



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA BIOMÉDICA

PROYECTO DE IMPLEMENTACIÓN DE UN SISTEMA DE BAJO COSTE Y CÓDIGO ABIERTO PARA ESTIMACIÓN DEL NIVEL DE CONSCIENCIA

AUTORA: ANA DE LA TORRE GARCÍA

TUTOR: JOSÉ JOAQUÍN RIETA IBÁÑEZ

COTUTORA: GEMMA MAS SESÉ

Curso Académico: 2020-21

Agradecimientos

A mis padres y mis abuelos, por creer en mí y apoyarme en todo momento

A mis amigas y compañeras de carrera, por acompañarme durante estos cuatro años y hacer de ellos
una experiencia inolvidable

Por último, a mi tutor, por confiar en mí y ofrecerme la oportunidad de participar en este proyecto

Resumen

La práctica de la anestesia se origina, históricamente, en la necesidad de resolver problemas tales como el dolor y la consciencia, ambos relacionados con el sistema nervioso, durante la práctica clínica. Cabe destacar la importancia de una correcta monitorización de la profundidad de la anestesia, con el fin de disminuir la incidencia de complicaciones asociadas a ésta. A pesar de que la función cerebral está directamente relacionada con la reactividad del paciente, la señal de electroencefalograma (EEG) es de difícil interpretación, por lo que, generalmente, se opta por la monitorización de otras variables, como la frecuencia cardíaca, la saturación de oxígeno, la presión arterial, etc. No obstante, la medida de la profundidad anestésica puede verse distorsionada ante cambios fisiológicos imperceptibles para el anestesiólogo.

En general, el EEG de pacientes anestesiados presenta cambios en amplitud y frecuencia, motivando así la búsqueda de parámetros representativos del nivel de profundidad anestésica en que se encuentra el paciente. Así, fue desarrollado empíricamente el índice biespectral (BIS), que ofrece una gradación de 0 a 100 sobre el nivel de consciencia del paciente. Sin embargo, a pesar de la existencia de equipos comerciales que calculan el BIS a partir de la señal de EEG, éstos presentan un elevado coste. Además, no ha sido revelado el algoritmo exacto con que se calcula el parámetro.

Por tanto, este TFG tiene por objetivo implementar un sistema de captación y procesamiento de EEG para valorar el nivel de consciencia de los pacientes, de bajo coste y código abierto. De este modo, se pretende determinar qué parámetros son de interés, para los profesionales sanitarios, en el estudio de la actividad cerebral, y diseñar un software de análisis que los calcule y presente en una interfaz de usuario de fácil manejo. Asimismo, mediante una clasificación basada en los rangos del BIS que representan el nivel de consciencia, se realizará una estimación de éste y se compararán los resultados con los proporcionados por un equipo comercial.

Palabras clave: nivel de consciencia, profundidad anestésica, índice biespectral, electroencefalograma

Resum

La pràctica de l'anestèsia s'origina, històricament, en la necessitat de resoldre problemes com ara el dolor i la consciència, tots dos relacionats amb el sistema nerviós, durant la pràctica clínica. Cal destacar la importància d'un correcte monitoratge de la profunditat de l'anestèsia, amb la finalitat de disminuir la incidència de complicacions associades a aquesta. A pesar que la funció cerebral està directament relacionada amb la reactivitat del pacient, el senyal d'electroencefalograma (EEG) és de difícil interpretació, per la qual cosa, generalment, s'opta pel monitoratge d'altres variables, com la freqüència cardíaca, la saturació d'oxigen, la pressió arterial, etc. No obstant això, la mesura de la profunditat anestèsica pot veure's distorsionada davant canvis fisiològics imperceptibles per a l'anestesiòleg.

En general, el EEG de pacients anestesiats presenta canvis en amplitud i freqüència, motivant així la cerca de paràmetres representatius del nivell de profunditat anestèsica en què es troba el pacient. Així, va ser desenvolupat empíricament l'índex biespectral (BIS), que ofereix una gradació de 0 a 100 sobre el nivell de consciència del pacient. No obstant això, malgrat l'existència d'equips comercials que calculen el BIS a partir del senyal de EEG, aquests presenten un elevat cost. A més, no ha estat revelat l'algorisme exacte amb què es calcula el paràmetre.

Per tant, aquest TFG té per objectiu implementar un sistema de captació i processament de EEG per a valorar el nivell de consciència dels pacients, de baix cost i codi obert. D'aquesta manera, es pretén determinar quins paràmetres són d'interès, per als professionals sanitaris, en l'estudi de l'activitat cerebral, i dissenyar un programari d'anàlisi que els calcule i presente en una interfície d'usuari de fàcil maneig. Així mateix, mitjançant una classificació basada en els rangs del BIS que representen el nivell de consciència, es realitzarà una estimació d'aquest i es compararan els resultats amb els proporcionats per un equip comercial.

Paraules clau: nivell de consciència, profunditat anestèsica, índex biespectral, electroencefalograma

Abstract

The anesthesia practice originates, historically, in the need to solve problems such as pain and consciousness, both related with the nervous system, during the clinical practice. It is noteworthy to underline the importance of a correct monitorization of the anesthesia depth, in order to diminish the incidence of related complications. Despite the fact that cerebral function is directly associated with the patient's reactivity, the electroencephalogram (EEG) signal is difficult to interpret so, generally, it is chosen to monitor other variables, such as cardiac frequency, oxygen saturation, blood pressure, etc. Nevertheless, the anesthesia depth measure can be distorted by imperceptible for the anesthesiologist physiological changes.

Generally, anesthetized patients' EEG presents amplitude and frequency changes, motivating the search of parameters that are representative of the patients' anesthesia depth level. Thus, the bispectral index (BIS) was empirically developed, offering a scale from 0 to 100 of the patients' level of consciousness. However, despite the existence of commercial devices able to calculate the BIS from the EEG signal, they present a significant cost. Moreover, the exact algorithm to obtain this parameter has not been revealed.

Therefore, this final degree project aims to implement a low cost and open-source EEG capture and processing system to evaluate the patients' level of consciousness. Thus, it is intended to determine which parameters are interesting to the healthcare professionals, in the study of cerebral activity, and design an analysis software to calculate and present them in a user-friendly interface. Additionally, with a classification based on the BIS ranges representing the level of consciousness, an estimation of this parameter will be carried out and the results will be compared to those obtained by a commercial device.

Key words: level of consciousness, anesthesia depth, bispectral index, electroencephalogram

Índice general

| | |
|---|-----------|
| I. MEMORIA..... | 1 |
| 1. Motivación y objetivos..... | 3 |
| 2. Principios de electroencefalografía | 5 |
| 2.1. Potencial de acción neuronal | 5 |
| 2.2. Propagación del potencial de acción | 7 |
| 2.3. Electroencefalograma..... | 8 |
| 2.4. Registro de potenciales en la superficie corporal..... | 10 |
| 3. Conceptos básicos de la anestesia | 13 |
| 3.1. Bases de la anestesia | 13 |
| 3.2. Control de la profundidad de la anestesia..... | 15 |
| 3.3. Índice Biespectral | 15 |
| 4. Diseño del sistema de captura..... | 19 |
| 4.1. Configuración del sistema hardware | 19 |
| 4.1.1. EEG Click | 19 |
| 4.1.2. AD8232 | 21 |
| 4.1.3. Electroodos | 23 |
| 4.1.4. Arduino Uno..... | 23 |
| 4.1.5. Montaje | 25 |
| 4.2. Desarrollo del software de control y adquisición | 27 |
| 4.2.1. Preprocesado..... | 27 |
| 4.2.2. Cálculo de subparámetros | 28 |
| 4.2.3. Estimación del índice biespectral..... | 29 |
| 5. Diseño de la interfaz de usuario | 32 |
| 5.1. Descripción | 32 |
| 5.2. Comprobación del funcionamiento..... | 38 |
| 5.3. Comparación con un dispositivo comercial..... | 41 |
| 5.4. Limitaciones..... | 44 |
| 6. Conclusiones | 46 |
| 7. Líneas futuras..... | 47 |
| Bibliografía | 48 |
| II. MANUAL DE USUARIO | 51 |
| 1. Nuevo registro | 53 |
| 2. Revisión de un registro previo | 55 |
| 3. Importación de archivo externo | 55 |

| | | |
|-------------|--|-----------|
| III. | ANEJOS..... | 57 |
| IV. | Presupuesto..... | 77 |
| 1. | Cuadro de precios Mano de Obra | 79 |
| 2. | Cuadro de precios Maquinaria | 80 |
| 3. | Cuadro de precios unitarios..... | 81 |
| 4. | Cuadro de precios descompuestos | 82 |
| 5. | Cuadro de presupuestos parciales | 85 |
| 6. | Presupuesto de ejecución material..... | 86 |

Índice de figuras

| | |
|--|----|
| Figura 1. Esquema de la morfología de la neurona..... | 5 |
| Figura 2. Esquema del funcionamiento de las sinapsis químicas..... | 6 |
| Figura 3. Potencial de acción neuronal. La línea discontinua representa el potencial umbral. | 6 |
| Figura 4. Micrografías electrónicas de células de Schwann y oligodendrocitos envolviendo axones mielínicos. (a) Imagen de microscopio electrónico en falso color y (c) esquema de un axón neuronal mielínico con un oligodendrocito formando vainas de mielina para múltiples segmentos axónicos. (b) Imagen de microscopio electrónico en falso color y (d) esquema de un axón neuronal mielínico con una célula de Schwann formando una vaina de mielina para un único segmento axónico..... | 8 |
| Figura 5. Comparación de la longitud del axón en la que ocurre la despolarización en axones amielínicos (arriba) y mielínicos (bajo)..... | 8 |
| Figura 6. Ejemplo de neurona piramidal del córtex prefrontal; la posición relativa de la célula se muestra mediante líneas discontinuas indicando las capas corticales (I – V). (a) Fotomicrografía de una neurona piramidal etiquetada intracelularmente en la capa III de la subárea prelímbica (hemisferio izquierdo). (b) Reconstrucción de la neurona representada en (a)..... | 9 |
| Figura 7. Sistema 10-20 de posicionamiento de electrodos para EEG..... | 11 |
| Figura 8. Esquemas de sistemas de registro en modo unipolar con cinco canales..... | 11 |
| Figura 9. Esquemas de sistemas de registro en modo promediado con cinco canales..... | 12 |
| Figura 10. Esquemas de sistemas de registro en modo bipolar con cuatro canales..... | 12 |
| Figura 11. Diagrama de bloques del procedimiento seguido para la obtención del BIS..... | 16 |
| Figura 12. Diagrama de bloques del procedimiento a seguir para el cálculo del BIS..... | 17 |
| Figura 13. Esquema del montaje interno de la placa EEG Click..... | 19 |
| Figura 14. Placa EEG Click utilizada para captura de la señal de EEG..... | 20 |
| Figura 15. Esquema del montaje interno de la placa AD8232..... | 21 |
| Figura 16. Placa AD8232 utilizada para captura de ECG..... | 22 |

| | |
|---|----|
| Figura 17. Ejemplos de disposición de los electrodos para medida de ECG..... | 22 |
| Figura 18. Electrodo de gel empleados para las medidas de EEG y ECG..... | 23 |
| Figura 19. Esquema del montaje interno de la placa AD8232..... | 24 |
| Figura 20. Placa Arduino Uno empleada en el montaje..... | 24 |
| Figura 21. Diagrama de bloques del procedimiento seguido en la programación del Arduino Uno..... | 25 |
| Figura 22. Esquema del montaje final de los elementos hardware..... | 26 |
| Figura 23. Diagrama de bloques de la etapa de preprocesado de la señal de EEG para cada subparámetro..... | 28 |
| Figura 24. Esquema del árbol de decisión implementado para la clasificación en rangos de BIS..... | 30 |
| Figura 25. Interfaz de usuario diseñada..... | 33 |
| Figura 26. Menú <i>File</i> y modos de utilización de la interfaz de usuario propuestos..... | 33 |
| Figura 27. Panel de selección del puerto serie..... | 33 |
| Figura 28. Mensaje emergente al tratar de finalizar el registro..... | 35 |
| Figura 29. Vista de la interfaz de usuario para la revisión de un registro previo..... | 37 |
| Figura 30. Vista de la interfaz de usuario para la importación de un registro de EEG externo..... | 38 |
| Figura 31. Captura de pantalla de la interfaz de usuario registrando en tiempo real..... | 39 |
| Figura 32. Captura de pantalla del contenido de la variable almacenada..... | 39 |
| Figura 33. Captura de pantalla de la interfaz de usuario al revisar una sesión previa..... | 40 |
| Figura 34. Captura de pantalla de la interfaz de usuario al importar un EEG externo..... | 40 |
| Figura 35. Sistema de monitorización completa BIS..... | 41 |
| Figura 36. Características de la pantalla de datos de tendencias del monitor BIS..... | 43 |
| Figura 37. Esquema de la colocación de los electrodos de EEG..... | 53 |
| Figura 38. Esquema de la colocación de los electrodos de ECG..... | 53 |

Índice de tablas

| | |
|--|----|
| Tabla 1. Definición de las bandas de frecuencias del EEG | 10 |
| Tabla 2. Efectos de los fármacos en las fases de la anestesia (las flechas indican en qué nivel afectan a cada fase, Ø indica que no se produce ningún efecto) | 13 |
| Tabla 3. Relación entre el rango de BIS y el nivel de profundidad anestésica..... | 30 |
| Tabla 4. Modelos de regresión lineal para calcular el BIS en cada uno de los cinco rangos..... | 31 |
| Tabla 5. Cuadro de mano de obra..... | 79 |
| Tabla 6. Cuadro de maquinaria..... | 80 |

I. MEMORIA

1. Motivación y objetivos

En las últimas décadas, el control de la profundidad de la anestesia durante las intervenciones quirúrgicas ha ganado protagonismo en diversos proyectos de investigación (Gruenewald et al., 2021; Johansen, 2006; Lee et al., 2019; Rampil, 1998; Schneider et al., 2014; Shepherd et al., 2013). Esto es, durante la anestesia general, se induce mediante fármacos una pérdida de consciencia controlada y reversible en el paciente (Tranquilli & Grimm, 2015). No obstante, la respuesta individual ante estos fármacos es muy difícil de predecir, ya que son muchos los factores que influyen, como la coadministración de múltiples fármacos de manera sinérgica y la variabilidad de la respuesta de cada individuo frente a éstos. Además, es muy importante administrar la dosis adecuada de cada fármaco para evitar complicaciones y un mayor riesgo de morbilidades asociados a una cantidad excesiva o insuficiente. Una de las mayores preocupaciones es que el paciente no quede completamente inconsciente y sea capaz de recordar lo sucedido durante la intervención (Shepherd et al., 2013). Por tanto, anteriormente, los anestesiólogos administraban con excesiva frecuencia, una dosis de fármaco superior a la necesaria (Johansen, 2006). Por otro lado, tradicionalmente se estimaba la profundidad anestésica en base a respuestas fisiológicas como la frecuencia cardiaca, la saturación de oxígeno, la presión arterial, etc. (Gruenewald et al., 2021; Schneider et al., 2014).

Así, surgió la necesidad de hallar una medida objetiva y eficaz para el control de la profundidad anestésica. En este contexto, se demostró que la señal de electroencefalograma (EEG) contiene información de gran importancia acerca del nivel de consciencia (Gruenewald et al., 2021). De este modo, surgió empíricamente el índice biespectral (BIS), un parámetro desarrollado por la empresa Aspect Medical Systems, Inc., asentada en Norwood, MA, Estados Unidos. Éste ofrece una gradación entre 0 y 100 acerca del nivel de consciencia y se obtiene al combinar subparámetros extraídos del EEG utilizando funciones no lineales (Rampil, 1998). No obstante, no ha sido revelada la totalidad del algoritmo que permite obtener el BIS, por lo que éste no es reproducible (Lee et al., 2019). Así, la interpretación de los resultados es más compleja cuando se desconoce por qué se ha obtenido un cierto valor de BIS.

Por otro lado, el coste de los monitores BIS se encuentra en torno a los 9000 € (Abenstein, 2009). Esto implica que, en los centros con menor presupuesto, no se puedan adquirir todos los monitores necesarios para ofrecer una atención de calidad a todos los pacientes. Por tanto, en estos centros, se atendería a un menor número de pacientes por no disponer de suficientes equipos y se alargarían las listas de espera.

A pesar de que fue diseñado para su uso en quirófano como monitor del nivel de profundidad de la anestesia, también se está utilizando en proyectos de investigación en el ámbito de la rehabilitación neurológica. Por ejemplo, en el contexto de las terapias de estimulación de pacientes con Estados Alterados de Consciencia (EAC), la monitorización del BIS durante las terapias de estimulación podría ser significativa de la respuesta de los pacientes a éstas. Así, para la elaboración de este proyecto, se ha contado con la colaboración de la Dra. Gemma Mas Sesé, neuróloga del Hospital la Pedrera, Denia. Dada su experiencia con estos equipos, la Dra. Mas ha podido observar las ventajas y limitaciones que

presentan los equipos comerciales, para así transmitir las especificaciones que debería cumplir un equipo de monitorización del nivel de consciencia basado en el BIS.

En el ámbito de la investigación, es esencial comprender por qué se obtienen ciertos valores, en este caso de BIS, para poder interpretarlos. Así, es imprescindible que un dispositivo de medida del nivel de consciencia represente cada uno de los elementos que se emplean para determinar el valor de BIS, así como otros parámetros específicos de estudios electroencefalográficos. Además, también resulta útil disponer, no solo de la señal de EEG, sino también de la señal de electrocardiograma (ECG), con el fin de poder correlacionar ambas.

En el presente proyecto, se pretende implementar un sistema de bajo coste y código abierto para estimación del nivel de consciencia basado en el BIS. Además, se pretende que su uso no se limite únicamente al control de la profundidad de la anestesia en quirófano, sino que también pueda emplearse en estudios de investigación donde se requiera monitorizar esta variable.

Así, los objetivos principales de este proyecto son:

- Implementar sistema de captación de EEG y ECG de bajo coste.
- Diseñar un software de análisis de código abierto para estimación del BIS y de otros parámetros de interés para los profesionales sanitarios.
- Diseñar una interfaz de usuario de fácil manejo, accesible y útil para estimación del nivel de consciencia tanto en quirófano, como en investigación.

Para lograr estos objetivos principales, se han establecido unos objetivos secundarios:

- Diseñar un algoritmo para calcular los subparámetros que permiten obtener el BIS.
- Diseñar un algoritmo para estimar el BIS a partir de los subparámetros calculados previamente.
- Determinar los parámetros de interés para los profesionales sanitarios en el estudio de la actividad cerebral y diseñar un algoritmo para obtenerlos.
- Comprobar el funcionamiento del sistema implementado.
- Comparar el sistema para estimación del nivel de consciencia implementado con un equipo comercial.

2. Principios de electroencefalografía

La electroencefalografía es una técnica no invasiva de monitorización electrofisiológica del cerebro basada en el registro de la actividad bioeléctrica en la superficie del cuero cabelludo. Esta señal captada es representativa de la actividad eléctrica generada por las neuronas que se encuentran en las capas más superficiales del encéfalo. Por tanto, para poder interpretar este registro, es importante comprender las bases fisiológicas de generación de señales eléctricas a nivel de las neuronas, su propagación y cómo se compone la señal registrada en la superficie corporal.

2.1. Potencial de acción neuronal

La neurona es la unidad fundamental del cerebro y del sistema nervioso. A nivel histológico, son células alargadas que pueden presentar tamaños y morfologías variables dependiendo de su función y localización. Poseen tres componentes esenciales: el soma o cuerpo neuronal, las dendritas y el axón, representados en la Figura 1. Su función principal es la de procesar y comunicar información. Para ello, cada neurona integra la información que recibe a través de miles de sinapsis, la procesa y la comunica a otras neuronas, generando una red neuronal (Woodruff, 2019).

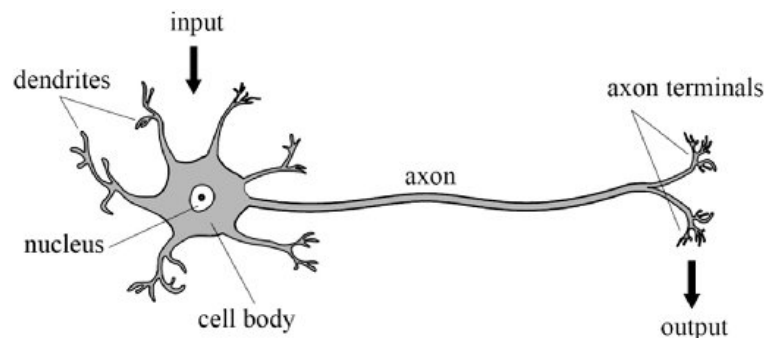


Figura 1. Esquema de la morfología de la neurona (Neves et al., 2018).

Las sinapsis son zonas especializadas de contacto donde tiene lugar la transmisión del impulso eléctrico mediada por un transmisor. Se produce una transmisión unidireccional de la información desde el botón sináptico del axón de la neurona presináptica hacia la dendrita de la neurona postsináptica. Las sinapsis químicas son las más comunes; en ellas, la señal eléctrica presináptica alcanza el terminal del axón, provocando la liberación de neurotransmisores a la hendidura sináptica y su unión a los receptores que se encuentran en la superficie de las dendritas postsinápticas, como se muestra en la Figura 2. De este modo, se estimula la emisión de un nuevo impulso eléctrico postsináptico (Izaurieta & Saavedra, s. f.; Williams et al., 2017).

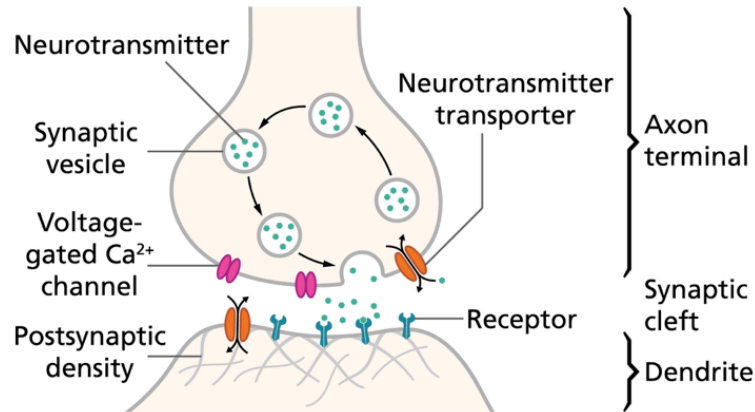


Figura 2. Esquema del funcionamiento de las sinapsis químicas (Williams et al., 2017).

Normalmente, el medio intracelular de la neurona es negativo en comparación con el medio extracelular, con un potencial de membrana en reposo alrededor de -70 mV. La bicapa lipídica de la neurona actúa como capacitor, mientras que los canales iónicos que la atraviesan actúan como resistores. Cuando éstos últimos se abren, se produce un flujo de iones hacia el interior de la célula, alterando su potencial. De este modo, ya que se trata principalmente de cationes (sodio y potasio), el potencial se vuelve menos negativo y, si alcanza un cierto umbral de disparo, se produce la despolarización de la célula, generando un potencial de acción. Si bien es necesario alcanzar el umbral para que se dispare, una vez generado, el potencial de acción evoluciona con independencia del estímulo inicialmente recibido. El potencial de acción se compone de tres fases: despolarización, repolarización e hiperpolarización, como se muestra en la Figura 3 (Chen & Lui, 2021).

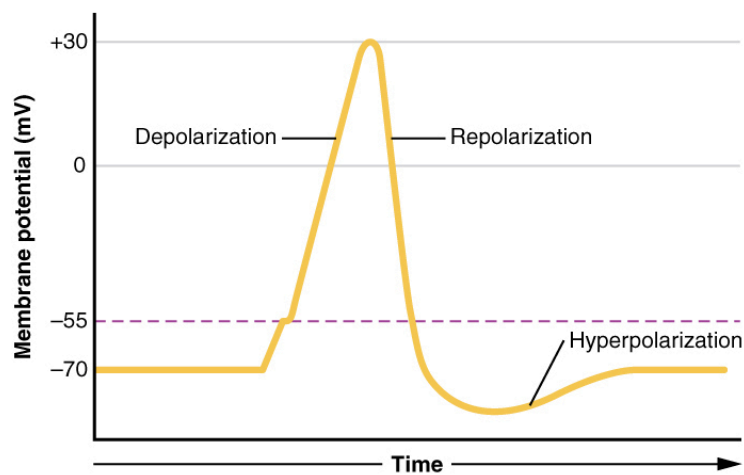


Figura 3. Potencial de acción neuronal. La línea discontinua representa el potencial umbral (Chen & Lui, 2021).

La unión de neurotransmisores específicos como la acetilcolina a los receptores de la neurona postsináptica desencadena la apertura de canales iónicos de sodio. Así, se produce un flujo entrante de sodio, lo que constituye una corriente de estímulo que eleva progresivamente el potencial de membrana. Si la amplitud y duración del estímulo no son suficientes para alcanzar el umbral, el estímulo se denomina subumbral y el potencial de membrana retorna al de reposo. En cambio, si se alcanza el umbral, el estímulo se considera supraumbral y permite el disparo del potencial de acción. Este potencial umbral se encuentra en torno a -55 mV y puede alcanzarse mediante distintas

combinaciones de duración y amplitud del estímulo, de manera que a mayor amplitud, menor es la duración requerida para alcanzar el umbral. Asimismo, una única sinapsis no puede desencadenar por sí sola un potencial de acción, sino que cada neurona presenta una media de mil sinapsis que participan en la variación del potencial de membrana (Chen & Lui, 2021; Ferrero de Loma-Osorio, 2019a).

Una vez superado el umbral, se abren completamente los canales de sodio, generando un mayor flujo de este ion, hasta llegar a un equilibrio de concentraciones intracelular y extracelular de sodio. Esta fase de despolarización termina al alcanzar aproximadamente +40 mV, cerrando rápidamente los canales y extinguiendo la corriente de sodio. A continuación, se desencadena la apertura de los canales de potasio y un flujo de salida del ion hacia el medio extracelular. Debido a la salida de cargas positivas hacia el medio extracelular, el potencial de membrana disminuye (fase de repolarización) hasta alcanzar el potencial de equilibrio del potasio en torno a -100 mV (fase de hiperpolarización), igualando las concentraciones intracelular y extracelular. Los canales de potasio se cierran entonces, extinguiendo esta corriente. Finalmente, la membrana recupera progresivamente su potencial de reposo (Chen & Lui, 2021; Ferrero de Loma-Osorio, 2019a).

2.2. Propagación del potencial de acción

Una vez se genera el potencial de acción a nivel local, éste puede despolarizar segmentos adyacentes de la membrana, superando el umbral de disparo y provocando la propagación del potencial de acción a lo largo del axón. La velocidad de propagación es crucial para lograr un correcto procesamiento de la información y depende de las dinámicas de los canales iónicos que se encuentran en el axón, pero también de sus propiedades geométricas (Schmidt & Knösche, 2019). Cabe destacar que existen dos tipos de axones: los amielínicos y los mielínicos. Los primeros están únicamente compuestos por la membrana celular y el citoplasma. En cambio, los axones mielínicos poseen unas estructuras que rodean el axón, aislando la membrana e impidiendo el paso de iones a través de ella. Estas estructuras son las vainas de mielina, en forma de células de Schwann o de oligodendrocitos (Figura 4), y únicamente pueden hallarse en los axones neuronales (Ferrero de Loma-Osorio, 2019b).

En los axones amielínicos, el potencial de acción que se ha generado a nivel local constituye una corriente de estímulo para el siguiente tramo de membrana que no ha sido despolarizada. Así, si se supera el umbral de disparo, se abren los canales de sodio, permitiendo la entrada del catión al medio intracelular y, como consecuencia, se genera un nuevo potencial de acción que va a continuar propagándose por el axón. Cabe destacar la importancia de la presencia de canales de sodio a lo largo de todo el axón, pues, si éstos no estuvieran, no podría llevarse a cabo la propagación del potencial de acción y la corriente se extinguiría (Ferrero de Loma-Osorio, 2019b; Giulliodori & DiCarlo, 2004).

Por otro lado, las vainas de mielina contribuyen a crear una mayor resistencia de la membrana, evitando que haya corrientes de fuga a través de canales iónicos. No obstante, el potencial de acción sólo puede propagarse debido a la entrada de sodio a través de estos canales, por lo que se crean huecos entre las vainas de mielina, llamados nodos de Ranvier. En ellos, se halla una gran concentración de canales iónicos para recuperar la conducción del potencial de acción a lo largo del axón. Es lo que se conoce como conducción saltatoria. Así, la despolarización de la membrana se produce únicamente a nivel de los nodos de Ranvier, reduciendo tanto la longitud como el área en las

que se produce en comparación con los axones amielínicos, como se observa en la Figura 5. Por tanto, a igualdad de diámetro de axón, la presencia de las vainas de mielina permite una mayor velocidad de propagación del potencial de acción a lo largo del axón (Chen & Lui, 2021; Giuliadori & DiCarlo, 2004).

