en 2016 se han producido en accidentes que fueron salidas de la vía, el 20% en accidentes con colisión trasera y múltiple y el 17% en atropello a peatones.

Es de vital importancia, teniendo en cuenta estos datos, evitar a toda costa situaciones de despiste en la carretera que pueden ser debidas a una situación de estrés, cansancio físico o fatiga mental en el conductor. Sistemas que detecten estos estados y disparen un aviso directamente al conductor, podrían evitar, en la mayoría de casos, la que hasta ahora es la mayor causa de accidentes mortales.

## **2.2. Riesgos por fatiga mental y nivel de estrés**

El conductor es la persona encargada de tomar el control del vehículo y las decisiones durante la conducción. Es importante que este se encuentre en las mejores condiciones mentales y físicas a la hora de conducir. Un estado de somnolencia o estrés podría provocar graves riesgos durante la conducción que serán perjudiciales para el mismo y para todos los usuarios que se encuentren en la vía en dicho momento.

### **2.2.1. Fatiga mental**

La fatiga mental se describe como un cambio en los estados tanto psicológicos como fisiológicos que un individuo experimenta durante el desarrollo de una actividad cognitiva que demanda alta concentración durante

un prologado periodo de tiempo. Estos cambios principalmente se manifiestan en un deterioro de la eficiencia durante la realización de una labor. A consecuencia de esto, la fatiga mental está vinculada a un estado de disminución de alerta, donde gradualmente predomina el cansancio y la falta de energía.

Por lo tanto, la fatiga se muestra como un indicador de un problema implícito, la pérdida de recursos básicos de un individuo tales como reaccionar de forma oportuna y apropiadamente ante una emergencia o un imprevisto. [4]

- Causas de la fatiga:

Esta pérdida o disminución de recursos fisiológicos y de atención puede estar ocasionada o influenciada por una serie de factores propios del sujeto o de su entorno: [5]

- Cantidad de trabajo y horas laborales: tanto la sobrecarga de tareas como el trabajo monótono o repetitivo podrían condicionar la fatiga.
- Falta de descansos: la ausencia de descansos, tanto como la falta de horas de sueño, son un principal factor.
- Conducción nocturna.
- Organización: tener un deficiente diseño de turnos y horarios de trabajo y de descanso, en particular de trabajo nocturno.
- Ambiente del lugar: ruido, vibraciones, temperaturas inadecuadas (frío o calor), inadecuada iluminación, entre otros.

- Riesgos de la fatiga:

Las escalas utilizadas para medir la fatiga mental han mostrado ser subjetivas, y la autoevaluación de este mismo estado mental lo es más aún. Contar con una medición objetiva y reproducible permitiría identificar individuos en riesgo de posibles accidentes.

Una dificultad en la auto detección del nivel de fatiga es que habitualmente tiende a ser subestimada por quienes la padecen. Esto se debe a que tras una primera fase de “alarma” que avisa de la necesidad de descanso,

## 2 | SEGURIDAD EN LA CONDUCCIÓN, RIESGOS Y CAUSAS

paradójicamente aparece una fase de “resistencia” que se caracteriza por una aparente recuperación y mejora del rendimiento del individuo.

A medida que una persona padece fatiga mental, aparecen los siguientes síntomas: [5]

- Movimientos lentos o torpes.
- Disminución de la velocidad motora de reacción.
- Aparición de visión borrosa o doble.
- Dificultad para concentrarse o permanecer alerta.
- Dificultad para recordar.

Esto implica que aquel individuo que bajo los efectos de la fatiga realice una actividad donde es necesario un alto nivel de atención, como es la conducción, puede generar un accidente.

Un estado de fatiga mental durante la conducción implica un riesgo de distintos tipos de accidentes como son choques con otros automóviles o elementos de la vía, atropellos a peatones o animales y salidas de la vía con diferentes tipos de consecuencias.

Según informa el anuario de accidentes del año 2015 de la DGT (Dirección General de Tráfico), el número de accidentes y las víctimas en estos tanto en vías urbanas como interurbanas se pueden observar en la siguiente figura [3].

| TIPO DE ACCIDENTE                                   | TOTAL                   |                    |              |                        |                           |
|---|-------------------------|--------------------|--------------|------------------------|---------------------------|
|   | Accidentes con víctimas |                    |              | Víctimas               |                           |
|   | Total                   | Mortales a 30 días | Fallecidos   | Heridos hospitalizados | Heridos no hospitalizados |
| Frontal   | 2.767                   | 160                | 209          | 695                    | 4.158                     |
| Fronto-lateral                                      | 20.016                  | 154                | 174          | 1.538                  | 26.960                    |
| Lateral   | 7.610                   | 16                 | 16           | 365                    | 9.573                     |
| Por alcance   | 18.700                  | 99                 | 108          | 676                    | 27.619                    |
| Múltiple o en caravana                              | 5.266                   | 52                 | 61           | 331                    | 10.342                    |
| Colisión contra obstáculo o elemento de la vía      | 2.269                   | 23                 | 24           | 171                    | 2.678                     |
| Atropello a personas                                | 13.040                  | 303                | 306          | 1.824                  | 12.596                    |
| Atropello a animales                                | 433                     | 5                  | 5            | 38                     | 505                       |
| Vuelco  | 2.454                   | 16                 | 16           | 164                    | 2.566                     |
| Caída   | 4.034                   | 22                 | 22           | 348                    | 4.044                     |
| Sólo salida de la vía                               | 9                       | 0                  | 0            | 1                      | 16                        |
| Salida de la vía por la izquierda con colisión      | 2.237                   | 107                | 113          | 376                    | 2.787                     |
| Salida de la vía por la izquierda con despeñamiento | 241                     | 23                 | 23           | 67                     | 264                       |
| Salida de la vía por la izquierda con vuelco        | 1.374                   | 60                 | 64           | 224                    | 1.645                     |
| Salida de la vía por la izquierda, otro tipo        | 1.077                   | 30                 | 31           | 144                    | 1.202                     |
| Salida de la vía por la derecha con colisión        | 3.261                   | 142                | 150          | 508                    | 3.835                     |
| Salida de la vía por la derecha con despeñamiento   | 358                     | 27                 | 30           | 102                    | 394                       |
| Salida de la vía por la derecha con vuelco          | 2.156                   | 71                 | 72           | 285                    | 2.594                     |
| Salida de la vía por la derecha otro tipo           | 4.603                   | 35                 | 39           | 257                    | 5.474                     |
| Otro tipo de accidente                              | 5.851                   | 214                | 226          | 1.381                  | 5.708                     |
| <b>TOTAL</b>  | <b>97.756</b>           | <b>1.559</b>       | <b>1.689</b> | <b>9.495</b>           | <b>124.960</b>            |

Figura 3. Accidentes y víctimas según tipo de accidente.

### 2.2.2. Estrés

El estrés es el proceso automático y natural del cuerpo que se inicia frente a situaciones desafiantes o amenazadoras, a las cuales se les debe dar una respuesta apropiada mediante determinados recursos del organismo.

Se puede describir como una situación de tensión o preocupación que en determinadas ocasiones puede aumentar la motivación para llevar a cabo una tarea o mejorar el rendimiento. Sin embargo, también puede ser dañino si provoca sobre-estrés o afecta gravemente al desarrollo de la vida diaria.

Cuando el organismo se ve sometido a situaciones estresantes, se produce una respuesta por parte del sistema nervioso junto con ciertas hormonas. El inicio se produce en el hipotálamo, donde se envían señales a las glándulas adrenales para producir adrenalina y cortisol. Estas hormonas viajan por el torrente sanguíneo y producen variaciones sobre la frecuencia cardíaca, la frecuencia respiratoria, la presión arterial o el metabolismo. Además, el

## 2 | SEGURIDAD EN LA CONDUCCIÓN, RIESGOS Y CAUSAS

organismo produce más sudor. Al conjunto de cambios producidos se le conoce como respuesta al estrés [5].

Como se puede observar en la figura 4, existen cuatro fases en el proceso del estrés: partiendo de la fase de normalidad del individuo, aparece una fase de resistencia en la que la energía comienza a disminuir para llegar a la fase de reacción de alarma, en la que se produce el máximo gasto de energía. A consecuencia de esto, aparece la fase de agotamiento en la cual el organismo está exhausto.



Figura 4. Evolución y fases del estrés en el ser humano.

En este proyecto se presenta un enfoque del estrés considerándolo únicamente presente en eventos que producen respuestas fisiológicas. Este comportamiento se denomina Síndrome de Adaptación General y consta de tres fases: alarma, resistencia y agotamiento. Solo se consideran estímulos estresores aquellos que producen una respuesta de reajuste o reequilibrio. Dentro de la variedad de estímulos estresores están contenidos los estímulos emocionales.

- Tipos de estrés:

## 2 | SEGURIDAD EN LA CONDUCCIÓN, RIESGOS Y CAUSAS

Se encuentran hasta tres tipos de estrés. Cada uno de ellos tiene una duración, síntomas, tratamientos y características:

El estrés agudo es la forma más típica de estrés. Suele aparecer bajo determinadas situaciones y puede llegar a resultar estimulante y ayudar al desarrollo de la actividad. Sin embargo, este tipo de estrés de baja duración puede derivar en problemas más graves. Todo ello va a depender del estímulo estresor y la capacidad del individuo para actuar frente a él.

Otro tipo es el estrés agudo episódico que es similar al anterior, pero se da de forma recurrente y suele aparecer en personas con demasiadas responsabilidades, no pueden organizarlas correctamente y tienden a actuar de esta forma.

Por último, el estrés crónico se trata de un estrés que deteriora la salud de la persona, tanto a nivel mental como físico. Es un estrés a largo plazo que puede provocar infartos, apoplejía, violencia, etc. [5]

### - Estrés durante la conducción:

Durante la conducción, se llevan a cabo una serie de eventos que pueden afectar al nivel de estrés. Existen determinados eventos predecibles, como fijarse tanto en los retrovisores como en la carretera cuando se va a cambiar de carril, que aumentan el estrés. Situaciones como frenar rápidamente para evitar una colisión o tener un coche demasiado cerca no pueden predecirse y afectan en mayor medida a ese estrés. Todo ello hace que la conducción sea una de las actividades más estresantes.

Conducir es considerada como una actividad más dentro de la vida cotidiana, pero el entorno que genera puede ser mucho más estresante que para el resto de actividades. En un vehículo la movilidad se encuentra bastante limitada y existe un riesgo de no poder ejercer un control completo sobre dicho vehículo. A esto se le añade la presencia de otros conductores que pueden actuar incorrectamente.

### - Previsiones del estrés al volante:

## 2 | SEGURIDAD EN LA CONDUCCIÓN, RIESGOS Y CAUSAS

Se pueden seguir algunos consejos para evitar el estrés en la medida de lo posible antes de coger el coche.

- Organizarse y tener tiempo para afrontar con tranquilidad cualquier imprevisto que pueda surgir.
- Dormir las horas necesarias (7-8 aproximadamente)
- No imponerse metas inalcanzables y priorizar las necesidades. Resolver las tareas una tras otra, planificando el tiempo para descansar.
- Llevar un estilo de vida saludable, tener el organismo listo para enfrentarse a cualquier exigencia y descargar tensiones.
- Evitar el consumo de alcohol, tabaco, drogas o fármacos.
- Al realizar un viaje largo, tomar el tiempo suficiente y realizar paradas frecuentes.

La importancia de una buena salud mental es, por tanto, un elemento muy influyente durante la actividad de conducción. Por ello, a lo largo de los años han sido varios los estudios planteados para la evaluación de estrés o fatiga mental. En ellos se han extraído diferentes parámetros sobre un conjunto de señales fisiológicas relevantes ante la presencia de dichas situaciones.

En este proyecto se propone el análisis de las señales de electrocardiograma (ECG), electromiograma (EMG) y respuesta galvánica de la piel (GSR).

## 3. SENSORES FISIOLÓGICOS SHIMMER

---

Los sensores se definen como dispositivos capaces de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y mediante un transductor transformarlas en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo, la temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, etc. Una magnitud eléctrica puede ser un voltaje, una corriente o una impedancia. Consiste en una señal analógica y necesita una interfaz con el sistema para saber cuál es la medida que interesa. En la mayoría de casos estas interfaces son específicas o puede ser que el sensor utilice una tarjeta de adquisición.

La monitorización en tiempo real de los parámetros fisiológicos mediante sensores en los seres humanos lleva muchos problemas consigo, barreras que superar para realizar una adquisición correcta de las señales. Hace un tiempo la realización de un estudio de este tipo requería del desplazamiento de la persona hasta un centro donde se pudieran realizar estas pruebas, como pueden ser un electrocardiograma, por ejemplo. Este factor introduce mayores dificultades si, de alguna manera, se quiere llevar a cabo un diagnóstico fiable con la recopilación de datos durante un largo periodo de tiempo.

Hoy en día y con el desarrollo de la tecnología como apoyo a las aplicaciones médicas y cuidado de salud, se llega a tener una rama que cada vez tiene más auge, esta es la Telemedicina. Mediante pequeños dispositivos y la comunicación mediante redes inalámbricas podemos obtener información en cualquier lugar y cualquier momento lo que facilita y soluciona los problemas que anteriormente se exponían.

En este proyecto, como se ha planteado ya, se utilizan este tipo de sensores fisiológicos para el estudio y análisis del nivel de fatiga mental y estrés de un conductor ya que nos dan la posibilidad de hacer un seguimiento de las constantes fisiológicas del conductor y así poder analizar los cambios que se van dando en distintos escenarios y momentos de conducción.



### **3.1. Sensores de la gama *shimmer*.**

*Shimmer* es una plataforma de sensores inalámbricos desarrollados por la empresa del mismo nombre, que se utilizan en aplicaciones dedicadas a la investigación biomédica. Los sensores son dispositivos compactos y bastante ligeros. Estas características hacen que estos sensores sean muy adecuados en aplicaciones de detección fisiológica, ya que se pueden colocar fácilmente en el cuerpo humano y, sobre todo, porque son inalámbricos, y se comunican a través de *Bluetooth*.

#### **3.1.1. Equipamiento Hardware**

En un vistazo general, la parte hardware ofrecida por la plataforma *Shimmer* consta de la Unidad básica, el dispositivo *Shimmer Dock 2* y los distintos módulos de expansión que se pueden incorporar a la unidad básica y así obtener las señales de interés.

- Unidad Básica de *Shimmer*.

Estos sensores están formados por una placa base que contiene un sensor de aceleración en los tres ejes (X, Y, Z), un led que nos indica el estado en el que se encuentra el dispositivo: si este está en color verde, indica que el sensor se encuentra encendido y conectado y si está rojo indica que la batería de este es baja y necesita recargarse. Por último, cuando parpadea en naranja, nos informa de que el dispositivo está en fase de transmisión de datos. También tiene un botón para reiniciar el sensor. Incluyen una batería recargable de litio de 450 mAh, que es bastante duradera, ya que estos dispositivos son de baja potencia, por lo que consumen muy poca energía.

En la placa base se puede encontrar un conector interno que da la posibilidad de añadir diferentes módulos, como ya se ha descrito. Entre los posibles módulos posibles están el giróscopo, magnetómetro, ECG (electrocardiograma), EMG (electromiograma) y GSR (Respuesta galvánica de la piel). Para terminar, tienen una ranura para memorias Micro SD y un conector para cargar la batería, actualizar el *firmware* del sensor o conectar el

### 3 | SENSORES FISIOLÓGICOS SHIMMER

sensor al ordenador para leer los datos de la memoria SD. En la figura 5 se muestra la unidad básica y todas las características descritas.



Figura 5. Unidad básica de *Shimmer*

#### - *Shimmer Dock 2:*

Este dispositivo base llamado *Shimmer Dock* se utiliza para cargar la batería del sensor y es un aparato cuadrado que consta de cuatro botones y tres leds. El primero de ellos es para indicar si el sensor está encendido, con un color verde, el segundo indica si el sensor se está cargando, con color naranja y el último, si estamos accediendo a la memoria del sensor, parpadea con color azul, y si aparece alguna anomalía se ilumina de color rojo. Este dispositivo también sirve como intermediario para conectar los sensores al ordenador vía USB y realizar una posible actualización del *firmware* o una posible lectura de datos guardados en la memoria.

En la figura 6 se muestra el dispositivo *Shimmer Dock* indicando todos sus componentes.

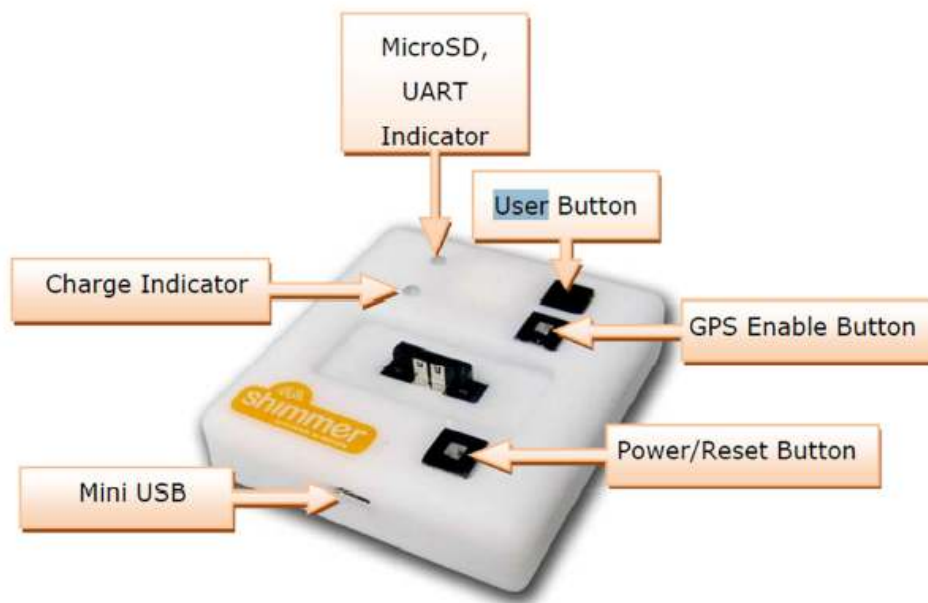


Figura 6. Shimmer Dock 2

- Módulos de expansión:

Se describen los distintos módulos de expansión que se pueden incorporar a la unidad básica.

- ECG (Electrocardiograma):

Este sensor denominado RN42-A8ED es el dispositivo que se emplea para medir y mostrar de forma gráfica la actividad eléctrica del corazón, evaluando el ritmo y la función cardiaca. La definición y características de la señal de electrocardiograma se explican en un capítulo posterior a la hora de implementar su adquisición en tiempo real.

El módulo ECG se añade a la unidad básica y ofrece la posibilidad de analizar la respuesta del corazón ante situaciones distintas durante la conducción [7].

En la figura 7 se muestra, de forma simplificada, el diagrama de bloques por los que está compuesto este sensor.

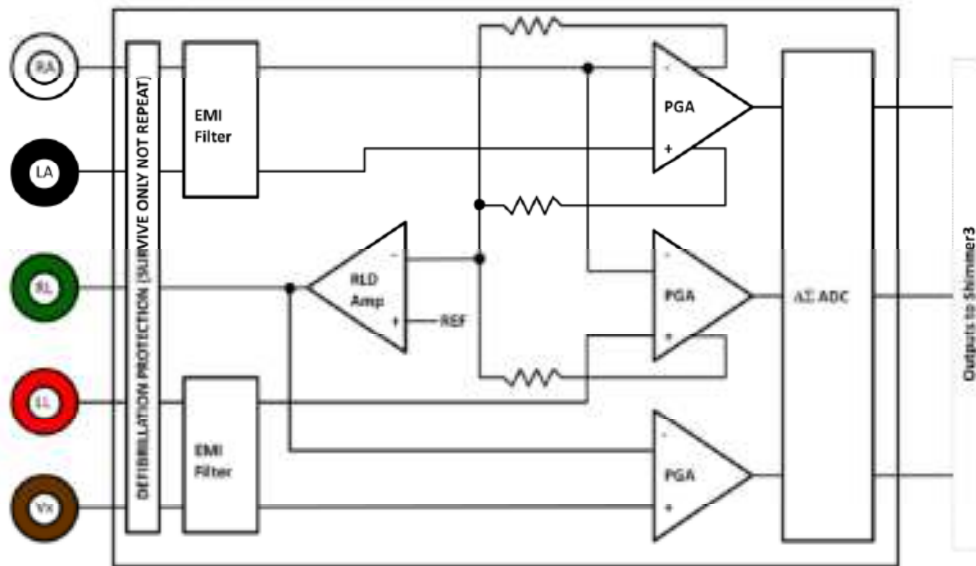


Figura 7. Diagrama de bloques simplificado del sensor ECG

La adquisición de esta señal se realiza por medio de electrodos adheridos al cuerpo de sujeto y se colocan en ciertas posiciones específicas que se denominan “derivaciones”, y dependiendo de la derivación es como se va a obtener la forma de onda del ECG.

Existen tres tipos de derivaciones: monopares, bipolares y precordiales. El primer tipo consiste en tan solo la medición del ECG con respecto a una referencia. Esto solamente utiliza dos electrodos: uno para medir y otro que actúa de “tierra”. El segundo caso agrega un tercer electrodo que mide la señal en polaridad negativa, entonces se tienen dos polos (positivo y negativo) y una tierra, lo que da un total de tres electrodos.

Las derivaciones precordiales utilizan seis electrodos adheridos a la piel. Su principal ventaja es que entregan distintas tomas o registros para una misma secuencia de latidos, por lo que este tipo de derivaciones es el utilizado para exámenes médicos.

En la figura 8 se muestran las distintas derivaciones que se acaban de explicar.

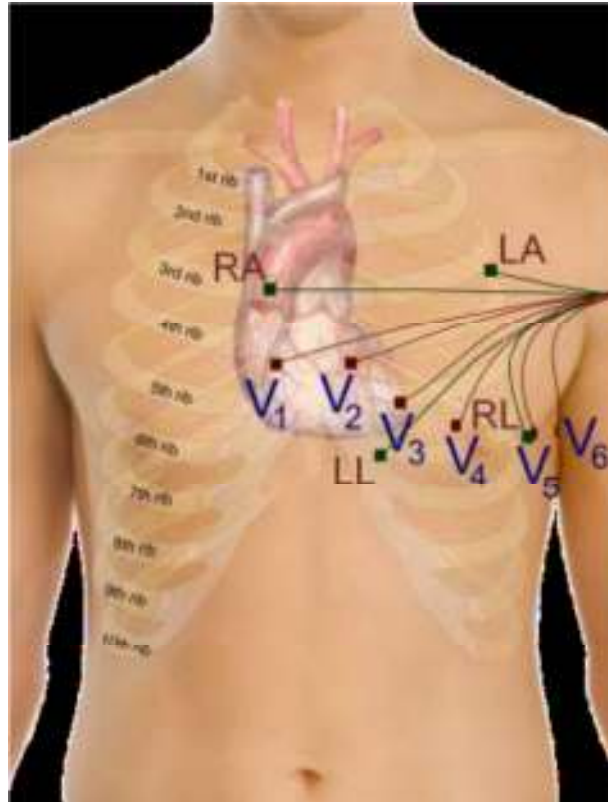


Figura 8. Ejemplo del posicionamiento de los electrodos para la medida ECG

Siguiendo el manual de usuario del sensor ECG utilizaremos el método denominado como las derivaciones bipolares utilizando cuatro electrodos:

- LA *Left Arm*
- RA *Right Arm*
- LL *Left Leg*
- RL *Right Leg*

Por lo tanto, en la unidad del sensor, podremos obtener las medidas llamadas:

- (LA-RA), es la señal del vector ECG medida desde la posición RA (brazo derecho) hasta la posición LA (brazo izquierdo) y esta se emite en el canal ExG1 Ch2 de la unidad ECG.
- (LL-RA), es la señal del vector ECG medida desde la posición RA (brazo derecho) hasta la posición LL (pierna izquierda) y esta se emite en el canal ExG1 Ch1 de la unidad ECG.
- (LL-LA), es la señal del vector ECG medida desde la posición LA (brazo izquierdo) hasta la posición LL (pierna izquierda) y esta se obtiene

### 3 | SENSORES FISIOLÓGICOS SHIMMER

restando la primera derivación (LA-RA) de la segunda derivación (LL-RA).

En la figura 9 se puede observar el módulo del sensor ECG y una ilustración con la posición que deben tener los electrodos.

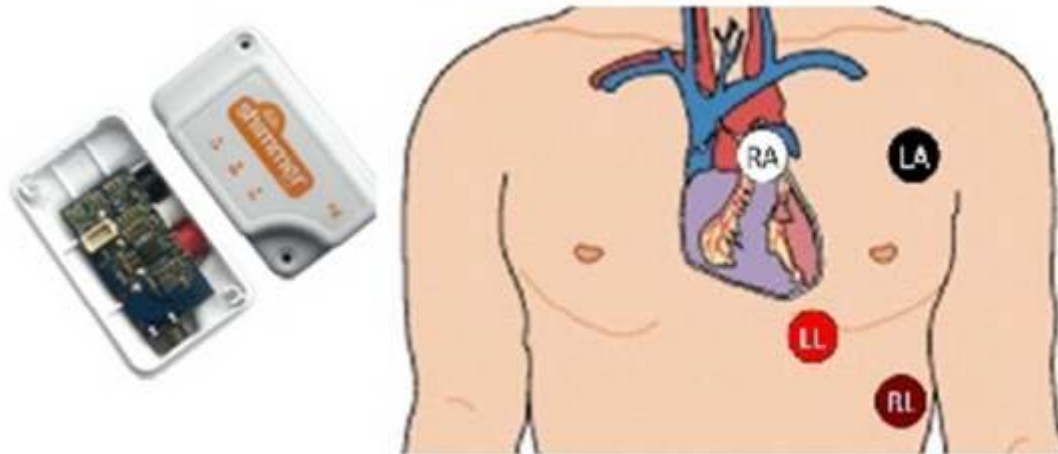


Figura 9. Módulo ECG y colocación de electrodos

Ambas señales obtenidas con las dos primeras medias o derivaciones tienen una forma de onda muy similar, aunque en este caso se recoge con mayor amplitud la LA-RA por lo que se ha decidido utilizar esta señal en el capítulo posterior de adquisición y procesamiento de la señal.

- EMG (Electromiograma):

Este sensor se utiliza para registrar y medir la actividad eléctrica asociada a las contracciones musculares. Puede evaluar la conducción nerviosa, la respuesta del músculo, el nivel de activación o se puede utilizar para analizar y medir la biomecánica del movimiento humano.

El módulo RN42-B53C es un sensor de superficie y, por lo tanto, es una representación de la actividad de todo el músculo. Para lograr el funcionamiento correcto hay que seguir una serie de normas a la hora de colocar los electrodos encima del músculo [8]:

### 3 | SENSORES FISIOLÓGICOS SHIMMER

- Los electrodos positivos y negativos deben colocarse en paralelo con las fibras musculares del músculo que se está midiendo.
- El electrodo de referencia debe ser colocado en un punto eléctricamente neutral del cuerpo, tan lejos como sea razonablemente posible del músculo que se está midiendo. El tobillo o la muñeca suelen ser recomendables.
- El electrodo no se debe colocar ni cerca ni sobre el tendón del músculo. Tampoco se debe colocar sobre el punto motor, ni próximo al “borde” del músculo.
- La distancia máxima entre los electrodos positivo y negativo no debe superar 10 cm. Se recomienda la distancia más corta posible.

En la figura 10 se muestra el módulo EMG y un ejemplo de la posición que deben tener los electrodos.



Figura 10. Módulo EMG y colocación de electrodos

En este proyecto se ha decidido situar este sensor en el músculo trapecio de la parte izquierda del cuerpo. De esta forma, registraremos la sobrecarga que pueda tener este tras un periodo largo de conducción o si se producen maniobras bruscas debido a correcciones en la ruta.

- GSR (Respuesta Galvánica de la piel):

Los cambios en la resistencia galvánica de la piel dependen de ciertos tipos de glándulas sudoríparas que son abundantes en las manos y los dedos.

### 3 | SENSORES FISIOLÓGICOS SHIMMER

Esto es la respuesta galvánica o conductancia de la piel, que consiste en una forma de medir las respuestas del sistema nervioso autónomo, específicamente del sistema nervioso simpático. Esto se detallará más adelante en el capítulo de su implementación.

El módulo RN42-938E supervisa la resistencia de la piel entre dos electrodos unidos a dos dedos de una mano, como se puede ver en la figura 11 [9].



Figura 11. Módulo GSR y colocación de electrodos

En el caso de este proyecto, la configuración de este módulo se puede hacer en 5 tipos de configuración dependiendo del rango de resistencia o conductancia que se quiera adquirir. Se ajusta la configuración al llamado “*Auto-Range*” que establecerá el rango al valor más adecuado para la lectura actual.

En el software ofrecido por *Shimmer* están implementados los métodos que permiten convertir entre unidades de resistencia y conductancia. En este caso elegiremos la medida de resistencia en KOhm.

- 9DOF:

El módulo 9DOF de *Shimmer* combina la función de un giróscopo con la de un magnetómetro para ofrecer una potente solución de detección cinemática. Este módulo, combinado con el acelerómetro incorporado en la unidad básica, puede proporcionar una orientación estática y dinámica, de medición inercial y convertirse en un sensor complejo de movimientos en 3D.

Este módulo es altamente configurable, y ofrece la posibilidad de realizar una calibración. El sensor magnético reduce el error de deriva en los cálculos



cinemáticos, permitiendo así capturar excelentes datos para una gran gama de aplicaciones.

El giróscopo da la velocidad de los cambios de orientación con respecto a los tres ejes. Esta velocidad la ofrece en ángulo recorrido por segundo.

En la figura 12 se muestra una ilustración del módulo 9DOF.



Figura 12. Módulo 9DOF

#### 3.1.2. Emparejamiento de los sensores

Para poder trabajar con los sensores *Shimmer*, se debe configurar la conexión *Bluetooth*. A este proceso se le llama “emparejamiento”. Es un proceso similar al de los dispositivos móviles o *smartphones*. Uno de los requisitos es obviamente un adaptador *Bluetooth* para efectuar la conexión.

Desde el PC se debe seguir la ruta Inicio -> Panel de control -> Hardware y sonido -> Agregar un dispositivo. En la ventana que se abre se pueden observar los distintos dispositivos encontrados dentro del área de cobertura. Se debe seleccionar uno de estos y, cuando se pida la clave de asociación, introducir “1234”. Después de aceptar el sensor, ya estaría emparejado con el ordenador correspondiente.

En cualquier dispositivo Android el proceso de emparejamiento es muy similar y se debe realizar desde el apartado Ajustes -> Conectividad -> Bluetooth. Una vez activado, aparecen los dispositivos en su área de cobertura, al seleccionarlo se introduce la clave y se acepta. Los sensores están correctamente emparejados y listos para ser utilizados.

### 3.1.3. Calibración

Los sensores pueden necesitar de una calibración previa para mejorar su precisión antes de ser utilizados. Para esto se utiliza la aplicación *Shimmer 9DOF Calibration*, que es una aplicación desarrollada para permitir a los usuarios calcular los parámetros de compensación para el acelerómetro, la sensibilidad y la alineación del eje y. En general permite un proceso de calibración automatizado de las señales del acelerómetro, ya que se almacenan los parámetros calculados en la memoria del sensor. En la figura 13 aparece la interfaz de la aplicación utilizada.



Figura 13. Aplicación Shimmer 9DOF

## 4. SISTEMA OPERATIVO: ANDROID

---

### 4.1. Introducción y Arquitectura

En este momento, Android es el sistema operativo móvil más utilizado en *smartphones* y dispositivos electrónicos de todo el mundo, compite directamente con iOS, el sistema operativo de Apple, y Windows Phone, el sistema de Microsoft.

Según el ONTSI (Observatorio Nacional de las Telecomunicaciones y de la SI) [10], la mayoría de los *smartphones* tienen sistema operativo Android (más de un 80%) seguido de iOS (con casi un 10%), sumando entre ambos más de un 90%, mientras que en las tabletas, los dispositivos con iOS son un 20% y los que integran Android son el 60%. Se determina que Android es el sistema operativo de mayor alcance para los usuarios y, por lo tanto, el adecuado para realizar el tipo de aplicaciones como la que desarrollamos en este proyecto. Estos datos se pueden observar a continuación en la figura 14.

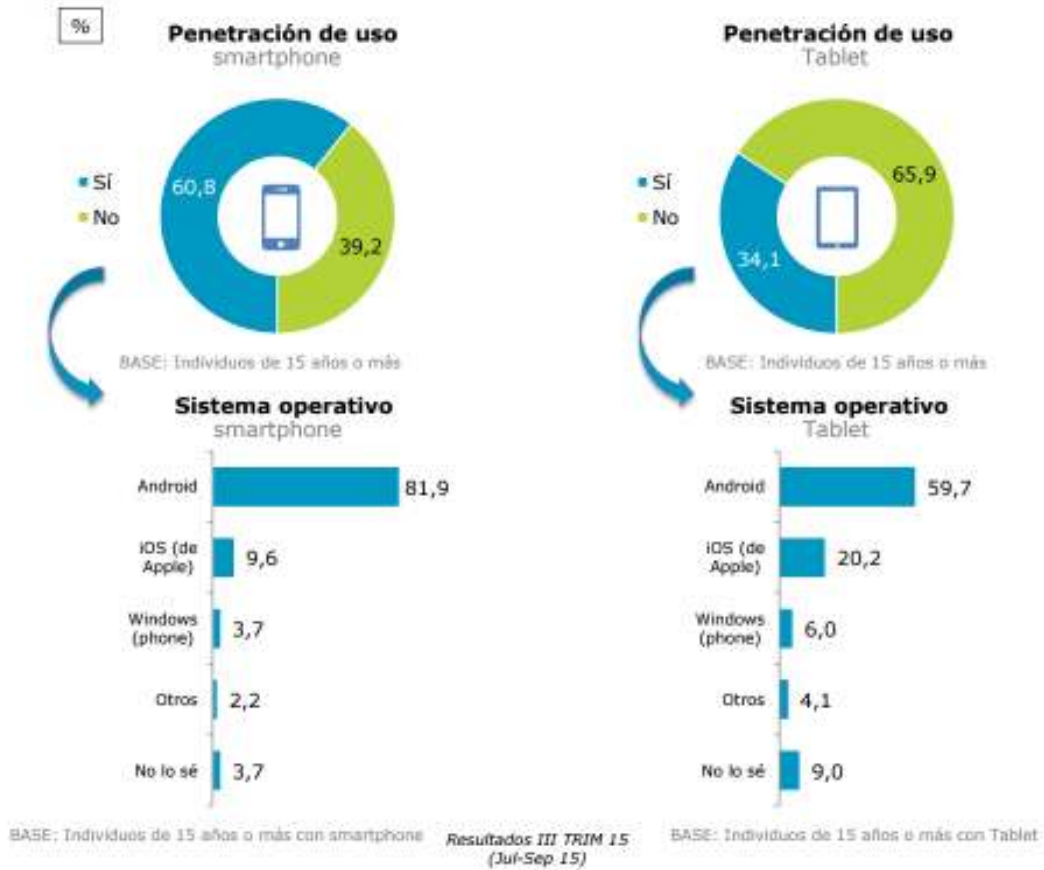


Figura 14. Estudio de uso de Smartphone y Tablets en la sociedad

Este sistema se basa en el núcleo de Linux, diseñado fundamentalmente para dispositivos móviles con pantallas táctiles. Fue desarrollado por la compañía Android Inc., que posteriormente fue comprada por Google en 2005. Android fue presentado en noviembre de 2007 junto a la fundación *Open Handset Alliance*, un conjunto de compañías de software, hardware y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. El primer dispositivo móvil con Android (HTC Dream) salió a la venta en octubre de 2008.