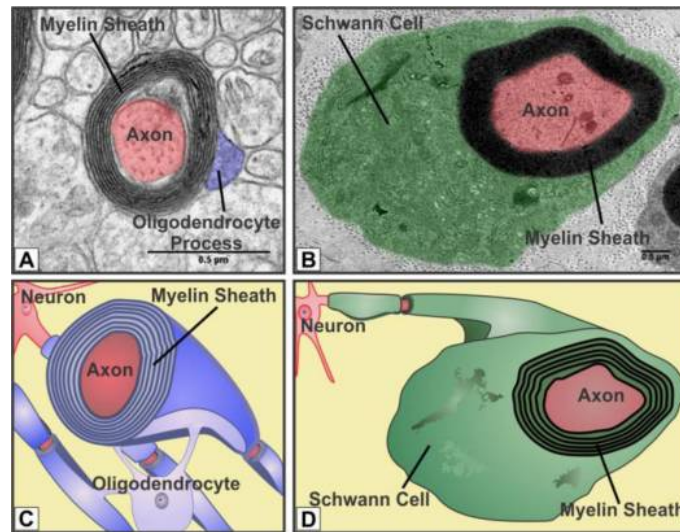


Figura 4. Micrografías electrónicas de células de Schwann y oligodendrocitos envolviendo axones mielínicos. (a) Imagen de microscopio electrónico en falso color y (c) esquema de un axón neuronal mielínico con un oligodendrocito formando vainas de mielina para múltiples segmentos axónicos. (b) Imagen de microscopio electrónico en falso color y (d) esquema de un axón neuronal mielínico con una célula de Schwann formando una vaina de mielina para un único segmento axónico (Clayton & Popko, 2016).

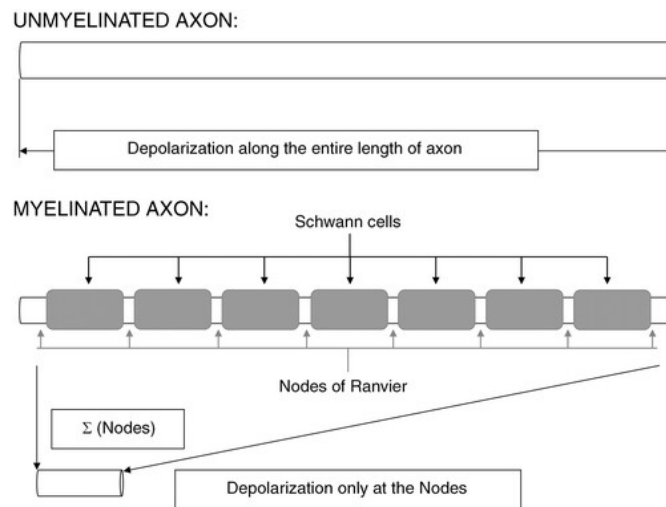


Figura 5. Comparación de la longitud del axón en la que ocurre la despolarización en axones amielínicos (arriba) y mielínicos (bajo) (Giuliadori & DiCarlo, 2004).

2.3. Electroencefalograma

El EEG es el resultado de registrar la actividad eléctrica generada por neuronas corticales, de manera no invasiva. Cada electrodo de medida registra, como mínimo, en torno a 6 cm² de actividad cortical síncrona. Ésta representa el sumatorio de potenciales, inhibitorios o excitatorios, de miles de neuronas

de las capas superiores del córtex. Puesto que los campos eléctricos generados por las neuronas se atenúan con la distancia, únicamente las capas del córtex más cercanas al cráneo pueden influir en el EEG. Para poder producir un potencial eléctrico en la superficie, miles de neuronas deben estar alineadas y disparar potenciales simultáneamente (Keenan et al., 2013; Louis et al., 2016).

La actividad eléctrica de las neuronas puede dividirse en dos categorías: potenciales de acción y potenciales postsinápticos. Estos últimos se producen cuando los neurotransmisores liberados por la neurona presináptica alteran la permeabilidad de los canales iónicos de la membrana celular de la neurona postsináptica y, como consecuencia del flujo de iones a través de estos canales, se altera el potencial transmembrana. Así, la magnitud de los potenciales postsinápticos es proporcional al número de receptores postsinápticos a los que se han unido los neurotransmisores (Rampil, 1998).

El registro de EEG se basa en las corrientes de los potenciales postsinápticos. Individualmente, las corrientes locales de una única neurona serían muy difíciles de detectar desde la superficie del cráneo. No obstante, la anatomía del córtex cerebral posibilita el registro de señales relativamente robustas. Las neuronas corticales se clasifican por su morfología, siendo las neuronas piramidales la clase prevalente. Se caracterizan por presentar dendritas largas apicales que se extienden a través de las capas corticales desde el soma, directamente hasta la superficie de la piamadre (Figura 6). De este modo, neuronas piramidales vecinas presentan dendritas prácticamente paralelas, que reciben miles de conexiones sinápticas de otras neuronas. Por tanto, cuando estas neuronas vecinas presentan alteraciones sincronas de potencial de membrana, las corrientes se combinan de manera aditiva en el fluido extracelular, creando un flujo de corriente local mucho mayor que puede ser detectado por el voltaje que crea en el cráneo (Rampil, 1998).

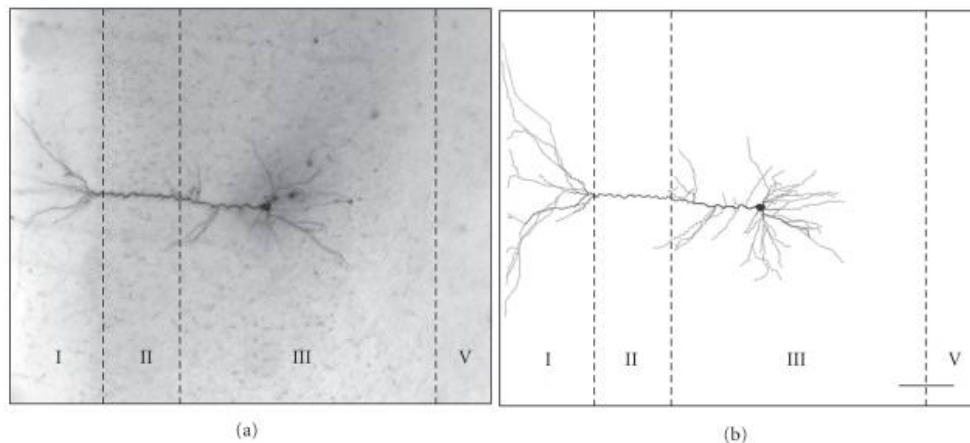


Figura 6. Ejemplo de neurona piramidal del córtex prefrontal; la posición relativa de la célula se muestra mediante líneas discontinuas indicando las capas corticales (I – V). (a) Fotomicrografía de una neurona piramidal etiquetada intracelularmente en la capa III de la subárea prelímbica (hemisferio izquierdo). (b) Reconstrucción de la neurona representada en (a) (Perez-Cruz et al., 2007).

Puesto que los potenciales postsinápticos decaen con el tiempo tras producirse, los potenciales detectados en la superficie del cráneo mediante el EEG cambian con el tiempo. En condiciones normales, se disparan millones de potenciales postsinápticos por todo el córtex, sumándose para crear una señal compleja que no puede volver a descomponerse en potenciales postsinápticos individuales. Así, a diferencia del ECG, un EEG normal no presenta patrones repetitivos y su forma de onda no se correlaciona con fenómenos específicos subyacentes. Esto es, la señal de EEG es aparentemente

aleatoria, lo que dificulta considerablemente su interpretación. No obstante, se pueden medir características extraídas de la señal para posibilitar una monitorización cuantitativa e indirecta de ciertos aspectos de la función cerebral (Rampil, 1998).

Una mayor función cortical se asocia generalmente con la desincronización, es decir, señales eléctricas de frecuencias más altas y menores amplitudes, pues las neuronas actúan de manera más independiente en la creación de actividad consciente. En cambio, una menor función cortical se asocia con la sincronía, es decir, bajas frecuencias y mayores amplitudes. Anatómicamente, la sincronía y el nivel de consciencia se encuentran fuertemente relacionados con circuitos neuronales que conectan el córtex con el bulbo raquídeo y el tálamo (Keenan et al., 2013; Rampil, 1998).

Así, el EEG se clasifica en bandas de frecuencias de interés, representadas en la Tabla 1, y presenta una amplitud entre 5 y 200 μ V.

| Banda de Frecuencias | Rango (Hz) |
|----------------------|-------------|
| Delta (δ) | 0.5 – 3.5 |
| Theta (θ) | 3.5 – 7.0 |
| Alfa (α) | 7.0 – 13.0 |
| Beta (β) | 13.0 – 30.0 |
| Beta2 (β_2) | 30.0 – 50.0 |

Tabla 1. Definición de las bandas de frecuencias del EEG (Rampil, 1998).

2.4. Registro de potenciales en la superficie corporal

Una vez generadas, las corrientes postsinápticas agregadas deben atravesar el fluido cerebroespinal, el cráneo y el cuero cabelludo para poder ser detectadas por los electrodos de superficie. El fluido cerebroespinal y el cuero cabelludo son relativamente conductores en comparación con el cráneo, y el efecto global de la transmisión a través de estas capas es un conjunto de diferencias de potencial en el espacio. Esto implica que la señal de un electrodo de EEG refleja la actividad de un área relativamente amplia, y no solo la zona directamente bajo el electrodo (Rampil, 1998).

Se requieren electrodos de aguja metálica o de gel para que actúen como transductores, de modo que conviertan la corriente iónica fisiológica del EEG en una corriente electrónica que pueda ser procesada posteriormente por el equipo de monitorización de EEG. Las señales voltaicas como el EEG siempre se miden como diferencia de potencial entre dos puntos, por lo que se requiere un amplificador bioeléctrico con dos entradas (una positiva, y otra negativa), además de la destinada al electrodo de referencia. Puesto que la actividad eléctrica del córtex es topográficamente heterogénea, se suele medir en distintas localizaciones del cuero cabelludo. En neurología diagnóstica, existen distintos sistemas de posicionamiento de electrodos, aunque el más utilizado es el sistema 10-20 (Figura 7), que se basa en los meridianos que cruzan la cabeza. La nomenclatura utiliza una letra como prefijo, indicando la región del cerebro en que se encuentra (Fp: frontopolar, F: frontal, T: temporal, C: central, P: parietal, O: occipital), y un número, indicando la distancia relativa a la línea media. Los electrodos

situados en el hemisferio derecho reciben números pares, mientras que los que se ubican en el hemisferio izquierdo reciben números impares. En cambio, los electrodos que se encuentran en la línea media se designan mediante la letra Z en lugar de un número (Rampil, 1998; Rojas et al., 2018).

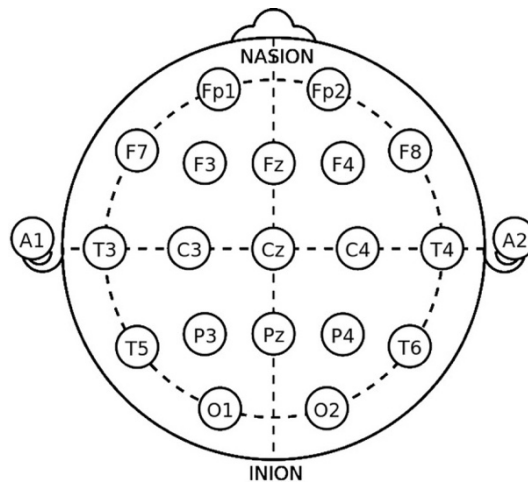


Figura 7. Sistema 10-20 de posicionamiento de electrodos para EEG (Rojas et al., 2018).

Existen tres modos de registro de la señal de EEG: unipolar, promediado y bipolar. En el primero, cada canal capta la diferencia de potencial entre la señal proveniente de un electrodo y un electrodo de referencia, que es común para todos los canales (Figura 8). Existen distintas posibilidades en cuanto a la localización del electrodo de referencia, aunque una de las aproximaciones más utilizadas es la de tomar el promedio del potencial en los lóbulos de las orejas, ya que se consideran eléctricamente inactivos. Este modo de registro permite detectar la región en que se producen descargas localizadas, ya que la amplitud de la señal será mayor cuanto más próximo se encuentre el electrodo de medida.

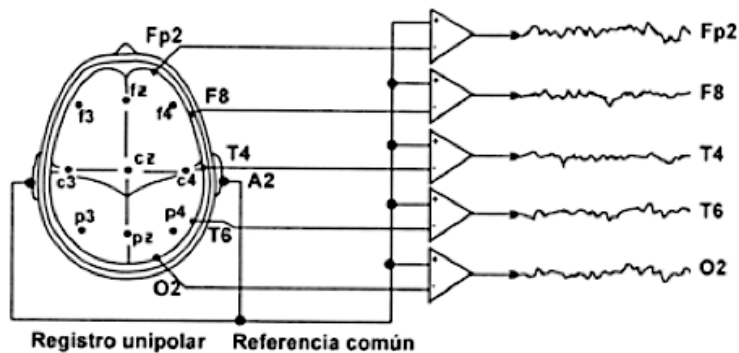


Figura 8. Esquemas de sistemas de registro en modo unipolar con cinco canales (Reta, s. f.).

En el modo de registro promediado, cada canal capta la diferencia de potencial entre el potencial de un electrodo y el promedio de todos los demás (Figura 9). Esto es, se detectan las desviaciones con respecto al promedio instantáneo. De manera análoga al registro unipolar, este modo de registro permite detectar la región en que se producen descargas localizadas, ya que la amplitud de la señal será mayor cuanto más próximo se encuentre el electrodo de medida.

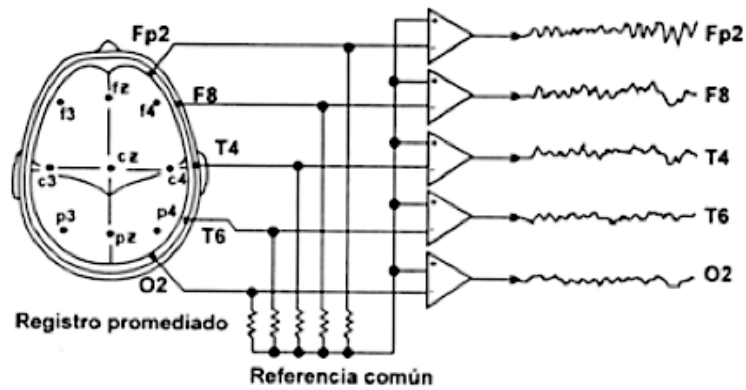


Figura 9. Esquemas de sistemas de registro en modo promediado con cinco canales (Reta, s. f.).

Por otro lado, en el modo de registro bipolar, se emplean dos electrodos activos situados en el cuero cabelludo y se registra la diferencia de potencial entre ambos electrodos (Figura 10). Una de las ventajas de este modo de registro es el rechazo al modo común que se produce. Esto es, cualquier interferencia externa se produce en los dos electrodos al mismo tiempo, por lo que su amplitud y fase se anulan y se logra mejorar la resolución espacial.

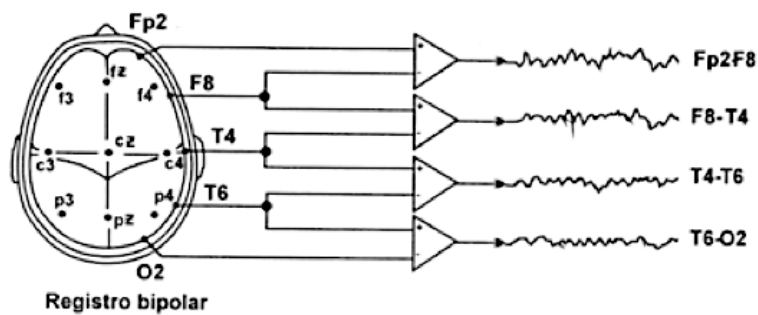


Figura 10. Esquemas de sistemas de registro en modo bipolar con cuatro canales (Reta, s. f.).

3. Conceptos básicos de la anestesia

El término anestesia, que proviene de la palabra griega *anaesthesia* y significa “insensibilidad”, se utiliza para describir la pérdida de sensación en una parte o en la totalidad del cuerpo (Tranquilli & Grimm, 2015). A continuación, se describen las bases de esta técnica, el control de la profundidad de la anestesia y el desarrollo de un parámetro implementado con este fin, BIS.

3.1. Bases de la anestesia

La anestesia se induce con fármacos que inhiben la actividad del tejido nervioso a nivel local, regional, o a nivel del sistema nervioso central. La anestesia general se define como una pérdida de consciencia inducida por fármacos que se caracteriza por una depresión controlada y reversible del sistema nervioso central y de la percepción, afectando a todo el cuerpo (Tranquilli & Grimm, 2015).

En la práctica médica, se pasa por cuatro fases: hipnosis, analgesia, amnesia y relajación muscular (Alwardt et al., 2005). La hipnosis es una condición de sueño o trance inducido artificialmente, que resulta de una depresión moderada del sistema nervioso central. La analgesia se define como la ausencia de dolor en respuesta a una estimulación que, en condiciones normales, sería dolorosa. La amnesia es la pérdida de memoria, no necesariamente a causa de la hipnosis o la sedación (Tranquilli & Grimm, 2015). La relajación muscular conlleva el bloqueo de los impulsos nerviosos que llegan a los músculos, impidiendo su contracción y, por lo tanto cualquier posible movimiento, tanto del músculo esquelético, como del músculo liso (Alwardt et al., 2005).

Cada una de las fases de la anestesia puede llevarse a cabo mediante múltiples fármacos y no existe una técnica universalmente aceptada para la gestión de la anestesia. Así, los fármacos o combinación de fármacos empleados se seleccionan en base al estado fisiológico del paciente y a la propia experiencia del anestesista (Longnecker & Cheung, 1998). En la Tabla 2, se reúnen los efectos farmacológicos de los diferentes compuestos en cada una de las fases de la anestesia.

| Fármaco | Hipnosis | Analgesia | Amnesia | Relajación muscular |
|-----------------------|----------|-----------|---------|---------------------|
| Anestésicos volátiles | ↑ | ↑ | ↑ | ↑ |
| Barbitúricos | ↑↑ | ∅ | ↑ | ∅ |
| Ketamina | ↑↑ | ↑ | ↑↑ | ∅ |
| Propofol | ↑↑ | ↑ | ↑ | ↑ |
| Etomidato | ↑↑ | ↑ | ∅ | ∅ |
| Benzodiacepinas | ↑↑ | ∅ | ↑↑ | ∅ |
| Opioides | ↑ | ↑↑ | ∅ | ↓ |
| Relajantes musculares | ∅ | ∅ | ∅ | ↑↑ |

Tabla 2. Efectos de los fármacos en las fases de la anestesia (las flechas indican en qué nivel afectan a cada fase, ∅ indica que no se produce ningún efecto) (Alwardt et al., 2005).

Pese a que un mismo fármaco puede inducir varias de las fases de la anestesia previamente mencionadas, en ocasiones, se continúan utilizando combinaciones de fármacos para obtener un mayor efecto. Por ejemplo, los anestésicos inhalatorios por sí mismos son capaces de producir todas las condiciones necesarias para anestesiarse de manera segura a un paciente durante una cirugía. No obstante, debido a sus efectos cardiodepresivos, se suelen utilizar bajas dosis de éstos en combinación con agentes intravenosos capaces de producir la hipnosis inicial. Las fases de analgesia y relajación muscular vienen inducidas generalmente por narcóticos y relajantes musculares, especialmente durante momentos de alta estimulación, como puede ser la intubación o las incisiones en la piel. Por otro lado, en ocasiones, los agentes amnésicos son difícilmente diferenciables de los agentes hipnóticos, aunque algunos agentes hipnóticos como las benzodiacepinas, han demostrado mayores propiedades amnésicas que otros (Alwardt et al., 2005).

Actualmente, tres agentes inhalatorios (isoflurano, desflurano y sevoflurano) y un gas (óxido nitroso) son los que se emplean generalmente para alcanzar la fase de hipnosis por vía respiratoria, aunque también se han llegado a utilizar halotano y enflurano (Alwardt et al., 2005). Por otro lado, entre los agentes intravenosos utilizados para la hipnosis se encuentran los barbitúricos (Russo & Bressolle, 1998), la ketamina (Kohrs & Durieux, 1998), el propofol (Morison, 1993), el etomidato (Hensley et al., 2013), y las benzodiacepinas (Alwardt et al., 2005).

En cuanto a la amnesia, no es necesariamente un producto de la hipnosis o la sedación. A pesar de que muchos fármacos hipnóticos también producen amnesia, no siempre se da el caso. Sin embargo, esta fase es muy importante, ya que, si no se induce, existe un alto riesgo de que el paciente recuerde la intervención quirúrgica, lo que es bastante común cuando se emplean narcóticos puros en la fase de hipnosis. Fármacos como la ketamina y las benzodiacepinas se utilizan comúnmente como agentes intravenosos con propiedades amnésicas (Alwardt et al., 2005).

En la fase de analgesia, se emplean generalmente opioides, que se definen como cualquier derivado, natural o sintético, del opio, la morfina o cualquier sustancia que presente los mismos efectos. Los narcóticos tienen efectos analgésicos asociados con el estado de ánimo y el comportamiento, por lo que existe un cierto riesgo de dependencia y tolerancia tras una administración repetida, es decir, se debe controlar su administración (Alwardt et al., 2005). Existen tres receptores principales de opioides (μ , κ y δ). Los de tipo μ y κ se han localizado en áreas del cerebro asociadas con los mecanismos del dolor. En cambio, los de tipo δ se han encontrado en órganos distintos, aunque los niveles más altos se hallan en el corazón. La activación de los receptores de opioides inhibe la liberación de neurotransmisores a las neuronas asociadas con el dolor, así como la respuesta a éstos (Pugsley, 2002).

Los agentes bloqueantes neuromusculares, o relajantes musculares, se administran para facilitar la intubación y prevenir cualquier movimiento del paciente durante los periodos de alta estimulación. De nuevo, el orden de administración de los fármacos es crucial. Concretamente, los relajantes musculares deben administrarse tras la pérdida de consciencia e interrumpir sus efectos antes de que el paciente recupere la consciencia, ya que, de no ser así, éste estaría despierto pero sus músculos estarían paralizados, lo que podría ser traumatizante (Alwardt et al., 2005).

3.2. Control de la profundidad de la anestesia

Cuando un paciente se somete a intervenciones quirúrgicas que emplean anestesia general, es muy importante que la anestesia no sea ni demasiado ligera, ni demasiado profunda. En el primer caso, el paciente podría no estar totalmente inconsciente y cabría la posibilidad de que recordara lo sucedido durante la operación, lo que conllevaría mayores secuelas postoperatorias. Además, se correría el riesgo de no suministrar suficientes fármacos analgésicos. En cambio, si la anestesia es demasiado profunda, habría un mayor riesgo de morbilidades asociadas con la administración de fármacos (Shepherd et al., 2013).

Que un paciente esté consciente durante una intervención quirúrgica es poco frecuente, pero supone una complicación de la anestesia general con graves consecuencias para el paciente. Al utilizar agentes bloqueantes neuromusculares, necesarios para lograr la relajación muscular, puede suceder que el paciente se halle paralizado pero no inconsciente. En esta situación, el paciente no podría comunicar su estado a los sanitarios a pesar de ser consciente de todo lo que sucede durante la intervención, lo que tendría grandes secuelas (Abenstein, 2009).

La respuesta de un paciente a la sedación y a la hipnosis es difícil de predecir, ya que depende de un complejo conjunto de factores, que incluye la coadministración de múltiples fármacos de manera sinérgica y la variabilidad de la respuesta de cada individuo frente a los distintos fármacos. Muchos anestesiólogos fueron entrenados para administrar anestésicos hipnóticos hasta que las respuestas hemodinámicas y de movimiento fueran suprimidas, lo que suponía, en la mayoría de los casos, un exceso de fármaco para el paciente (Johansen, 2006). Tradicionalmente, se estimaba la profundidad de la anestesia en base a respuestas fisiológicas como la frecuencia cardíaca, la saturación de oxígeno, la presión arterial, etc. (Gruenewald et al., 2021; Schneider et al., 2014). Esto es, alcanzar un nivel adecuado de profundidad anestésica y prevenir los posibles efectos secundarios suponía un reto.

El término de adecuación de la anestesia refleja la intención de individualizar la práctica de la anestesia, basándose en distintas variables que reflejen la profundidad de la hipnosis, un nivel de analgesia óptimo y el grado de relajación muscular, reduciendo así memoria, respuestas hemodinámicas o movimiento indeseados y sus consecuencias (Gruenewald et al., 2021).

Como se ha mencionado previamente, determinar la cantidad necesaria de fármacos hipnóticos en una situación concreta puede resultar una tarea complicada, ya que la mayoría de los anestésicos tienen una gran variabilidad en la producción y mantenimiento del estado de hipnosis. Así, la evaluación individual de la farmacocinética y la farmacodinámica de los anestésicos es crucial. Se ha demostrado que monitorizar variables extraídas del EEG aporta información de gran importancia respecto al efecto hipnótico de los fármacos, especialmente los de vía intravenosa (Gruenewald et al., 2021).

3.3. Índice Biespectral

El BIS fue desarrollado por la empresa Aspect Medical Systems, Inc., asentada en Norwood, MA, Estados Unidos, y ofrece una gradación entre 0 y 100 acerca del nivel de profundidad anestésica. Se

trata de un complejo parámetro que se obtiene gracias a una combinación de subparámetros del dominio del tiempo, de la frecuencia y de alto orden espectral. Es capaz de integrar descriptores dispares del EEG en una única variable. Ésta se desarrolló a partir de un gran volumen de datos clínicos, con el fin de hallar una combinación de parámetros que correlacione una valoración del comportamiento de la sedación y la hipnosis, siendo sensible a los agentes anestésicos específicos empleados (Rampil, 1998).

Concretamente, el conjunto de subparámetros necesarios para calcular el BIS se obtuvo empíricamente a partir de una base de datos de EEG y escalas comportamentales, que representan aproximadamente 1500 administraciones anestésicas (lo que equivale a unas 5000 h de registro) y emplearon una amplia variedad de protocolos anestésicos. En primer lugar, se recolectó un conjunto de registros de EEG libres de artefactos. A continuación, se calculó una serie de subparámetros y se evaluó su correlación con el comportamiento anestésico. Los parámetros con el mejor rendimiento se introdujeron en un análisis multivariante para la creación de un parámetro final, el BIS. Para el cálculo final del BIS, estos subparámetros se combinan empleando funciones no lineales, cuyos coeficientes se determinan mediante el proceso descrito en la Figura 11. El rendimiento del BIS se mejoró con un proceso iterativo en el que intervinieron al menos tres nuevas recolecciones de conjuntos de datos, modelado y refinamiento progresivo. El BIS se evaluó posteriormente en otras poblaciones para comprobar su eficacia y se ha demostrado que se correlaciona con medidas comportamentales de sedación y anestesia ligera debidas a una amplia variedad de fármacos anestésicos (Rampil, 1998).

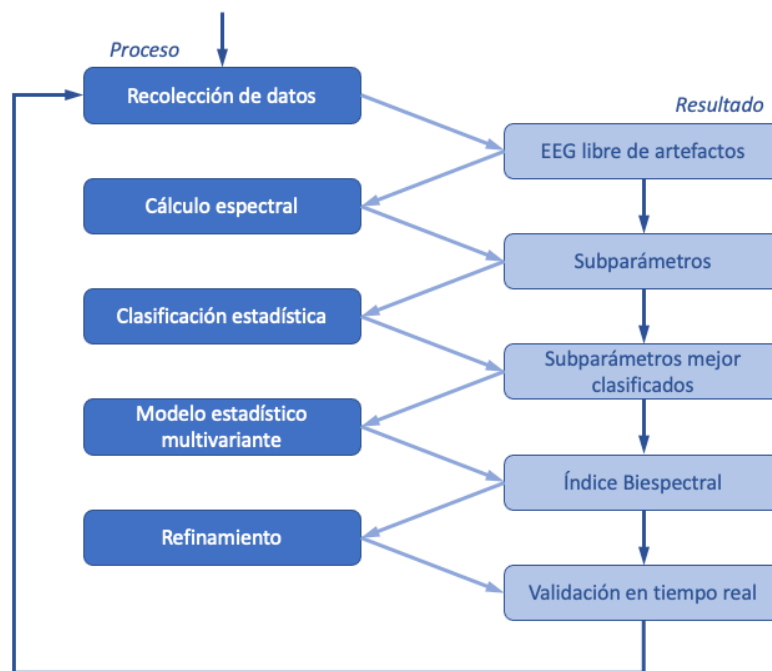


Figura 11. Diagrama de bloques del procedimiento seguido para la obtención del BIS.

El cálculo del BIS comienza con un preprocesado de la señal, en el que se busca filtrarla para excluir los artefactos de altas y bajas frecuencias, y se divide en segmentos de 2 s de duración. Cabe destacar que Rampil (1998) no ha publicado las frecuencias de corte, ni los métodos de filtrado de la señal. Asumiendo que el segmento de EEG está libre de artefactos, se utiliza el dominio del tiempo para calcular el grado de supresión de brotes con dos algoritmos separados: ratio de supresión de brotes

(BSR, *Burst Suppression Ratio*) y “QUAZI”. A continuación, se convierte al dominio de frecuencias con una función de ventana Blackman y se calcula la transformada de Fourier y el biespectro del segmento de EEG. Así, se calculan los subparámetros basados en el dominio de frecuencia “SynchFastSlow” y “Relative β -Ratio”. El BIS resultante se define como una combinación de estos cuatro subparámetros, donde cada uno de ellos se seleccionó de manera que presenten un mejor rendimiento en un rango específico del nivel de profundidad anestésica. En la Figura 12, se presenta el procedimiento a seguir para el cálculo de los subparámetros y del BIS (Rampil, 1998).