Actualmente, Android consta de más de 1 millón de aplicaciones disponibles en la tienda Google Play, donde más de dos tercios son gratuitas. Tiene una gran comunidad de desarrolladores que continuamente están aportando aplicaciones con nuevas funcionalidades.

Android como sistema operativo consta de varias partes o componentes principales:

- Núcleo Linux: Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los *drivers* necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Android se apoya en Linux para la mayoría de servicios de bajo nivel, como seguridad, gestión de memoria, de procesos y el modelo de controladores utilizados.
- Tiempo de ejecución: Este sistema proporciona la mayor parte de las funciones disponibles en Java. Lo constituyen las *Core Libraries*, que son librerías con multitud de clases Java y la máquina virtual Dalvik, en la que cada aplicación ejecuta su propio proceso.
- Librerías: Estas han sido escritas utilizando el lenguaje de programación C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Estas librerías son parte del corazón de Android.
- Marco de trabajo: Esta capa permite la reutilización de componentes, ya que cada aplicación puede publicar sus capacidades y cualquier otra puede hacer uso de estas. Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Todas las aplicaciones que se desarrollan utilizan el mismo conjunto de APIs y el mismo “framework”, representado por este nivel.
- Aplicaciones: Este nivel contiene, tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo. Todas estas aplicaciones utilizan los servicios, las APIs y las librerías de los niveles anteriores.

La mayoría de aplicaciones de Android se desarrollan en Java, aunque también existe otra herramienta que permite el desarrollo de estas en C o C++. Esto es una de las razones por las que este sistema operativo es tan utilizado.

### 4.2. Versiones de Android

En cuanto a las versiones de este sistema operativo, una aplicación desarrollada para una versión de Android es compatible tanto con esa versión como con las posteriores a esta. Es decir, cuanto menor es la versión para la

cual se desarrolla la aplicación, mayor será el número de dispositivos móviles con los que será compatible.

Alguna de estas versiones son las denominadas JellyBean (Android 4.1), KitKat (Android 4.4), Lollipop (Android 5.0), Marshmallow (Android 6.0) y actualmente se ha desarrollado la versión Oreo (Android 8.0). A continuación, en la figura 15 se muestra el logotipo de la última versión de Android (Oreo).



Figura 15. Logotipo Android 8.0.0

Las principales ventajas que tiene desarrollar una aplicación en Android es el uso de programación Java ya que es extendido y de fácil comprensión y se basa en la licencia Apache y, por lo tanto, tiene licencia libre en la que cualquier programador puede modificar las partes más internas. Además, se pueden crear aplicaciones sin tener que pagar ninguna cuota, al contrario que en iOS.

### **4.3. Herramientas de desarrollo**

El desarrollo de aplicaciones para Android se hace habitualmente con un lenguaje de programación Java, por lo que se debe instalar un kit de desarrollo Java SE. Actualmente se utiliza el entorno de desarrollo Android Studio proporcionado por Google de manera gratuita y que proporciona todas las herramientas necesarias para la creación de aplicaciones. Incluye un editor de

#### 4 | SISTEMA OPERATIVO: ANDROID

código de primer nivel, depurador, y es un sistema instantáneo de compilación e implementación. La instalación de Android Studio conlleva la instalación también del SDK de Android. Su logotipo se muestra en la figura 16.



Figura 16. Logotipo Android SDK

El SDK es un conjunto de herramientas de desarrollo que incluye la API, para todas las versiones de Android disponibles en el mercado. En este proyecto se ha instalado y hecho compatible la aplicación hasta la última y más reciente versión 8.0.

Ya que partimos de la aplicación base desarrollada por Amine Khadmaoui [1], (Se utiliza el IDE (*Integrated Development Environment*) Eclipse, por simplicidad a la hora de importar todos los archivos del proyecto.

Se ha utilizado la última versión disponible de esta herramienta. Eclipse Java Oxygen, a la que hay que instalarle el complemento ADT (*Android Development Tools Plugin*) para poder tener las herramientas relacionadas para el desarrollo de aplicaciones Android.

En la figura 17 se muestra el logo de la nueva versión de Eclipse.



Figura 17. Logotipo Eclipse Java Oxygen

## 5. IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

---

### 5.1. PROCESADO ONLINE

#### 5.1.1. Aplicación de Android base

El proyecto parte de una aplicación Android ya existente creada por Amine Khadmaoui (2014) [1] y que fue desarrollada para el control y manejo de datos que nos ofrecen los sensores de la gama Shimmer.

En esta aplicación ya están implementadas diferentes funcionalidades de las que nosotros partiremos y utilizaremos:

- Desde la pantalla principal de la app se controla el Bluetooth de nuestro dispositivo Android, de tal forma que se escanean y muestran todos los dispositivos Shimmer que se encuentren encendidos.



Figura 18. Pantalla principal de la aplicación Shimmer App

- Una vez realizado el escaneado de sensores, se permite seleccionar uno o varios para el manejo de datos de estos dispositivos. Tenemos



## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

otra pestaña donde se puede elegir la configuración de cada dispositivo (Modulo, frecuencia de muestreo) y también permite iniciar o detener la recogida de datos para cada uno de ellos. En otra pestaña se muestran las gráficas en tiempo real con los datos básicos que se recogen, como puede ser el módulo de acelerómetro y modulo propio para cada sensor.

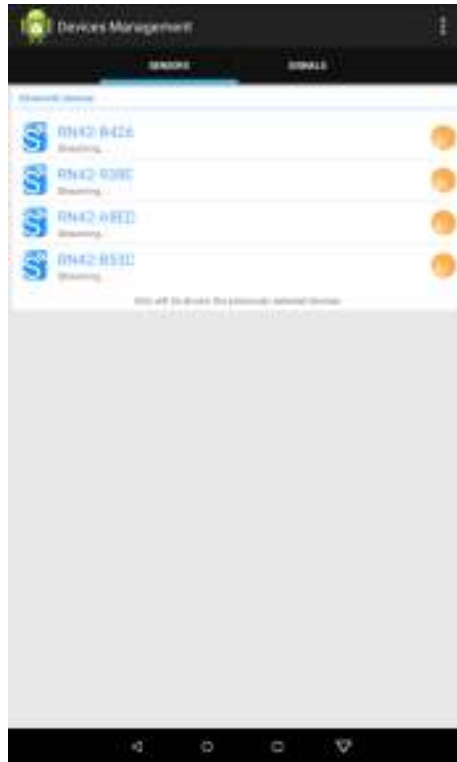


Figura 19. Pantalla de manejo de los dispositivos en la aplicación

- Otra funcionalidad implementada es el almacenamiento en fichero de los datos provenientes de los sensores y la gestión de ficheros que se crean desde la propia aplicación pudiendo ser agrupados, eliminados y compartirlos con otra aplicación.



Figura 20. Pantalla de manejo de ficheros en la aplicación

### 5.1.2. Introducción sobre las nuevas implementaciones

Utilizando las funcionalidades ya presentes en la aplicación base, comenzaremos con la finalidad de este proyecto: desarrollar las nuevas implementaciones principales para los sensores que nos proporcionan las señales ECG, EMG y GSR. Estas implementaciones son:

- Recoger de forma más extensa datos para el sensor RN42-A8ED (ECG), RN42-B53C (EMG) y RN42-938E (GSR).
- Desarrollar un algoritmo para el procesamiento en tiempo real de la señal de electrocardiograma, calculando los parámetros que nos sean de mayor utilidad para detectar un estado de fatiga en el individuo durante la conducción y alertarnos si existe un riesgo elevado por ello. En este caso, se realiza una extracción de parámetros en el dominio del tiempo y también en el dominio de la frecuencia.
- Desarrollar un algoritmo para el procesamiento en tiempo real de las señales del electromiograma (EMG) y resistencia galvánica de la piel

(GSR), de tal forma que obtengamos los parámetros de mayor utilidad para detectar un estado de fatiga en el sujeto que esté siendo monitorizado.

- Mostrar todos los parámetros calculados para cada señal en tiempo real a través de la interfaz de la aplicación. Generar y almacenar en ficheros todos los datos y parámetros correspondientes a cada señal.

El primero de los puntos descritos es la extracción de datos desde los sensores. Una vez que estos estén conectados e iniciados correctamente, adaptamos la configuración de cada sensor en concreto. A cada uno se le asigna automáticamente su módulo, que tiene el mismo nombre que la señal, por el cual a través de la clase *GraphView.java* y los diferentes métodos que se encuentran en ella, como puede ser *setAdjustement()*, se representa la señal de datos extraída en las gráficas de la interfaz, ajustándola adecuadamente dependiendo del sensor que provenga.

Otro aspecto de la configuración es la frecuencia de muestreo que se utiliza para extraer los datos. En el caso del sensor GSR, una frecuencia baja de 10Hz basta para la adquisición de la señal [9]. Para las señales ECG y EMG se recomiendan frecuencias de muestreo relativamente altas para detectar y adquirir correctamente los impulsos eléctricos provenientes del corazón y los músculos sobre los que se esté haciendo la medición. En nuestro caso se utiliza una frecuencia de 128Hz, aunque la señal EMG puede ser extraída correctamente con una frecuencia más baja de 50Hz. [7] [8]

Los datos de los sensores se estructuran de la forma que nos indica la API de la librería de Shimmer. Se tiene una clase *ObjectCluster* que contiene un *Multimap* de JAVA llamado *PropertyCluster*, donde se tiene una lista de elementos a los que accedemos con una clave, que representa cada propiedad y obtenemos el valor o *FormatCluster* de esa propiedad ya que contiene el formato, las unidades y los datos de esta. La estructura explicada se resume en la siguiente figura:

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

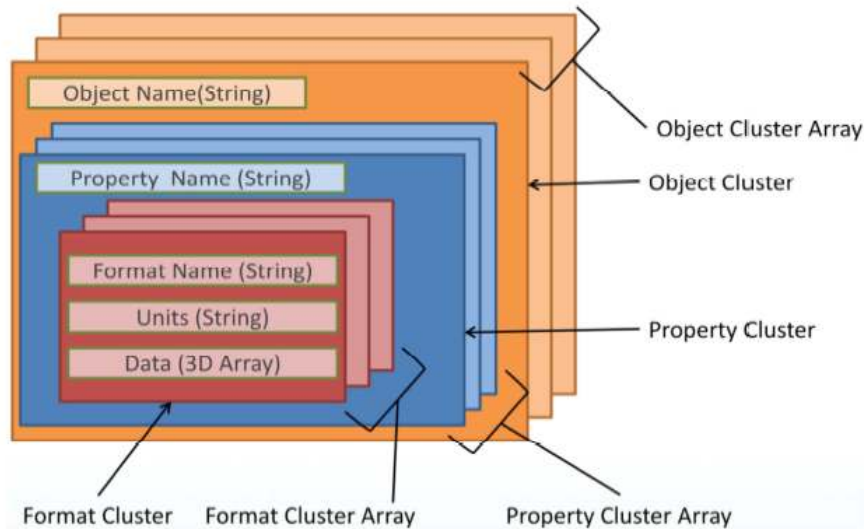


Figura 21. Estructura de datos en librería Shimmer

Las líneas de código que se corresponden con la extracción de datos serán las siguientes, en este caso se muestran para la adquisición de la señal EMG:

```

for (j=0; j<ModuleNames[Dev].length; j++) {
    formats =
    objectCluster.mPropertyCluster.get (ModuleNames[Dev][j]);
    formatCluster =
    ((FormatCluster)ObjectCluster.returnFormatCluster(formats, "RAW"));
    // Datos para dibujar la gráfica, sabemos límites y se escalan mejor.
    if (formatCluster!=null) ModuleDataArray[j] = (int)formatCluster.mData;
    formats =
    objectCluster.mPropertyCluster.get (ModuleNames[Dev][j]);
    formatClus =
    ((FormatCluster)ObjectCluster.returnFormatCluster(formats, "CAL"));
    // Datos para guardar en fichero, unidades conocidas.
    if (formatCluster!=null) TotalDataArray[i+j] = formatCluster.mData;
}

//Sacamos datos del EMG para procesarlos.

Collection<FormatCluster>
EMGprocessingFormats=objectCluster.mPropertyCluster.get ("EMG");
//Datos para procesar: sacar medias, etc.
formatCluster=((FormatCluster)ObjectCluster.returnFormatCluster(EMGpro
cessingFormats, "CAL"));
if (formatCluster!=null) emg = (double)formatCluster.mData; //Guardamos
en emg el valor sacado del sensor.
myEMG.setValorEMG(emg); //Llamamos al método de la clase que va a
procesar los datos del EMG y le pasamos el valor sacado

formats = objectCluster.mPropertyCluster.get ("Timestamp");
formatCluster =
((FormatCluster)ObjectCluster.returnFormatCluster(formats, "CAL"));
// Datos para guardar en fichero, unidades conocidas.
if (formatCluster!=null) tiempoEmg = formatCluster.mData;
myEMG.setTimeofexec (tiempoEmg);

```

Una vez ya tenemos la adquisición de datos para cada sensor y que nos proporcionan las correspondientes señales ECG, EMG y GSR, comenzamos con el procesado de estas.

### 5.1.3. Procesado online de la señal ECG

La señal de electrocardiograma es la representación gráfica de la actividad eléctrica del latido cardiaco. Esta actividad es adquirida mediante electrodos adheridos en partes estandarizadas del cuerpo, como se ha descrito en el capítulo 2, y que captan los cambios de voltaje en mV.

Al escoger un determinado tipo de derivaciones para la adquisición de la señal, como se explica en el capítulo 3, esta no recoge al 100% la actividad eléctrica del corazón, la captura de la señal es tan débil que se encontrará con diferentes perturbaciones que alteran los resultados deseados: [11].

1. Interferencia de línea de potencia: es el ruido en la banda de 50Hz o 60Hz y sus armónicos procedente de la red eléctrica y que se acopla como interferencia. También puede proceder de la interferencia electromagnética de complejos cercanos de alta potencia de radio, televisión, etc.
2. Ruido por contacto de electrodos: es el ruido causado por las pérdidas de contacto entre el electrodo y la piel. Se puede dar el caso de pérdida permanente o intermitente.
3. Contracción muscular (EMG): las contracciones musculares causan potenciales de niveles de milivoltios. La línea base de electromiograma está en el rango de los micro-voltios por lo que es usualmente insignificante, aunque las señales resultantes de esta contracción pueden ser asumidas como transitorios y hacerse notables en el espectro de la señal.
4. Desplazamiento de la línea base: este desplazamiento de la línea base puede ser representado como una componente sinusoidal en la frecuencia de respiración adherida a la señal. La amplitud de la señal ECG puede variar 15% debido a la respiración y esta variación puede

producir modulación de la amplitud del ECG. El espectro de la señal contaminada muestra componentes adicionales de 0.15Hz hasta 0.3Hz por lo que es de especial cuidado el procesado de este tipo de ruidos.

Tomando todo esto en consideración y siguiendo las especificaciones recomendadas en el manual de usuario ECG de Shimmer [7], realizaremos el filtrado y limpieza de la señal implementando dos tipos de filtro.

El primero es un filtro *band-stop* que elimine la componente frecuencial de 50Hz y sus armónicos provenientes de la red eléctrica. En el correspondiente procesamiento *online* utilizaremos una librería que contiene las funciones de filtrado y clases predefinidas con los distintos tipos de filtro: Bessel, Butterworth y Chebysev de tipo I y II.

Esta librería llamada IIR filter library está escrita en Java y tiene la correspondiente licencia (Apache License, Versión 2.0. January 2004)

Para la implementación del filtro Notch o “*band stop*” se define un objeto de clase Butterworth. Este objeto utiliza el método *bandstop()* con el que definimos las propiedades del filtro como la frecuencia central de 50Hz y la frecuencia de muestreo y el orden de dicho filtro, en este caso de orden 2.

También se realiza la definición de un segundo filtro también de la clase Butterworth pero esta vez utilizando el método *highPass()* ya que siguiendo las recomendaciones del manual de usuario ECG es necesario eliminar las componentes de baja frecuencia de la señal para eliminar el ruido por el desplazamiento de la línea base utilizando un filtro paso alto. La frecuencia de corte para este filtro es de 0.5Hz y también tendrá un orden 2:

```
float Fs = myPrefs.getFloatSAMPLINGRATE(); //Obtenemos la
frecuencia de muestreo aplicada a esta señal.

filtroNotch.bandStop(2, Fs, 50, 1); //Definición del filtro
"bandstop" de orden 2, frecuencia central 50Hz y ancho de banda
1Hz.
filtroECG.highPass(2, Fs, 0.5); //Definición del fitro paso alto
de orden 2 y con frecuencia de corte 0.5Hz
```

Para aplicar estos filtros a la señal se utiliza el método *filter()* que realiza un filtrado muestra a muestra en tiempo real a medida que vamos adquiriendo cada dato de la señal:

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

```
Collection<FormatCluster>
ECGprocessingFormats=objectCluster.mPropertyCluster.get("ECG RA-
LL"); //Datos para procesar.

formatCluster=((FormatCluster)ObjectCluster.returnFormatCluster(
ECGprocessingFormats, "CAL"));
if(formatCluster!=null)    ecg = (double)formatCluster.mData;
//Guardamos en ecg el valor sacado del sensor.

ecg = filtroNotch.filter(ecg); //Filtro para eliminar el ruido
de 50Hz de la red.
ecg = filtroECG.filter(ecg); //Filtro paso alto para eliminar
las componentes de baja frecuencia (baseling wondering)
```

Tras haber eliminado las interferencias y perturbaciones de la señal de electrocardiograma, es conveniente analizar y estudiar las características de esta. En la siguiente figura se muestran las ondas que componen la señal y su nomenclatura[12].

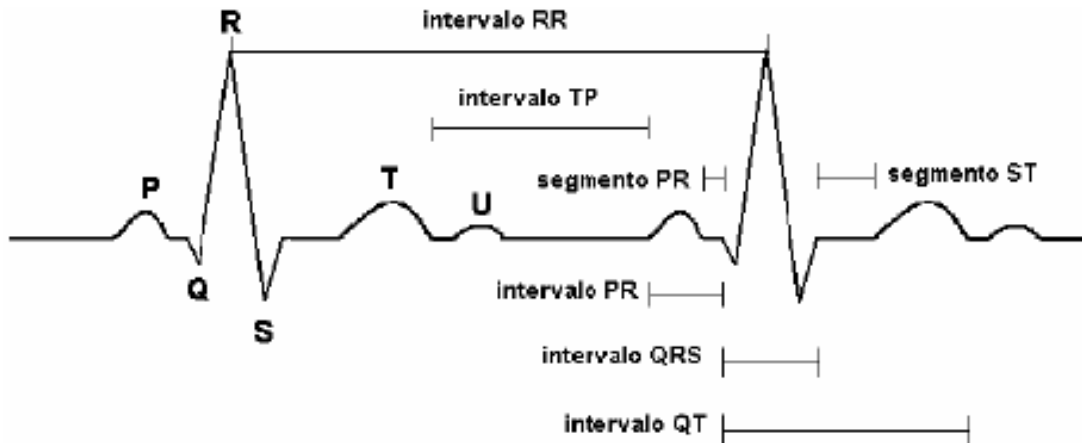


Figura 22. Forma de onda de la señal ECG

La primera onda que se detecta es la onda P. Está relacionada con la despolarización de las aurículas. En general, su duración no debe exceder los 100ms y el voltaje máximo que alcanza es de 0,25 mV.

Seguidamente aparece el complejo QRS formado por la unión de las ondas Q, R y S. Representa la despolarización de los ventrículos, aunque coincide

también con la repolarización de las aurículas. Es el de un periodo de mayor amplitud pudiendo alcanzar los 3,5 mV y con una duración comprendida entre 80 y 120 ms:

- La onda Q es la primera que aparece en el complejo y tiene valores negativos.
- La onda R es la que sigue a la onda Q, es positiva y presenta la mayor amplitud de tres y de todo el periodo de señal ECG.
- La onda S es la última onda del complejo también negativa.

El segmento S-T es el tiempo durante el cual los ventrículos se contraen y vacían y tiene una duración aproximada de entre 70 y 80 ms.

La onda T representa la repolarización de los ventrículos con una duración máxima de 200 ms y una amplitud menor que el complejo QRS, alcanzando los 0,5 mV.

Por último, con una escasa amplitud se encuentra la onda U que suele aparecer en las derivaciones precordiales, explicadas en el capítulo 2 del presente proyecto. Su amplitud es un tercio de la onda T.

### **5.1.3.1. Detección de la onda R**

El primer paso que se debe dar en el procesado de la señal ECG con la finalidad de obtener los parámetros de interés para el objetivo de hallar indicadores acerca del estado físico y mental del individuo es la detección del complejo QRS, más en concreto detectar la onda R y almacenar su localización temporal. Para ello seguimos los pasos establecidos en el estándar internacional definido en 1996 [13].

Por esto es necesario implementar un algoritmo eficiente de detección de la onda o los picos R en tiempo real. Este es el primer objetivo de cualquier algoritmo de procesado de señales ECG y es el primer paso después del filtrado definido en el estándar [13].

Existen varios algoritmos para la detección automática de los picos R que se han propuesto durante las últimas tres décadas y que se ponen en práctica.



Por poner algún ejemplo, podemos citar el método *Pan-Tompkins*, en el que se realiza un primer filtrado de la señal, las sucesivas diferencias entre muestras y luego el cuadrado de estas para luego ser integradas a través de un integrador de ventana móvil.

Otros de los métodos son los basados en la *transformada Wavelet*, en los que esta transformada es definida como la integral de los productos entre la señal y una función estandarizada. El cálculo del área bajo la curva o la descomposición de modo empírico de la señal son algunos de los otros muchos algoritmos que se pueden encontrar [14].

Pero lo que se va a desarrollar en este proyecto es un algoritmo que tenga en cuenta aspectos relacionados con el procesamiento en tiempo real de señales que puedan mantener algún tipo de perturbaciones o ruido tras las etapas de filtrado y que tenga un bajo coste computacional relativamente hablando. Se realiza una adaptación, de acuerdo a las características de la señal obtenida desde el sensor Shimmer, del método propuesto en el artículo de referencia [15], que define el llamado detector triangular.

- *Detector triangular del complejo QRS:*

Ya que el complejo QRS es definido por una onda o pico positivo (onda R) en vecindad de los valores más pequeños de la señal como son la onda Q y la onda S y la duración de este complejo está comprendida entre 80 y 120 ms, para implementar la detección del pulso R se propone que para cada muestra de la señal localice y compare dos muestras de la señal situadas un tiempo mínimo anterior y posterior siempre dentro del tiempo de duración QRS. Si las tres muestras comparadas forman un triángulo con su vértice en un rango positivo, superando un valor umbral y los picos que forman la base situados en un rango negativo también superando el umbral correspondiente, habremos localizado de forma correcta el complejo y, en concreto, la onda o pico R.

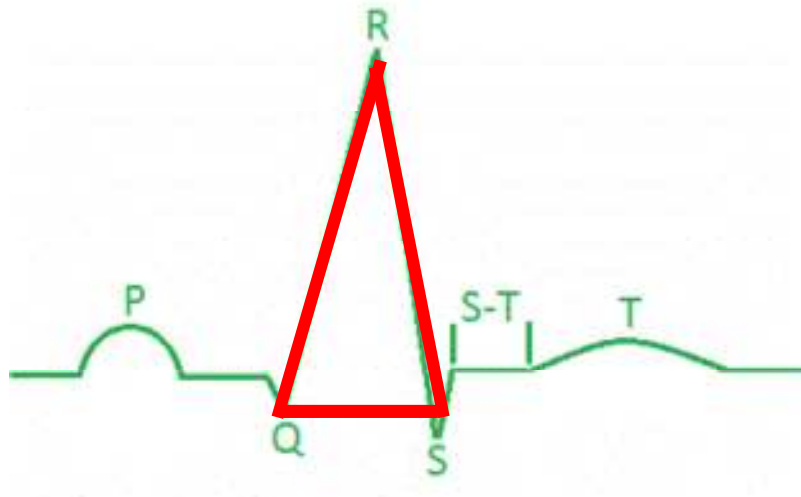


Figura 23. Triángulo formado por el complejo QRS

No todas las muestras de la señal son correctas para la detección del complejo QRS debido a las morfologías que puede sufrir la señal.

En la adaptación que se realiza en este proyecto de este método se tiene en cuenta que la señal proveniente del sensor Shimmer RN42-A8ED recoge de una forma diferenciable al resto de muestras los impulsos para la onda R y la onda S, dejando así la onda Q con una amplitud muy similar a la de la línea base.

Por lo tanto, no se centra en la detección entera del complejo QRS sino solo de la parte que forma la onda R y la S.

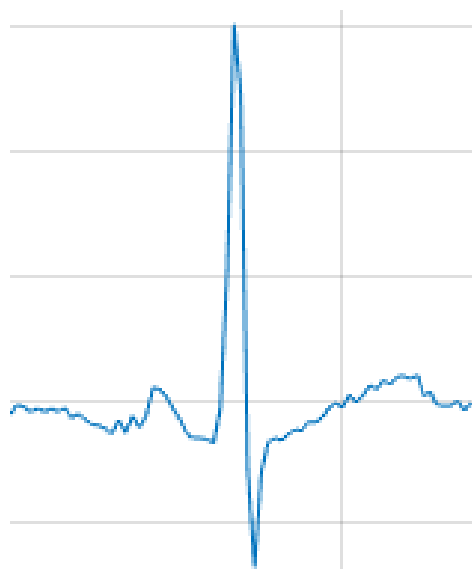


Figura 24. Forma de onda de la señal ECG obtenida del sensor fisiológico Shimmer

Para el desarrollo del procesado de esta señal se ha creado una clase encargada de ello, *ECGDataprocessing.java*, que contiene diferentes métodos con algoritmos y cálculo de parámetros. El método principal *setValorECG()* es el encargado de la detección del pico R mediante el algoritmo que hemos definido. Este método utiliza dos parámetros como son el valor y el tiempo de la muestra de la señal que está siendo adquirida en ese instante.

Durante el primer segundo de adquisición esas muestras serán utilizadas para encontrar el máximo y mínimo absoluto de la señal con lo que almacenaremos el valor de un pico R y el valor de un pico S y así calcular los umbrales a partir de los cuales estarán comprendidos las siguientes ondas R y ondas S (Threshold MAX y MIN).

Estos umbrales son calculados durante el primer segundo de adquisición de la señal. Para ello, se almacena el máximo y mínimo valor de esta en dicho periodo de tiempo y tanto el umbral máximo como mínimo será un 50% de los valores máximos y mínimos localizados. Esto permite adaptar los umbrales a cada usuario ya que la señal para cada uno de ellos se recogerá con niveles de amplitud diferentes en las ondas.

```
//Bloque de 1 segundo para calibrar los umbrales Máximos y
Mínimos

    if(cont<=128) {

        if(valorecg>calibratemax) {
            calibratemax = valorecg;
            thresholdMAX = valorecg*0.5;
        }
        if(valorecg<calibratemin) {
            calibratemin = valorecg;
            thresholdMIN = valorecg*0.3;
        }

        cont++;
    }
```

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

Seguidamente cada muestra se procesará siguiendo el algoritmo anteriormente definido y adaptado a la señal proveniente del sensor que se está utilizando. Cada muestra se compara de tal forma que solamente se detectará y se dará por válido un pico R cuando este sea el máximo absoluto localizado en el rango de valores positivos superiores al umbral máximo anteriormente calculado y que, seguidamente y en un periodo de tiempo limitado por la duración del complejo QRS, se encuentre un mínimo absoluto en el rango de valores inferiores al valor umbral mínimo, es decir, se localice el pico S.

Se limita la detección del pico R y el almacenamiento de su posición temporal exclusivamente a esta secuencia descrita, eliminando cualquier otra posibilidad que pueda aparecer debido a perturbaciones en la señal.

```
//-----Algoritmo en tiempo real de detección de picos R-----  
//  
if(this.valorecg>=peakMax && this.valorecg >= thresholdMAX) {  
//La primera condición es superar el umbral que hemos  
determinado y el valor de pico almacenado anteriormente para  
encontrar así el máximo pico.  
  
    peakMax = this.valorecg; //Si se cumple la condición  
almacenamos este nuevo valor como el máximo pico.  
    locMax = this.timeofexec; //También almacenamos su  
instante temporal en una variable  
    max=true; //Se ha encontrado un máximo  
}  
  
else if(this.valorecg<peakMax && this.valorecg < thresholdMAX) {  
  
    if(this.valorecg<=peakMin && this.valorecg<=thresholdMIN) {  
        //Cuando superemos el umbral para mínimos que hemos  
determinado y así encontrar de la misma manera el mínimo pico.  
        peakMin=this.valorecg; //Almacenamos el  
valor y el instante temporal en las siguientes variables  
        locMin=this.timeofexec;;  
        min=true; //Se ha encontrado un mínimo.  
    }  
  
    else if(this.valorecg>peakMin &&this.valorecg>thresholdMIN) {  
  
        if(max==true && min==true && (locMin-locMax)<RS_Time  
&& (locMin-locMax)>0) { //Si finalmente hemos detectado  
un máximo, seguido de un mínimo dentro de que ambos tengan  
una diferencia temporal entre 0 y 120 ms. Guardamos el  
valor del instante temporal del máximo valor, es decir del  
pico R.  
            Locs.add(locMax); //Añadimos el instante  
temporal a esta array.        }  
    }  
}
```

```
        dif=CalculateHRV();//Llamamos a la función que
calcula el HRV.
        max=false;
        min=false;
    }
    peakMin=0.0;
    peakMax=0.0; //Ponemos a 0 el valor del pico
máximo y mínimo para localizar los siguientes picos R y S.
}
}
```

### 5.1.3.2. Cálculo de la variabilidad del ritmo cardíaco (HRV)

Una vez se va obteniendo y almacenando la posición temporal de los picos R, se procede a seguir el siguiente paso dentro del estándar [13], que es el cálculo de los intervalos entre pico y pico, más conocido como HRV (*Heart Rate Variability*).

La variabilidad del ritmo cardíaco se define como la medición de las variaciones o diferencia temporales que ocurren en un conjunto de ciclos cardíacos. Su construcción consiste en obtener el tiempo transcurrido en todos los intervalos RR secuencialmente a lo largo de un registro ECG. Para cada intervalo de tiempo medido, se tiene un punto que permite general la señal de HRV [16].

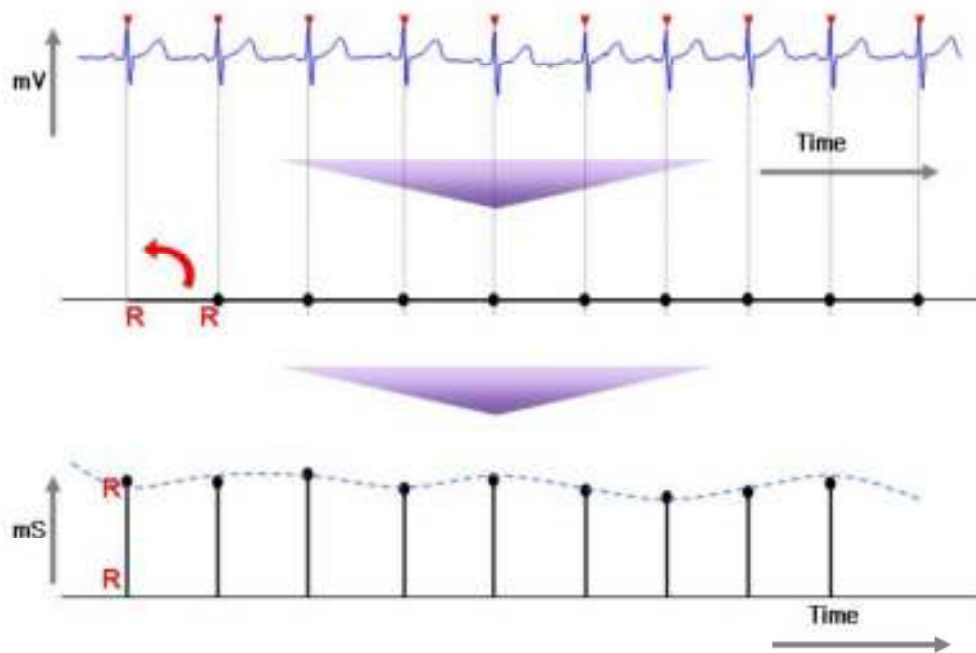


Figura 25. Cálculo y representación de la señal HRV

En este proyecto, a medida que se va almacenando la posición de los picos R, también se utiliza el método dentro de la clase *ECGDataProcessing* llamado *CalculateHRV()* y cuyas líneas de código se muestran a continuación.

```
public double CalculateHRV() {

    double diferencia=0.0;
    int length=0;

    length=Locs.size();//Obtenemos la longitud del vector donde
    están almacenadas las posiciones de los picos R.

    if(length>1) {
        diferencia = Locs.get(length-1)-Locs.get(length-2);
        //Realizamos las sucesivas diferencias entre las posiciones
        donde se encuentran los picos R.

        if((diferencia>=400)&&(diferencia<=1000))hrv.add(diferencia);
        //Almacenamos en el array hrv los valores de los intervalos
        RR.
    }
```

```
}  
  
    return diferencia;  
}
```

Esta función devuelve cada punto de la señal HRV calculada para ser representada en una gráfica dentro del *layout* correspondiente a este sensor.