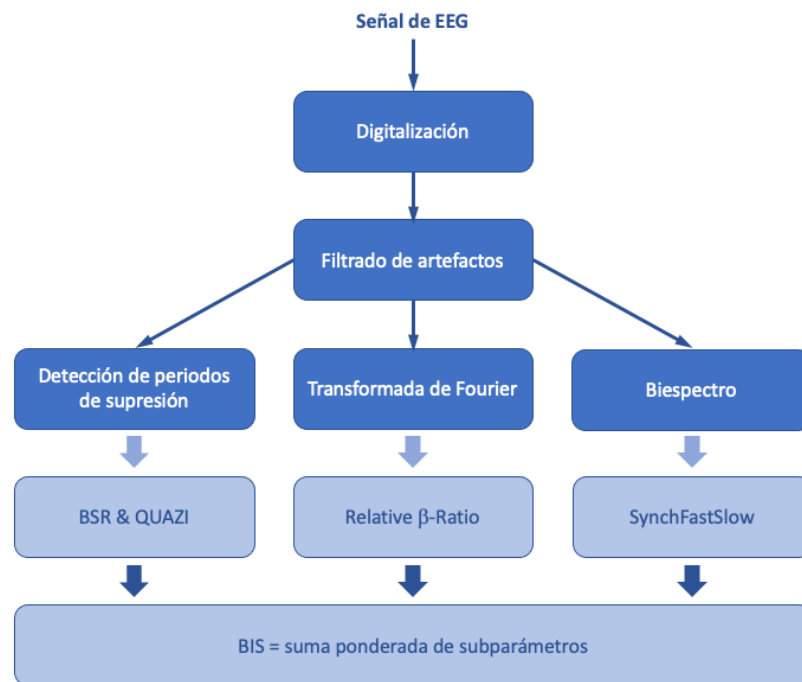


Figura 12. Diagrama de bloques del procedimiento a seguir para el cálculo del BIS.

Así, el parámetro “SynchFastSlow” se correlaciona mejor con las respuestas comportamentales durante una sedación moderada o anestesia ligera. Por tanto, el algoritmo del BIS aporta un mayor peso a este parámetro durante la activación del EEG y durante estados leves de hipnosis. Asimismo, el parámetro “Relative β -Ratio” tiene un mayor peso ante un EEG que presente características de sedación ligera. Por otro lado, los parámetros BSR y “QUAZI” detectan la anestesia profunda (Rampil, 1998).

A continuación, se va a explicar independientemente cada uno de los subparámetros empleados para el cálculo del BIS.

El BSR es un parámetro extraído del dominio del tiempo de la señal de EEG desarrollado para cuantificar el fenómeno de supresión de brotes. Durante la anestesia profunda, el EEG puede desarrollar un patrón de actividad particular, observable en el dominio del tiempo. Este patrón, conocido como supresión de brotes, se caracteriza por alternar periodos de voltaje normal o incluso alto, con periodos de muy bajo voltaje que puede incluso alcanzar la isoelectricidad. Tras traumatismos en la cabeza o isquemia cerebral, este patrón conlleva un pronóstico grave, aunque también puede ser inducido por altas dosis de anestésicos generales, en cuyo caso, la supresión de brotes se asocia con una reducción de la demanda metabólica del cerebro y una posible protección contra la isquemia.

Para calcular el BSR, se reconoce como supresión aquellos periodos más largos de 0.5 s, en los que el voltaje del EEG no supere $\pm 5.0 \mu\text{V}$. Se mide el tiempo que se pasa en este estado y se calcula el BSR como la fracción de la longitud del segmento en que el EEG está suprimido (Rampil, 1998).

El índice de supresión “QUAZI” fue diseñado para detectar la supresión de brotes en presencia de una oscilación de la línea base. Este parámetro incorpora información de baja frecuencia ($< 1.0 \text{ Hz}$) derivada del dominio de frecuencia para detectar actividad superpuesta sobre estas ondas de baja frecuencia. En esta situación, el algoritmo de BSR no sería capaz de detectar la supresión, ya que se superaría el umbral de voltaje establecido en $\pm 5.0 \mu\text{V}$ (Rampil, 1998).

El subparámetro “Relative β -Ratio” se obtiene a partir de la Ecuación 1:

$$\text{Relative } \beta\text{-Ratio} = \log \left(\frac{P^{30-47 \text{ Hz}}}{P^{11-20 \text{ Hz}}} \right) \quad (1)$$

Donde, $P^{x-y \text{ Hz}}$ es la suma de la potencia espectral en la banda de frecuencias de x a y Hz (Rampil, 1998).

El subparámetro “SynchFastSlow” se obtiene del análisis biespectral. Mientras que el espectro de fase producido por el análisis de Fourier mide la fase de las distintas componentes de frecuencia con respecto al inicio del segmento de EEG, el biespectro mide la correlación de fase entre distintas componentes de frecuencia. El análisis biespectral presenta distintas características adicionales que pueden resultar ventajosas para el procesado de señales de EEG: las fuentes de ruido gaussiano se suprimen, mejorando la relación señal-ruido para un EEG no gaussiano, y es posible identificar no linealidades, lo que puede ser importante en el proceso de generación de señales. Concretamente, el análisis biespectral analiza las relaciones existentes entre las sinusoides a dos frecuencias primarias (f^1 y f^2), y un componente de modulación a la frecuencia $f^1 + f^2$. El conjunto de estas tres componentes se conoce como un triplete. Para cada triplete, el biespectro, $B(f^1, f^2)$, representa un valor que incorpora información tanto de la fase como de la potencia, y se calcula a partir de la Ecuación 2 (Rampil, 1998).

$$B(f^1, f^2) = |X(f^1) \cdot X(f^2) \cdot X^*(f^1 + f^2)| \quad (2)$$

Donde $X(f^i)$ es la amplitud del espectro a la frecuencia f^i tras aplicar la transformada de Fourier y $X^*(f^1 + f^2)$ es el complejo conjugado del valor espectral a la frecuencia de modulación. Así, si a cada frecuencia en el triplete hay una alta amplitud espectral y los ángulos de fase están alineados, el producto tendrá un valor elevado. En cambio, si una de las componentes frecuenciales es de baja amplitud o inexistente, o bien los ángulos de fase no se encuentran alineados, el producto tendrá un valor pequeño (Rampil, 1998).

Una vez calculado el biespectro para cada par de frecuencias, el subparámetro “SynchFastSlow” se calcula a partir de la Ecuación 3 (Rampil, 1998).

$$\text{SynchFastSlow} = \log \left(\frac{B^{0.5-47 \text{ Hz}}}{B^{40-47 \text{ Hz}}} \right) \quad (3)$$

Donde, $B^{x-y \text{ Hz}}$ es la suma de la actividad biespectral en el área ocupada por la banda de frecuencias de x a y Hz.

4. Diseño del sistema de captura

La metodología llevada a cabo para la implementación del sistema de estimación del nivel de consciencia puede dividirse en la configuración del sistema hardware y el posterior desarrollo del software de control y adquisición. A continuación, se describen las herramientas empleadas para la elaboración de cada uno de los subsistemas.

4.1. Configuración del sistema hardware

Para la implementación del sistema hardware, se han empleado circuitos electrónicos para captación de EEG y ECG por separado. Se ha decidido utilizar placas electrónicas comerciales, pues se adaptan perfectamente a las necesidades de este proyecto y presentan un bajo coste, por lo que se cumple con el objetivo inicial del proyecto. Asimismo, esta decisión facilita la reproducibilidad del sistema implementado. Por otro lado, se emplea una placa Arduino para el muestreo de las señales, su conversión analógico-digital y su transmisión al ordenador. De nuevo, se opta por una placa comercial por su reducido precio y la facilidad de utilización y programación. A continuación, se describen los distintos elementos hardware empleados.

4.1.1. EEG Click

El registro de la señal de EEG analógica se lleva a cabo con una placa EEG Click (Figuras 13 y 14). Se trata de una placa Click board™, desarrollada por la empresa MikroElektronika (cuya sede se encuentra en Belgrado, Serbia) para monitorización de la actividad cerebral con un único canal de entrada. Está equipada con un circuito de alta sensibilidad que amplifica señales de muy baja amplitud, como es el caso del EEG. Las dimensiones de la placa son de 57.15 x 25.4 mm² y se requiere una tensión de alimentación de 5 V (Burr Brown, 1998; MikroElektronika, 2021).

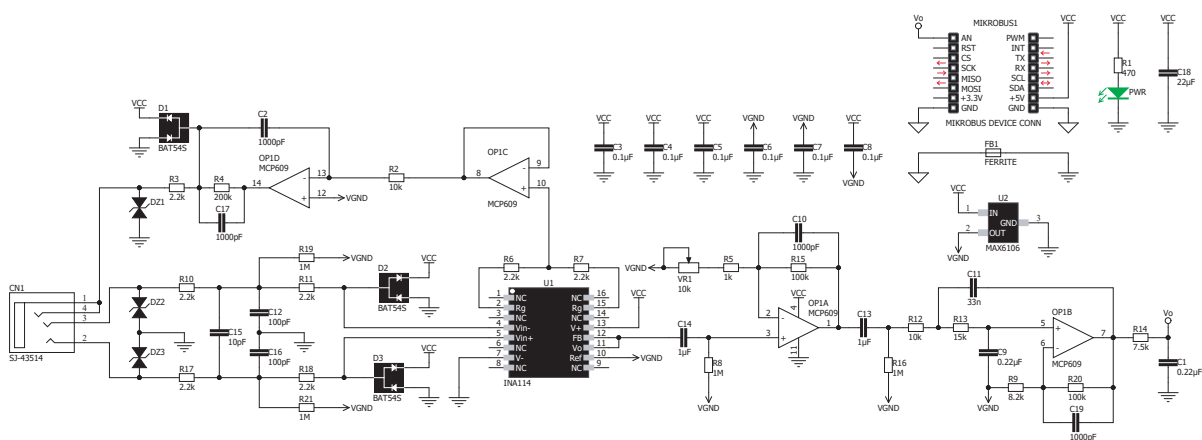


Figura 13. Esquema del montaje interno de la placa EEG Click (MikroElektronika, 2021).

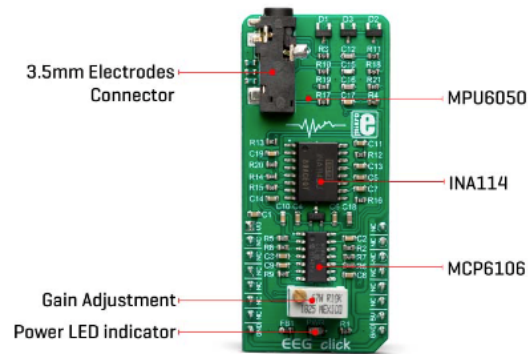


Figura 14. Placa EEG Click utilizada para captura de la señal de EEG (MikroElektronika, 2021).

La primera etapa de amplificación se lleva a cabo con un amplificador de instrumentación INA114, desarrollado por Burr Brown®, una compañía tecnológica de Tucson, AZ, Estados Unidos, especializada en el diseño e implementación de circuitos integrados de alto rendimiento para señales analógicas y mixtas. Concretamente, el INA114 ofrece muy bajo ruido, ajuste por láser (*laser trimming*) para reducción del voltaje de offset (que alcanza 50 μV como máximo) y muy buen factor de rechazo al modo común (115 dB como mínimo). Emplea una única resistencia para establecer la ganancia, que puede alcanzar fácilmente un valor de 10 000. En EEG Click, la ganancia se ha establecido a un valor de 12. Amplificaciones posteriores, así como el filtrado de señal, se llevan a cabo por amplificadores operacionales de tipo MCP609, con cuatro canales, de manera que la ganancia final alcanza 7 800 (Burr Brown, 1998).

Al tratarse de una señal de EEG con una amplitud del orden de microvoltios, es necesaria una gran amplificación. Con el fin de ajustarla, un potenciómetro de precisión permite regular la ganancia de la fase de amplificación intermedia entre 10 y 100. Asimismo, dado que las ondas que provienen del cerebro pueden ser tanto positivas como negativas, se emplea una tierra virtual de tipo MAX6106, con un potencial de referencia de 2.048 V. Esto ayuda a reducir el ruido de la señal, ofreciendo registros de mayor calidad. Finalmente, la señal analógica de actividad cerebral amplificada está disponible en el pin V0 de salida de la placa EEG Click (Burr Brown, 1998; MikroElektronika, 2021).

Idealmente, las medidas de EEG deberían realizarse en una habitación eléctricamente aislada, ya que cualquier tipo de interferencia electromagnética podría corromper la señal registrada. No obstante, el amplificador INA114 ofrece cierta protección contra esta interferencia, ya que, como se ha comentado previamente, presenta un factor de rechazo al modo común muy bueno y permite cancelar la mayoría de las interferencias (MikroElektronika, 2021).

El sistema está protegido contra picos de tensión que pueden aparecer como resultado de una descarga electrostática y voltajes transitorios en contacto con el cuerpo humano. Así, se han instalado una serie de diodos de supresión de descargas electrostáticas y diodos de supresión de voltajes transitorios, con el fin de proteger los amplificadores de la placa ante este tipo de eventos (MikroElektronika, 2021).

El EEG Click utiliza un sistema de tres electrodos para la captura de la señal, que pueden conectarse a través de un conector jack de 3.5 mm a la placa. Según el fabricante, dos de los electrodos deben colocarse detrás de las orejas, son los que se encargan de registrar la señal diferencial y constituyen las dos entradas diferenciales del INA114. El tercero es el electrodo de referencia, que ayuda a eliminar

la interferencia de modo común, y se coloca alejado de los otros dos, en el centro de la frente (MikroElektronika, 2021).

4.1.2. AD8232

El registro de la señal de ECG se realiza con una placa AD8232 (Figuras 15 y 16), desarrollada por la empresa Analog Devices, cuya sede corporativa se encuentra en Wilmington, MA, Estados Unidos. Ha sido diseñada para extraer, amplificar y filtrar señales bioeléctricas de baja amplitud, concretamente el ECG. Las dimensiones de la placa pueden apreciarse en la Figura 16. EL AD8232 opera con una única fuente de alimentación a 3.3 V y emplea una tierra virtual con un potencial de referencia de 1.5 V (Analog Devices, 2013).

El AD8232 presenta un filtro paso alto de segundo orden, con el fin de eliminar tanto artefactos de movimiento, como el potencial de media celda de los electrodos. Este filtro ha sido diseñado de manera que permita alcanzar una ganancia elevada (con un valor de 100) y el filtrado paso alto, con frecuencia de corte de 0.5 Hz, en una única etapa con un amplificador de instrumentación, lo que permite ahorrar tanto espacio en la placa, como coste. Para reducir en mayor medida el ruido de la señal, un amplificador operacional implementa un filtro paso bajo de segundo orden con frecuencia de corte de 40 Hz. Esta segunda etapa de filtrado está configurada para una ganancia de 11, por lo que la ganancia total del sistema es de 1100. Para optimizar el rango dinámico del sistema, el nivel de ganancia es ajustable dependiendo de la amplitud de la señal de entrada (que puede variar en función del posicionamiento de los electrodos) y del rango de amplitudes que admita a la entrada el convertidor analógico-digital al que se conecte posteriormente (Analog Devices, 2013).

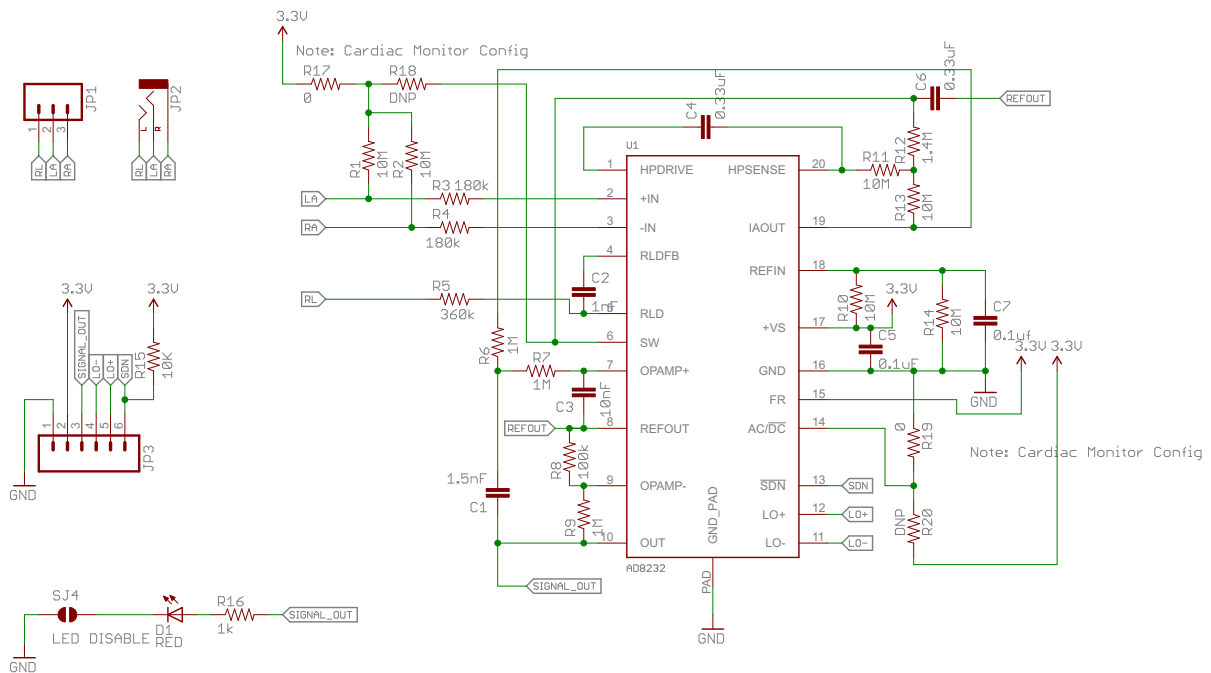


Figura 15. Esquema del montaje interno de la placa AD8232 (Analog Devices, 2013).

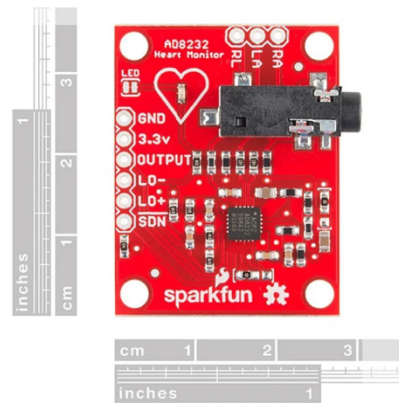


Figura 16. Placa AD8232 utilizada para captura de ECG (Botland, s. f.).

En un sistema de ECG, una gran cantidad de interferencia electromagnética se acopla al cuerpo del paciente a través de la piel. No obstante, el AD8232 ha sido diseñado con el objetivo de mejorar el rechazo al modo común y, por lo tanto, reducir esta interferencia, incluyendo un amplificador para implementación de realimentación activa. Esto es, se extrae la señal de modo común de las entradas del amplificador de instrumentación y, mediante un amplificador operacional, se envía una señal opuesta hacia el paciente a través del electrodo de pierna derecha. Esta funcionalidad mantiene la tensión entre el paciente y el AD8232 prácticamente invariante, mejorando considerablemente el factor de rechazo al modo común, que presenta un valor de 80 dB en el rango de frecuencias de continua (DC) a 60 Hz. Como medida de seguridad, se coloca una resistencia entre el electrodo y la salida de la placa, para garantizar que la corriente nunca sobrepase los $10 \mu\text{A}$. Esta resistencia se calcula para que tenga un valor igual a la tensión de alimentación dividida por $10 \mu\text{V}$, es decir, $330 \text{ k}\Omega$ (Analog Devices, 2013).

El sistema admite dos configuraciones de electrodos posibles: bien empleando únicamente dos electrodos (brazo izquierdo y brazo derecho), bien añadiendo un electrodo adicional en la pierna derecha para reducción de la interferencia de modo común. Los electrodos pueden colocarse en las extremidades o en el torso, como se muestra en la Figura 17. Para conectarlos al AD8232, puede emplearse un conector jack de 3.5 mm, o a través de los pines de entrada destinados a cada uno de los electrodos. Finalmente, la señal analógica de ECG está disponible en el pin OUTPUT de la placa AD8232 (Analog Devices, 2013).

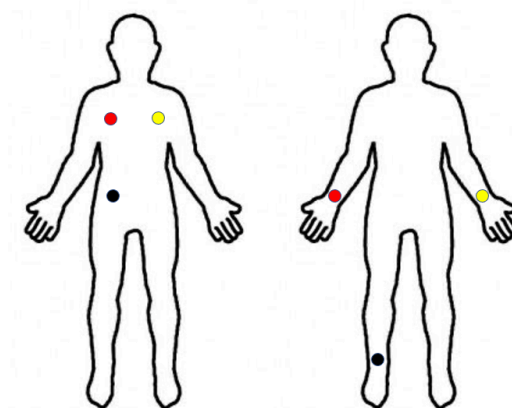


Figura 17. Ejemplos de disposición de los electrodos para medida de ECG.

4.1.3. *Electrodos*

Un electrodo es un material conductor, generalmente metálico, aunque también los hay textiles y plásticos, que constituye una interfaz entre el cuerpo y el equipo de medida, convirtiendo la corriente iónica del organismo en corriente eléctrica. Existen tres grandes tipos de electrodos: de superficie, de aguja y microelectrodos. No obstante, en el presente proyecto se ha decidido emplear electrodos de superficie, que, como su nombre indica, se colocan en la superficie de la piel, por lo que no se compromete la integridad de la misma. Entre los electrodos de superficie, se distinguen los electrodos secos y los de gel. Los primeros se utilizan en situaciones en las que la geometría o el tamaño de los electrodos no permitan la aplicación de gel. En este caso, se emplean electrodos de gel para registrar las señales bioeléctricas tanto de EEG, como ECG (Córdova, 2013).

Debido a la elevada impedancia que presenta la piel, junto con la irregularidad de su superficie, se emplea un gel electrolítico en la interfaz entre el electrodo y la piel, de manera que baje la impedancia de la piel y se maximice el contacto con el electrodo. El metal más común en los electrodos es el de plata - cloruro de plata, ya que se introduce un menor ruido eléctrico en las medidas (Córdova, 2013).

Por otro lado, los electrodos pueden ser desechables o reutilizables, aunque, en este proyecto, se han utilizado desechables. Concretamente, se han utilizado electrodos de gel (Figura 18) de Comepa Industries, cuya sede se encuentra en Bagnolet, Francia. El sistema está diseñado para lograr la máxima conductividad a través de la piel a corto y largo plazo, pues el gel conductor presenta un poder hidroabsorbente. Además, es biocompatible y no deja residuos adhesivos al retirarlo (Comepa Industries, 2020).

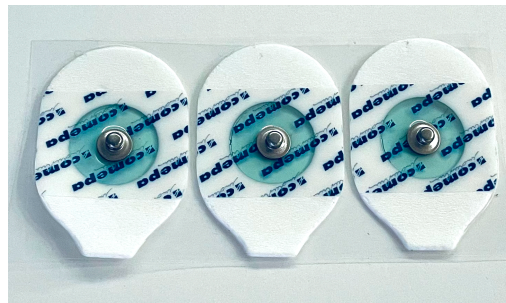


Figura 18. Electrodos de gel empleados para las medidas de EEG y ECG.

4.1.4. *Arduino Uno*

Las etapas de conversión analógico-digital y transmisión al ordenador se llevan a cabo por una placa de tipo Arduino Uno (Figuras 19 y 20), de la marca Arduino®. Arduino es una plataforma de desarrollo electrónico de acceso abierto basada en sistemas hardware y software de fácil manejo. La idea fue concebida en el Instituto de Diseño Interactivo de Ivrea, Italia, como una herramienta para prototipado rápido y fácil, enfocada a estudiantes sin ninguna base de electrónica ni programación. A medida que se expandía su uso, las placas Arduino comenzaron a cambiar para adaptarse a las nuevas necesidades, diferenciando su oferta desde placas simples de 8 bits, a productos para aplicaciones de internet de

las cosas (IoT, *Internet of Things*). Todas las placas Arduino son de código abierto, permitiendo a los usuarios programarlas y adaptarlas a sus necesidades particulares. Asimismo, el software también es de código abierto, lo que permite la colaboración de los usuarios (Arduino, 2021b).

Debido a su accesibilidad y facilidad de uso, Arduino puede utilizarse por principiantes, pero es suficientemente flexible para que usuarios con más experiencia puedan aprovecharlo (Arduino, 2021b).

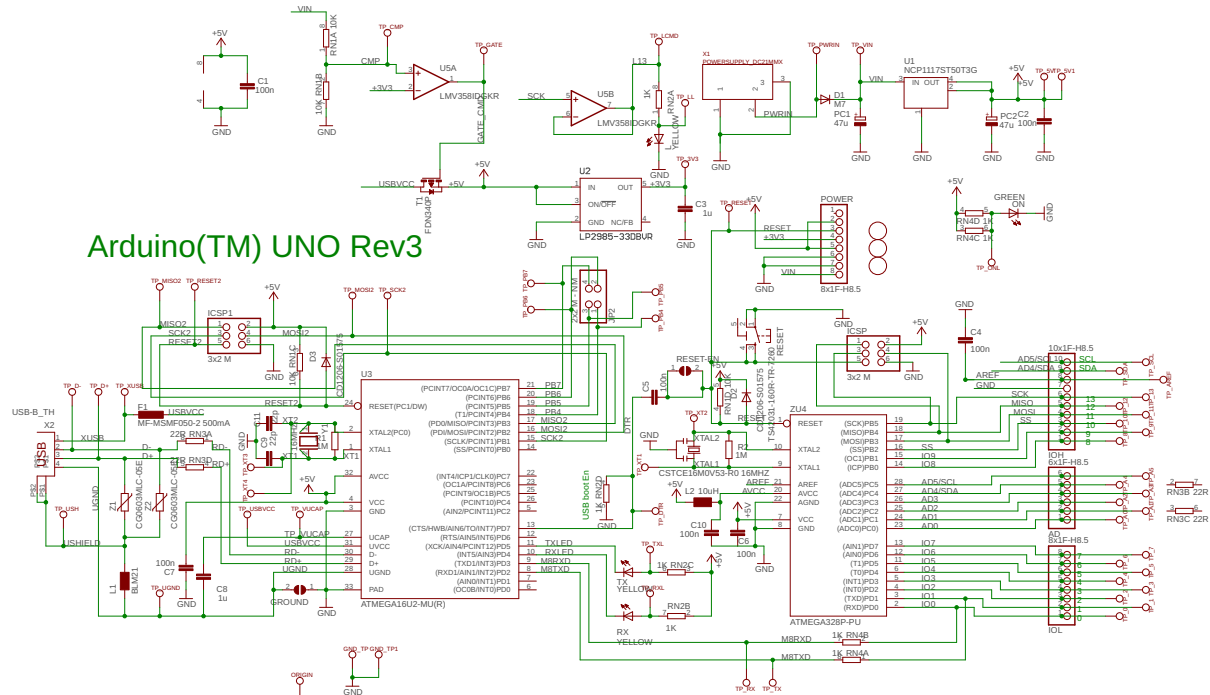


Figura 19. Esquema del montaje interno de la placa AD8232 (Arduino, 2021b).

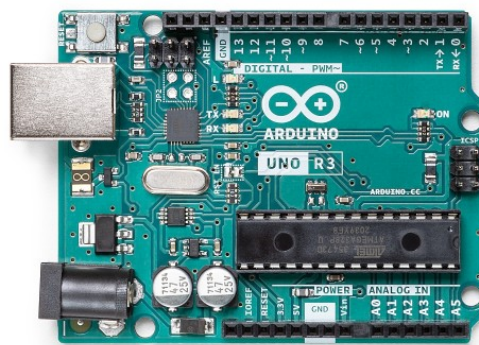


Figura 20. Placa Arduino Uno empleada en el montaje (Arduino, 2021a).

La placa Arduino Uno presenta un microcontrolador de tipo ATmega328P, con 14 pines digitales (de 0 a 13), que pueden emplearse como entrada o salida, y de los cuales 6 pueden ser utilizados como salidas de modulación de ancho de pulsos (PWM, *Pulse Width Modulation*), y 6 pines de entrada analógica (de A0 a A5). Así, la placa puede recibir múltiples entradas, tanto analógicas como digitales, y generar múltiples salidas digitales. Además, cuenta con un resonador cerámico de 16 MHz (CSTCE16MOV53-R0). Respecto a la alimentación, puede realizarse mediante conexión USB, o por un

conector de carga jack, y se envía a los pines específicos de 5 V y 3.3 V. Puede conectarse directamente a un ordenador con la conexión USB, enchufarse a la red eléctrica con un adaptador AC-DC, o a una batería (Arduino, 2021a).

Una de las principales ventajas de la placa Arduino es su bajo coste en comparación con otros microcontroladores y plataformas de microcontroladores con funcionalidades similares. Además, existen versiones que se ensamblan a mano y son incluso más económicas (Arduino, 2021a).

Por otro lado, el software de Arduino permite programar la placa desde cualquier sistema operativo (Windows, Macintosh OSX o Linux), lo que supone una ventaja frente a otras plataformas de microcontroladores que están limitadas a Windows. Para programar el Arduino, simplemente se debe conectar al ordenador mediante conexión USB, identificar el puerto de conexión, escribir el código y compilarlo. El entorno de programación es simple, de código abierto y extensible. El lenguaje de programación puede desarrollarse a través de librerías C++, y aquellos que deseen investigar acerca de detalles técnicos pueden cambiar al lenguaje de programación AVR C en el que se ha basado (Arduino, 2021a).