Una vez generada la señal de HRV, es posible implementar una serie de métodos que permiten extraer diversas características de la señal. Estos se pueden dividir entre aquellos que realizan un análisis temporal (dominio del tiempo) y los que realizan un análisis frecuencial (dominio de la frecuencia).

- **Análisis temporal:**

A través del análisis temporal se puede hacer un estudio estadístico de las propiedades de la variabilidad del ritmo cardíaco. En este caso, medir variables tales como el promedio de los intervalos RR del registro de la señal, o la desviación estándar de estos mismos intervalos, así como el cálculo de la frecuencia cardíaca (*beats per minute*) [17].

En relación a la fatiga física y mental que presenta un individuo, el HRV puede proporcionar datos útiles para cuantificar la somnolencia fisiológicamente. Según otros estudios [20], una disminución de la desviación estándar de una serie de intervalos RR, que es equivalente a una disminución de la variabilidad del ritmo cardíaco, es un indicador de un aumento de la carga de trabajo mental. Dicha fluctuación puede ocurrir, por ejemplo, en individuos que realizan tareas monótonas y de alta demanda cognitiva por un prolongado período de tiempo. En este proyecto se puede dar ese caso en la tarea de conducción por vía interurbana o autovía durante varias horas.

Por lo tanto, en la aplicación Android se realizará el cálculo de los tres parámetros anteriormente citados cada 30 segundos, de manera que este valor se irá mostrando por pantalla y almacenando en fichero, como veremos más adelante. Estos son los métodos utilizados para su correspondiente cálculo:

```
//IntervaloRR, HR y HRV:  
  
TextViewintervalRR=(TextView) findViewById(R.id.IntervalRR_num);
```

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

```
TextView heartrate = (TextView) findViewById(R.id.HR_num);  
TextView hrv = (TextView) findViewById(R.id.HRV_num);  
  
boolean calculateornot=myECG.getCalculateornot(); //Llamada a la  
función para decidir si hacer el cálculo de parametros.  
    if(calculateornot==true) {  
  
intervalRR.setText(String.format(Locale.ENGLISH, "%.2f", myECG.get  
RRinterval())); //Llamada a la función que calcula el intervalo  
RR  
heartrate.setText(String.format(Locale.ENGLISH, "%.2f", myECG.getH  
eartrate())); //Llamada a la función que calcula el HR.  
hrv.setText(String.format(Locale.ENGLISH, "%.2f", myECG.getHRVvalu  
e()));} //Llamada a la función que calcula el HRV.
```

El método *getRRinterval()* se corresponde al cálculo promedio de los intervalos RR y el método *getHRVvalue()* es la desviación estándar entre estos intervalos. Para el cálculo del *Heart Rate* en el método *getHeartRate()* utilizamos la fórmula que nos viene dada en el manual de usuario de ECG y la convertimos a unidades de bpm (*beats per minute*).

$$HR (BPM) = 60 * (NR - 1) / ((S2 - S1) / Fs) \quad [7]$$

NR es el número total de ondas R almacenadas, S2 es la última onda R almacenada y S1 la primera. Fs es la frecuencia de muestreo.

- **Análisis frecuencial:**

El análisis frecuencial o espectral consiste en el estudio de la señal en el dominio de la frecuencia, cuya principal ventaja es poder observar fenómenos o propiedades de una señal que en el dominio del tiempo no son evidentes de estudiar, tales como descomponer una señal en sus armónicos, ver la energía confinada en distintos intervalos de frecuencia o analizar la distribución de esta a lo largo de todo un espectro.

Para comenzar este análisis y así seguir el siguiente paso del estándar internacional [13], es necesario utilizar el operador algebraico conocido como la “Transformada de Fourier” y, como su nombre lo señala, permite la



## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

transformación de una señal periódica desde el dominio del tiempo al dominio de la frecuencia.

En la aplicación que se ha desarrollado, se llevará a cabo la transformada rápida de Fourier (FFT, “*Fast Fourier Transform*”) que consiste en un algoritmo implementado para calcular la transformada de Fourier de señales discretas (en este caso, la señal que se quiere analizar es la señal HRV). Destaca principalmente por su eficacia y velocidad de cálculo. Se utiliza entonces un método dentro de la clase *FastFourierTransform.java* llamado *transform()* en el que, como parámetros, tenemos la parte real e imaginaria del vector de muestras discretas HRV.

```
double[] real = new double[hrv.size()]; //Vector definido
para la parte real de HRV.
double[] imag = new double[hrv.size()]; //Vector definido
para la parte imaginaria de HRV.

for(int j=0; j<hrv.size(); j++) { //Se les da valores a
ambos vectores.
    real[j] = hrv.get(j);
    imag[j] = 0;
}

FastFourierTransform.transform(real, imag); //Llamada a la
función que realiza la FFT
```

El siguiente paso nos lleva al cálculo de la densidad espectral de potencia o PSD. Es una herramienta estadística utilizada para conocer cómo se distribuye la potencia de una señal definida en el tiempo a lo largo de un intervalo de frecuencias. Se calcula como el módulo al cuadrado de la Transformada de Fourier, como se muestra en la ecuación [18]:

$$P(w) = |TF\{f(t)\}|^2 = \left| \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt \right|^2$$

A continuación, se muestran las líneas de código en las que se procede al cálculo de la PSD y la creación de un vector que actúa como eje de frecuencias cuya longitud es la misma que el vector que contiene la densidad espectral.

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

```
double[] psd = new double[real.length]; //Vector definido
para almacenar la PSD.

for(int j=0; j<real.length; j++) {
    //Algoritmo para el calculo de la PSD.
    psd[j] = Math.hypot(real[j], imag[j]); //Modulo de la
Transformada de Fourier
    psd[j] = psd[j]*psd[j];
    //Cuadrado del modulo calculado anterior.
}

double[] f = new double[psd.length]; // Creamos un vector
que representa el eje de frecuencias.
while(cont<=(0.5*(f.length-1))){ //Bucle para
crear el eje de frecuencias. Vector entre 0 y 0.5 con N puntos
(Siendo N la longitud del vector PSD)
    f[i]=cont/(f.length-1);
    cont=cont+0.5;
    i++;
}
}
```

Antes de extraer y realizar un estudio sobre los parámetros de interés dentro del análisis en frecuencia, que es el siguiente paso dentro del estándar, debemos explicar qué es el sistema nervioso autónomo (SNA), ya que es uno de los principales sistemas que regula la frecuencia a la que late el corazón.

El sistema nervioso autónomo transmite impulsos nerviosos desde el sistema nervioso central hasta la periferia, estimulando los aparatos y sistemas orgánicos periféricos. Sus vías neuronales actúan sobre la frecuencia cardíaca y respiratoria, además de la contracción y dilatación de vasos sanguíneos, digestión, salivación, etc. [19].

El SNA se divide funcionalmente en:

- **Sistema simpático:** Está implicado en actividades que requieren gasto de energía. También es llamado sistema adrenérgico, ya que es el que prepara al individuo para reaccionar ante una situación de estrés.
- **Sistema parasimpático:** Está encargado de almacenar y conservar la energía. Es llamado también sistema colinérgico, ya que es el que mantiene al individuo en situaciones normales que no implican peligro o estrés. Este sistema es opuesto al simpático.

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

Ambos sistemas están en constante equilibrio y juntos regulan el nivel de alerta o estrés del individuo.

Estudios demuestran [20] que al realizar un análisis espectral sobre la señal de HRV, existen unas variaciones de densidad espectral de potencia en las bandas de frecuencia:

- “High Frequency” (HF, comprendida entre 0,15 y 0,4 Hz)
- “Low Frequency” (LF, comprendida entre 0,04 y 0,15 Hz)
- “Very Low Frequency” (VLF, comprendida entre 0 y 0,4 Hz)

Estas variaciones son una consecuencia directa de las fluctuaciones entre la actividad neuronal, simpática y parasimpática. En concreto, en presencia de un aumento de fatiga mental, se puede evidenciar una bajada en la densidad espectral de potencia en la banda LF, la cual es generada por cambios en la actividad del sistema simpático y parasimpático. Para el mismo estado de fatiga, se puede ver un aumento en la banda HF, la cual está dada a partir de un cambio en la actividad autónoma parasimpática [20].

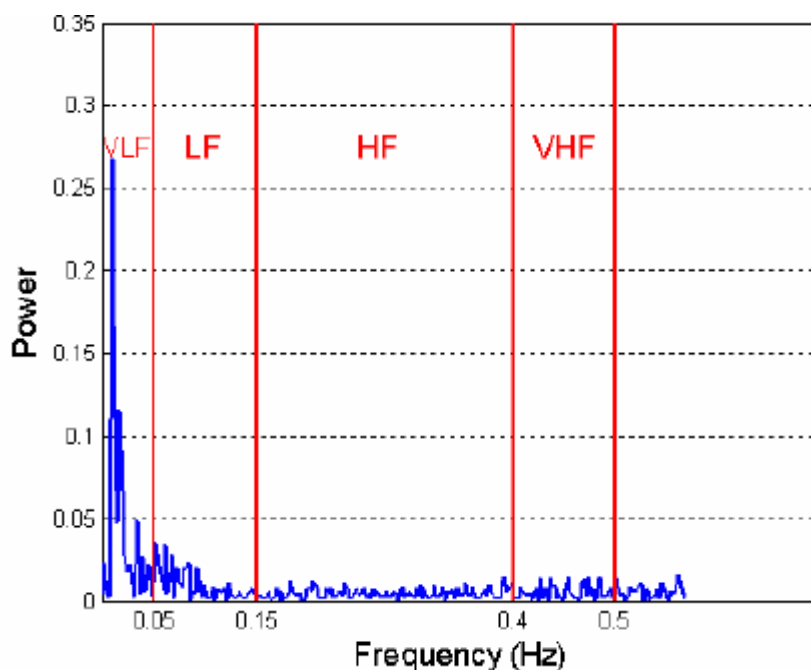


Figura 26. PSD y bandas de frecuencia de la señal HRV

Otro indicador de las consecuencias de la fatiga mental en el sistema nervioso autónomo en el balance simpático y parasimpático, que consiste en la tasa entre la banda HF y la banda LF.

$$\text{Ratio} \left( \frac{LF}{HF} \right) = \frac{\sum_0^{0.05} PSD(f)}{\sum_{0.05}^{0.4} PSD(f)}$$

Diversos estudios concluyen que individuos con un aumento del nivel de fatiga mental, registran una disminución de este valor. Aunque hay que tener en cuenta que, en periodos con un estado de estrés, actuará el sistema nervioso simpático, por lo que este valor debería verse aumentado [21].

El desarrollo para el cálculo de los valores de estas bandas de frecuencia en el vector generado de la PSD y el análisis frecuencial en general se realiza igualmente que el cálculo de parámetros en el dominio del tiempo, cada 30 segundos y se implementa de la siguiente forma:

```
//Buscamos las posiciones para delimitar las bandas de
frecuencias de la señal PSD.

i=0;
while(VLFmin==0) {
    if(f[i]>=0.001) VLFmin = i;           //Banda para VLF
    i++;
}

i=0;
while(LFmin==0) {
    if(f[i]>=0.04) LFmin = i;           //Banda para LF
    i++;
}

i=0;
while(HFmin==0) {
    if(f[i]>=0.15) HFmin = i;           //Banda para HF
    i++;
}

i=0;
while(HFmax==0) {
    if(f[i]>=0.4) HFmax = i;
    i++;
}
```

```
//Una vez obtenidos los valores de las bandas hacemos la
integral en cada banda de la PSD calculada anteriormente.
for(int k=VLFmin; k<=LFmin; k++) {
    VLF = VLF + psd[k];
}
for(int k=LFmin; k<=HFmin; k++) {
    LF = LF + psd[k];
}
for(int k=HFmin; k<=HFmax; k++) {
    HF = HF + psd[k];
}
}
```

Con todo esto, se ha concluido la implementación en tiempo real de la extracción, procesado y cálculo de parámetros de la señal ECG proveniente del sensor RN42-A8ED. En un apartado posterior, se mostrará el resultado visual en la interfaz creada para esta aplicación Android.

### 5.1.4. Procesado online de la señal EMG

El electromiograma es una prueba que consiste en el registro de la actividad eléctrica de los músculos mediante el uso de electrodos. Cuando un músculo se somete a un estímulo voluntario o involuntario como, por ejemplo, un movimiento realizado por el sujeto, se generan una serie de potenciales de acción que producen actividad eléctrica. Dicha actividad se puede registrar con unos electrodos colocados en diferentes localizaciones.

Esta señal EMG es una señal complicada y no estacionaria que es controlada por el sistema nervioso ya que es el responsable de la actividad muscular. La forma de esta va a depender de diferentes factores como el músculo de medida, el tipo de electrodo o la fuerza del sujeto. En general, la amplitud típica suele oscilar de 50  $\mu$ V a 1 mV con un ancho de banda de 10 Hz a 3000Hz [22].

La adquisición se puede realizar colocando los electrodos sobre la piel o insertándolos dentro de un músculo obteniendo así una medida invasiva. La medida superficial es el método más empleado y el que se utiliza en este

proyecto, ya que no requiere de ninguna incisión y no se corren riesgos. Estos dos métodos se denominan electrodos de superficie o de aguja:

- **Electrodos de superficie:** se pueden emplear electrodos activos caracterizados por incluir amplificadores integrados que reducen la impedancia de contacto o electrodos pasivos que requieren de geles de contacto para mejorar la adquisición de la señal [23].

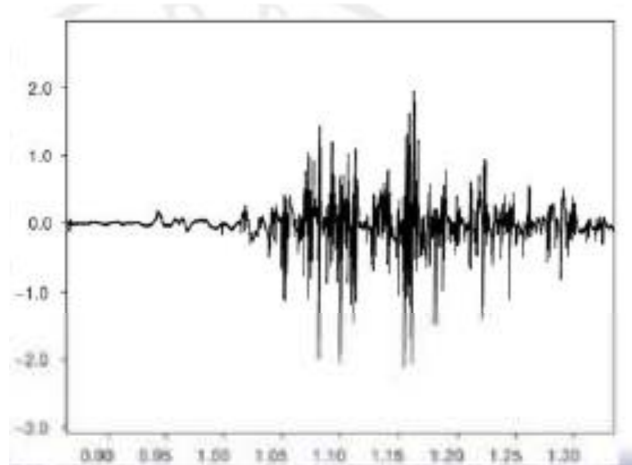


Figura 27. Señal EMG adquirida por medio de electrodos de contacto

- **Electrodos de aguja:** requiere su inserción en el músculo donde se va a realizar la medición. Constan de una aguja que puede estar aislada en toda su longitud menos en la punta o que puede llevar dos conductores metálicos en su interior aislados entre sí [23].

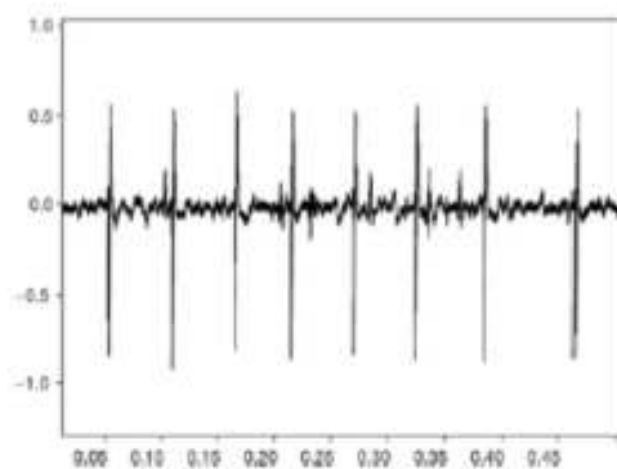


Figura 28. Señal EMG adquirida por medio de electrodos de aguja

El uso de cada tipo de electrodos va a depender del músculo y de la información que se desee obtener. También el acceso al músculo en ocasiones estará limitado por su localización. En general los electrodos de aguja o invasivos recogen información exclusivamente de la zona del músculo de medida mientras que los electrodos de superficie adquieren un mayor número de potenciales y se puede obtener información de músculos adyacentes.

Los músculos se pueden clasificar en tres categorías dependiendo de su participación en un movimiento:

- **Agonistas:** también referidos como motores primarios ya que son los principales responsables de generar un movimiento específico porque inician el movimiento y también generan la mayor parte de la fuerza.
- **Sinérgicos:** Estos ayudan a neutralizar el movimiento extra de los agonistas para asegurarse de que se utiliza la fuerza adecuada. Contribuyen al control del movimiento.
- **Antagonistas:** Actúan en oposición al movimiento específico generado por el agonista y son los responsables de regresar a la posición inicial.

Un movimiento ocurre cuando los músculos agonistas y sinérgicos se contraen y el antagonista se relaja.

En el caso del presente proyecto, se realizan medidas el músculo trapecio como ya se mencionó en el capítulo 3, por lo que se utiliza el método de adquisición de la señal por electrodos de contacto.

Al igual que para la señal de electrocardiograma en el procesado en tiempo real dentro de la aplicación Android, se incluyen los filtros convenientes para el procesado y adquisición correcta de la señal EMG.

El análisis de la señal EMG se realiza con el cálculo de determinados parámetros que permiten analizar la fuerza realizada durante los periodos de conducción. Se desarrolla un algoritmo, dentro una nueva clase llamada

*EMGDataprocessing.Java*, para almacenar y registrar cada muestra procedente del sensor RN42-B53C:

```
public void setValorEMG(Double emg){
    this.valoremg=emg;
    this.datosemg[this.muestra_n]=emg;
    this.totalsamples=this.totalsamples+1;
    //Actualizamos el numero de muestras totales sacadas

    this.muestra_n=(this.muestra_n==N-1) ? 0:this.muestra_n+1;
    //(Actualizamos el numero de muestra actual antes de salir
    -estructura circular- ->muestras de 0 a N-1)
}
```

El primer parámetro es la media de la señal registrada, que se obtiene a través del sumatorio de todos los valores. En este caso será el promedio del conjunto de 128 muestras almacenadas, que equivale a 1 minuto de adquisición de datos al utilizar la frecuencia de 128 Hz.

$$Media = \frac{\sum_i^n x_i}{N}$$

Con el análisis de dicho parámetro se estima que, ante una situación de estrés o fatiga mental, así como una maniobra brusca durante la conducción o una recuperación de la trayectoria tras un despiste, aumenta el movimiento corporal y con ello cabe esperar un aumento de la señal.

Por otra parte, también se realiza como un refuerzo al parámetro anterior un cálculo de la desviación típica. Esta es una medida del grado de dispersión de los datos con respecto al valor promedio o media calculada.

$$Desviación\ típica = \sqrt{\frac{\sum_i^n (x_i - Media)^2}{N}}$$

La implementación del cálculo de dichos parámetros en la aplicación se realiza por medio de la clase *MeanAndSD.Java*, que incluye los métodos *meanDouble()* y *standardDeviationDouble()*.



## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

Además, se realiza un almacenamiento de las sucesivas medias y desviaciones calculadas para que, también en tiempo real, se haga un cómputo global del valor de dichos parámetros. Esto se lleva a cabo añadiendo un nuevo método a la clase como es *meanDoubleArray()* que trabaja con un *ArrayList Double*, un vector cuyo tamaño es dinámico.

```
public double getMeanArrayEMG() {
    double media=0.0; //Método para
    calcular la media del array de giros
    media = calculate.meanDouble(this.datosemg);
    Mean.add(media);
    return media;
}

public double getMeanEMG() {
    return calculate.meanDoubleArray(Mean);
}

public double getSDArrayEMG() { //Método
    para calcular la desviacion estandard del array de giros
    double sd = 0.0;
    sd = calculate.standardDeviationDouble(this.datosemg);
    SD.add(sd);
    return sd;
}

public double getSDEMG() { //Método para
    calcular la desviacion estandard del array de giros
    return calculate.meanDoubleArray(SD);
}
```

### 5.1.5. Procesado online de la señal GSR

La señal GSR (*Galvanic Skin Response*) se define como el cambio en las propiedades eléctricas de la piel. Su medición es relativamente simple y se utiliza como herramienta para el estudio del comportamiento del SNA (sistema nervioso autónomo), así como del SNP (sistema nervioso periférico). Esta señal se puede describir en términos de resistencia, conductancia o potencial electrofisiológico, ya que proviene de las glándulas sudoríparas cuya secreción

es la causante de la variación en la resistencia y conductividad de la piel. Está formada por varias componentes que se pueden relacionar con diferentes estímulos [24].

- La primera de ellas es la componente tónica que tiene lugar cuando no existen acontecimientos o estímulos en el medio que rodea al sujeto. Por lo tanto, cada individuo tiene una conductancia en la piel determinada que varía a lo largo de los años y del estado emocional en que se encuentra.
- La conductancia fásica muestra la respuesta a un estímulo específico. Cambia la conductancia de la piel para, después de este evento, volver al estado de la componente tónica o mantenerse, dependiendo de la duración del estímulo.

Para calcular el estado emocional general es útil realizar el seguimiento de la amplitud media de esta señal.

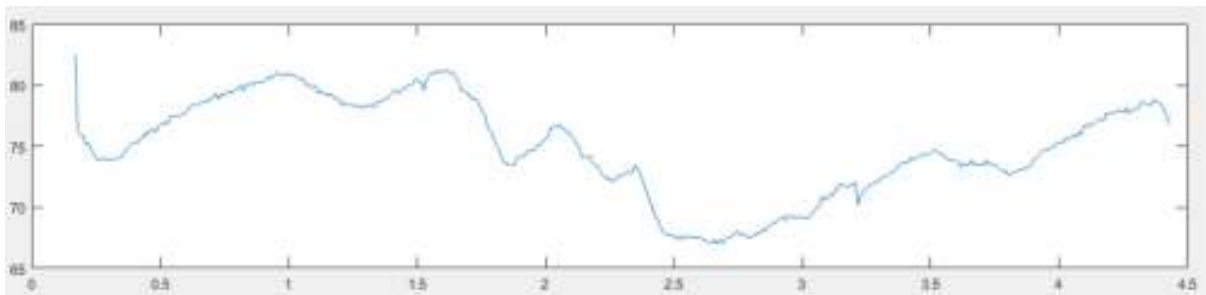


Figura 29. Forma de onda de la señal GSR obtenida del sensor Shimmer

El GSR refleja la activación del SNA, que se encarga del control de funciones viscerales tales como la frecuencia cardíaca, digestión, respiración, etc. Como se ha explicado anteriormente, el SNA está formado por el sistema nervioso simpático y parasimpático. El primero será el encargado de mediar en la respuesta de estrés mientras que el segundo trata de mantener un estado de descanso o relajación tras la realización de un esfuerzo. Para registrar la señal GSR, existen varios métodos:

- Medición electrofísica como el ECG o EMG: los electrodos empleados son los mismos que los utilizados para la adquisición de estas señales. En este caso, el electrodo activo se coloca en la

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

palma de la mano y el electrodo referencia podría situarse en la parte dorsal de la misma mano

- Variación en la resistencia o conductividad de la piel: medida de la variación de la resistencia o la conductividad mediante unos electrodos situados, por ejemplo, en los dedos.

Aunque una combinación de ambos métodos resulta muy eficiente, en este proyecto se utiliza la segunda técnica ya que resulta fácilmente reproducible y es más económica.

El análisis de la señal GSR se realiza de igual forma que el análisis de la señal EMG. En primer lugar, se desarrolla un algoritmo en la aplicación Android, dentro de una nueva clase llamada *GSRDataprocessing.Java* para almacenar y registrar cada muestra procedente del sensor RN42-938E.

```
public void setValorGSR(Double gsr) {
    this.valorgsr=gsr;
    this.datosgsr[this.muestra_n]=gsr;
    this.totalsamples=this.totalsamples+1;
    //Actualizamos el número de muestras totales sacadas

    this.muestra_n=(this.muestra_n==N-1) ? 0:this.muestra_n+1;
    //(Actualizamos el número de muestra actual antes de salir
    -estructura circular- ->muestras de 0 a N-1)
}
```

Seguidamente se procede al cálculo de la media y la desviación. En este caso, será el promedio del conjunto de 20 muestras almacenadas.

El análisis de este parámetro implica que a medida que un conductor se encuentra más activo y tiende a un estado de estrés, el nivel de resistencia o conductancia de la piel aumentará progresivamente. Sin embargo, cuando el sujeto que conduce se encuentra más cansado y tiende hacia un estado de relajación, el nivel disminuye. Esto nos va a permitir conocer el estado del conductor y si entra en un nivel de estrés o por el contrario un nivel de fatiga mental [24].

La implementación del cálculo de dichos parámetros en la aplicación se realiza por medio de los mismos métodos que para la señal de EMG, utilizando los

mismos algoritmos de cálculo de media y desviación típica, tal y como se ha explicado anteriormente.

```
public double getMeanArrayGSR() {
    double media=0.0;//Método para calcular la media del array
de giros
    media = calculate.meanDouble(this.datosgsr);
    Mean.add(media);
    return media;
}

public double getMeanGSR() {
    return calculate.meanDoubleArray(Mean);
}

public double getSDArrayGSR(){//Método para calcular la
desviacion estandard del array de giros
    double sd = 0.0;
    sd = calculate.standardDeviationDouble(this.datosgsr);
    SD.add(sd);
    return sd;
}

public double getSDGSR(){ //Método para calcular la
desviacion estandard del array de giros
    return calculate.meanDoubleArray(SD);
}
```

### 5.1.6. Interfaz gráfica y generación de ficheros del procesado en tiempo real de las señales ECG, EMG y GSR.

- **Interfaz gráfica**

Para poder reflejar todo el procesado y análisis de la señal en tiempo real que se ha implementado, es necesario crear una interfaz gráfica dentro de la aplicación que represente y actualice gráficamente las señales y muestre en tiempo real los correspondientes parámetros calculados.

Es necesario crear nuevos archivos *plot\_fragment.xml* donde definimos el *layout* para cada sensor y cómo va a estar estructurado. También se deben

definir los correspondientes *Strings* para mostrar el texto que se desee en dicho *layout*.

En todos hay algo en común y es que se representan las correspondientes gráficas del módulo acelerómetro y se representa la gráfica del módulo para cada señal del sensor del que se esté adquiriendo la señal en ese momento. El tiempo de ejecución desde que empieza el procesado (*Runtime*) y el número de muestras que se van tomando (*current N*) también es algo visible en todos los *layouts*.

Pero hacemos distinciones, comentadas a continuación, en función del sensor que haya conectado:

- RN42-A8ED (Señal ECG):

En este *layout* se muestran los tres principales parámetros del análisis temporal descrito anteriormente. Se muestra el valor para el intervalo entre picos R (Intervalo RR), la frecuencia cardiaca (*Heart Rate*) y el valor de la variabilidad de la frecuencia cardiaca (HRV). Además, se genera otra gráfica en la cual se representan los valores de esta variabilidad, para ello se ha creado un nuevo objeto de la clase *GraphView.Java* con el correspondiente ajuste de resolución de la señal.

Tras una separación se muestran los parámetros calculados del análisis frecuencial (VLF, LF, HF y Ratio), así como el porcentaje repartido entre la banda LF y la banda HF en tiempo real.

Esto va acompañado de un indicador de fatiga que se mantiene en color verde cuando el conductor se encuentra en un estado de alerta y pasa a color rojo cuando el estado de fatiga o somnolencia se hace evidente, es decir cuando hay un porcentaje de más del 50% en la banda de HF respecto a la banda LF.

A continuación, se muestra una figura con el *layout* correspondiente a dicho sensor en pleno funcionamiento:



Figura 30. *Layout* de la pestaña correspondiente al sensor ECG en la aplicación

- RN42-B53C (Señal EMG) y RN42-938E (Señal GSR):

En este *layout* se mostrarán los parámetros que se calculan en tiempo real, como se ha descrito anteriormente. Estos son la media aritmética *Last Interval Mean* y la desviación típica o estándar *Last Interval Deviation*. Se muestran dichos parámetros calculados usando las N muestras establecidas y también el cómputo global progresivo de dichos parámetros *Total Mean* y *Total Standard Deviation*.

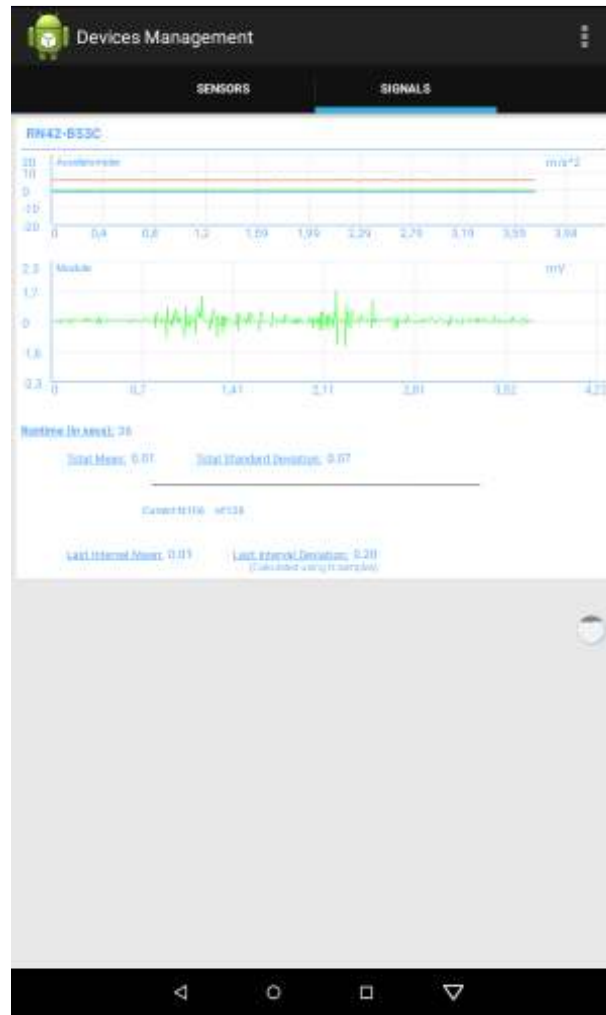


Figura 31. *Layout* de la pestaña correspondiente al sensor ECM en la aplicación



Figura 32. *Layout* de la pestaña correspondiente al sensor GSR en la aplicación

- **Generación de ficheros**

Al mismo tiempo que se hace la representación gráfica de señales y parámetros también se realiza la generación de ficheros donde se almacenan los datos extraídos y calculados.

Para cada señal del sensor se almacenan en un fichero principal los valores para el módulo acelerómetro y su propio modulo acompañados del instante de tiempo en que se recogen. A continuación se muestra un ejemplo de nombre que puede tener este fichero:



*RN42-A8ED\_23\_08\_2017.15-19.dat*

Para el sensor RN42-A8ED y, por lo tanto, la señal ECG, se generan dos ficheros más que contienen los parámetros calculados en el análisis temporal y el análisis frecuencial respectivamente y el instante de tiempo en que son almacenados:

*IntervalRR&HeartRate&HRV\_23\_08\_2017.15-19.dat*

*FrequencyAnalysisECG\_23\_08\_2017.15-19.dat*

Para los sensores con las señales EMG y GSR también se generan dos ficheros con las medias y desviaciones típicas por cada intervalo de muestras calculados y el cómputo de medias y desviaciones totales, todo acompañado siempre del instante de tiempo en que son almacenados. Estos dos ficheros son:

*RN42-B53C-TotalMean&SDEMG\_23\_08\_2017.15-19.dat*

*RN42-B53C-LastIntervalMean&SDEMG\_23\_08\_2017.15-19.dat*

De esta manera, podemos utilizar todos los datos y parámetros generados durante el procesado *online* de dichas señales para un posterior análisis de ellos o incluso un procesado *offline* de la señal.

## **5.2. PROCESADO OFFLINE**

### **5.2.1. Introducción y herramienta de desarrollo**

Tras la generación de los ficheros desde la aplicación Android en los que se almacenan todos los datos de las señales y sus correspondientes parámetros calculados en tiempo real, estos pueden ser utilizados para un procesado *online* de las señales utilizando la herramienta de software matemático Matlab R2015b, que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio.

Esta herramienta se caracteriza por la manipulación de matrices y vectores, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario y la comunicación con otros programas en otros lenguajes y con otros dispositivos hardware. En los últimos años, ha aumentado su número de prestaciones, entre las que se incluye programar directamente procesadores de señal, como es el caso de este proyecto.

En el presente proyecto se crean diferentes *scripts*, uno por cada sensor, para el procesado de la señal proveniente del fichero donde se encuentra almacenada.

Se comienza con la importación de los ficheros. Matlab incluye una herramienta encargada de la importación de archivos de datos, en la cual se generan automáticamente variables con los datos de cada una de las columnas del fichero y como nombre el que estas columnas tengan como cabecera. Por lo tanto, en el espacio de trabajo de este software ya estarán listos todos los datos recogidos en vectores o variables para trabajar con ellos, una vez que se realice la importación.

### **5.2.2. Procesado offline de la señal ECG**

Anteriormente ya se ha definido la señal ECG y las características de esta misma. En esa definición se encuentran las dificultades de adquisición de la señal debido a las perturbaciones a las que está sometida. En ese caso, se definen y aplican unos determinados filtros en tiempo real para la correcta extracción de la señal.

Los datos almacenados en fichero corresponden con la señal en bruto por lo que debemos realizar el procesado de la señal correcto para seguidamente procesarla de manera adecuada.

Tras definir la frecuencia de muestreo con la que la señal ha sido adquirida y hacer el cálculo del periodo, se definen los siguientes filtros siguiendo el estándar establecido [13].

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

```
%Filtrado de la señal

%FILTRO FIR rechaza banda para eliminar el ruido de los 50Hz de
la red
fN1=45/(Fs/2);
fN2=55/(Fs/2);
rechazaBanda = fir1(50,[fN1,fN2],'stop');
ECGSin50 = filter(rechazaBanda,1,ECGRALL);

%FILTRO FIR paso alto para eliminar la componente DC
fN1=1/(Fs/2);
pasaAlto = fir1(100,fN1,'high');
y = filter(pasaAlto,1,ECGSin50);
```

Se ha definido, al igual que en la aplicación Android, un filtro rechaza banda, en este caso de tipo FIR [25] y se aplica en la banda de 50Hz para rechazar el ruido proveniente de la red eléctrica.

También se aplica un filtro de tipo FIR paso alto con frecuencia de corte en 1 Hz, en este caso para eliminar las componentes DC. El orden de los filtros se ajusta para conseguir el mejor resultado en esta etapa de adaptación y procesamiento de la señal.