Para la programación del Arduino Uno, se ha seguido el procedimiento descrito en la Figura 21.

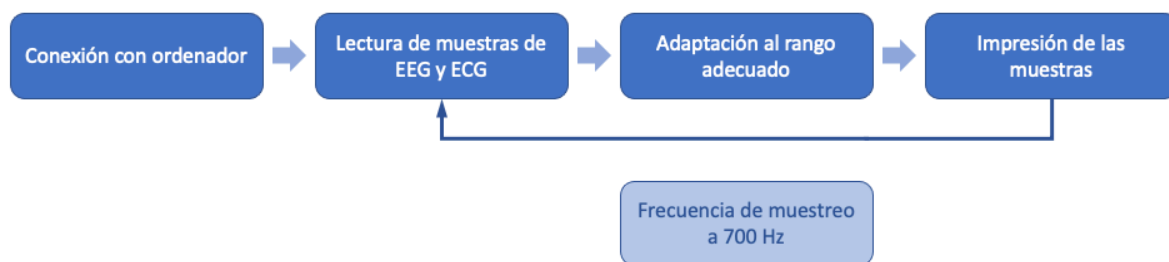


Figura 21. Diagrama de bloques del procedimiento seguido en la programación del Arduino Uno

4.1.5. Montaje

Una vez descritos todos los elementos hardware necesarios para implementar el sistema de captura, se procede con el montaje final (Figura 22).

En primer lugar, se conectan los pines de tierra del Arduino Uno con los de las placas de EEG y ECG (representados con el acrónimo GND). En la Figura 22, estas conexiones se han representado en negro.

A continuación, se conecta la alimentación de las dos placas de captura. En el caso del EEG Click, requiere una alimentación de 5 V y, en el caso del AD8232, la alimentación es de 3.3 V, por lo que se conecta cada uno al pin del Arduino Uno que proporciona la tensión necesaria. En la Figura 22, estas conexiones se han representado de color rojo.

Seguidamente, se conectan las señales de salida del EEG Click y el AD8232 a los pines de entrada del Arduino Uno. Dado que estas señales son analógicas, se tendrán que emplear los pines de entrada analógica. Concretamente, se conecta la salida del EEG Click (V0) a la entrada A0 del Arduino Uno y se representa en azul en la Figura 22. La salida del AD8232 (OUTPUT) se conecta a la entrada A1 del Arduino Uno y se representa en morado en la Figura 22.

Por último, se colocan los electrodos en el paciente y se conectan a las placas de ECG y EEG mediante los cables naranjas y verdes respectivamente (Figura 22).

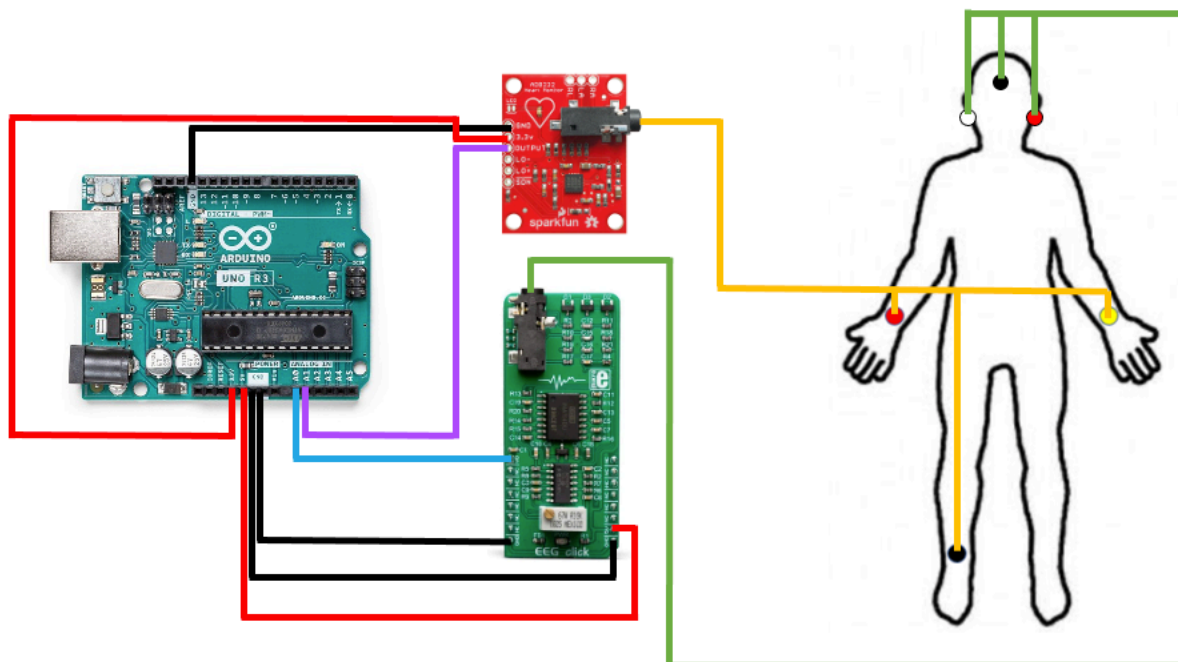


Figura 22. Esquema del montaje final de los elementos hardware (Arduino, 2021a; Botland, s. f.; MikroElektronika, 2021).

Por último, se debe programar la placa Arduino Uno, con el fin de que sea capaz de capturar las señales analógicas, muestrearlas, convertirlas a digital y transmitir las al ordenador a través del puerto serie. Para ello, el primer paso es abrir el software de Arduino previamente descargado. A continuación, se conecta al ordenador la placa Arduino mediante USB y se selecciona el puerto al que se ha conectado. Una vez seleccionado, se escribe el código y se compila.

Las señales de entrada son analógicas, es decir, son continuas y pueden ser medidas con cualquier grado de precisión en cualquier instante. En cambio, el ordenador únicamente puede procesar señales digitales, que representan puntos discretos en el tiempo y su valor se limita a una resolución fija. Cuando se convierte de analógico a digital, esta conversión se produce únicamente en instantes determinados del tiempo, es lo que se conoce como muestreo y, por lo general, se realiza a intervalos regulares. Estos intervalos de tiempo se llaman periodo de muestreo y su inversa es la frecuencia de muestreo, que se expresa en Hercios (Hz) (Rampil, 1998).

Cuando el periodo de muestreo es demasiado largo, y por tanto, la frecuencia de muestreo demasiado pequeña, las ondas más rápidas no se identifican correctamente. Si esto ocurre, se produce lo que se conoce como *aliasing*, que distorsiona la señal digital. Por tanto, es importante muestrear a la frecuencia adecuada. Según el teorema de Shannon, se debe muestrear a una frecuencia mayor al doble de la frecuencia más alta de la señal analógica. No obstante, en la práctica, se suele muestrear a una frecuencia de muestreo entre 4 y 10 veces mayor. En este caso, se considera que, tanto el EEG como el ECG tienen un ancho de banda de 0 a 100 Hz, por lo que, manteniendo una postura conservadora, se selecciona una frecuencia de muestreo 7 veces superior a la máxima de las señales, es decir, 700 Hz, lo que equivale a un periodo de muestreo de 1.4 ms (Rampil, 1998).

4.2. Desarrollo del software de control y adquisición

Para el desarrollo del software, se ha empleado el entorno de programación Matlab® de MathWorks, cuyas oficinas centrales se encuentran en Natick, MA, Estados Unidos. Se han desarrollado una serie de funciones que llevan a cabo el preprocesado de la señal de EEG, el cálculo de los diferentes subparámetros necesarios y la estimación del BIS. El conjunto de los comentarios que describen el código se ha redactado en inglés ya que la comprensión de éste por cualquier usuario y programador es una parte fundamental para la interpretación de los resultados. Uno de los objetivos de este proyecto es que el sistema sea accesible para todos, por lo que se considera que cualquier usuario o programador, independientemente de su origen, debe ser capaz de comprender el contenido de este software.

Como se ha comentado previamente, el BIS es un estimador del nivel de consciencia, ampliamente utilizado para el control de la profundidad de la anestesia. No obstante, únicamente ha sido revelada una porción del algoritmo registrado, por lo que éste no es reproducible. El fabricante de los equipos BIS no ha desvelado el método exacto con que se calcula el índice de supresión “QUAZI”, los criterios de selección de las diferentes ecuaciones de regresión, ni los pesos asociados a cada uno de los subparámetros. Por tanto, se ha llevado a cabo una investigación acerca de métodos para estimación del BIS y se ha decidido utilizar el sistema implementado por Lee et al. (2019). En éste, se propone un algoritmo empleando distintos subparámetros electroencefalográficos, con distintos pesos en función del rango de BIS en el que se haya clasificado previamente la señal con un árbol de decisión (Lee et al., 2019). A continuación se van a describir los pasos seguidos durante el proyecto para ello.

4.2.1. Preprocesado

Para la estimación del BIS a partir del algoritmo diseñado por Lee et al. (2019), se necesita calcular el BSR, el “Relative β -Ratio”, la potencia de electromiografía (BIS-EMG) y la frecuencia límite espectral al 95% (95% SEF, 95% *Spectral Edge Frequency*), cuyos algoritmos se describirán posteriormente. Cada uno de estos parámetros requiere un preprocesado específico por lo que se llevan a cabo tres metodologías en paralelo, representadas en la Figura 23. En primer lugar, se filtra la señal de EEG de entrada. A continuación, se divide en segmentos de 2 s, ya que el BIS se actualizará con este intervalo. Finalmente, para cada segmento, se realiza la transformada de Fourier tras aplicar la ventana Blackman. Esta última etapa no es necesaria para la obtención del BIS-EMG, ya que se obtiene a partir del dominio del tiempo (Lee et al., 2019).

Esta etapa de preprocesado se introduce en Matlab® como una función (“eeg_processing”), cuyas entradas son el EEG sin filtrar y la frecuencia de muestreo. A la salida, se devuelven tres objetos de tipo “struct”, es decir array de estructura. Cada uno de ellos contiene el resultado de cada etapa del preprocesado (filtrado, división en segmentos de 2 s y transformada de Fourier).

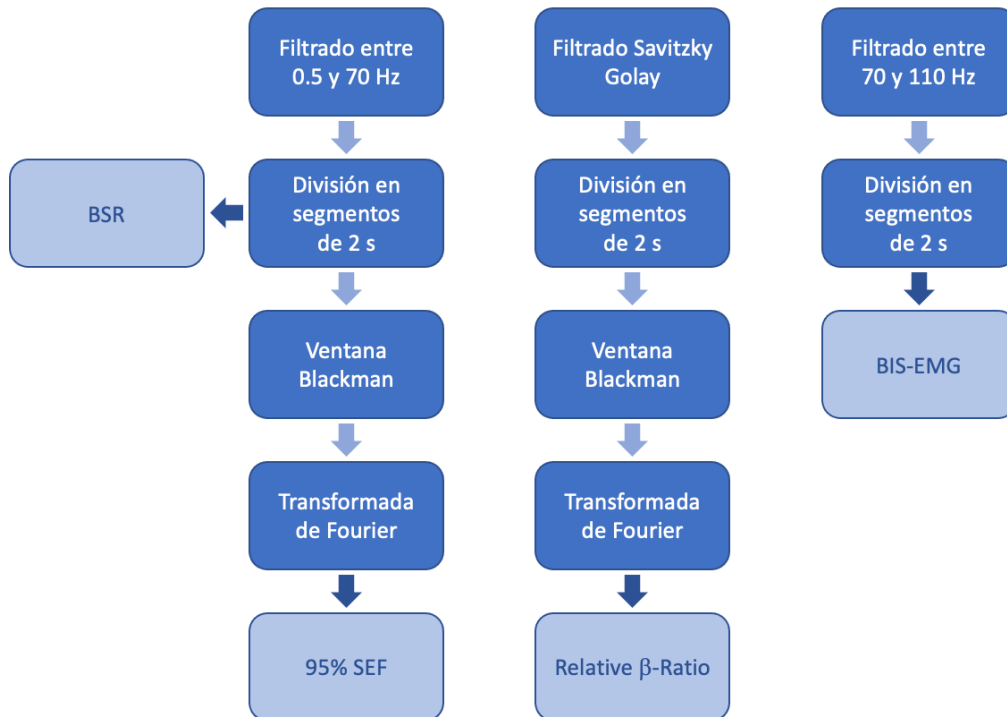


Figura 23. Diagrama de bloques de la etapa de preprocesado de la señal de EEG para cada subparámetro.

4.2.2. Cálculo de subparámetros

Una vez preprocesada la señal de EEG, se procede al cálculo de los subparámetros que se emplean para la estimación del BIS.

BSR

Como se ha mencionado previamente, los periodos de supresión son aquellos en los que no se supera un voltaje de $\pm 5.0 \mu\text{V}$ durante periodos que superan 0.5 s. Por tanto, en base a esta condición, se diseña el algoritmo de cálculo del BSR a partir de la señal de EEG en el dominio del tiempo. Se ha creado una función en Matlab® (“bsr_calculation”) con este fin, cuyas entradas son el EEG filtrado entre 0.5 y 70 Hz y la frecuencia de muestreo. Se calcula el BSR del segmento de entrada, según la Ecuación 4, como la fracción de la longitud del segmento en que el EEG está suprimido (Rampil, 1998).

$$BSR = \frac{\text{longitud de EEG en supresión}}{\text{longitud total del segmento}} \quad (4)$$

Relative β -Ratio

Para la obtención del subparámetro “Relative β -Ratio”, se crea una función en Matlab® (“b_ratio_calculation”) que calcule, a partir de la transformada de Fourier de la señal de EEG filtrada con el filtro Savitzky Golay, la potencia espectral en las bandas de frecuencias de 30 - 40 Hz y de 11 - 20 Hz, y aplique la Ecuación 1.

BIS-EMG

El subparámetro BIS-EMG se define como la potencia absoluta (en dB) del EEG en el rango de frecuencias de 70 - 110 Hz y se calcula según la Ecuación 5, donde x es el valor de cada muestra del EEG de entrada (Lee et al., 2019). Por tanto, se diseña una función de Matlab® (“EMG_power”) que lo calcule a partir del EEG filtrado en ese rango de frecuencias.

$$BIS - EMG = 20 \cdot \log_{10} \left(\sum x^2 \right) \quad (5)$$

95% SEF

El subparámetro 95% SEF se corresponde con la frecuencia por debajo de la cual se halla el 95% de la potencia total (Lee et al., 2019). En el estudio realizado por Lee et al. (2019), únicamente se tiene en cuenta el rango de frecuencias de 0.5 – 30 Hz. No obstante, dado que en este proyecto se está trabajando con un EEG en el rango de 0.5 – 50 Hz (según las bandas de frecuencias definidas previamente de acuerdo a (Rampil, 1998)), se ha decidido continuar utilizando este último rango para el cálculo de 95% SEF. Así, el valor obtenido será más representativo de las características de la señal en cuestión y se facilitará la interpretación por parte de los profesionales sanitarios. Por tanto, de acuerdo a esta definición, se implementa una función en Matlab® (“spectral_edge_freq95”) que calcula el 95% SEF a partir de la transformada de Fourier de la señal filtrada entre 0.5 y 70 Hz, aunque en el interior de la función se reduce este rango al especificado anteriormente: 0.5 – 50 Hz.

4.2.3. Estimación del índice biespectral

Una vez definidas las funciones que permiten obtener los distintos subparámetros, se procede a la estimación del BIS. El algoritmo original del BIS selecciona la ecuación de regresión apropiada tras determinar la profundidad de la anestesia en base a unos valores de referencia de los subparámetros extraídos del EEG. No obstante, como se ha mencionado anteriormente, este algoritmo no ha sido revelado, por lo que se ha decidido utilizar el sistema de estimación de este parámetro implementado por Lee et al. (2019). Para desarrollarlo, se llevó a cabo un análisis basado en árboles de decisión, con el objetivo de hallar los subparámetros del EEG y los umbrales que determinan el nivel de profundidad anestésica, así como un análisis de regresión para estimar las diferentes ecuaciones de regresión para cada nivel de profundidad anestésica (Lee et al., 2019).

El análisis basado en árboles de decisión es un modelo de clasificación de *machine learning* supervisado que puede entenderse como un conjunto de reglas, para el que se requiere conocer el resultado concreto asociado a cada una de las entradas. En este caso, el resultado es el rango de BIS, que corresponde a un nivel de profundidad anestésica determinado. Concretamente, se distinguen cinco rangos representados en la Tabla 3. Además, se emplearon como entradas de los árboles de decisión los subparámetros descritos previamente (BSR, 95% SEF, Relative β -Ratio y BIS-EMG). En total, se crearon diez árboles de decisión de acuerdo a un algoritmo de Árboles de Clasificación y Regresión (CART, *Classification and Regression Trees*) optimizado, en los que se llevan a cabo separaciones binarias a partir de los subparámetros de entrada. Por último, se empleó el método de *Bootstrap aggregating (bagging)*, que consiste en promediar los resultados de los distintos árboles generados

para reducir la varianza y minimizar el sobreajuste del modelo. El árbol de decisión resultante de este análisis se representa en la Figura 24 (Lee et al., 2019).

| Rango de BIS | Nivel de profundidad anestésica |
|-----------------------------|---------------------------------|
| $0 \leq \text{BIS} \leq 21$ | Isoelectricidad |
| $21 < \text{BIS} \leq 41$ | Anestesia profunda |
| $41 < \text{BIS} \leq 61$ | Anestesia adecuada |
| $61 < \text{BIS} \leq 78$ | Sedación ligera |
| $78 < \text{BIS} \leq 98$ | Despierto |

Tabla 3. Relación entre el rango de BIS y el nivel de profundidad anestésica.

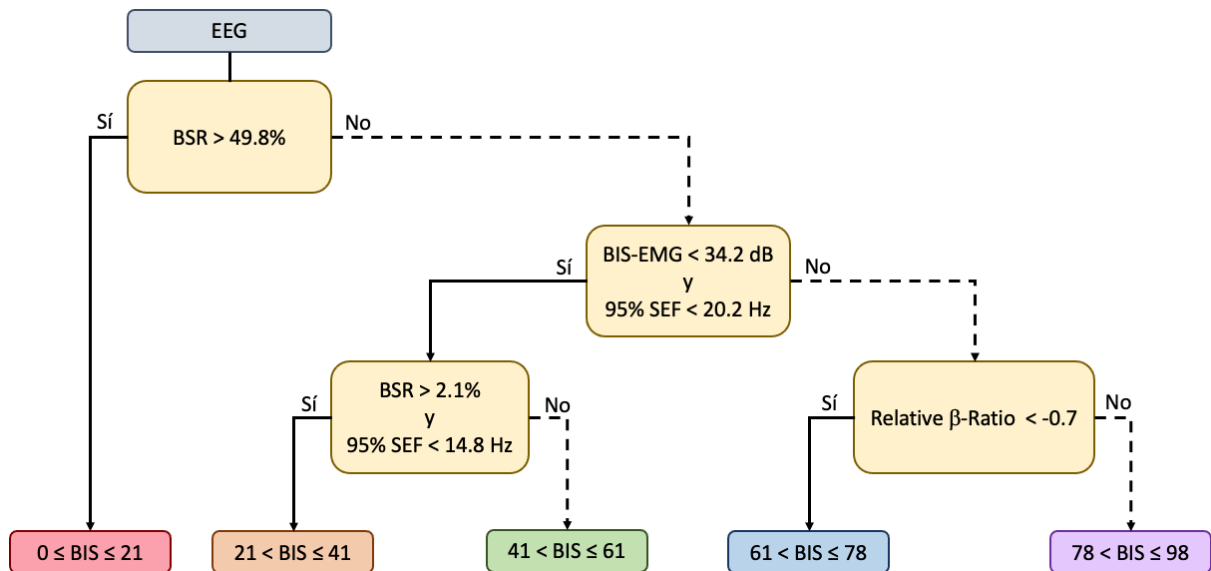


Figura 24. Esquema del árbol de decisión implementado para la clasificación en rangos de BIS.

Una vez desarrollado el clasificador, se llevó a cabo un análisis de regresión lineal múltiple para cada uno de los rangos de BIS, cuyas entradas son los subparámetros seleccionados como entrada del árbol y la salida es una estimación del valor del BIS. Para evitar errores debido a posibles *outliers*, se realizó una regresión de tipo *RANdom Sample Consensus* (RANSAC), con unas 1000 muestras, 100 iteraciones y un margen para los *outliers* de 5. Los resultados del análisis de regresión se presentan en la Tabla 4 como coeficientes para los subparámetros extraídos del EEG y una intercepción para cada uno de los rangos de BIS (Lee et al., 2019).

| Rango de BIS | BSR | 95% SEF | Relative β-Ratio | BIS-EMG | Intercepción |
|-----------------------------|------------|----------------|--|----------------|---------------------|
| $0 \leq \text{BIS} \leq 21$ | -0.42 | 0.00 | 0.01 | 0.00 | 42.1 |
| $21 < \text{BIS} \leq 41$ | -0.42 | 0.91 | 3.06 | 0.04 | 29.9 |
| $41 < \text{BIS} \leq 61$ | -3.01 | 3.84 | -8.70 | 0.96 | -57.6 |
| $61 < \text{BIS} \leq 78$ | -1.43 | 2.55 | 4.26 | 0.41 | 5.3 |
| $78 < \text{BIS} \leq 98$ | -1.97 | 0.88 | 7.89 | -0.07 | 65.2 |

Tabla 4. Modelos de regresión lineal para calcular el BIS en cada uno de los cinco rangos.

En base a estos datos, se programa una función en Matlab® (“BIS_decision_tree”) que es capaz de clasificar una señal de EEG en cada uno de estos cinco rangos y atribuir un coeficiente a cada subparámetro para calcular una estimación del BIS. Para ello, se introducen como entradas de la función los valores de BSR, 95% SEF, Relative β -Ratio y BIS-EMG.

5. Diseño de la interfaz de usuario

Para el desarrollo de la interfaz de usuario, se ha utilizado el entorno AppDesigner de Matlab®. En primer lugar, se ha diseñado el formato de la interfaz, a partir de las recomendaciones de la Dra. Mas. Para que cada uno de los elementos que componen la interfaz desempeñen una cierta función, se han añadido los comandos correspondientes en la vista del código. Una vez finalizada la interfaz de usuario, se han realizado las pruebas pertinentes para comprobar su funcionamiento. A continuación, se describe la interfaz de usuario creada, se comprueba su funcionamiento, se plantea una comparación con el equipo comercial de BIS del grupo Medtronic, cuya sede operativa se encuentra en Minneapolis, MN, Estados Unidos, y se exponen las limitaciones del sistema implementado.

5.1. Descripción

A partir de las especificaciones de la Dra. Mas, se ha diseñado una interfaz de usuario intuitiva y de fácil manejo. Con el objetivo de que sea accesible a cualquier profesional sanitario, independientemente de su nacionalidad, la interfaz de usuario ha sido diseñada en inglés.

La ventana emergente con la interfaz de usuario diseñada (Figura 25) está dividida en dos. A la derecha, se encuentran las gráficas sobre las que se representarán las señales de EEG y ECG, los subparámetros de obtención del BIS, y el propio BIS. A la izquierda, se hallan los distintos elementos con los que puede interactuar el usuario durante los registros (botón de parada, campo de texto para la introducción del identificador del paciente, y opciones específicas de análisis electroencefalográfico). Para la colocación de los distintos elementos en la pantalla, se han dispuesto en una cuadrícula que evita que éstos se desplacen al abrir la interfaz de usuario en modo pantalla completa, o en un monitor de mayor tamaño.

En primer lugar, en la parte superior izquierda de la pantalla, se encuentra un menú desplegable (*File*), que permite elegir entre tres modos de utilización: nuevo registro en tiempo real (*New*), revisión de un registro previo (*Revise Existing File*) y estimación del nivel de consciencia sobre un EEG almacenado (*Import External File*) (Figura 26). Es imprescindible seleccionar una de las tres opciones para que funcione la aplicación. A continuación, se van a describir cada una de las tres alternativas.

Quando se pretende registrar en tiempo real una nueva señal de EEG, se tiene que haber conectado previamente la placa Arduino Uno del sistema hardware implementado al ordenador, mediante conexión USB. Así, para que la interfaz de usuario reconozca la placa y pueda empezar a registrar, se ha diseñado un panel para introducción del puerto serie al que se ha conectado (Figura 27) y que únicamente se hace visible cuando se selecciona la opción *New* del menú *File*. Contiene cuatro elementos: un botón *X*, un campo de texto, una etiqueta y un botón *OK*. En la etiqueta, se indica el formato con el que se debe introducir el puerto serie de conexión del Arduino en función del sistema operativo que se esté utilizando. Así, el campo de texto está destinado a la introducción del nombre de este puerto serie. Al presionar el botón *X*, el panel se hace invisible sin realizar ninguna modificación. En cambio, si se presiona el botón *OK*, la interfaz de usuario reconoce el puerto serie introducido y comienza a registrar los datos que provienen de la placa Arduino Uno.

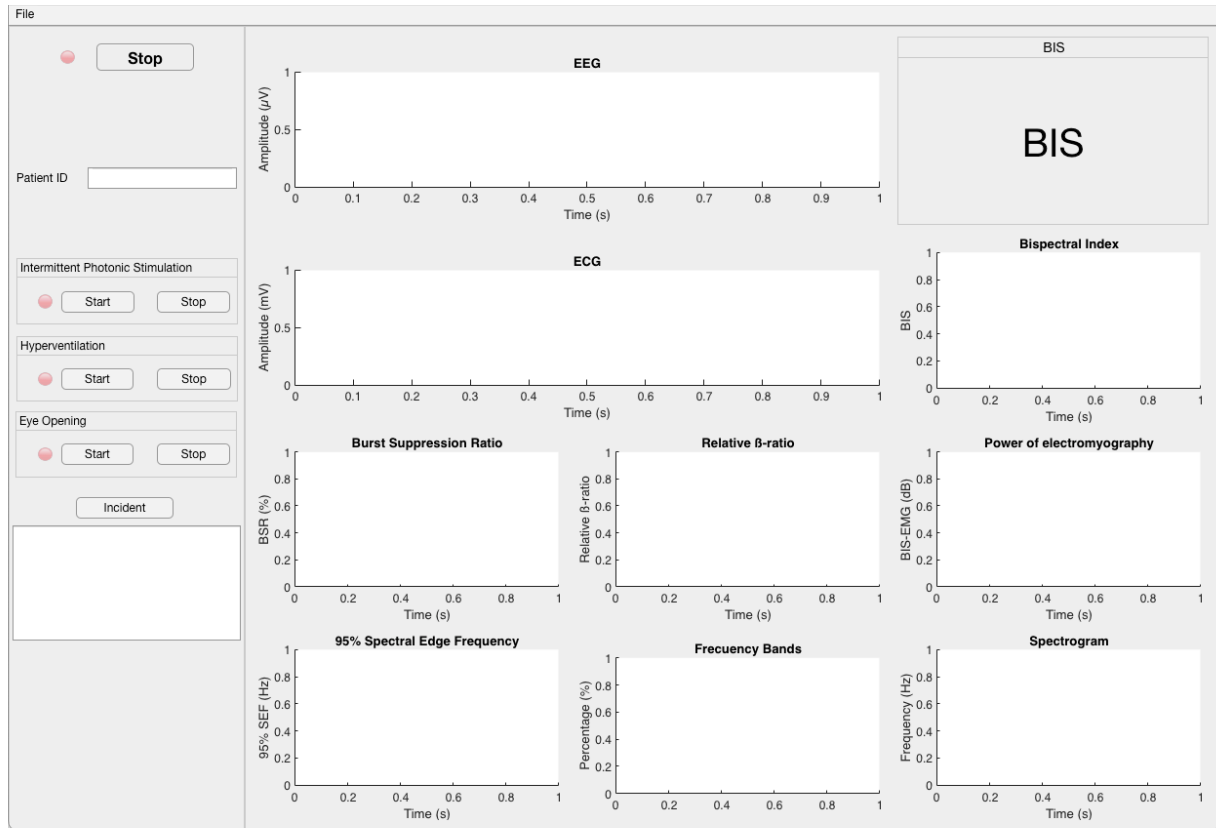


Figura 25. Interfaz de usuario diseñada.

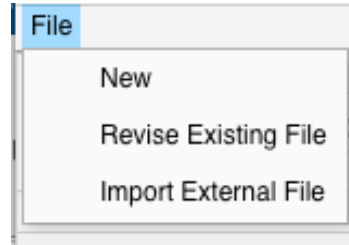


Figura 26. Menú *File* y modos de utilización de la interfaz de usuario propuestos.

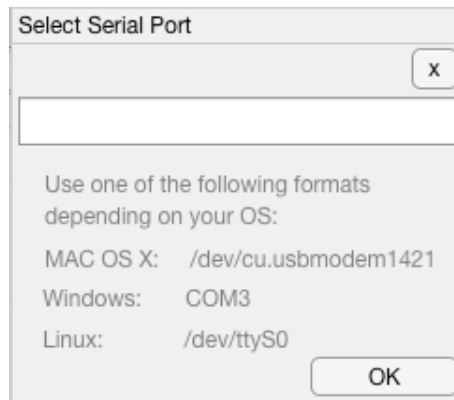


Figura 27. Panel de selección del puerto serie.