Los resultados conseguidos se muestran en la siguiente figura:

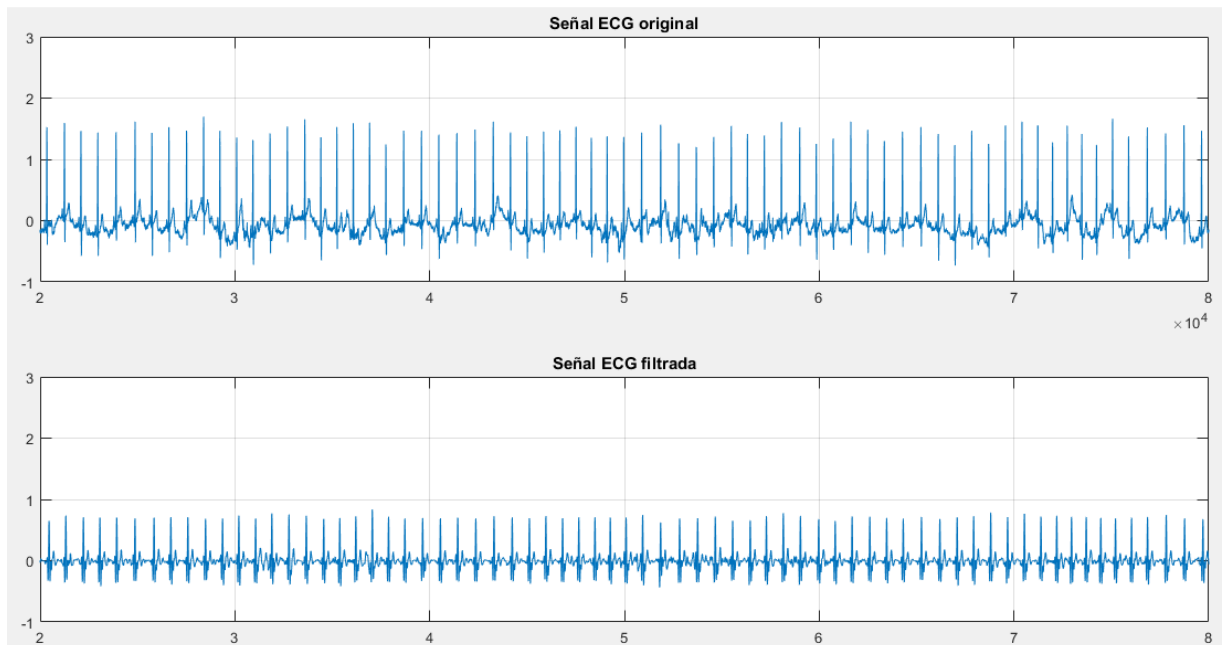


Figura 33. Comparativa de la señal ECG original con la señal tras pasar por los filtros FIR en Matlab

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

A continuación, se desarrolla el algoritmo de detección de picos u onda R. En este caso se utiliza la función definida por la herramienta Matlab como *findpeaks()* [26], que actúa localizando los máximos locales de la señal estableciendo un umbral que puede ser adaptado en función de cada señal una vez que esta haya pasado por los filtros.

```
%Obtencion de los picos RR
[peaks,locs] = findpeaks(y1,'MinPeakHeight',0.5);
%Funcion Matlab que localiza los picos de la señal por encima
del umbral que se quiera
%En el vector peaks se almacena el valor del pico y en el vector
locs se almacena su posición.
```

En la siguiente figura se puede ver la detección de picos R, que almacena en dos vectores el valor del pico y su localización temporal:

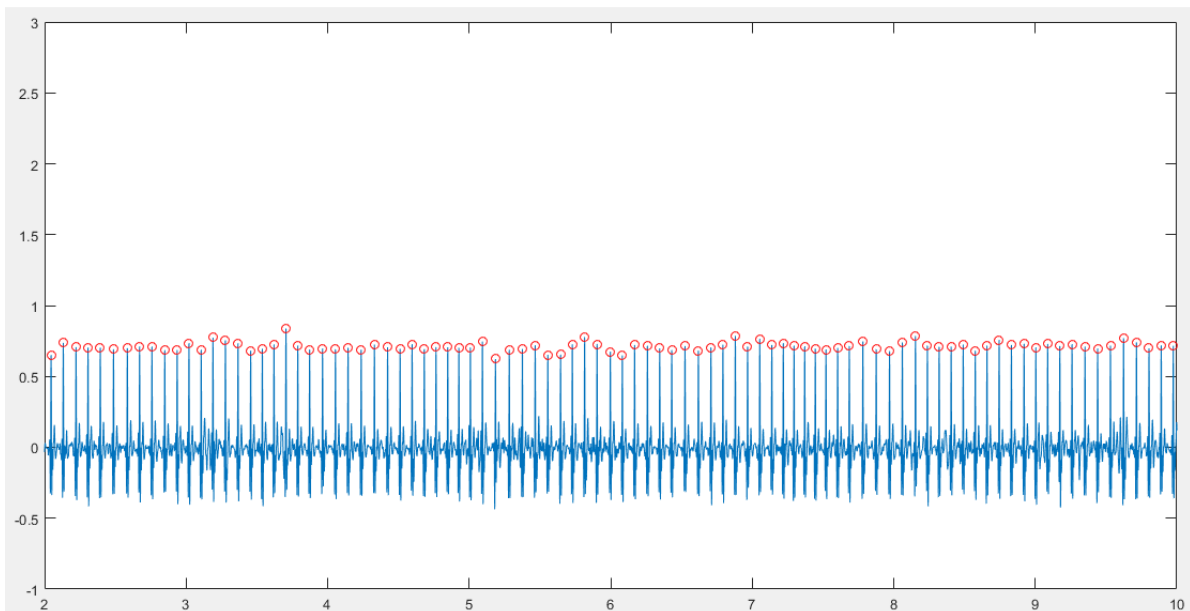


Figura 34. Detección de los picos R usando la función *findpeaks()*

Una vez tenemos la posición de picos R de la señal ECG almacenados, realizaremos el análisis temporal y frecuencial para la obtención de los parámetros de interés.

- Análisis temporal:

En este caso se va a realizar el análisis en dos partes, una en la que obtendremos el HRV (variabilidad de la frecuencia cardiaca), para así poder optimizar la detección de picos R y, por lo tanto, eliminar los falsos picos

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

detectados, y otra en la que, una vez se haya realizado la corrección, se realice el cálculo de parámetros.

Para ello implementamos el algoritmo en el cual obtenemos el vector de muestras de HRV calculando las sucesivas diferencias entre las localizaciones temporales de los picos R. A continuación se realiza el cálculo de la media y la desviación típica de esta variabilidad a través de las funciones *mean()* y *std()* que incorpora Matlab, para así utilizarlas en un algoritmo de detección y eliminación de falsos picos R como especifica el estándar.

```
%Cálculo de HRV
%En el vector hrv almacenamos la diferencia entre los distintos
picos de la
%señal.

for data = 1:1:(length(Timestamp(locs))-1)
    hrv(data)=Timestamp(locs(data+1))-Timestamp(locs(data));
end
```

En este algoritmo se eliminan los picos R cuya variabilidad entre ellos sea menor que 4 veces la desviación estándar [13].

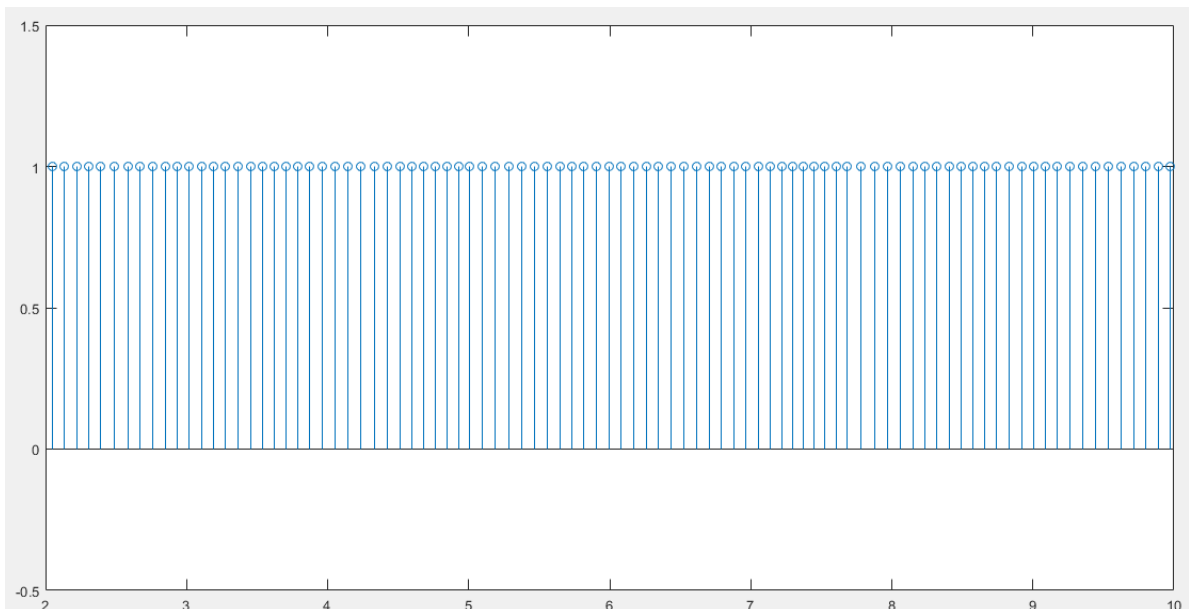


Figura 35. Obtención y normalización de los picos R de forma definitiva

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

Ahora ya se puede realizar de nuevo el cálculo correcto del vector HRV y su correspondiente representación utilizando de nuevo el mismo algoritmo descrito anteriormente:

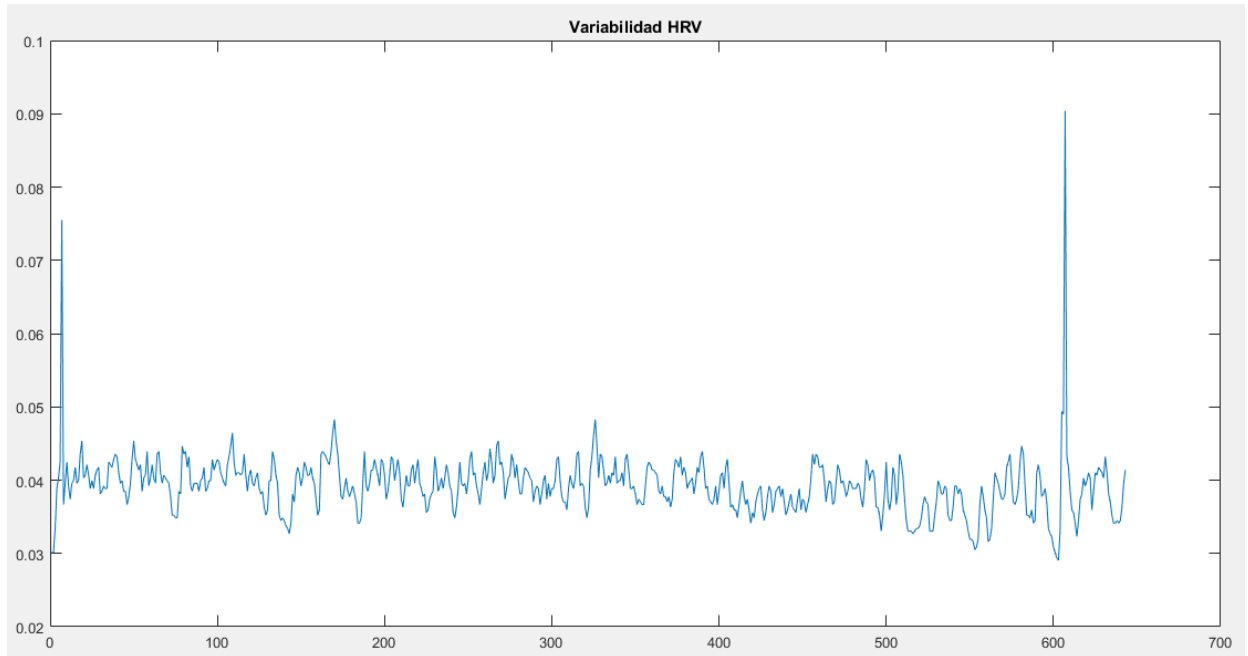


Figura 36. Representación de la señal HRV

Siguiendo los cálculos descritos en el análisis *online* y con la ayuda de las funciones que ofrece esta herramienta, se realiza el cálculo de los parámetros como son la media y desviación típica del HRV (Intervalo RR y HRV *value*) y también el cálculo de la frecuencia cardíaca (*Heart Rate*) siguiendo la fórmula que la define.

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

```
%HR
distanciaTotalPicos = Timestamp(locs1(length(locs1)))-
Timestamp(locs1(1)); %Distancia entre el primer y último pico
IntervaloRR = mean(hrv1); %Media de la variación de la
frecuencia cardiaca.
HeartRate = 60 / (IntervaloRR*0.001); %Formula para el calculo de
la frecuencia cardiaca. (HR = 60 / IntervaloRR(s))

fprintf('\nIntervalo RR = %0.2f mseg\n',IntervaloRR);
%Intervalo RR en milisegundos.
fprintf('Heart Rate = %0.2f beats/min\n',HeartRate);
%Frecuencia cardiaca.

%HRV (Representa la desviación del intervalo entre picos R)

desviacion = std(hrv1,1); %Desviación de la variación de la
frecuencia cardiaca.
fprintf('HRV = %0.2f ms\n\n',desviacion);
```

La obtención de parámetros se muestra en la línea de comandos de Matlab con las unidades correspondientes.

- Análisis frecuencial:

Este análisis se facilita con creces con la utilización de las funciones que proporciona este software. En este caso se realiza la transformada de Fourier del vector HRV que previamente ha sido normalizado utilizando la función *fft()* [27].

Para hallar la densidad espectral de potencia se implementan los cálculos de igual forma que en el análisis *online*, normalizándolo a continuación. Para la creación del eje de frecuencias se utiliza la función *linspace()*.

```
%Calculo de la FFT y densidad espectral de potencia
N = length(hrv1);
hrvFreq = fft(hrv1); %Transformada de Fourier
hrvFreq = hrvFreq(1:ceil(N/2));
PSD = hrvFreq.*conj(hrvFreq); %Calculo de la PSD
PSD = PSD/norm(PSD);
f=linspace(0,1/2,length(hrvFreq)); %eje de frecuencias
```

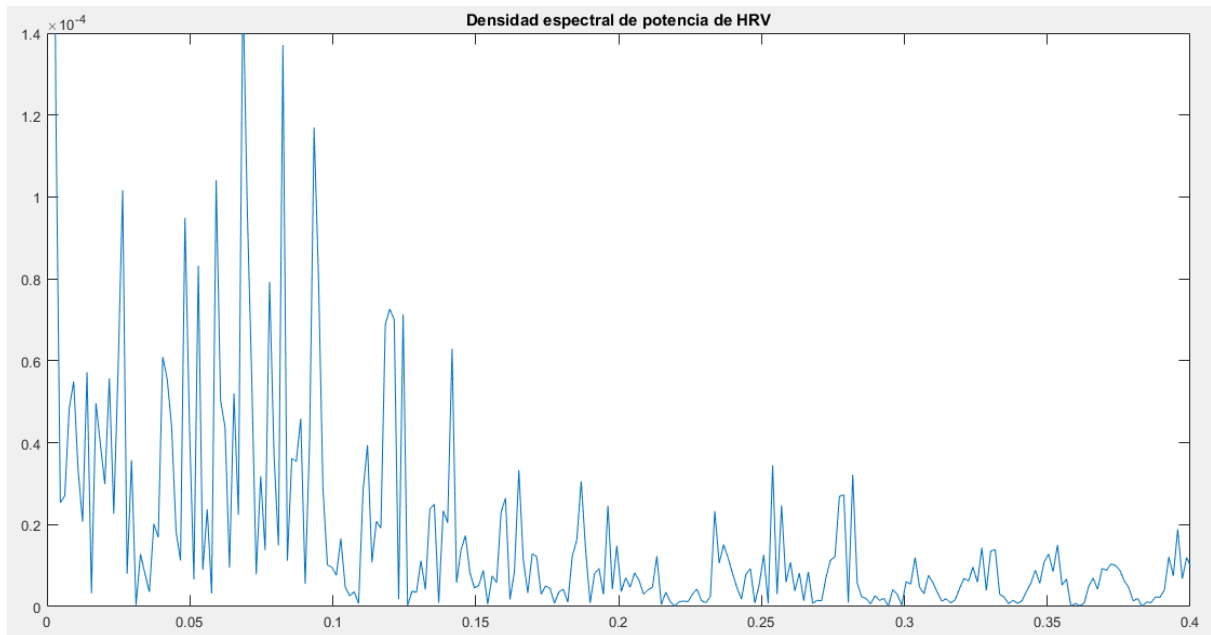


Figura 37. Representación de la densidad espectral de potencia (PSD)

Para la obtención de los parámetros relacionados con este análisis se realiza la integral discreta de las bandas de frecuencia de interés y se procede con la manipulación y el cálculo de los demás parámetros que se muestran en la línea de comandos del programa.

```

VLFmin=find(f>=0.0001,1);
VLFmaxLFmin=find(f>=0.04,1);
LFmaxHFmin=find(f>=0.15,1);
HFmax=find(f>=0.4,1);

VLF = sum(PSD(VLFmin:VLFmaxLFmin));
LF = sum(PSD(VLFmaxLFmin:LFmaxHFmin));
HF = sum(PSD(LFmaxHFmin:HFmax));
Ratio = LF/HF;

fprintf('LF(AU) = %0.2d\n',LF); %sympathetic nervous activity
fprintf('LF(NU) = %0.2f\n', (LF/(HF+LF))*100); %Contribution of
sympathetic nervous activity
fprintf('HF(AU) = %0.2d\n',HF); %parasympathetic nervous
activity
fprintf('HF(NU) = %0.2f\n', (HF/(HF+LF))*100); %Contribution of
parasympathetic nervous activity
fprintf('VLF(AU) = %0.2d\n',VLF); %Very Low Frequency
fprintf('RATIO = %0.2f\n',Ratio);

```



Este *script*, llamado *ECGprocessing.m*, contiene la función para importar los datos de fichero y todo el procesamiento de la señal, generando las gráficas y parámetros mencionados.

### 5.2.3. Procesado offline de las señales EMG y GSR

Se explica la implementación de los *scripts* para estas dos señales en un solo punto ya que ambos procesados y obtenciones de parámetros son similares, como se ha detallado anteriormente en el procesamiento en tiempo real en la aplicación Android.

Ambas señales son tratadas con los mismos filtros definidos en el script de la señal ECG y que son adaptados para cada una de estas dos señales.

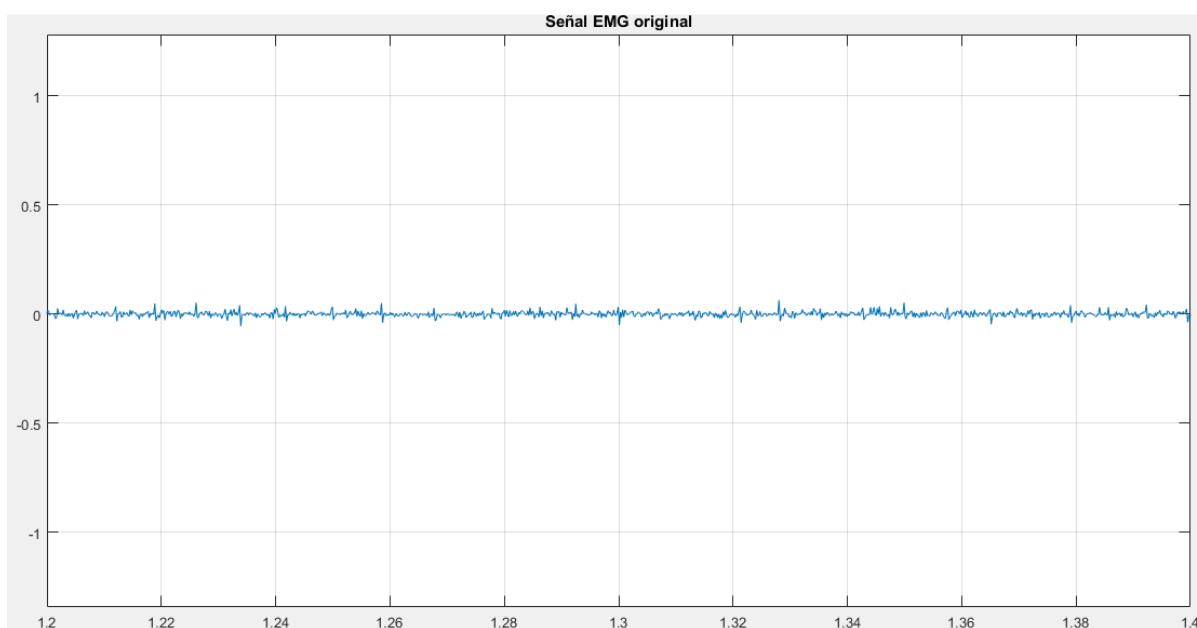


Figura 38. Representación de la señal EMG original

En este caso, se implementa el cálculo de la media y la desviación típica de ambas señales, lo que resulta fácilmente realizable con la ayuda de las funciones incorporadas en Matlab como son *mean()* y *std()*.

```
MediaEMG = mean(EMGSinDC); %Media de la señal EMG.  
fprintf('Media EMG = %0.2d\n',MediaEMG);
```

## 5 | IMPLEMENTACIÓN DEL PROCESADO DE LAS SEÑALES FISIOLÓGICAS

```
stdEMG = std(EMGSinDC); %Desviacion tipica de la señal EMG.  
fprintf('Desviacion tipica EMG = %0.2d\n',stdEMG);
```

Los parámetros calculados son mostrados en la línea de comandos de Matlab con sus correspondientes unidades.

## 6. BALANCE ECONÓMICO DEL PROYECTO

---

En este capítulo se va a hacer una aproximación económica al coste supuesto para este proyecto. Habrá que tener en cuenta no solo la inversión que hubiera sido necesaria en los medios hardware y software empleados, sino el elevado número de horas invertidas en la realización de este extenso proyecto.

En cuanto a los costes hardware, los sensores fisiológicos *Shimmer v2* empleados tienen un coste de 3.000 €, a lo que se puede sumar 600 € aproximados del resto de hardware empleado: coste de PCs, portátil, dispositivos Android como Tablet y smartphone, etc.

No se tiene coste por parte del software, por lo que se obtiene un total en medios de 3.600 €.

Para la realización del proyecto se estiman las horas invertidas en 500. Consideramos un período de trabajo aproximado de 7 meses. Si se tiene en cuenta el salario de ingeniero junior, que se puede estimar en 8 € la hora con los salarios actuales, se obtiene un gasto total de 4.000 €.

Si se considera al trabajador como autónomo, teniendo en cuenta que la cuota de la seguridad social en régimen de autónomos es de 264,44 € al mes, a lo largo de esos 7 meses ascendería el total a 1.851,08.

Por tanto, tendríamos un coste total de mano de obra de  $4.000 + 1.851,08 = 5.851,08$  €.

En resumen, el presupuesto final estimado para el proyecto sería el siguiente:

Coste total en medios Hardware y Software = 3.600 €

Coste total en mano de obra = 5.851,08 €

---

**Coste total estimado para el proyecto = 9.451,08 €**

## 7. CONCLUSIONES Y LÍNEAS FUTURAS

---

En este último capítulo se comentan todas las conclusiones obtenidas durante la realización del presente proyecto. Se tratan los aspectos relacionados con el estudio que se ha llevado a cabo para poder realizar el desarrollo de la aplicación. También se detallan los pasos necesarios para llevar a cabo la implementación de esta aplicación. Para terminar, se expondrán líneas de futuro a seguir para este proyecto, proponiendo mejoras para el mismo e inclusiones de nuevas funcionalidades con el objetivo de mejorar la aplicación.

### 7.1. Conclusiones

El objetivo que se planteó en este proyecto era el estudio y desarrollo de una aplicación compatible con el sistema operativo Android para poder monitorizar en tiempo real los datos proporcionados mediante los sensores fisiológicos de la plataforma *Shimmer* y así obtener un análisis que detectase estados de fatiga mental o estrés en el conductor. Partiendo de una aplicación base, se ha desarrollado la adquisición y el procesado de datos en tiempo real para los sensores ECG (Electrocardiograma), EMG (Electromiograma) y GSR (Respuesta galvánica de la piel). También se ha creado una interfaz para mostrar todos los datos calculados referidos a las distintas señales y su correspondiente almacenamiento en ficheros.

El almacenamiento de los datos extraídos en ficheros ha permitido realizar un análisis *offline* de los datos, para así tener otro método de procesado de las señales en caso de que fuera necesario.

En primer lugar, ha sido necesario realizar un acercamiento y estudio del funcionamiento de los sensores proporcionados de la gama *Shimmer*. Con ayuda de los manuales de usuario se aprendió a sincronizar dichos dispositivos mediante *Bluetooth* con el PC y con los dispositivos portables utilizados. Se realizó un estudio detallado del método y formato de adquisición de las señales en las que se ha enfocado el proyecto, la señal ECG, EMG y GSR, analizando

el rango de valores que capturan los sensores para trabajar adecuadamente con ellas.

Para el desarrollo de este proyecto ha sido imprescindible hacer un estudio sobre los conceptos tanto básicos como avanzados de Android, aprendiendo el lenguaje de programación *Java* y su entorno de programación, como ha sido *Eclipse Oxygen*. De este modo, se ha obtenido una base técnica suficiente para empezar a desarrollar la aplicación.

Comenzando con el desarrollo de la aplicación, ya se ha mencionado que se partía de una aplicación base, la cual realizaba la función de comunicación y recogida de datos básicos de los sensores. Por lo tanto, fue necesario hacer un estudio sobre la estructura y las principales funciones de esta aplicación base, para adaptar el desarrollo realizado a lo que ya había implementado. También ha sido necesario profundizar en el funcionamiento de la librería correspondiente al fabricante *Shimmer*, empleada ya en esta aplicación base.

Para desarrollar la adquisición y procesado de las señales fisiológicas, fue necesario un correspondiente estudio de estas mismas, caracterizando sus formas de onda y principales características.

Para la señal ECG fue necesario encontrar librerías que tuvieran implementados filtros para el procesado y limpieza de esta señal eliminando los ruidos que se podían capturar. Seguidamente, siguiendo los artículos de referencia, se implementó en el código de la aplicación una clase encargada de procesar y obtener la señal HRV (Variabilidad de la frecuencia cardiaca). Se implementaron diferentes algoritmos, como por ejemplo uno para la detección de la onda R de la señal ECG u otro para el cálculo y almacenamiento de la señal HRV. A continuación, se realizó un análisis temporal y frecuencial de dicha señal, lo cual se hace mediante diferentes métodos encargados de la obtención de los parámetros correspondientes que cumpliesen con el objetivo de este proyecto.

Para las señales EMG y ECG se creó una clase encargada de la obtención de la media y desviación de ambas señales. Todo esto es mostrado a través de una interfaz gráfica desarrollada para cada sensor y los parámetros calculados se almacenan en distintos ficheros para su posterior análisis *offline*.

## 7 | CONCLUSIONES Y LÍNEAS FUTURAS

El procesado *offline* se desarrolló utilizando la herramienta matemática *Matlab*, con la cual también se tuvo que hacer un estudio previo de su funcionamiento para la importación de los ficheros generados con la aplicación y la correspondiente implementación del procesado de las señales fisiológicas a partir de estos.

Durante el desarrollo de esta memoria, en conclusión, se han redactado los objetivos principales de este proyecto y las fases y métodos seguidos para su desarrollo. Se pone en contexto el objetivo hablando de los diversos factores que afectan a la seguridad en la conducción y al estado físico del conductor, explicando la importancia de este tipo de proyecto en la mejora de estos factores. Seguidamente se incluye todo lo relacionado con los sensores con los que se ha trabajado, así como un estudio del lenguaje y la plataforma que utiliza la aplicación a desarrollar. Por último, se comenta toda la implementación realizada con las distintas herramientas y el proceso seguido.

Para finalizar, y a título personal, la realización de este proyecto me ha supuesto conocer las diferentes fases que constituyen el desarrollo de un proyecto real de software. También, me ha permitido adquirir nuevos conocimientos como la programación en *Java* y el desarrollo de aplicaciones para el sistema operativo *Android*. También he aprendido el procedimiento de adquisición y procesado de señales y cómo trabajar con ellas para lograr demostrar los objetivos planteados. El proyecto me ha resultado entretenido, teniendo variedad de tareas y pudiendo trabajar con dispositivos reales, como son los sensores para poner en práctica lo que se ha desarrollado.

## 7.2. Líneas futuras

Al tratarse de un proyecto de investigación, las líneas de futuro que se pueden llevar a cabo pueden ser infinitas. Aunque, en este caso, se intentan concentrar en las posibilidades a seguir en lo que a la aplicación Android se refiere.

Una de las posibles líneas a desarrollar en un futuro es el estudio más detallado de las señales EMG y GSR, para la obtención de un mayor número de parámetros que sean más indicativos del estado físico y mental del conductor.

Para la señal ECG se podría optimizar el proceso de filtrado de esta, ya que no se permite aplicar filtros demasiado agresivos debido al rendimiento de la aplicación. También, si cabe, realizar distintos algoritmos para la detección de la onda R y para el cálculo del HRV. Existen diferentes métodos para ello y podrían implementarse de tal forma que la aplicación permita escoger el método de detección que deseemos.

Se puede implementar también un algoritmo que detecte y trabaje con la frecuencia respiratoria para así obtener más información sobre el estado del conductor.

Por último, se podría proponer realizar un estudio en el que se obtuviese una métrica para el estrés y la fatiga mental. En esa métrica se desglosarían todos los parámetros provenientes de los sensores y se calcularía de qué manera cada uno de ellos contribuye a dicha métrica, es decir, cuánto porcentaje del total de cada parámetro contribuye a alcanzar un estado de fatiga o estrés mental. Esta métrica y porcentajes, teniendo en cuenta todos los sensores, podrían verse reflejados en tiempo real en la aplicación.

# BIBLIOGRAFÍA

---

[1] Amine Khadmaoui. Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad de Valladolid. (2014). *Desarrollo de una aplicación para el control de sensores fisiológicos mediante un dispositivo Android*.

[2] Nota de prensa: Balance de Siniestralidad y Seguridad Vial 2016 – DGT (Dirección General de Tráfico). Ministerio del Interior.

[3] Anuario Estadístico de Accidentes 2015 – DGT (Dirección General de Tráfico). Ministerio del Interior.

[4] Sistema detector de fatiga en la conducción - J. L. García, E. Rogado, R. Barea, L. M. Bergasa, E. López, M. Ocaña and D. Schleicher

[5] Otros factores de riesgo: La fatiga – Perfecto Sánchez Pérez Catálogo general de publicaciones oficiales – Subdirección general de intervención y políticas viales. Unidad de intervención educativa (DGT). Ministerio del Interior.

[6] Otros factores de riesgo: El estrés – Perfecto Sánchez Pérez Catálogo general de publicaciones oficiales – Subdirección general de intervención y políticas viales. Unidad de intervención educativa (DGT). Ministerio del Interior.

[7] ECG User Guide (Revision 1.1.1) – Shimmer.

[8] EMG User Guide (Revision 1.1.0) – Shimmer.

[9] GSR User Guide (Revision 1.12) – Shimmer.



- [10] Estudio de demanda y uso de Servicios de Telecomunicaciones y Sociedad de la Información. ONTSI (Observatorio nacional de las telecomunicaciones y de la SI).  
<https://www.ontsi.red.es/ontsi/sites/ontsi/files/LIV%20Oleada%20del%20Panel%20Hogares.pdf>
- [11] Analog Filtering Schemes Analysis for ECG Signals – Ricardo Henao, Leydy Laura Álvarez, Edison Duque – Universidad Tecnológica de Pereira.
- [12] ELECTROCARDIOGRAFÍA: Técnica de interpretación básica- Manuel Portillo Márquez.
- [13] Changes in Physiological Parameters Induced by Indoor Simulated Driving: Effect of Lower Body Exercise at Mid-Term Break (Appendix A) - Wen Chieh Liang, John Yuan, Deh Chuan Sun and Ming Han Lin.
- [14] Efficient R-peak detection algorithm for real-time analysis of ECG in portable Devices (April 2016) – Claudio Crema, Alessandra Flammini, Alessandro Depari, Angelo Vezzoll. Università degli Studi di Brescia.
- [15] Simple real-time QRS detector with the MaMeMi filter (2015) – David Castells-Rufas, Jordi Carrabina. Universitat Autònoma de Barcelona.
- [16] Heart Rate Variability Analysis System. Clinical Information (Version 3.0). MEDICORE.
- [17] Extracción y análisis de la señal de variabilidad del ritmo cardíaco (HRV) mediante filtrado (2003) – Yusely Ruiz González, Juan Lorenzo Ginori, Alberto Taboada Crispí.

[18] Capítulo 2: Análisis espectral de señales (2011) – Sistemas de Comunicaciones – Mario Alfredo Ibarra Castillo.

[19] Sistema Nervioso Autonomo (Febrero, 2001) – Prof. Dr. Miguel Ramos. Revista de Postgrado de la Cátedra VIa medicina. (En línea) [http://med.unne.edu.ar/revista/revista101/sist\\_nervioso\\_autonomo.htm](http://med.unne.edu.ar/revista/revista101/sist_nervioso_autonomo.htm)

[20] Detecting Stress During Real-World Driving Tasks (2005) - Healey Rosalind W. Picard.

[21] Power spectrum analysis of heart rate variability: a tool to explore neural regulatory mechanisms – Editorial. British Heart Journal (1994).

[22] An Explanatory study of the parameters to be measured from EMG signal – Ruchika, Shalini Dhingra.

[23] Electromyography Fundamentals – Gregory S.Rash, EdD.

[24] A stress sensor base don Galvanic Skin Response (GSR) controlled by ZigBee (2012) – María Viqueira Villarejo, Begoña García Zapirain and Amaia Méndez Zorrilla.

[25] FIR filter Desing (Online). <https://es.mathworks.com/help/signal/ug/fir-filter-design.html>

[26] Find local máxima(Online) [https://es.mathworks.com/help/signal/ref/findpeaks.html?searchHighlight=findpeaks&s\\_tid=doc\\_srchtile](https://es.mathworks.com/help/signal/ref/findpeaks.html?searchHighlight=findpeaks&s_tid=doc_srchtile)

[27] Fast Fourier Transform (Online) [https://es.mathworks.com/help/matlab/ref/fft.html?searchHighlight=fft&s\\_tid=doc\\_srchtile](https://es.mathworks.com/help/matlab/ref/fft.html?searchHighlight=fft&s_tid=doc_srchtile)