Una vez presionado el botón *OK*, se selecciona el puerto serie introducido en el campo de texto para la creación y apertura de un objeto de puerto serie y se hace invisible el panel de selección del puerto. Inmediatamente, se ilumina un LED que indica el inicio del registro y se comienza a representar en tiempo real las señales de EEG y ECG capturadas, mediante líneas animadas, en las gráficas dispuestas para ello. Cada dos segundos, periodo de duración de los segmentos de EEG, se preprocesa el segmento de señal con la función implementada para ello, se calculan tanto los subparámetros con los que se obtiene el BIS, como éste y se representan en pantalla, también mediante líneas animadas. Resulta imprescindible representar tanto el BIS, como los subparámetros, ya que, en caso de que se detectara algún valor anómalo, sería más sencillo clasificarlo como tal e identificar el motivo por el que se ha producido. Además de representar la evolución del BIS en el tiempo mediante una línea animada, se ha añadido un panel en el que aparece el valor instantáneo, que se actualiza cada dos segundos. Asimismo, para proporcionar más información sobre la señal de EEG, se representa el espectrograma de frecuencias de la señal, así como un diagrama de barras en el que aparezca el porcentaje de señal que pertenece a cada una de las bandas de frecuencias, y ambos se actualizan cada dos segundos.

Por otro lado, cabe destacar que el sistema puede estar destinado tanto a un control de la profundidad de la anestesia durante intervenciones quirúrgicas, como a proyectos de investigación. Así, para la segunda situación, resulta interesante incluir funcionalidades predeterminadas de los sistemas de análisis de EEG, que se encuentran en la parte izquierda de la pantalla. Se ha añadido una opción que permite anotar incidencias de cualquier tipo y asociarlas directamente al instante de tiempo en que sucedieron. Para ello, se tiene que redactar la incidencia en el campo de texto destinado a ello y, cuando se quiera introducir en el sistema, se presiona el botón *Incident*. Además, se pueden registrar los periodos de Estimulación Luminosa Intermitente (ELI, que consiste en la exposición directa a trenes de pulsos de luz (Taoufiqi et al., 2016)), hiperventilación y apertura ocular. Para ello, se han implementado paneles específicos para cada una de las opciones (*Intermittent Photonic Stimulation*, *Hyperventilation* y *Eye Opening* respectivamente). En cada uno de estos paneles, hay dos botones, *Start* y *Stop*, que se deben pulsar para marcar el inicio y el final de estos periodos. Cuando se pulsa el botón *Start*, se enciende un LED en el panel, indicando al usuario que se está contabilizando el tiempo transcurrido para ese parámetro. En cambio, cuando se pulsa el botón *Stop*, se apaga el LED.

Una vez se quiere finalizar el registro, se pulsa el botón *Stop* ubicado en la parte superior del panel izquierdo. Para evitar cometer errores, no se deja de registrar inmediatamente, sino que aparece una etiqueta que informa al usuario de que se va a dejar de registrar y le pregunta si quiere guardar lo que se ha registrado hasta el momento (Figura 28). Bajo la etiqueta, se presentan tres botones: *Cancel*, *No* y *Yes*. Si se presiona el botón *Cancel*, desaparece el mensaje emergente y se continúa con el registro. Se ha desarrollado esta opción por si el usuario hubiera presionado el botón *Stop* accidentalmente. En caso de que se hubiera comenzado un registro por error, o se quisiera repetir sin guardar lo que se ha registrado, el botón *No* finaliza el registro, borra sus datos y apaga el LED, indicando que se ha dejado de registrar. Por último, si el registro se ha realizado correctamente y se quiere guardar en el disco duro del ordenador, se presiona el botón *Yes*.

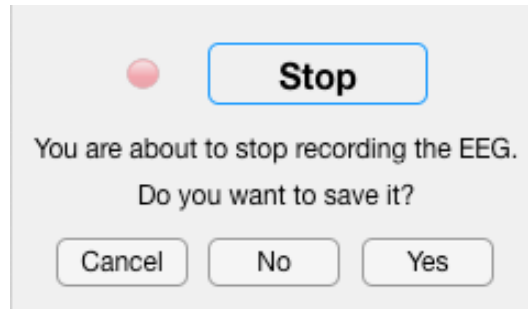


Figura 28. Mensaje emergente al tratar de finalizar el registro.

Así, pulsando el botón *Yes* se finaliza el registro y se crea un objeto de tipo *struct* con todos los datos recabados hasta el momento. En éste, se incluye:

- *TimeEEG*: un vector de tiempo con los instantes en que se ha tomado cada muestra de EEG y ECG.
- *RawEEG*: la señal de EEG sin filtrar.
- *ECG*: la señal de ECG.
- *Incident*: un vector de misma longitud que el vector *TimeEEG* con las incidencias anotadas en los instantes de tiempo en que se pulsó el botón *Incident*; para los instantes de tiempo en que no se introdujo ninguna incidencia, se anota un espacio en blanco.
- *IntermittentPhotonicStimulation*: un vector de misma longitud que el vector *TimeEEG* que presenta un 1 en los periodos en que se llevaba a cabo la ELI (coincidiendo con la iluminación del LED correspondiente) y un 0 en los que no.
- *Hyperventilation*: un vector de misma longitud que el vector *TimeEEG* que presenta un 1 en los periodos en que se producía hiperventilación (coincidiendo con la iluminación del LED correspondiente) y un 0 en los que no.
- *EyeOpening*: un vector de misma longitud que el vector *TimeEEG* que presenta un 1 en los periodos en que se producía la apertura ocular (coincidiendo con la iluminación del LED correspondiente) y un 0 en los que no.
- *ParamTime*: un vector de tiempo con los instantes en que se calculan los distintos parámetros.
- *BIS*: un vector con los valores de BIS calculados durante toda la duración del registro.
- *BSR*: un vector con los valores de BSR calculados durante toda la duración del registro.
- *BetaRatio*: un vector con los valores de Relative β -Ratio calculados durante toda la duración del registro.
- *BIS-EMG*: un vector con los valores de BIS-EMG calculados durante toda la duración del registro.
- *SEF95*: un vector con los valores de 95% SEF calculados durante toda la duración del registro.
- *Delta*: un vector con los valores de porcentaje de señal en la banda δ calculados durante toda la duración del registro.

- *Theta*: un vector con los valores de porcentaje de señal en la banda θ calculados durante toda la duración del registro.
- *Alpha*: un vector con los valores de porcentaje de señal en la banda α calculados durante toda la duración del registro.
- *Beta*: un vector con los valores de porcentaje de señal en la banda β calculados durante toda la duración del registro.
- *Beta2*: un vector con los valores de porcentaje de señal en la banda β_2 calculados durante toda la duración del registro.

Esta estructura creada se guarda con el nombre “EEG_XXXX_dd-mmm-aaaa HH:MM:SS.mat” en el disco duro del ordenador, donde, “XXXX” es el identificador del paciente que debe haberse introducido previamente en el campo de texto *Patient ID* y “dd-mmm-aaaa HH:MM:SS” es la fecha y hora exactas en que se finalizó el registro. Así, cada registro va asociado a un paciente y a un instante determinados y son fácilmente identificables por los profesionales sanitarios que tengan que gestionarlos posteriormente. Además, la extensión del archivo es “.mat” para poder tratar posteriormente los datos almacenados desde la aplicación Matlab®.

Una vez se ha almacenado un registro, éste puede volver a revisarse desde la interfaz de usuario (Figura 29), sin necesidad de volver a calcular cada parámetro. Para ello, se selecciona la opción *Revise Existing File* del menú *File* y aparece una ventana emergente que permite al usuario seleccionar un archivo de la memoria del ordenador. Ese archivo se carga en el espacio de trabajo y se extraen las distintas variables que contiene para representarlas. Así, se representa el conjunto de datos, registrados en una sesión anterior, en las gráficas destinadas a las variables EEG, ECG, BIS, BSR, Relative β -Ratio, BIS-EMG y 95% SEF. Además, se representa el porcentaje promedio de señal en cada una de las bandas de frecuencias a lo largo de la sesión, mediante un diagrama de barras, y el espectrograma de frecuencias de toda la señal de EEG.

En cambio, dado que esta opción ha sido diseñada únicamente para revisar un registro previo, no se pueden realizar modificaciones sobre las incidencias, el identificador del paciente, o los periodos de ELI, hiperventilación y apertura ocular, por lo que se ocultan los elementos destinados a ello. Asimismo, para no guardar accidentalmente dos veces el mismo registro, también se oculta el botón de parada *Stop*. Por otro lado, el BIS únicamente se representa gráficamente con el fin de poder apreciar cómo ha evolucionado en función del tiempo durante el registro, por lo que el panel destinado a mostrar su valor instantáneo se hace invisible.

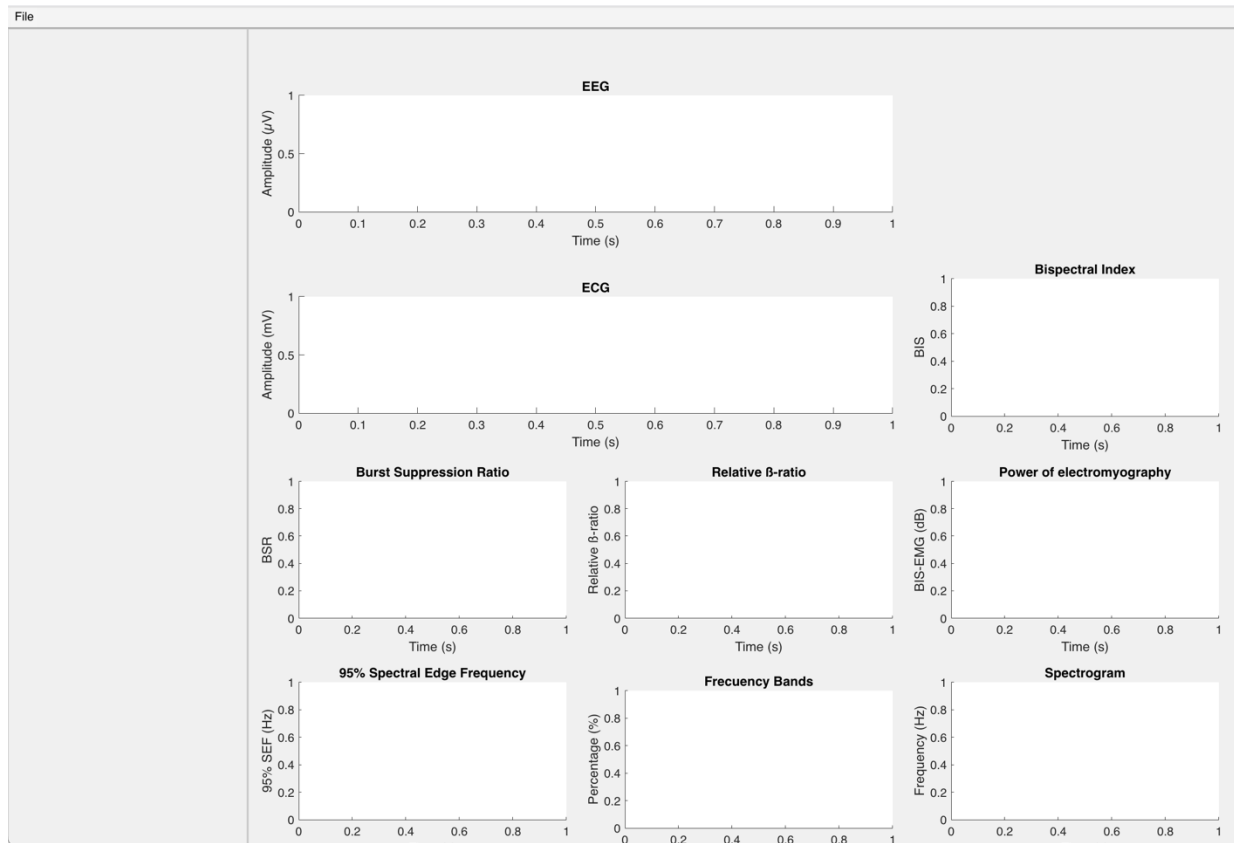


Figura 29. Vista de la interfaz de usuario para la revisión de un registro previo.

Finalmente, si se quiere estudiar el nivel de consciencia que tuvo previamente un paciente durante un registro rutinario de EEG, se selecciona la opción *Import External File* del menú *File* (Figura 30). En esta situación, se entiende que únicamente se dispone de una señal de EEG sin filtrar, por lo que el primer paso es preprocesar esta señal con la función implementada para ello. Durante el preprocesado, se divide la señal en segmentos de 2 s, de manera que se puedan calcular los distintos parámetros en cada uno de los segmentos. Así, se representa, en las distintas gráficas, tanto los subparámetros del BIS como éste. De manera análoga a la opción *Revise Existing File*, se representa el porcentaje promedio de señal en cada una de las bandas de frecuencias a lo largo de la señal, mediante un diagrama de barras, y el espectrograma de frecuencias de todo el EEG.

Asimismo, no es posible anotar incidencias, ni registrar los periodos de ELI, hiperventilación o apertura ocular, por lo que se hacen invisibles estos campos. Además, puesto que únicamente se dispone del EEG, se oculta la gráfica en la que se debería representar el ECG. En cambio, sí que se presenta la opción de almacenar los nuevos parámetros calculados de manera análoga a la opción *New*, por lo que el botón de parada *Stop* sí que está visible y, al pulsarlo, da la opción de guardar el registro. Así, se mantiene visible el campo de texto para la introducción del identificador del paciente *Patient ID*, con el fin de que se pueda identificar correctamente el archivo, una vez almacenado.

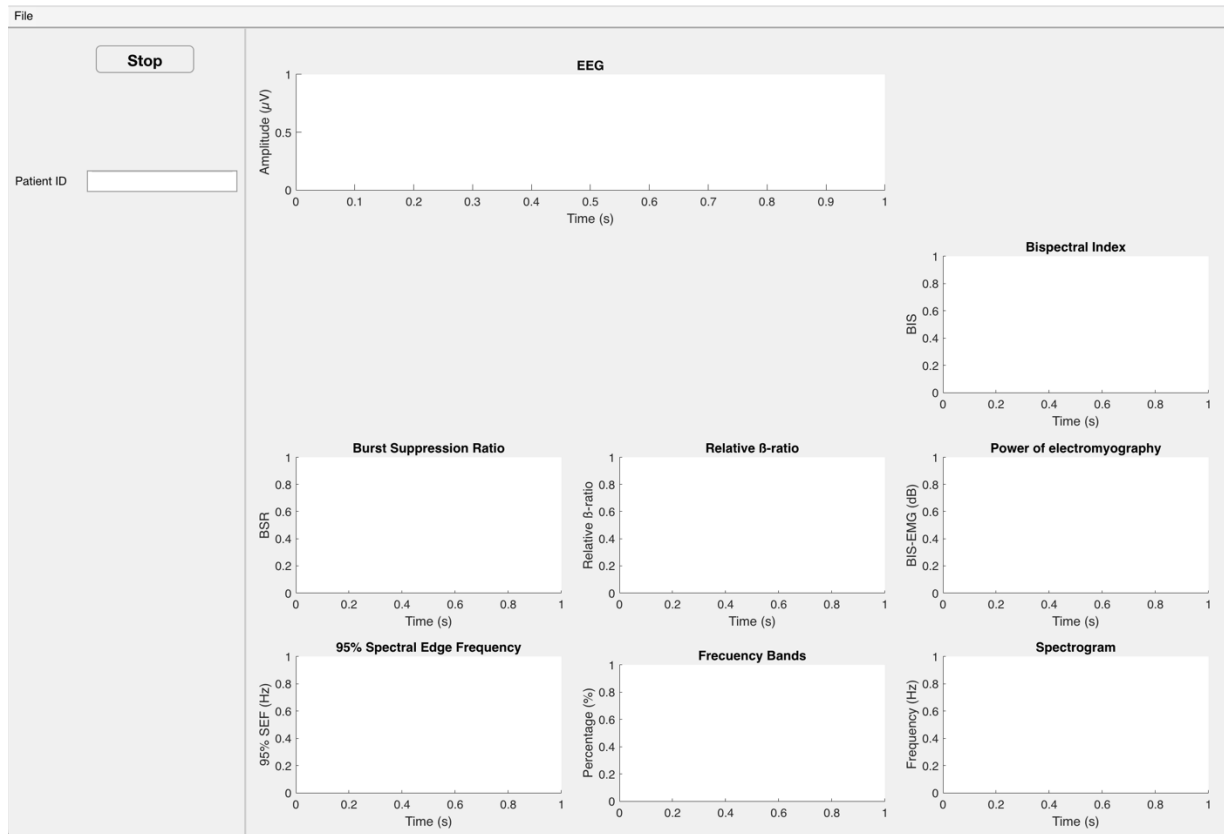


Figura 30. Vista de la interfaz de usuario para la importación de un registro de EEG externo.

5.2. Comprobación del funcionamiento

Una vez diseñada la interfaz de usuario, se han realizado pruebas para comprobar el funcionamiento del sistema implementado. Debido a cuestiones prácticas, no se ha tenido acceso a pacientes con distintos niveles de consciencia sobre los que probar el sistema. Por tanto, únicamente se han llevado a cabo registros en pacientes despiertos y sin ningún tipo de afectación en el cerebro.

En la Figura 31, se puede observar que el sistema es capaz de registrar las señales de EEG y ECG en tiempo real y calcular los distintos parámetros de interés a medida que los representa en las respectivas gráficas. Además, se ha comprobado que todos los botones funcionan correctamente, registrando los periodos de ELI, hiperventilación y apertura ocular, e introduciendo las incidencias. Por otro lado, en la Figura 32, puede observarse el contenido de la variable almacenada tras indicar que se quiere guardar el registro. Así, todos los parámetros se han guardado correctamente en el disco duro del ordenador. Por otro lado, en la Figura 33, se observa lo que sucede si se trata de revisar el contenido de una sesión previa. Esto es, todos los parámetros se han representado correctamente en sus respectivas gráficas. Por último, cuando se importa un EEG externo, la Figura 34 muestra cómo se calculan los distintos parámetros para cada segmento de señal de 2 s y se representan correctamente.

Por tanto, se ha diseñado un sistema funcional, capaz de registrar señales de EEG y ECG y estimar el nivel de consciencia del paciente.

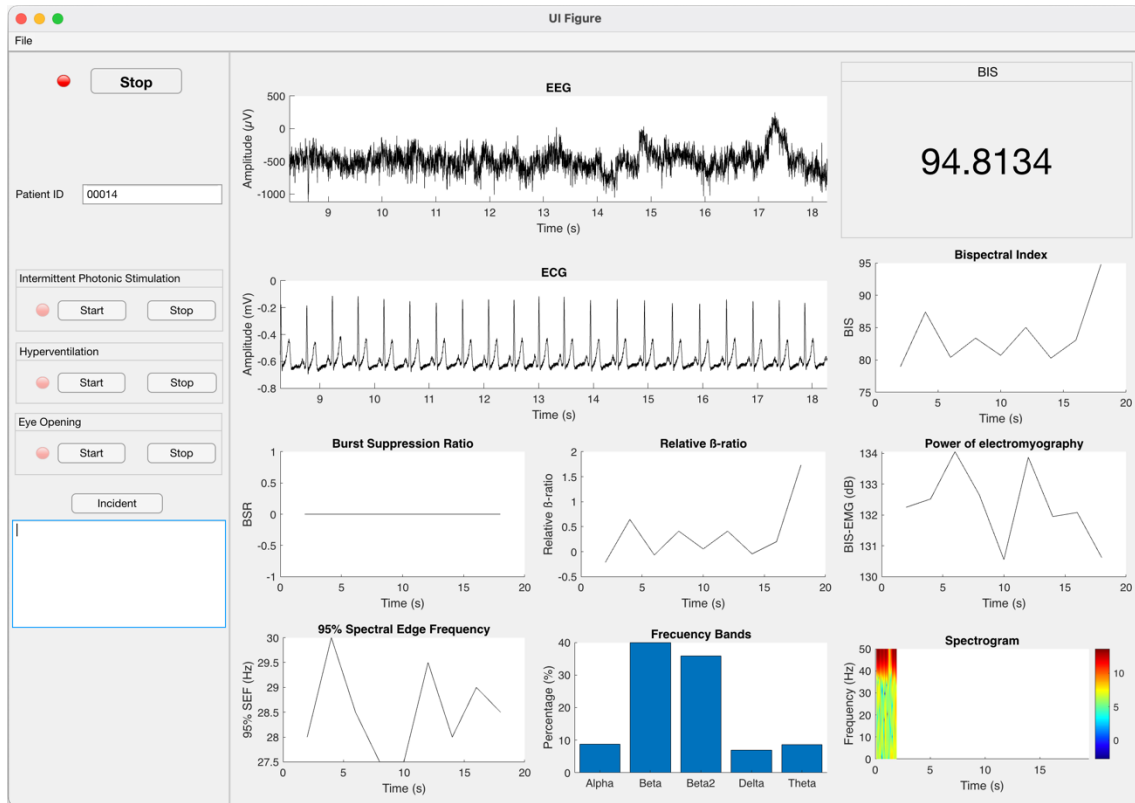


Figura 31. Captura de pantalla de la interfaz de usuario registrando en tiempo real.

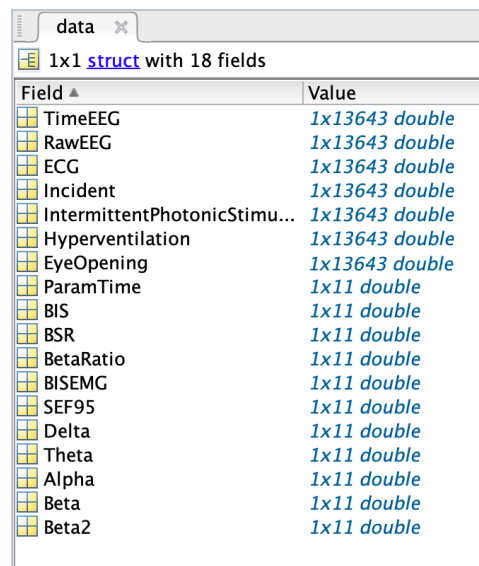


Figura 32. Captura de pantalla del contenido de la variable almacenada.

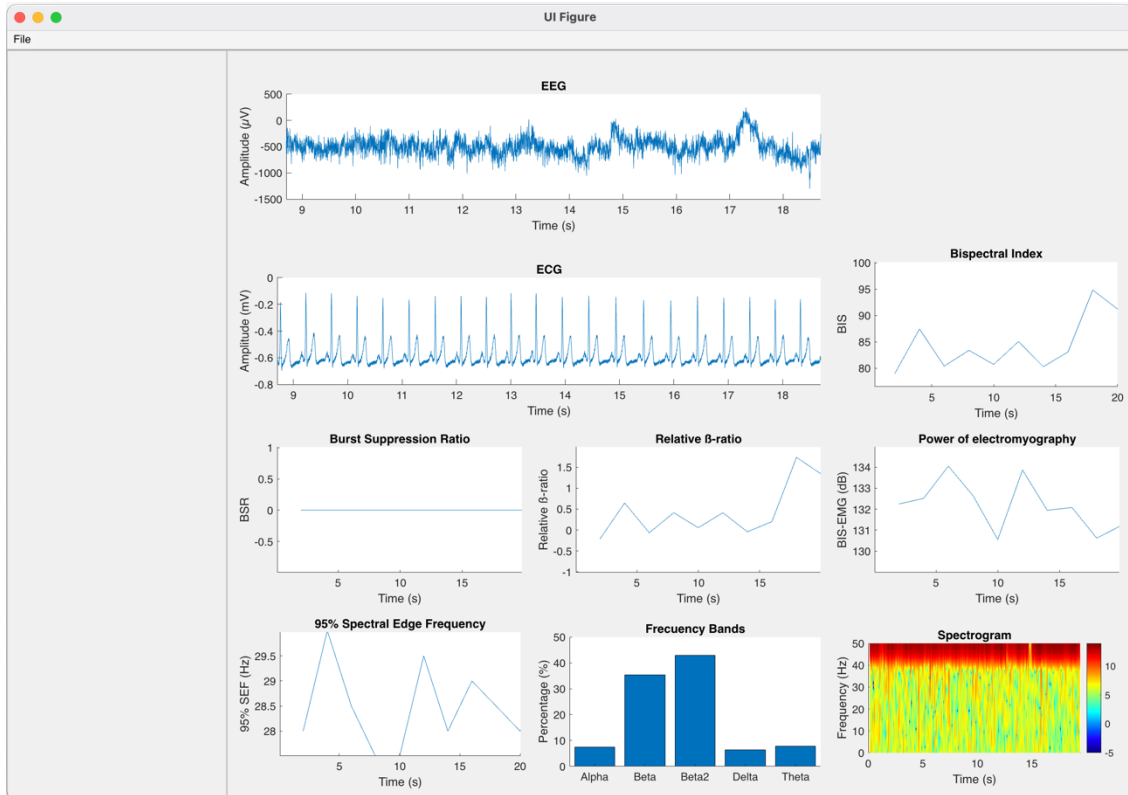


Figura 33. Captura de pantalla de la interfaz de usuario al revisar una sesión previa.



Figura 34. Captura de pantalla de la interfaz de usuario al importar un EEG externo.

5.3. Comparación con un dispositivo comercial

El sistema para estimación del nivel de consciencia implementado en el presente proyecto se ha basado en el monitor BIS comercial del grupo Medtronic, y, como se ha comentado previamente, se ha desarrollado este proyecto en base a las recomendaciones de la Dra. Mas. Debido a cuestiones prácticas, no pueden compararse los resultados proporcionados por ambos sistemas, ya que el equipo comercial no permite el almacenamiento de la señal de EEG registrada durante la sesión. Por tanto, a continuación, se realiza una comparación entre los dos equipos a nivel de hardware y de interfaz de usuario.

En primer lugar, el equipo de monitorización de Medtronic (Figura 35) recibe la señal a través del cable de interfaz del paciente (4 en la Figura 35), la transmite a la unidad BISX™ (5 en la Figura 35), donde se filtra y procesa la señal de EEG del paciente y se transmite al monitor BIS (2 en la Figura 35) a través del cable de interfaz del monitor (1 en la Figura 35). El monitor está diseñado de manera que el acceso a todos los controles puede realizarse tocando una zona concreta de la pantalla, es decir, una tecla táctil, y funciona incluso si el usuario lleva puestos guantes de exploración. El equipo es portátil y puede desplazarse con facilidad aunque, generalmente, se coloca en un carro para evitar caídas accidentales que puedan dañarlo. Además, los datos pueden exportarse para almacenarlos en una memoria USB, o directamente a un ordenador a través de un puerto serie (Medtronic, 2020).



Figura 35. Sistema de monitorización completa BIS (Medtronic, 2020).

En cambio, el sistema implementado en el presente proyecto requiere una conexión a ordenador a través de puerto serie para poder registrar los datos. Esto es, al no disponer de un monitor propio al equipo, es necesario tanto visualizar la interfaz de usuario, como almacenar el registro, directamente a través del monitor de un ordenador. Por tanto, a pesar de que el equipo hardware implementado es más ligero y puede transportarse con mayor facilidad, resulta indispensable llevar consigo un ordenador portátil, o disponer de un ordenador allí donde se quiera realizar un registro.

Por otro lado, la interfaz de usuario del equipo de Medtronic se presenta en la Figura 36. Los distintos elementos que la componen se enumeran a continuación (Medtronic, 2020):

1. Tecla táctil de comprobación del sensor: permite comprobar si los electrodos están colocados correctamente.

2. Tecla táctil para acceder al menú.
3. Tecla táctil de silencio/pausa de alarma: la alarma suena cuando el valor del BIS se sale del rango objetivo.
4. Etiquetas de la unidad de tendencias del BIS.
5. Nombre de la variable principal.
6. Valor del BIS: se muestra y se actualiza continuamente.
7. Indicador de calidad de la señal: medida de la calidad de la señal para la fuente del canal de EEG, que se calcula a partir de los datos de impedancia, las interferencias y otras variables.
8. Indicador de EMG: muestra la potencia (en dB) en el intervalo de frecuencias de 70 a 110 Hz.
9. Forma de onda del EEG.
10. ID del caso: se asigna un nuevo ID único cada vez que se conecta un sensor al sistema.
11. Variable secundaria: es posible visualizar una variable secundaria, seleccionándola desde el menú (puede ser la tasa de supresión, la potencia de EMG, la calidad de la señal o el tiempo de supresión).
12. Hora y fecha actuales.
13. Rango objetivo: se puede establecer un rango objetivo, que se muestra como una barra de color o dos líneas horizontales que muestran los límites.
14. Etiquetas de la unidad de la variable secundaria.
15. Escala de tiempo de la tendencia.
16. Tendencia secundaria.
17. Marcador de eventos instantáneos.
18. Tendencia del BIS.
19. Barra de interferencias.
20. Flecha de revisión: sirve para desplazarse lateralmente y visualizar los valores previos.

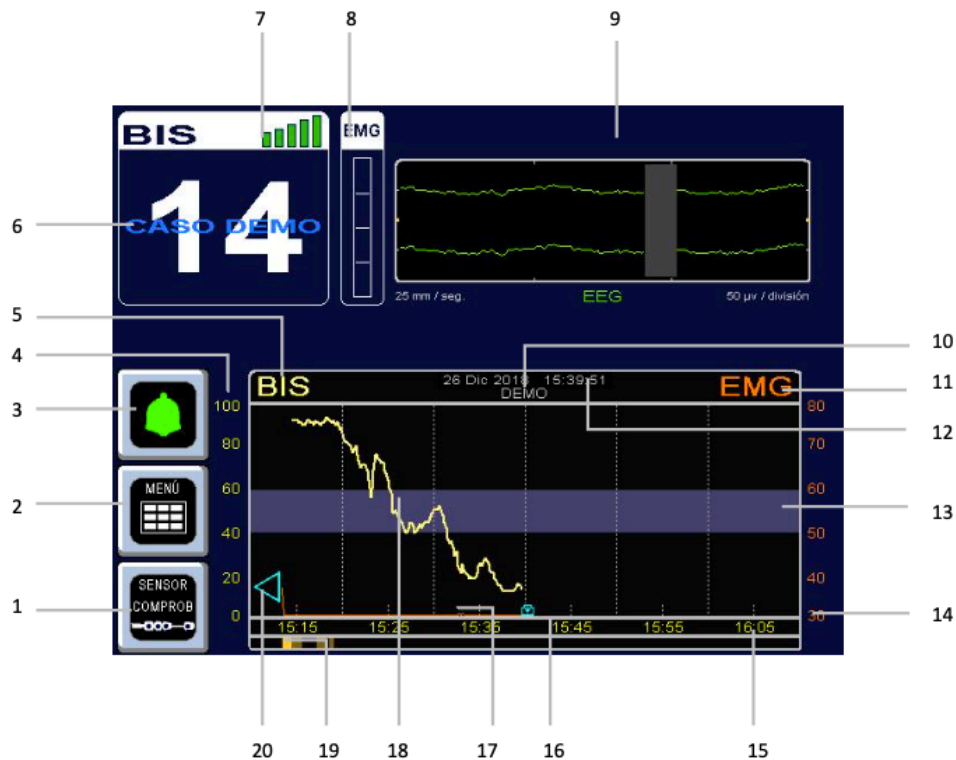


Figura 36. Características de la pantalla de datos de tendencias del monitor BIS (Medtronic, 2020).

Así, la interfaz de usuario de este equipo comercial es compleja y requiere que el usuario interactúe con ella para seleccionar qué variable desea visualizar, con el inconveniente de que únicamente puede ver simultáneamente dos de ellas: la principal y la secundaria (Medtronic, 2020). Esta limitación se ha solventado en la interfaz de usuario diseñada en este proyecto, ya que se presentan los distintos parámetros en una única pantalla, sin necesidad de ir cambiando las opciones continuamente. Esto supone una gran ventaja respecto a los equipos comerciales y resulta más cómodo e intuitivo para el usuario.

Por otro lado, en el equipo del grupo Medtronic, el identificador del paciente se asigna automáticamente en cada sesión de registro. Se trata de un código único que no puede ser elegido por el usuario (Medtronic, 2020). Por tanto, no se puede asociar un registro a un número de historia clínica de paciente para localizarlo más fácilmente, sino que el usuario debe modificarlo manualmente desde el propio ordenador una vez almacenada la sesión. Esto se ha solucionado en la interfaz de usuario diseñada, pues es el propio usuario el que debe introducir el ID del paciente. En función del contexto en que se realice el registro, puede elegir introducir el número de historia clínica del paciente, el nombre completo, o un identificador anónimo. Los registros son fácilmente identificables una vez almacenados, pues en el propio nombre del archivo se incluye tanto el identificador seleccionado, como la fecha y hora en que se realizó el registro.

Además, de acuerdo a las especificaciones de la Dra. Mas, se han incluido funcionalidades que no están presentes en el dispositivo comercial y que, sin embargo, resultan de gran utilidad en el día a día. Entre ellas, se encuentra la posibilidad de visualizar el ECG simultáneamente para poder correlacionar ambas señales y poder interpretar más fácilmente los resultados. Por otro lado, el dispositivo comercial no ofrece la posibilidad de almacenar la señal de EEG registrada a lo largo de la sesión, sino que

únicamente se almacenan los distintos parámetros calculados. Esto supone una gran limitación a la hora de revisar una sesión anterior, ya que únicamente podrían volver a estudiarse estos parámetros y no se podrían relacionar los resultados con la señal de EEG. Asimismo, esto impide realizar análisis adicionales de la señal de EEG, limitando cada estudio a los resultados que proporciona el monitor. Por tanto, una de las prioridades del sistema implementado en este proyecto ha sido ofrecer la posibilidad de almacenar tanto las señales de EEG y ECG, como cada uno de los parámetros calculados, con el fin de poder realizar posteriormente las acciones que el usuario considere oportunas.

5.4. Limitaciones

A pesar de la utilidad y el potencial del sistema implementado para la monitorización del nivel de consciencia, tanto el parámetro del BIS, como el sistema en sí, presentan diversas limitaciones que se exponen a continuación.

Bajo ciertas circunstancias, los valores obtenidos del BIS pueden no reflejar el estado hipnótico concreto en que se encuentra el paciente. Los monitores BIS únicamente constituyen un sistema de ayuda a la decisión clínica, pero en ningún caso la pueden sustituir. Esto es, los profesionales sanitarios deben ser capaces de identificar y responder ante situaciones en las que el BIS pueda no representar con exactitud el nivel de consciencia. Por ejemplo, valores superiores a 60 pueden producirse como resultado de artefactos externos, algunos agentes farmacológicos, u otras causas, más que reflejar la consciencia del paciente. Asimismo, valores de BIS menores a 40 pueden desarrollarse como consecuencia de condiciones clínicas graves, y no únicamente por el estado hipnótico del paciente. Alteraciones en el estado fisiológico pueden reducir el metabolismo cerebral (hipotensión, hipoxia, hipotermia o isquemia), lo que conllevaría una reducción del valor del BIS (Medtronic, 2019).

Así, diversos factores pueden afectar al valor del BIS. Un excesivo tono muscular en los músculos faciales puede resultar en un incremento del valor del BIS, debido a artefactos del electromiograma (EMG). En cambio, la administración de agentes bloqueantes neuromusculares puede disminuir el valor del BIS, aunque estos efectos no son apreciables una vez se estabiliza la anestesia. Por otro lado, pueden aparecer artefactos electromecánicos que, bajo ciertas condiciones, pueden incrementar el valor del BIS. Esto puede producirse, por ejemplo cuando el paciente tiene implantado un marcapasos o cuando se están utilizando otro tipo de dispositivos electrónicos sobre el paciente. Además, estados anormales del paciente, como estado postictal, demencia, lesiones cerebrales severas o muerte cerebral, pueden disminuir el valor del BIS, mientras que, durante crisis epilépticas, puede registrarse un valor incrementado de este parámetro (Medtronic, 2019).

De manera análoga, el valor del BIS puede verse afectado con la administración de ciertos fármacos anestésicos (Medtronic, 2019):

- La ketamina puede incrementar temporalmente el BIS debido a una activación del EEG.
- El etomidato puede incrementar temporalmente el BIS.
- La administración de halotano resulta en valores de BIS superiores que los obtenidos con isoflurano o sevoflurano.

- El óxido nitroso tiene un efecto mínimo en los valores de BIS.
- La efedrina puede incrementar el valor del BIS.

Por tanto, cabe destacar la importancia de considerar estas condiciones cuando se evalúen valores anómalos de BIS.

Por otra parte, se han hallado dificultades respecto al desarrollo del software de obtención del BIS, pues el algoritmo completo para calcularlo no ha sido publicado. Concretamente, no ha sido posible diseñar una función para el cálculo el subparámetro "QUAZI". Asimismo, se desconocen los coeficientes de las diferentes ecuaciones de regresión empleadas en el algoritmo de obtención del BIS. Por ello, se ha tenido que emplear una metodología distinta para su estimación. Así, los resultados obtenidos por el sistema implementado en este proyecto suponen una aproximación teórica al valor que se obtendría con un equipo comercial, por lo que no son completamente fiables. Los profesionales sanitarios que tuvieran que analizar los resultados obtenidos deberían examinar cuidadosamente cada uno de los subparámetros calculados y valorar si el valor de BIS es coherente con las características extraídas de la señal de EEG y el nivel aparente de consciencia del paciente.

Además, en un principio, se pretendía valorar la correlación existente entre los resultados del equipo comercial y el sistema implementado en este proyecto. Para ello, se trató de extraer las señales de EEG obtenidas del estudio dirigido por la Dra. Mas, junto con los parámetros calculados, con el fin de introducir las señales en el nuevo sistema y estimar el nivel de consciencia en base al software desarrollado. A continuación, se compararían los valores de BIS reales, obtenidos del equipo comercial, con los experimentales, calculados por el nuevo sistema, de manera que podría determinarse el error cometido. No obstante, se han encontrado limitaciones en el acceso a los datos de los pacientes para realizar esta comparación, ya que el equipo comercial del grupo Medtronic no tiene la capacidad de almacenar el EEG registrado, sino que únicamente proporciona los parámetros calculados a lo largo de la sesión. Así, la comparativa con respecto al equipo comercial únicamente ha podido realizarse a nivel de hardware e interfaz de usuario.

6. Conclusiones

En conclusión, en el presente proyecto se ha logrado implementar un sistema hardware de bajo coste completamente funcional, capaz de registrar tanto EEG, como ECG y transmitirlos al ordenador a través de un puerto serie. Para ello, se han empleado placas electrónicas comerciales, ya que su función se adapta a los requisitos de este proyecto, presentan un bajo coste, y se facilita la reproducibilidad del sistema. No obstante, cabe destacar que se trata de un prototipo y, a nivel de hardware, presenta diversas carencias en comparación a los equipos comerciales, como la falta de un monitor que evite depender de la disponibilidad de un ordenador en el momento de realizar el registro. Así, en este sentido, se debería optimizar el dispositivo de cara a extender su uso por parte de profesionales sanitarios.

Por otro lado, se ha diseñado un software de análisis electroencefalográfico para estimación del nivel de consciencia, basado en el BIS. Durante su desarrollo, se han hallado ciertas dificultades debido a que el algoritmo completo para calcular este parámetro no ha sido publicado. No obstante, se ha logrado implementar un sistema para estimarlo. Además, se han determinado qué variables resultan de interés para los profesionales sanitarios en la monitorización del nivel de consciencia y se ha diseñado una interfaz de usuario de fácil manejo, intuitiva y accesible que permite representar simultáneamente todas ellas. Asimismo, se ha buscado que ésta sea útil para los profesionales sanitarios tanto en el contexto de monitorización de la profundidad de la anestesia, para lo que han sido diseñados los equipos comerciales, como para investigación en otras áreas no relacionadas con la anestesia.

Así, se ha implementado un sistema funcional de monitorización del nivel de consciencia con un gran potencial para ayudar a la decisión médica. Por una parte, respecto a la monitorización de la anestesia, el uso de este tipo de dispositivos supone una guía para determinar el nivel de hipnosis óptimo en que se debe encontrar un paciente, y así administrar la dosis adecuada de fármacos anestésicos. Por tanto, se evitan riesgos innecesarios relacionados con una dosis excesiva o insuficiente de fármacos y se proporciona una mejor atención médica a los pacientes.

Por otra parte, el sistema implementado presenta un gran potencial en el ámbito de la investigación. Esto es, además de proporcionar el nivel de consciencia mediante el valor del BIS, se han incluido parámetros de interés para los profesionales sanitarios, incluyendo variables específicas de los estudios de electroencefalografía, ausentes en los dispositivos comerciales de monitorización del BIS. Así, se ha buscado facilitar la interpretación de los resultados por parte de los usuarios, ofreciéndoles las herramientas necesarias para comprender los valores de BIS calculados por el software diseñado.

En definitiva, se ha logrado cumplir con los objetivos establecidos al inicio del proyecto y se ha implementado un sistema, de bajo coste y código abierto, para la estimación del nivel de consciencia, con el potencial de ayudar a los profesionales sanitarios.

7. Líneas futuras

Existen varias líneas de desarrollo futuro que, por cuestiones prácticas y de tiempo, no han podido ser abarcadas en el presente proyecto. En primer lugar, no se ha podido evaluar la robustez de los resultados proporcionados por el sistema implementado. Por tanto, de cara al futuro, sería necesario hallar una manera de compararlos con los de un dispositivo comercial. Así, se debería implementar un protocolo para registrar el EEG durante la sesión de monitorización del BIS, y poder introducir esa señal en el sistema desarrollado en el presente proyecto. Además, únicamente se ha podido comprobar su funcionamiento en individuos despiertos y sin ninguna afectación cerebral, por lo que se plantea probar el sistema en un mayor grupo de pacientes, con distintos niveles de consciencia y evaluar el rendimiento del sistema para cada uno de estos niveles.

Por otro lado, se ha visto que, a nivel de hardware, el sistema implementado presenta diversas carencias en comparación a los equipos comerciales. Por tanto, podría implementarse un dispositivo con un diseño hardware optimizado y a medida. Un aspecto importante sería incluir un monitor propio al dispositivo, que permita la interacción con la interfaz de usuario y la visualización de los resultados. Así, se podría elegir si conectar el sistema a un ordenador o no, de manera que se facilite el transporte del dispositivo. Además, se incluiría la posibilidad de almacenar los datos registrados en una memoria USB externa, en aquellos casos en los que se decidiera no conectar el dispositivo al ordenador.

Asimismo, la plataforma Matlab® es muy productiva para plantear prototipos aunque de cara a desarrollar una versión más eficiente del software, se plantea generar una versión compilada de la interfaz de usuario. De este modo, los usuarios finales, es decir, los profesionales sanitarios, podrían ejecutar la aplicación de manera gratuita, sin que se requiera una licencia de Matlab®.

Además, con el fin de que la interfaz de usuario resulte de mayor utilidad para los usuarios y ayudarles más en lo que respecta a la interpretación de los resultados proporcionados, se podría incluir un mayor número de variables de interés. Así, se valoraría la opción de hallar otras variables que pudieran presentar cambios significativos ante variaciones en el nivel de consciencia del paciente. Por ejemplo, se podría medir la frecuencia cardiaca en base a la señal de ECG ya registrada, la presión arterial, la saturación de oxígeno o incluso la conductancia de la piel. Esto es, se buscaría desarrollar un monitor multiparamétrico que ayudara a comprender en mayor medida lo que sucede a nivel fisiológico para los diferentes niveles de consciencia.

Finalmente, se podría investigar más acerca del parámetro BIS y concretamente, cómo se obtiene y cómo varía ante cada tipo de artefacto. De este modo, se podría tratar de desarrollar un sistema que lo vuelva más resistente a estos artefactos. Por ejemplo, para evitar los cambios en el BIS debidos a la administración de ciertos fármacos, sería interesante estudiar cómo se producen estas variaciones, y desarrollar un sistema que, al indicar los fármacos administrados, elimine el artefacto producido, presentando en pantalla el valor real de BIS.

Bibliografía

- Abenstein, J. P. (2009). Is BIS monitoring cost-effective? *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 7041-7044.
- Alwardt, C. M., Redford, D., & Larson, D. F. (2005). General anesthesia in cardiac surgery: a review of drugs and practices. *The journal of extra-corporeal technology*, 37(2), 227-235.
- Analog Devices. (2013). *Data Sheet AD8232*.
- Arduino. (2021a). *Arduino Uno Rev3*.
- Arduino. (2021b). *What is Arduino?* Recuperado 21 de junio de 2021, de <https://www.arduino.cc/en/Guide/Introduction>
- Botland. (s. f.). *Sensor for measuring heart activity - AD8232*. Recuperado 21 de junio de 2021, de <https://botland.store/medical-sensors/2654-sensor-for-measuring-heart-activity-ad8232-sparkfun-sen-12650.html>
- Burr Brown. (1998). *INA114 Data Sheet*.
- Chen, I., & Lui, F. (2021). Neuroanatomy, Neuron Action Potential. *StatPearls*.
- Clayton, B. L. L., & Popko, B. (2016). Endoplasmic reticulum stress and the unfolded protein response in disorders of myelinating glia. *Brain research*, 1648(Pt B), 594-602.
- Comepa Industries. (2020). *Diagnostic and monitoring electrodes*. Recuperado 21 de junio de 2021, de <https://www.comepa.com/copie-de-gamme-monitoring>
- Córdova, F. (2013). *Desarrollo de un sistema para la evaluación de la actividad muscular mediante electrodos de superficie*.
- Ferrero de Loma-Osorio, J. M. (2019a). *Fundamentos bioeléctricos del potencial de acción*. Universitat Politècnica de València.
- Ferrero de Loma-Osorio, J. M. (2019b). *Propagación del potencial de acción en el sistema nervioso*. Universitat Politècnica de València.
- Giuliodori, M. J., & DiCarlo, S. E. (2004). MYELINATED VS. UNMYELINATED NERVE CONDUCTION: A NOVEL WAY OF UNDERSTANDING THE MECHANISMS. *Advances in Physiology Education*, 28(2), 80-81.
- Gruenewald, M., Harju, J., Preckel, B., Molnár, Z., Yli-Hankala, A., Roskopf, F., Koers, L., Orban, A., & Bein, B. (2021). Comparison of adequacy of anaesthesia monitoring with standard clinical practice monitoring during routine general anaesthesia. *European Journal of Anaesthesiology*, 38(1), 73-81.
- Hensley, F. A., Martin, D. E., & Gravlee, G. P. (2013). *A practical approach to cardiac anesthesia* (5.^a ed.). Lippincott Williams & Wilkins.
- Izaurieta, F., & Saavedra, C. (s. f.). *Redes Neuronales Artificiales*.
- Johansen, J. W. (2006). Update on bispectral index monitoring. *Best Practice and Research: Clinical Anaesthesiology*, 20(1), 81-99.
- Keenan, S. A., Carrillo, O., & Casseres, H. (2013). Electroencephalography. *Encyclopedia of Sleep*, 66-70.

- Kohrs, R., & Durieux, M. E. (1998). Ketamine: Teaching an Old Drug New Tricks. *Anesthesia & Analgesia*, 87(5), 1186-1193.
- Lee, H. C., Ryu, H. G., Park, Y., Yoon, S. Bin, Yang, S. M., Oh, H. W., & Jung, C. W. (2019). Data Driven Investigation of Bispectral Index Algorithm. *Scientific Reports*, 9(1), 13769.
- Longnecker, D. E., & Cheung, A. T. (1998). Pharmacology of inhalational anesthetics. En *Principles and Practice of Anesthesiology* (2.^a ed., p. 1123). Mosby.
- Louis, E. K. St., Frey, L. C., Britton, J. W., Frey, L. C., Hopp, J. L., Korb, P., Koubeissi, M. Z., Lievens, W. E., Pestana-Knight, E. M., & Louis, E. K. St. (2016). Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants. *American Epilepsy Society*.
- Medtronic. (2019). *Monitoring consciousness using the Bispectral Index™ (BIS™) brain monitoring system*.
- Medtronic. (2020). *Sistema de monitorización completa BIS - Manual de usuario*.
- MikroElektronika. (2021). Recuperado 16 de junio de 2021, de *EEG Click*. <https://www.mikroe.com/eeg-click>
- Morison, D. H. (1993). New iv induction anaesthetics. *Canadian journal of anaesthesia = Journal canadien d'anesthesie*, 40(5 Pt 2).
- Neves, A. C., González, I., Leander, J., & Karoumi, R. (2018). A New Approach to Damage Detection in Bridges Using Machine Learning. En *Lecture Notes in Civil Engineering* (pp. 73-84). Springer, Cham.
- Perez-Cruz, C., Müller-Keuker, J. I. H., Heilbronner, U., Fuchs, E., & Flügge, G. (2007). Morphology of pyramidal neurons in the rat prefrontal cortex: lateralized dendritic remodeling by chronic stress. *Neural plasticity*, 2007(46276).
- Pugsley, M. K. (2002). The diverse molecular mechanisms responsible for the actions of opioids on the cardiovascular system. *Pharmacology & therapeutics*, 93(1), 51-75.
- Rampil, I. J. (1998). A Primer for EEG Signal Processing in Anesthesia. *Anesthesiology*, 89(4), 980-1002.
- Reta, J. M. (s. f.). *Principios de Electroencefalografía*. Recuperado 19 de junio de 2021, de http://www.sase.com.ar/2012/files/2012/09/EEG_PETAC_publicar.pdf
- Rojas, G. M., Alvarez, C., Montoya, C. E., de la Iglesia-Vayá, M., Cisternas, J. E., & Gálvez, M. (2018). Study of Resting-State Functional Connectivity Networks Using EEG Electrodes Position As Seed. *Frontiers in neuroscience*, 12(235).
- Russo, H., & Bressolle, F. (1998). Pharmacodynamics and pharmacokinetics of thiopental. *Clinical Pharmacokinetics*, 35(2), 95-134.
- Schmidt, H., & Knösche, T. R. (2019). Action potential propagation and synchronisation in myelinated axons. *PLoS computational biology*, 15(10), e1007004.
- Schneider, G., Jordan, D., Schwarz, G., Bischoff, P., Kalkman, C. J., Kuppe, H., Rundshagen, I., Omerovic, A., Kreuzer, M., Stockmanns, G., & Kochs, E. F. (2014). Monitoring Depth of Anesthesia Utilizing a Combination of Electroencephalographic and Standard Measures. *Anesthesiology*, 120(4), 819-828.
- Shepherd, J., Jones, J., Frampton, G. K., Bryant, J., Baxter, L., & Cooper, K. (2013). Clinical effectiveness and cost-effectiveness of depth of anaesthesia monitoring (E-Entropy, Bispectral Index and Narcotrend): A systematic review and economic evaluation. *Health Technology Assessment*,

17(34), 1-264.

Taoufiqi, F. Z., Mounach, J., Satte, A., Ouhabi, H., & El Hessni, A. (2016). IPS Interest in the EEG of Patients after a Single Epileptic Seizure. *Neuroscience journal*, 2016, 5050278.

Tranquilli, W. J., & Grimm, K. A. (2015). Introduction: Use, Definitions, History, Concepts, Classification, and Considerations for Anesthesia and Analgesia. En *Veterinary Anesthesia and Analgesia* (5ª ed., pp. 1-10). John Wiley & Sons, Ltd.

Williams, S., Sah, P., Anggono, V., Lynch, J., & Meunier, F. (2017). *Action potentials and synapses*. Queensland Brain Institute - University of Queensland. Recuperado 16 de junio de 2021, de <https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/action-potentials-and-synapses>

Woodruff, A. (2019). *What is a neuron?* Queensland Brain Institute - The University of Queensland. Recuperado 16 de junio de 2021, de <https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron>

II. MANUAL DE USUARIO

Para utilizar correctamente el sistema implementado en el presente proyecto, el usuario debe seguir los siguientes pasos. Se ha separado cada uno de los tres modos de funcionamiento: nuevo registro, revisión de un registro previo e importación de archivo externo.

1. Nuevo registro

Paso 1: Colocar los electrodos de medida sobre el paciente. Para registrar el EEG, colocar los electrodos según lo indica la Figura 37. Para registrar el ECG, colocar los electrodos según las indicaciones de la Figura 38.

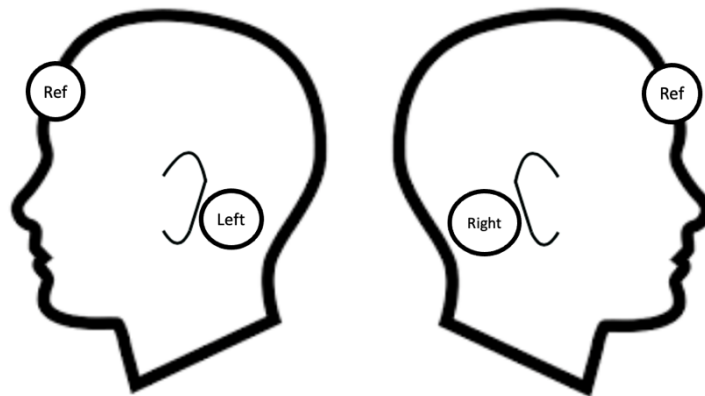


Figura 37. Esquema de la colocación de los electrodos de EEG.

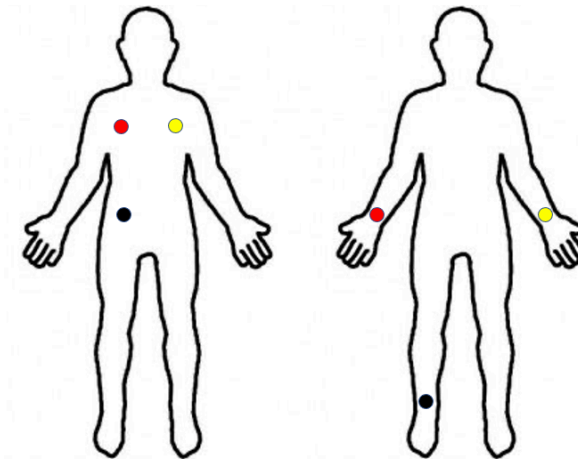


Figura 38. Esquema de la colocación de los electrodos de ECG.

Paso 2: Conectar la placa Arduino Uno al ordenador a través de un puerto USB.

Paso 3: Abrir la aplicación Matlab y escribir el nombre de la interfaz gráfica, sin la extensión “.mat”, en la ventana de comandos.

Paso 4: Seleccionar la opción *New* del menú *File*.

Paso 5: Introducir el puerto serie al que se ha conectado la placa Arduino en el campo de texto disponible para ello, con el formato correspondiente al sistema operativo que se esté utilizando, y

pulsar el botón *OK*, debería encenderse un LED junto al botón *Stop*, en la parte superior izquierda, indicando el comienzo del registro. A continuación se muestran ejemplos del formato a utilizar para la introducción del puerto serie en Mac OS X, Windows y Linux:

- Mac OS X: `/dev/cu.usbmodem1421`
- Windows: COM3
- Linux: `/dev/ttyS0`

Paso 6: Para introducir una incidencia, redactar el contenido en el campo de texto situado en la parte inferior del panel izquierdo y pulsar el botón *Incident*, se guardará automáticamente junto al instante de tiempo en que se pulsó el botón.

Paso 7: Para indicar el inicio de un periodo de ELI, pulsar el botón *Start* situado en el interior del panel *Intermittent Photonic Stimulation*, debería encenderse el LED junto al botón indicando que se está registrando el periodo. Para indicar el final del periodo, pulsar el botón *Stop* situado en el mismo panel, debería apagarse el LED, indicando la finalización del periodo.

Paso 8: Para indicar el inicio de un periodo de hiperventilación, pulsar el botón *Start* situado en el interior del panel *Hyperventilation*, debería encenderse el LED junto al botón indicando que se está registrando el periodo. Para indicar el final del periodo, pulsar el botón *Stop* situado en el mismo panel, debería apagarse el LED, indicando la finalización del periodo.

Paso 9: Para indicar el inicio de un periodo de apertura ocular, pulsar el botón *Start* situado en el interior del panel *Eye Opening*, debería encenderse el LED junto al botón indicando que se está registrando el periodo. Para indicar el final del periodo, pulsar el botón *Stop* situado en el mismo panel, debería apagarse el LED, indicando la finalización del periodo.

Paso 10: Para introducir el identificador del paciente, escribirlo en el campo de texto junto a la etiqueta *Patient ID*. Es necesario rellenar este campo para poder identificar el registro posteriormente.

Paso 11: Para parar el registro, pulsar el botón *Stop*, aparecerá un mensaje indicando que se va a finalizar el registro y preguntando si se quiere almacenar.

Paso 11-A: Si se ha pulsado accidentalmente el botón *Stop* y no se quiere parar el registro, pulsar el botón *Cancel*.

Paso 11-B: Si se quiere finalizar el registro pero no se quiere almacenar, pulsar el botón *No*, se apagará el LED junto al botón *Stop* y se borrará el contenido de las gráficas, indicando el final del registro.

Paso 11-C: Si se quiere finalizar el registro y almacenarlo, pulsar el botón *Yes*, se apagará el LED junto al botón *Stop* y se borrará el contenido de las gráficas, indicando el final del registro.

2. Revisión de un registro previo

Paso 1: Abrir la aplicación Matlab y escribir el nombre de la interfaz gráfica en la ventana de comandos.

Paso 2: Seleccionar la opción *Revise Existing File* del menú *File*, aparecerá una ventana emergente con los archivos de la carpeta actual.

Paso 3: Seleccionar el archivo que se quiera abrir para revisar el registro, aparecerá en las gráficas el contenido del archivo seleccionado.

3. Importación de archivo externo

Paso 1: Abrir la aplicación Matlab y escribir el nombre de la interfaz gráfica en la ventana de comandos.

Paso 2: Seleccionar la opción *Import External File* del menú *File*, aparecerá una ventana emergente con los archivos de la carpeta actual.

Paso 3: Seleccionar el archivo que se quiera abrir con el EEG, aparecerá en la gráfica de EEG la señal importada y se calculará cada uno de los parámetros para representarlos en las demás gráficas.

Paso 4: Para introducir el identificador del paciente, escribirlo en el campo de texto junto a la etiqueta *Patient ID*. Es necesario rellenar este campo para poder identificar el registro posteriormente.

Paso 5: Para finalizar, pulsar el botón *Stop*, aparecerá un mensaje indicando que se va a finalizar el registro y preguntando si se quiere almacenar.

Paso 5-A: Si se ha pulsado accidentalmente el botón *Stop* y no se quiere finalizar, pulsar el botón *Cancel*.

Paso 5-B: Si se quiere finalizar pero no se quiere almacenar el registro, pulsar el botón *No*, se borrará el contenido de las gráficas, indicando el final de la sesión.

Paso 5-C: Si se quiere finalizar y almacenar el registro, pulsar el botón *Yes*, se borrará el contenido de las gráficas, indicando el final de la sesión.

III. ANEJOS

Anejo 1. Código de Arduino

```
/*
  Read the values of the EEG board and the ECG board and send them to the serial
  port of Arduino

  _____
  |                                     |
  |               Programmed by Ana de la Torre               |
  |                   June 2021                               |
  |                                     |
  |_____

*/

float eeg;
float ecg;
void setup() {
  Serial.begin (115200);      // Conection with computer and baud rate specification
  analogReference (DEFAULT);
}

void loop() {
  // EEG
  eeg = analogRead (A0);      // get EEG sample between 0 and 1023
  eeg = eeg*5000/1024;        /* adaptation to voltage input range between 0 and 5 V
                               (EEG signal in  $\mu$ V)
  */
  eeg = eeg - 2500;          // center on 0 V and range between -2,5 V and 2,5 V

  // ECG
  ecg = analogRead(A1);      // get ECG sample between 0 and 1023
  ecg = ecg*3.3/1024;        /* adaptation to voltage input range between 0 and 3,3 V
                               (ECG signal in mV)
  */
  ecg = ecg - 1.65;         // center on 0 V and range between -1,65 V and 1,65 V

  // Print data in decimal format 4 digits
  Serial.print (eeg, 4);
  Serial.print (" ");
  Serial.println (ecg, 4);

  delay (1.4);              // sample period = 1,4 ms => sample rate = 700 Hz
}
```

Anejo 2. Código de AppDesigner

```

classdef TFG_V7 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        GridLayout              matlab.ui.container.GridLayout
        LeftPanel               matlab.ui.container.Panel
        LampStart               matlab.ui.control.Lamp
        PatientIDEditFieldLabel matlab.ui.control.Label
        PatientIDEditField      matlab.ui.control.EditField
        StopRegisteringButton    matlab.ui.control.Button
        StopEEGLabel            matlab.ui.control.Label
        SaveLabel                matlab.ui.control.Label
        YesButton                matlab.ui.control.Button
        NoButton                 matlab.ui.control.Button
        CancelButton            matlab.ui.control.Button
        IPSPanel                 matlab.ui.container.Panel
        StartIPSButton           matlab.ui.control.Button
        StopIPSButton           matlab.ui.control.Button
        LampIPS                  matlab.ui.control.Lamp
        HyperventilationPanel    matlab.ui.container.Panel
        StartHyperButton         matlab.ui.control.Button
        StopHyperButton          matlab.ui.control.Button
        LampHyper                matlab.ui.control.Lamp
        EyeOpeningPanel          matlab.ui.container.Panel
        StartEyeButton           matlab.ui.control.Button
        StopEyeButton            matlab.ui.control.Button
        LampEye                  matlab.ui.control.Lamp
        IncidentButton           matlab.ui.control.Button
        IncidentTextArea         matlab.ui.control.TextArea
        SelectSerialPortPanel    matlab.ui.container.Panel
        xButton                  matlab.ui.control.Button
        SerialPortTextArea       matlab.ui.control.TextArea
        FormatLabel               matlab.ui.control.Label
        FormatLabel2              matlab.ui.control.Label
        MACOSXLabel              matlab.ui.control.Label
        WindowsLabel             matlab.ui.control.Label
        LinuxLabel               matlab.ui.control.Label
        OKButton                 matlab.ui.control.Button
        RightPanel               matlab.ui.container.Panel
        GridLayout1              matlab.ui.container.GridLayout
        UIEEGAxes                matlab.ui.control.UIAxes
        UIBSRAxes                matlab.ui.control.UIAxes
        UIBRatioAxes             matlab.ui.control.UIAxes
        UIBISEMGAxes             matlab.ui.control.UIAxes
        UISEFAxes                matlab.ui.control.UIAxes
        UIBISAxes                matlab.ui.control.UIAxes
        BISPanel                 matlab.ui.container.Panel
        GridLayout2              matlab.ui.container.GridLayout
        BISLabel                  matlab.ui.control.Label
        UIFreqBandAxes           matlab.ui.control.UIAxes
        UISpectrAxes             matlab.ui.control.UIAxes
        UIECGAxes                matlab.ui.control.UIAxes
        FileMenu                 matlab.ui.container.Menu
        NewMenu                   matlab.ui.container.Menu
        ReviseExistingFileMenu    matlab.ui.container.Menu
        ImportExternalFileMenu    matlab.ui.container.Menu
    end

    % Properties that correspond to apps with auto-reflow
    properties (Access = private)
        onePanelWidth = 576;
    end

    properties (Access = public)

        % Internal variables initialization
        PatientID = {};           % Patient's identifier
        time_vector = [];         % Time vector for EEG
        SerialEEG = [];           % EEG
        SerialECG = [];           % ECG
        incident = [];            % Incident comments
        ips = [];                 % IPS periods
        hyperventilation = [];    % Hyperventilation periods
        eye_op = [];              % Eye opening periods
        time_parameters = [];     % Time vector for parameters
        BIS = [];                 % BIS
        bsr = [];                 % BSR
        b_ratio = [];             % Beta-Ratio
        bis_emg = [];             % BIS-EMG
        sef = [];                 % 95% SEF
        delta = [];               % Delta band
        theta = [];               % Theta band
        alpha = [];               % Alpha band
        beta = [];                % Beta band
        beta2 = [];               % Beta2 band
    end
end

```



```

% Buttons
Stop_button = 0;           % Don't Stop
Start_ips = 0;            % Don't Start
Start_hyper = 0;         % Don't Start
Start_eye = 0;           % Don't Start
Write_incident = 0;      % No incident

end

methods (Access = public)

function [eeg, eeg_bis_emg, br_eeg] = eeg_processing(eeg_no_filt, fs)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Inputs:
%   - eeg_no_filt: EEG signal without filtering
%   - fs: sampling frequency (Hz)
%
% This function conducts 3 types of preprocessing depending on
% the parameters that are going to be calculated. To do so,
% first, it applies 3 different filters separately. Then, each
% signal is divided into 2 s epochs and, finally, the Fourier
% Transformation is calculated for each epoch (this step is not
% necessary for the BIS-EMG calculation).
%
% It returns 3 structure array objects, each containing
% the results of filtering, dividing into epochs and
% Fourier Transformation.
%
%   _____
%   |                               |
%   |   Programmed by Ana de la Torre   |
%   |   May 2021                       |
%   |_____                          |
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Filtering for:

% BSR and SEF calculation
% 0.5 Hz highpass filter
eeg_filt_high = highpass(eeg_no_filt, 0.5, fs);

% 70 Hz lowpass filter
eeg_filt_low = lowpass(eeg_filt_high, 70, fs);

% BIS-EMG calculation
% 70 Hz highpass filter
eeg_bis_emg_filt_high = highpass(eeg_no_filt, 70, fs);

% 110 Hz lowpass filter
eeg_bis_emg_filt_low = lowpass(eeg_bis_emg_filt_high, 110, fs);

% Beta ratio calculation
% EEG detrending using the Savitzky Golay filter
br_eeg_filt = sgolayfilt(eeg_no_filt,9,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Division into 2 s epochs
epoch_length = fs * 2; % Length of a 2 s epoch

num_epochs = fix(length(eeg_no_filt)/epoch_length); % Number of epochs
% in the signal
DataLength = epoch_length * num_epochs; % Total length of the signal

% EEG filtered between 0.5 and 70 Hz
eeg_epochs = reshape(eeg_filt_low(1:DataLength), [epoch_length, ...
num_epochs]);

% EEG filtered between 70 and 110 Hz
eeg_bis_emg_epochs = reshape(eeg_bis_emg_filt_low(1:DataLength), ...
[epoch_length, num_epochs]);

% EEG filtered with the Savitzky-Golay filter
br_eeg_epochs = reshape(br_eeg_filt(1:DataLength), [epoch_length, ...
num_epochs]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fourier Transformation (this step isn't necessary for BIS-EMG
% calculation)

% Blackman window
w = ones(int64(num_epochs), int64(epoch_length)) *...
blackman(int64(epoch_length)); % Window
eeg_bw = eeg_epochs .* w; % EEG filtered between
% 0.5 and 70 Hz

```



```

size_window = 0.5 * fs;      % Number of samples in 0.5 s
umbral = 5;                 % 5µV
vector = abs(eeg) <= umbral; % Samples in the EEG that fulfill
                             % the condition

% Vector to store suppression periods
suppression = zeros(size(vector));

% First sample
if vector(1) == 1
    suppression(1) = 1;
end

for i = 2:length(vector)
    % If EEG doesn't exceed ± 5 µV, the number of samples that
    % fulfill the condition following each other are stored
    if vector(i) == 1
        suppression(i) = suppression(i-1) + 1;
        suppression(i-1) = 0;
    end

    % If the duration isn't longer than 0.5 s, the period
    % doesn't count for the BSR
    if suppression(i) == 0 && suppression(i-1) < size_window
        suppression(suppression < size_window) = 0;
    end
end

total_suppression = sum(suppression);

bsr = total_suppression / length(eeg);

end

function SEF = spectral_edge_freq95(eeg_fft,fs)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Inputs:
%   - eeg_fft: Fourier Transformation of the EEG signal
%             filtered between 0.5 and 70 Hz
%   - fs: sampling frequency (Hz)
%
% This function calculates the 95% Spectral Edge Frequency of
% the input signal, as the frequency below which 95% of the
% power in the spectrum resides.
%
% It returns the value of the 95% Spectral Edge Frequency.
%
%   _____
%   |                               |
%   |   Programmed by Ana de la Torre   |
%   |   March 2021                     |
%   |_____                          |
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Power Spectrum
power_spectrum = pspectrum(eeg_fft);

df = fs/length(eeg_fft);
i50 = (50/df)+1; % 50 Hz index

% 95% Spectral Edge Frequency
cum_power = cumsum(power_spectrum(1:i50)); % Cumulative power
                                                % spectrum
cum_power_norm = cum_power / cum_power(end); % Normalized cumulative
                                                % power spectrum
i95 = find(cum_power_norm >= 0.95,1); % 95% index
SEF = (i95-1)*df;

end

function BIS_EMG = EMG_power(eeg_bis_emg)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Inputs:
%   - eeg_bis_emg: EEG signal filtered between 70 and 110 Hz
%
% This function calculates the Power of Electromyography in the
% input signal, as the absolute power in the 70-110 Hz range
%
% It returns the value of the BIS-EMG in dB.
%
%   _____
%   |                               |
%   |   Programmed by Ana de la Torre   |
%   |   March 2021                     |
%   |_____                          |
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Absolute power
absolute_power = sum(eeg_bis_emg.^2);

% Sum of the absolute power from 70 Hz to 110 Hz (in dB)
BIS_EMG = 20*log10(absolute_power);

end

function BIS = BIS_decision_tree(BSR, BIS_EMG, SEF, b_ratio)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Inputs:
%   - BSR: Burst Suppression Ratio
%   - BIS_EMG: Power of Electromyography
%   - SEF: 95% Spectral Edge Frequency
%   - b_ratio: Relative Beta Ratio
%
% This function classifies the signal in one of the 5 BIS ranges
% using a decision tree and the BIS is estimated using multiple
% linear regression models with different weights depending on the
% assigned range.
%
% It returns the value of the BIS.
%
%   _____
%   |                               |
%   |   Programmed by Ana de la Torre   |
%   |   May 2021                       |
%   |_____                          |
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if BSR > 49.8
    % 0<=BIS<=21
    coef = [-0.42 0 0.01 0];
    intercept = 42.1;

else if BIS_EMG < 34.2 && SEF< 20.2
    if BSR > 2.1 || SEF< 14.8
        % 21<BIS<=41
        coef = [-0.42 0.91 3.06 0.04];
        intercept = 29.9;

    else
        % 41<BIS<=61
        coef = [-3.01 3.84 -8.7 0.96];
        intercept = -57.6;

    end

    else if b_ratio < (-0.7)
        % 61<BIS<=78
        coef = [-1.43 2.55 4.26 0.41];
        intercept = 5.3;

    else
        % 78<BIS<=98
        coef = [-1.97 0.88 7.89 -0.07];
        intercept = 65.2;

    end

end

% Linear regression model to calculate BIS
BIS = coef(1) * BSR + coef(2) * SEF + coef(3) * b_ratio
+ coef(4) * BIS_EMG + intercept;

end

function [delta, theta, alpha, beta, beta2] = band(eeg_fft,fs)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Inputs:
%   - eeg_fft: Fourier Transformation of the EEG signal
%             filtered between 0.5 and 70 Hz
%   - fs: sampling frequency (Hz)
%
% This function calculates the percentage of the signal
% corresponding to each frequency band (delta, theta, alpha,
% beta and beta2).
%
% It returns the value of these percentages.
%
%   _____
%   |                               |
%   |   Programmed by Ana de la Torre   |
%   |   March 2021                     |
%   |_____                          |
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Determination of the indexes associated to the limits of the
% frequency bands
df = fs/length(eeg_fft);

i05=round((0.5/df)+1); % 0.5 Hz
i35=round((3.5/df)+1); % 3.5 Hz
i7=round((7/df)+1); % 7 Hz
i13=round((13/df)+1); % 13 Hz
i30=round((30/df)+1); % 30 Hz
i50=round((50/df)+1); % 50 Hz

% Module of Fourier Transformation
module_eeg_fft = abs(eeg_fft);

% Percentage of the signal corresponding to each frequency band
delta = 100 * sum(module_eeg_fft(i05:i35))/sum(module_eeg_fft(i05:i50));
theta = 100 * sum(module_eeg_fft(i35+1:i7))/sum(module_eeg_fft(i05:i50));
alpha = 100 * sum(module_eeg_fft(i7+1:i13))/sum(module_eeg_fft(i05:i50));
beta = 100 * sum(module_eeg_fft(i13+1:i30))/sum(module_eeg_fft(i05:i50));
beta2 = 100 * sum(module_eeg_fft(i30+1:i50))/sum(module_eeg_fft(i05:i50));

end

function killports(Silent)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% KILLPORTS checks for exisiting open serial port objects. In case %
% of opened serial ports, they are closed and the corresponging %
% port object is deleted. Use this function to initialize serial %
% port communication. %
%
% KILLPORTS('Silent') performs the same task without displaying any %
% message. %
%
%          |          Version 1.0          | %
%          | Programmed by J.J. RIETA, February 2016 | %
%          |          jjrieta@upv.es          | %
%          |          |          | %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

OpenPort = instrfindall; % Check for exisiting serial port objects

if ~isempty(OpenPort) % Check for existing open port
    delete(OpenPort); % Delete the existing port objects
    if ~exist('Silent') % Activity message default case
        disp('Existing open ports killed')
    end
else
    if ~exist('Silent')
        disp('No open ports detected')
    end
end
end
end

% Callbacks that handle component events
methods (Access = private)

% Menu selected function: NewMenu
function NewMenuSelected(app, event)
% Make visible SelectSerialPortPanel
app.SelectSerialPortPanel.Visible = 'on';
app.SelectSerialPortPanel.OuterPosition
end

% Menu selected function: ReviseExistingFileMenu
function ReviseExistingFileMenuSelected(app, event)
% Make invisible elements that are not going to be used while
% revising a existing file.
app.IPSPanel.Visible = 'off';
app.HyperventilationPanel.Visible = 'off';
app.EyeOpeningPanel.Visible = 'off';
app.IncidentButton.Visible = 'off';
app.IncidentTextArea.Visible = 'off';
app.LampStart.Visible = 'off';
app.BISPanel.Visible = 'off';
app.StopRegisteringButton.Visible = 'off';
app.PatientIDEditField.Visible = 'off';
app.PatientIDEditFieldLabel.Visible = 'off';

% Clear everything
app.UIEGAXes.cla
app.UIECGAXes.cla
app.UIBSRAXes.cla
app.UIBRatioAxes.cla
app.UIBISEMGAXes.cla
app.UISEFAXes.cla
app.UIFreqBandAxes.cla
app.UISpectrAXes.cla

```

```

app.UIBISAxes.cla
app.PatientIDEditField.Value = '';
app.IncidentTextArea.Value = '';
app.BISLabel.Text = 'BIS';

% Make visible ECG graphic (it could have been made invisible
% previously)
app.UIECGAxes.Visible = 'on';

% Select and load file
[file, path] = uigetfile;
cd(path);
load(file, '-mat');

% Extraction of variables from file
app.time_vector = data.TimeEEG;
app.SerialEEG = data.RawEEG;
app.SerialECG = data.ECG;
app.time_parameters = data.ParamTime;
app.BIS = data.BIS;
app.bsr = data.BSR;
app.b_ratio = data.BetaRatio;
app.bis_emg = data.BISEMG;
app.sef = data.SEF95;
app.delta = data.Delta;
app.theta = data.Theta;
app.alpha = data.Alpha;
app.beta = data.Beta;
app.beta2 = data.Beta2;

% Variables initialization
fs = 700; % Sampling frequency in Hz
ts = 1/fs; % sampling period in s
DataLength = length(app.SerialEEG); % Length of the signal
TotalTime = DataLength * ts;

% Graphical representation
plot(app.UIEEGAxes, app.time_vector, app.SerialEEG);
plot(app.UIECGAxes, app.time_vector, app.SerialECG);
plot(app.UIBSRAxes, app.time_parameters, app.bsr);
plot(app.UIBRatioAxes, app.time_parameters, app.b_ratio);
plot(app.UIBISEMGAxes, app.time_parameters, app.bis_emg);
plot(app.UISEFAxes, app.time_parameters, app.sef);
plot(app.UIBISAxes, app.time_parameters, app.BIS);
bar(app.UIFreqBandAxes, categorical({'Delta', 'Theta', ...
    'Alpha', 'Beta', 'Beta2'}), [mean(app.delta), ...
    mean(app.theta), mean(app.alpha), mean(app.beta), ...
    mean(app.beta2)]);

% Spectrogram
[s,f,t] = pspectrum(app.SerialEEG, fs, 'spectrogram', ...
    'FrequencyLimits', [0, 50]);
imagesc(app.UISpectrAxes, t, flipud(f), log(abs(s)));
set(app.UISpectrAxes, 'YDir', 'normal');
app.UISpectrAxes.YLim = [0, 50];
app.UISpectrAxes.XLim = [0 TotalTime];
colormap(app.UISpectrAxes, 'jet');
colorbar(app.UISpectrAxes, "eastoutside");
    
```

end

```

% Menu selected function: ImportExternalFileMenu
function ImportExternalFileMenuSelected(app, event)
% Make invisible elements that are not going to be used while
% importing an external EEG file.
app.IPSPanel.Visible = 'off';
app.HyperventilationPanel.Visible = 'off';
app.EyeOpeningPanel.Visible = 'off';
app.IncidentButton.Visible = 'off';
app.IncidentTextArea.Visible = 'off';
app.LampStart.Visible = 'off';
app.BISPanel.Visible = 'off';
app.UIECGAxes.Visible = 'off';

% Clear everything
app.UIEEGAxes.cla
app.UIECGAxes.cla
app.UIBSRAxes.cla
app.UIBRatioAxes.cla
app.UIBISEMGAxes.cla
app.UISEFAxes.cla
app.UIFreqBandAxes.cla
app.UISpectrAxes.cla
app.UIBISAxes.cla
app.PatientIDEditField.Value = '';
app.IncidentTextArea.Value = '';
app.BISLabel.Text = 'BIS';

% Select and load file
[file, path] = uigetfile;
    
```

```

cd(path);
load(file, '-mat');

% Extraction of EEG from file
app.SerialEEG = data.RawEEG;

% Variable initialization
fs = 700;           % Sampling frequency in Hz
ts = 1/fs;         % sampling period in s
DataLength = length(app.SerialEEG); % Length of the signal
epoch_length = fs * 2; % Length of a 2 s epoch
TotalTime = DataLength * ts; % Total time
num_epochs = fix(DataLength/epoch_length); % Number of epochs

% EEG preprocessing
[eeg, eeg_bis_emg, br_eeg] = eeg_processing(app.SerialEEG, fs);

% Time vector
app.time_vector = (0:DataLength-1) * ts;
app.time_parameters = (0:num_epochs-1)*2;

% Graphical representation
plot(app.UIEEGAxes, app.time_vector, app.SerialEEG);

k = 1;
for i = 1:DataLength
    if i == k * epoch_length
        % BSR
        app.bsr(k) = bsr_calculation(eeg.epochs(k,:), fs);

        % Beta ratio
        app.b_ratio(k) = b_ratio_calculation(br_eeg.fft(k,:), fs);

        % EMG power
        app.bis_emg(k) = EMG_power(eeg_bis_emg.epochs(k,:));

        % 95% Spectral Edge Frequency
        app.sef(k) = spectral_edge_freq95(eeg.fft(k,:), fs);

        % Estimation of the patient's BIS
        app.BIS(k) = BIS_decision_tree(app.bsr(k), app.bis_emg(k), ...
            app.sef(k), app.b_ratio(k));

        k = k+1;
    else continue
    end
end

plot(app.UIBSRAxes, app.time_parameters, app.bsr);

plot(app.UIBRatioAxes, app.time_parameters, app.b_ratio);

plot(app.UIBISEMGAxes, app.time_parameters, app.bis_emg);

plot(app.UISEFAxes, app.time_parameters, app.sef);

plot(app.UIBISAxes, app.time_parameters, app.BIS);

% Calculation of the signal percentage corresponding to
% each frequency band
[app.delta, app.theta, app.alpha, app.beta, app.beta2] = band(app.SerialEEG, fs);
bar(app.UIFreqBandAxes, categorical({'Delta', 'Theta', 'Alpha', 'Beta', 'Beta2'}), ...
    [app.delta, app.theta, app.alpha, app.beta, app.beta2]);

% Spectrogram
[s, f, t] = pspectrum(app.SerialEEG((k-1) * epoch_length+1:i), ...
    fs, 'spectrogram', 'FrequencyLimits', [0, 50]);
imagesc(app.UISpectrAxes, t, flipud(f), log(abs(s)));
set(app.UISpectrAxes, 'YDir', 'normal');
app.UISpectrAxes.YLim = [0, 50];
colormap(app.UISpectrAxes, 'jet');
colorbar(app.UISpectrAxes, "eastoutside");

end

% Value changed function: PatientIDEditField
function PatientIDEditFieldValueChanged(app, event)
    % Store patient's identifier
    app.PatientID = app.PatientIDEditField.Value;
end

% Button pushed function: StartIPSButton
function StartIPSButtonPushed(app, event)
    % Start IPS and turn on the lamp
    app.Start_ips = 1;
    app.LampIPS.Enable = 'on';
end

% Button pushed function: StopIPSButton
function StopIPSButtonPushed(app, event)
    % Stop IPS and turn off the lamp

```



```

app.Start_ips = 0;
app.LampIPS.Enable = 'off';
end

% Button pushed function: StartHyperButton
function StartHyperButtonPushed(app, event)
% Start Hyperventilation and turn on the lamp
app.Start_hyper = 1;
app.LampHyper.Enable = 'on';
end

% Button pushed function: StopHyperButton
function StopHyperButtonPushed(app, event)
% Stop Hyperventilation and turn off the lamp
app.Start_hyper = 0;
app.LampHyper.Enable = 'off';
end

% Button pushed function: StartEyeButton
function StartEyeButtonPushed(app, event)
% Start Eye Opening and turn on the lamp
app.Start_eye = 1;
app.LampEye.Enable = 'on';
end

% Button pushed function: StopEyeButton
function StopEyeButtonPushed(app, event)
% Stop Eye Opening and turn off the lamp
app.Start_eye = 0;
app.LampEye.Enable = 'off';
end

% Button pushed function: IncidentButton
function IncidentButtonPushed(app, event)
app.Write_incident = 1; % Write Incident
end

% Button pushed function: OKButton
function OKButtonPushed(app, event)
% Make visible elements that are going to be used while
% recording a new EEG and could have been made invisible
% previously
app.BISPanel.Visible = 'on';
app.LampStart.Visible = 'on';
app.LampStart.Enable = 'on';
app.IPSPanel.Visible = 'on';
app.HyperventilationPanel.Visible = 'on';
app.EyeOpeningPanel.Visible = 'on';
app.IncidentButton.Visible = 'on';
app.IncidentTextArea.Visible = 'on';
app.PatientIDEditField.Visible = 'on';
app.PatientIDEditFieldLabel.Visible = 'on';
app.UIECGAxes.Visible = 'on';
app.StopRegisteringButton.Visible = 'on';

% Clear everything
app.UIEEGAxes.cla
app.UIECGAxes.cla
app.UIBSRAxes.cla
app.UIBRatioAxes.cla
app.UIBISEMGAxes.cla
app.UISEFAxes.cla
app.UIFreqBandAxes.cla
app.UISpectrAxes.cla
app.UIBISAxes.cla
app.PatientIDEditField.Value = '';
app.IncidentTextArea.Value = '';
app.BISLabel.Text = 'BIS';

% Restore Stop_button value
app.Stop_button = 0;

% Select the serial port and make invisible
% SelectSerialPortPanel
port = app.SerialPortTextArea.Value;
app.SelectSerialPortPanel.Visible = 'off';

% Check for existing open serial port objects and close them
killports('silent');

% Variable initialization
BaudRate = 115200; % Transmission speed (baud)
fs = 600; % Sampling frequency in Hz
ts = 1/fs; % sampling period in s
epoch_length = fs * 2; % Length of a 2 s epoch

% Starting point of time vectors
app.time_vector(1) = 0;
app.time_parameters(1) = 0;

```



```

% Creation and opening of serial port object
SerialObject = serial(port,'BaudRate',BaudRate);
fopen(SerialObject);

% Graphical representation with animated lines
xini = 0; % initial limit of the x axes
xfin = 10; % final limit of the x axes
EEG = animatedline(app.UIEEGAxes);
app.UIEEGAxes.XLim = [xini,xfin];

ECG = animatedline(app.UIECGAxes);
app.UIECGAxes.XLim = [xini,xfin];

BSR = animatedline(app.UIBSRAxes);

BR = animatedline(app.UIBRatioAxes);

BIS_EMGrep = animatedline(app.UIBISEMGAxes);

SEFrep = animatedline(app.UISEFAxes);

BISrep = animatedline(app.UIBISAxes);

% While the Stop button isn't pushed, the app keeps registering data.
i = 1;
k = 1;
while app.Stop_button == 0

    % Buttons
    % If the incident button is pushed, the text written in the
    % DescribetheincidentTextArea is stored on the exact moment
    % it was pushed
    if app.Write_incident == 1
        app.incident(i) = string(app.IncidentTextArea.Value);
        app.Write_incident = 0;
    else app.incident(i) = ' ';
    end

    % If the Start IPS button is pushed, the vector ips starts
    % filling itself with 1, until the Stop IPS button is
    % pushed
    if app.Start_ips == 1
        app.ips(i) = 1;
    else app.ips(i) = 0;
    end

    % If the Start Hyperventilation button is pushed, the
    % vector hyperventilation starts filling itself with 1,
    % until the Stop Hyperventilation button is pushed
    if app.Start_hyper == 1
        app.hyperventilation(i) = 1;
    else app.hyperventilation(i) = 0;
    end

    % If the Start Eye Opening button is pushed, the
    % vector eye_op starts filling itself with 1, until the
    % Stop Eye Opening button is pushed
    if app.Start_eye == 1
        app.eye_op(i) = 1;
    else app.eye_op(i) = 0;
    end

    % Time vector with ts seconds between samples
    if i>1
        app.time_vector(i) = app.time_vector(i-1) + ts;
    end

    % Read data from SerialObject
    SerialData = str2num(fscanf(SerialObject));
    app.SerialEEG(i) = SerialData(1);
    app.SerialECG(i) = SerialData(2);

    % If the new point fits in the window, it is represented
    % without modifications in the axes limits
    if app.time_vector(i)>=xini && app.time_vector(i)<=xfin
        xlim(app.UIEEGAxes,[xini, xfin]);
        xlim(app.UIECGAxes,[xini, xfin]);

    % If it doesn't fit, the axes limits are updated and the
    % window moves
    else
        xini = xini + ts;
        xfin = xfin + ts;
        xlim(app.UIEEGAxes, [xini, xfin]);
        xlim(app.UIECGAxes, [xini, xfin]);
    end

    % Sequential point representation
    addpoints(EEG,app.time_vector(i),app.SerialEEG(i));
    % Representation update

```

```

drawnow limitrate;

% Sequential point representation
addpoints(ECG,app.time_vector(i),app.SerialECG(i));
% Representation update
drawnow limitrate;

% For each epoch, every parameter is calculated and
% represented as animated lines
if i == k * epoch_length

    % Parameters time vector is updated
    app.time_parameters(k) = app.time_vector(i);

    % Preprocessing of the signal for the last epoch
    [eeg, eeg_bis_emg, br_eeg] = eeg_processing(app.SerialEEG((k-1) ...
        * epoch_length +1:i), fs);

    % BSR
    app.bsr(k) = bsr_calculation(eeg.filtered,fs);
    % Sequential point representation
    addpoints(BSR, app.time_parameters(k), app.bsr(k));
    % Representation update
    drawnow limitrate;

    % Beta ratio
    app.b_ratio(k) = b_ratio_calculation(br_eeg.fft, fs);
    % Sequential point representation
    addpoints(BR, app.time_parameters(k), app.b_ratio(k));
    % Representation update
    drawnow limitrate;

    % EMG power
    app.bis_emg(k) = EMG_power(eeg_bis_emg.filtered);
    % Sequential point representation
    addpoints(BIS_EMGrep, app.time_parameters(k), app.bis_emg(k));
    % Representation update
    drawnow limitrate;

    % 95% spectral edge frequency
    app.sef(k) = spectral_edge_freq95(eeg.fft,fs);
    % Sequential point representation
    addpoints(SEFrep, app.time_parameters(k), app.sef(k));
    % Representation update
    drawnow limitrate;

    % Frequency Bands
    [app.delta(k), app.theta(k), app.alpha(k), app.beta(k), ...
        app.beta2(k)] = band(app.SerialEEG((k-1) * epoch_length+1:i), fs);
    % Representation with bar graph
    bar(app.UIFreqBandAxes, categorical({'Delta', 'Theta', ...
        'Alpha', 'Beta', 'Beta2'}), [app.delta(k), app.theta(k), ...
        app.alpha(k), app.beta(k), app.beta2(k)])

    % Spectrogram
    [s,f,t] = pspectrum(app.SerialEEG((k-1) * epoch_length+1:i), fs, ...
        'spectrogram', 'FrequencyLimits', [0, 50]);
    imagesc(app.UISpectrAxes, t, flipud(f), log(abs(s)));
    set(app.UISpectrAxes, 'YDir', 'normal');
    app.UISpectrAxes.YLim = [0, 50];
    colormap(app.UISpectrAxes, 'jet');
    colorbar(app.UISpectrAxes, "eastoutside");

    % Estimation of the patient's BIS
    app.BIS(k) = BIS_decision_tree(app.bsr(k), app.bis_emg(k), ...
        app.sef(k), app.b_ratio(k));
    app.BISLabel.Text = string(app.BIS(k));
    % Sequential point representation
    addpoints(BISrep, app.time_parameters(k), app.BIS(k));
    % Representation update
    drawnow limitrate;

    k = k+1;

end
i=i+1;

end

%% Serial port disconnection
fclose(SerialObject);
delete(SerialObject);
clear SerialObject;
end

% Button pushed function: StopRegisteringButton
function StopRegisteringButtonPushed(app, event)
% Make visible Stop Menu
app.StopEEGLLabel.Visible = 'on';
app.SaveLabel.Visible = 'on';

```

```

app.YesButton.Visible = 'on';
app.NoButton.Visible = 'on';
app.CancelButton.Visible = 'on';
end

% Button pushed function: YesButton
function YesButtonPushed(app, event)
    app.Stop_button = 1; % Stop registering

    % Creation of a structure array object with all the parameters
    data = struct('TimeEEG',app.time_vector, 'RawEEG', app.SerialEEG, ...
        'ECG', app.SerialECG, 'Incident', app.incident, ...
        'IntermittentPhotonicStimulation',app.ips, 'Hyperventilation', ...
        app.hyperventilation, 'EyeOpening', app.eye_op, 'ParamTime', ...
        app.time_parameters, 'BIS', app.BIS, 'BSR', app.bsr, 'BetaRatio', ...
        app.b_ratio, 'BISEMG', app.bis_emg, 'SEF95', app.sef, 'Delta', ...
        app.delta, 'Theta',app.theta, 'Alpha', app.alpha, 'Beta', ...
        app.beta, 'Beta2', app.beta2);

    % Save as '.mat' file
    save("EEG_" + app.PatientID + "_" + datestr(now) + ".mat", 'data');

    % Make invisible Stop Menu
    app.StopEEGLabel.Visible = 'off';
    app.SaveLabel.Visible = 'off';
    app.YesButton.Visible = 'off';
    app.NoButton.Visible = 'off';
    app.CancelButton.Visible = 'off';
    app.LampStart.Visible = 'off';

    % Clear everything
    app.UIEEGAxes.cla
    app.UIECGAxes.cla
    app.UIBSRAxes.cla
    app.UIBRatioAxes.cla
    app.UIBISEMGAxes.cla
    app.UISEFAxes.cla
    app.UIFreqBandAxes.cla
    app.UISpectrAxes.cla
    app.UIBISAxes.cla
    app.PatientIDEditField.Value = '';
    app.IncidentTextArea.Value = '';
    app.BISLabel.Text = 'BIS';
end

% Button pushed function: NoButton
function NoButtonPushed(app, event)
    app.Stop_button = 1; % Stop registering
    app.LampStart.Enable = 'off'; % Turn off the lamp

    % Make invisible Stop Menu
    app.StopEEGLabel.Visible = 'off';
    app.SaveLabel.Visible = 'off';
    app.YesButton.Visible = 'off';
    app.NoButton.Visible = 'off';
    app.CancelButton.Visible = 'off';

    % Clear everything
    app.UIEEGAxes.cla
    app.UIECGAxes.cla
    app.UIBSRAxes.cla
    app.UIBRatioAxes.cla
    app.UIBISEMGAxes.cla
    app.UISEFAxes.cla
    app.UIFreqBandAxes.cla
    app.UISpectrAxes.cla
    app.UIBISAxes.cla
    app.PatientIDEditField.Value = '';
    app.IncidentTextArea.Value = '';
    app.BISLabel.Text = 'BIS';
end

% Button pushed function: CancelButton
function CancelButtonPushed(app, event)
    % Make invisible Stop Menu and keep registering
    app.StopEEGLabel.Visible = 'off';
    app.SaveLabel.Visible = 'off';
    app.YesButton.Visible = 'off';
    app.NoButton.Visible = 'off';
    app.CancelButton.Visible = 'off';
end

% Button pushed function: xButton
function xButtonPushed(app, event)
    % Make visible SelectSerialPortPanel
    app.SelectSerialPortPanel.Visible = 'off';
end
    
```

```

% Changes arrangement of the app based on UIFigure width
function updateAppLayout(app, event)
    currentFigureWidth = app.UIFigure.Position(3);
    if(currentFigureWidth <= app.onePanelWidth)
        % Change to a 2x1 grid
        app.GridLayout.RowHeight = {845, 845};
        app.GridLayout.ColumnWidth = {'1x'};
        app.RightPanel.Layout.Row = 2;
        app.RightPanel.Layout.Column = 1;
    else
        % Change to a 1x2 grid
        app.GridLayout.RowHeight = {'1x'};
        app.GridLayout.ColumnWidth = {242, '1x'};
        app.RightPanel.Layout.Row = 1;
        app.RightPanel.Layout.Column = 2;
    end
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.AutoResizeChildren = 'off';
    app.UIFigure.Position = [100 100 1241 845];
    app.UIFigure.Name = 'UI Figure';
    app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

    % Create GridLayout
    app.GridLayout = uigridlayout(app.UIFigure);
    app.GridLayout.ColumnWidth = {242, '1x'};
    app.GridLayout.RowHeight = {'1x'};
    app.GridLayout.ColumnSpacing = 0;
    app.GridLayout.RowSpacing = 0;
    app.GridLayout.Padding = [0 0 0 0];
    app.GridLayout.Scrollable = 'on';

    % Create LeftPanel
    app.LeftPanel = uipanel(app.GridLayout);
    app.LeftPanel.Layout.Row = 1;
    app.LeftPanel.Layout.Column = 1;

    % Create LampStart
    app.LampStart = uilamp(app.LeftPanel);
    app.LampStart.Enable = 'off';
    app.LampStart.Position = [53 805 15 15];
    app.LampStart.Color = [1 0 0];

    % Create PatientIDEditFieldLabel
    app.PatientIDEditFieldLabel = uilabel(app.LeftPanel);
    app.PatientIDEditFieldLabel.Position = [8 677 58 22];
    app.PatientIDEditFieldLabel.Text = 'Patient ID';

    % Create PatientIDEditField
    app.PatientIDEditField = uieditfield(app.LeftPanel, 'text');
    app.PatientIDEditField.ValueChangedFcn = createCallbackFcn(app, @PatientIDEditFieldValueChanged, true);
    app.PatientIDEditField.Position = [81 677 153 22];

    % Create StopRegisteringButton
    app.StopRegisteringButton = uibutton(app.LeftPanel, 'push');
    app.StopRegisteringButton.ButtonPushedFcn = createCallbackFcn(app, @StopRegisteringButtonPushed, true);
    app.StopRegisteringButton.BackgroundColor = [0.9412 0.9412 0.9412];
    app.StopRegisteringButton.FontSize = 16;
    app.StopRegisteringButton.FontWeight = 'bold';
    app.StopRegisteringButton.Position = [90 798 100 28];
    app.StopRegisteringButton.Text = 'Stop';

    % Create StopEEGLLabel
    app.StopEEGLLabel = uilabel(app.LeftPanel);
    app.StopEEGLLabel.HandleVisibility = 'off';
    app.StopEEGLLabel.HorizontalAlignment = 'center';
    app.StopEEGLLabel.Visible = 'off';
    app.StopEEGLLabel.Position = [7 767 228 22];
    app.StopEEGLLabel.Text = 'You are about to stop recording the EEG.';

    % Create SaveLabel
    app.SaveLabel = uilabel(app.LeftPanel);
    app.SaveLabel.HandleVisibility = 'off';
    app.SaveLabel.HorizontalAlignment = 'center';
    app.SaveLabel.Visible = 'off';
    app.SaveLabel.Position = [7 746 228 22];
    app.SaveLabel.Text = 'Do you want to save it?';

    % Create YesButton
    app.YesButton = uibutton(app.LeftPanel, 'push');
    app.YesButton.ButtonPushedFcn = createCallbackFcn(app, @YesButtonPushed, true);
    app.YesButton.HandleVisibility = 'off';
    
```

```

app.YesButton.Visible = 'off';
app.YesButton.Position = [160 716 60 22];
app.YesButton.Text = 'Yes';

% Create NoButton
app.NoButton = uibutton(app.LeftPanel, 'push');
app.NoButton.ButtonPushedFcn = createCallbackFcn(app, @NoButtonPushed, true);
app.NoButton.HandleVisibility = 'off';
app.NoButton.Visible = 'off';
app.NoButton.Position = [90 716 60 22];
app.NoButton.Text = 'No';

% Create CancelButton
app.CancelButton = uibutton(app.LeftPanel, 'push');
app.CancelButton.ButtonPushedFcn = createCallbackFcn(app, @CancelButtonPushed, true);
app.CancelButton.HandleVisibility = 'off';
app.CancelButton.Visible = 'off';
app.CancelButton.Position = [21 716 60 22];
app.CancelButton.Text = 'Cancel';

% Create IPSPanel
app.IPSPanel = uipanel(app.LeftPanel);
app.IPSPanel.Title = 'Intermittent Photonic Stimulation';
app.IPSPanel.Position = [8 535 227 69];

% Create StartIPSButton
app.StartIPSButton = uibutton(app.IPSPanel, 'push');
app.StartIPSButton.ButtonPushedFcn = createCallbackFcn(app, @StartIPSButtonPushed, true);
app.StartIPSButton.Position = [46 13 75 22];
app.StartIPSButton.Text = 'Start';

% Create StopIPSButton
app.StopIPSButton = uibutton(app.IPSPanel, 'push');
app.StopIPSButton.ButtonPushedFcn = createCallbackFcn(app, @StopIPSButtonPushed, true);
app.StopIPSButton.Position = [144 13 75 22];
app.StopIPSButton.Text = 'Stop';

% Create LampIPS
app.LampIPS = uilamp(app.IPSPanel);
app.LampIPS.Enable = 'off';
app.LampIPS.Position = [22 17 15 15];
app.LampIPS.Color = [1 0 0];

% Create HyperventilationPanel
app.HyperventilationPanel = uipanel(app.LeftPanel);
app.HyperventilationPanel.Title = 'Hyperventilation';
app.HyperventilationPanel.Position = [8 454 227 69];

% Create StartHyperButton
app.StartHyperButton = uibutton(app.HyperventilationPanel, 'push');
app.StartHyperButton.ButtonPushedFcn = createCallbackFcn(app, @StartHyperButtonPushed, true);
app.StartHyperButton.Position = [46 13 74 22];
app.StartHyperButton.Text = 'Start';

% Create StopHyperButton
app.StopHyperButton = uibutton(app.HyperventilationPanel, 'push');
app.StopHyperButton.ButtonPushedFcn = createCallbackFcn(app, @StopHyperButtonPushed, true);
app.StopHyperButton.Position = [144 13 76 22];
app.StopHyperButton.Text = 'Stop';

% Create LampHyper
app.LampHyper = uilamp(app.HyperventilationPanel);
app.LampHyper.Enable = 'off';
app.LampHyper.Position = [22 17 15 15];
app.LampHyper.Color = [1 0 0];

% Create EyeOpeningPanel
app.EyeOpeningPanel = uipanel(app.LeftPanel);
app.EyeOpeningPanel.Title = 'Eye Opening';
app.EyeOpeningPanel.Position = [7 375 227 69];

% Create StartEyeButton
app.StartEyeButton = uibutton(app.EyeOpeningPanel, 'push');
app.StartEyeButton.ButtonPushedFcn = createCallbackFcn(app, @StartEyeButtonPushed, true);
app.StartEyeButton.Position = [47 13 74 22];
app.StartEyeButton.Text = 'Start';

% Create StopEyeButton
app.StopEyeButton = uibutton(app.EyeOpeningPanel, 'push');
app.StopEyeButton.ButtonPushedFcn = createCallbackFcn(app, @StopEyeButtonPushed, true);
app.StopEyeButton.Position = [145 13 75 22];
app.StopEyeButton.Text = 'Stop';

% Create LampEye
app.LampEye = uilamp(app.EyeOpeningPanel);
app.LampEye.Enable = 'off';
app.LampEye.Position = [23 17 15 15];
app.LampEye.Color = [1 0 0];

% Create IncidentButton
app.IncidentButton = uibutton(app.LeftPanel, 'push');

```

```

app.IncidentButton.ButtonPushedFcn = createCallbackFcn(app, @IncidentButtonPushed, true);
app.IncidentButton.Position = [69 332 100 22];
app.IncidentButton.Text = 'Incident';

% Create IncidentTextArea
app.IncidentTextArea = uitextarea(app.LeftPanel);
app.IncidentTextArea.Position = [4 205 234 120];

% Create SelectSerialPortPanel
app.SelectSerialPortPanel = uipanel(app.LeftPanel);
app.SelectSerialPortPanel.Title = 'Select Serial Port';
app.SelectSerialPortPanel.Visible = 'off';
app.SelectSerialPortPanel.Position = [0 564 241 217];

% Create xButton
app.xButton = uibutton(app.SelectSerialPortPanel, 'push');
app.xButton.ButtonPushedFcn = createCallbackFcn(app, @xButtonPushed, true);
app.xButton.Position = [214 173 24.96 21.8809523809524];
app.xButton.Text = 'x';

% Create SerialPortTextArea
app.SerialPortTextArea = uitextarea(app.SelectSerialPortPanel);
app.SerialPortTextArea.Position = [4 143 234 26];

% Create FormatLabel
app.FormatLabel = uilabel(app.SelectSerialPortPanel);
app.FormatLabel.FontColor = [0.502 0.502 0.502];
app.FormatLabel.Position = [18 110 184 22];
app.FormatLabel.Text = 'Use one of the following formats ';

% Create FormatLabel2
app.FormatLabel2 = uilabel(app.SelectSerialPortPanel);
app.FormatLabel2.FontColor = [0.502 0.502 0.502];
app.FormatLabel2.Position = [18 95 184 18];
app.FormatLabel2.Text = 'depending on your OS:';

% Create MacOSXLabel
app.MACOSXLabel = uilabel(app.SelectSerialPortPanel);
app.MACOSXLabel.FontColor = [0.502 0.502 0.502];
app.MACOSXLabel.Position = [18 70 216 22];
app.MACOSXLabel.Text = 'MAC OS X: /dev/cu.usbmodem1421';

% Create WindowsLabel
app.WindowsLabel = uilabel(app.SelectSerialPortPanel);
app.WindowsLabel.FontColor = [0.502 0.502 0.502];
app.WindowsLabel.Position = [17 49 209 22];
app.WindowsLabel.Text = 'Windows: COM3';

% Create LinuxLabel
app.LinuxLabel = uilabel(app.SelectSerialPortPanel);
app.LinuxLabel.FontColor = [0.502 0.502 0.502];
app.LinuxLabel.Position = [17 27 162 22];
app.LinuxLabel.Text = 'Linux: /dev/ttyS0';

% Create OKButton
app.OKButton = uibutton(app.SelectSerialPortPanel, 'push');
app.OKButton.ButtonPushedFcn = createCallbackFcn(app, @OKButtonPushed, true);
app.OKButton.Position = [160 8 79 21];
app.OKButton.Text = 'OK';

% Create RightPanel
app.RightPanel = uipanel(app.GridLayout);
app.RightPanel.Layout.Row = 1;
app.RightPanel.Layout.Column = 2;
app.RightPanel.Scrollable = 'on';

% Create GridLayout1
app.GridLayout1 = uigridlayout(app.RightPanel);
app.GridLayout1.ColumnWidth = {'6.38x', '6.38x', '6.38x'};
app.GridLayout1.RowHeight = {'6.38x', '6.38x', '6.38x', '6.38x'};
app.GridLayout1.Scrollable = 'on';

% Create UIEEGAxes
app.UIEEGAxes = uiaxes(app.GridLayout1);
title(app.UIEEGAxes, 'EEG')
xlabel(app.UIEEGAxes, 'Time (s)')
ylabel(app.UIEEGAxes, 'Amplitude (µV)')
app.UIEEGAxes.PlotBoxAspectRatio = [5.09302325581395 1 1];
app.UIEEGAxes.Layout.Row = 1;
app.UIEEGAxes.Layout.Column = [1 2];

% Create UIBSRAxes
app.UIBSRAxes = uiaxes(app.GridLayout1);
title(app.UIBSRAxes, 'Burst Suppression Ratio')
xlabel(app.UIBSRAxes, 'Time (s)')
ylabel(app.UIBSRAxes, 'BSR')
app.UIBSRAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UIBSRAxes.Layout.Row = 3;
app.UIBSRAxes.Layout.Column = 1;

% Create UIBRatioAxes

```



```

app.UIBRatioAxes = uiaxes(app.GridLayout1);
title(app.UIBRatioAxes, 'Relative  $\beta$ -ratio')
xlabel(app.UIBRatioAxes, 'Time (s)')
ylabel(app.UIBRatioAxes, 'Relative  $\beta$ -ratio')
app.UIBRatioAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UIBRatioAxes.Layout.Row = 3;
app.UIBRatioAxes.Layout.Column = 2;

% Create UIBISEMGAxes
app.UIBISEMGAxes = uiaxes(app.GridLayout1);
title(app.UIBISEMGAxes, 'Power of electromyography')
xlabel(app.UIBISEMGAxes, 'Time (s)')
ylabel(app.UIBISEMGAxes, 'BIS-EMG (dB)')
app.UIBISEMGAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UIBISEMGAxes.Layout.Row = 3;
app.UIBISEMGAxes.Layout.Column = 3;

% Create UISEFAxes
app.UISEFAxes = uiaxes(app.GridLayout1);
title(app.UISEFAxes, '95% Spectral Edge Frequency')
xlabel(app.UISEFAxes, 'Time (s)')
ylabel(app.UISEFAxes, '95% SEF (Hz)')
app.UISEFAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UISEFAxes.Layout.Row = 4;
app.UISEFAxes.Layout.Column = 1;

% Create UIBISAxes
app.UIBISAxes = uiaxes(app.GridLayout1);
title(app.UIBISAxes, 'Bispectral Index')
xlabel(app.UIBISAxes, 'Time (s)')
ylabel(app.UIBISAxes, 'BIS')
app.UIBISAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UIBISAxes.Layout.Row = 2;
app.UIBISAxes.Layout.Column = 3;

% Create BISPanel
app.BISPanel = uipanel(app.GridLayout1);
app.BISPanel.TitlePosition = 'centertop';
app.BISPanel.Title = 'BIS';
app.BISPanel.Layout.Row = 1;
app.BISPanel.Layout.Column = 3;
app.BISPanel.FontSize = 14;

% Create GridLayout2
app.GridLayout2 = uigridlayout(app.BISPanel);
app.GridLayout2.ColumnWidth = {'1.02x', 170, '1.02x'};
app.GridLayout2.RowHeight = {'1.1x', '1.5x', '1.1x'};

% Create BISLabel
app.BISLabel = uilabel(app.GridLayout2);
app.BISLabel.HorizontalAlignment = 'center';
app.BISLabel.FontSize = 40;
app.BISLabel.Layout.Row = 2;
app.BISLabel.Layout.Column = 2;
app.BISLabel.Text = 'BIS';

% Create UIFreqBandAxes
app.UIFreqBandAxes = uiaxes(app.GridLayout1);
title(app.UIFreqBandAxes, 'Frequency Bands')
xlabel(app.UIFreqBandAxes, '')
ylabel(app.UIFreqBandAxes, 'Percentage (%)')
app.UIFreqBandAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UIFreqBandAxes.Layout.Row = 4;
app.UIFreqBandAxes.Layout.Column = 2;

% Create UISpectrAxes
app.UISpectrAxes = uiaxes(app.GridLayout1);
title(app.UISpectrAxes, 'Spectrogram')
xlabel(app.UISpectrAxes, 'Time (s)')
ylabel(app.UISpectrAxes, 'Frequency (Hz)')
app.UISpectrAxes.PlotBoxAspectRatio = [1.94573643410853 1 1];
app.UISpectrAxes.Layout.Row = 4;
app.UISpectrAxes.Layout.Column = 3;

% Create UIECGAxes
app.UIECGAxes = uiaxes(app.GridLayout1);
title(app.UIECGAxes, 'ECG')
xlabel(app.UIECGAxes, 'Time (s)')
ylabel(app.UIECGAxes, 'Amplitude (mV)')
app.UIECGAxes.PlotBoxAspectRatio = [5.09302325581395 1 1];
app.UIECGAxes.Layout.Row = 2;
app.UIECGAxes.Layout.Column = [1 2];

% Create FileMenu
app.FileMenu = uimenu(app.UIFigure);
app.FileMenu.Text = 'File';

% Create NewMenu
app.NewMenu = uimenu(app.FileMenu);
app.NewMenu.MenuSelectedFcn = createCallbackFcn(app, @NewMenuSelected, true);
app.NewMenu.Text = 'New';

```

```

% Create ReviseExistingFileMenu
app.ReviseExistingFileMenu = uimenu(app.FileMenu);
app.ReviseExistingFileMenu.MenuSelectedFcn = createCallbackFcn(app, @ReviseExistingFileMenuSelected, true);
app.ReviseExistingFileMenu.Text = 'Revise Existing File';

% Create ImportExternalFileMenu
app.ImportExternalFileMenu = uimenu(app.FileMenu);
app.ImportExternalFileMenu.MenuSelectedFcn = createCallbackFcn(app, @ImportExternalFileMenuSelected, true);
app.ImportExternalFileMenu.Text = 'Import External File';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = TFG_V7

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
end

```

IV. Presupuesto

1. Cuadro de precios Mano de Obra

Para la elaboración de este proyecto, se ha contado con un ingeniero biomédico junior, encargado de realizar el conjunto de tareas, y un ingeniero biomédico senior, cuya función es supervisar el trabajo que realiza el primero, como mano de obra.

Para el cálculo del coste por hora de cada uno, se ha considerado que el salario mensual de un ingeniero biomédico junior es de 1 500 € y el de un ingeniero biomédico senior es de 3 000 €. Sumando dos pagas extras de mismo valor que los salarios mensuales, el salario anual asciende a 21 000 € y 39 200 € respectivamente. Además, en concepto de Seguridad Social, el coste por los empleados asciende un 23.6% de contingencias comunes, 5.5% de seguro de desempleo, 0.6% de formación profesional, 0.2% de FOGASA y 1.65% de IT/IMS (Incapacidad Temporal e Invalidez, Muerte y Supervivencia).

Por otro lado, según la Dirección General de Trabajo, se establece un total de 210 días de trabajo anuales, con una jornada laboral de 8 h, lo que supone un total de 1680 h de trabajo anuales. Por tanto, el coste por hora de un ingeniero biomédico junior es de 16.44 € y el de un ingeniero biomédico senior es de 30.70 €.

| Cuadro de mano de obra | | | | Página 1 | |
|------------------------|--------|---------------------------------|--------|---------------------|----------|
| Num. | Código | Denominación de la mano de obra | Precio | Horas | Total |
| 1 | MO.IBS | Ingeniero Biomédico Senior | 30,70 | 18,000 h | 552,60 |
| 2 | MO.IBJ | Ingeniero Biomédico Junior | 16,44 | 193,000 h | 3.172,92 |
| | | | | Total mano de obra: | 3.725,52 |

Tabla 5. Cuadro de mano de obra.

2. Cuadro de precios Maquinaria

Para determinar los costes de maquinaria, se han considerado los costes asociados al ordenador portátil, las licencias de programas y el material asociado a la implementación del sistema hardware para captación de EEG y ECG. Se ha estimado el factor de amortización para cada elemento según la Ecuación 4.

$$\text{Factor de amortización} = \frac{\text{Tiempo amortizado (años)}}{\text{Amortización (años)}} \quad (6)$$

Para el ordenador portátil, se ha considerado un tiempo amortizado de 6 meses y una amortización de 6 años, para la licencia de Matlab®, un tiempo amortizado de 6 meses y una amortización de 3 años, para la licencia de Microsoft Office 365, un tiempo amortizado de 3 meses y una amortización de 3 años, para el material asociado a la implementación del sistema hardware, un tiempo amortizado de 2 meses y una amortización de 3 años. Puesto que los electrodos de medida son desechables, se considera que sólo tienen un uso y se introduce su coste por unidad.

| Cuadro de maquinaria | | | | Página 1 | |
|----------------------|-------------|-------------------------------|----------|-------------------|----------|
| Num. | Código | Denominación de la maquinaria | Precio | Cantidad | Total |
| 1 | MAQ.LMatlab | Licencia Matlab 2019b | 2.000,00 | 0,835 | 1.670,00 |
| 2 | MAQ.PC | Ordenador portátil | 1.000,00 | 0,996 | 996,00 |
| 3 | MAQ.LO365 | Licencia de Office 365 | 69,00 | 0,083 | 5,73 |
| 4 | MAQ.EEG | EEG Click | 40,00 | 0,112 | 4,48 |
| 5 | MAQ.Ard | Arduino Uno | 24,00 | 0,112 | 2,68 |
| 6 | MAQ.CE | Cable conexión electrodos | 10,00 | 0,112 | 1,12 |
| 7 | MAQ.ECG | AD8232 | 10,00 | 0,112 | 1,12 |
| 8 | MAQ.Ele | Electrodos de medida | 1,00 | 12,056 u | 12,06 |
| 9 | MAQ.Cab | Cable de puente macho-hembra | 0,10 | 0,112 | 0,02 |
| | | | | Total maquinaria: | 2.693,21 |

Tabla 6. Cuadro de maquinaria.

3. Cuadro de precios unitarios

| Nº | Designación | Importe | |
|-----|---|------------------|---|
| | | En cifra (Euros) | En letra (Euros) |
| | 1 Definición del proyecto | | |
| 1.1 | h Reunión inicial con el tutor del TFG | 184,37 | CIENTO OCHENTA Y CUATRO EUROS CON TREINTA Y SIETE CÉNTIMOS |
| 1.2 | h Reunión en el Hospital la Pedrera para definir especificaciones | 576,58 | QUINIENTOS SETENTA Y SEIS EUROS CON CINCUENTA Y OCHO CÉNTIMOS |
| | 2 Investigación del estado del arte | | |
| 2.1 | h Investigación BIS | 663,10 | SEISCIENTOS SESENTA Y TRES EUROS CON DIEZ CÉNTIMOS |
| 2.2 | h Apendizaje manejo de software | 690,14 | SEISCIENTOS NOVENTA EUROS CON CATORCE CÉNTIMOS |
| | 3 Implementación del sistema hardware | | |
| 3.1 | u Dispositivo | 39,16 | TREINTA Y NUEVE EUROS CON DIECISEIS CÉNTIMOS |
| 3.2 | h Programación placa Arduino Uno | 137,61 | CIENTO TREINTA Y SIETE EUROS CON SESENTA Y UN CÉNTIMOS |
| | 4 Implementación software | | |
| 4.1 | h Preprocesado de señal | 519,17 | QUINIENTOS DIECINUEVE EUROS CON DIECISIETE CÉNTIMOS |
| 4.2 | h Cálculo parámetros | 604,66 | SEISCIENTOS CUATRO EUROS CON SESENTA Y SEIS CÉNTIMOS |
| 4.3 | h Diseño y desarrollo de la interfaz de usuario | 690,14 | SEISCIENTOS NOVENTA EUROS CON CATORCE CÉNTIMOS |
| 4.4 | h Comprobación de funcionamiento | 485,25 | CUATROCIENTOS OCHENTA Y CINCO EUROS CON VEINTICINCO CÉNTIMOS |
| | 5 Redacción y defensa TFG | | |
| 5.1 | h Redacción de los documentos | 1.460,09 | MIL CUATROCIENTOS SESENTA EUROS CON NUEVE CÉNTIMOS |
| 5.2 | h Revisión de los documentos | 218,57 | DOSCIENTOS DIECIOCHO EUROS CON CINCUENTA Y SIETE CÉNTIMOS |
| 5.3 | h Preparación defensa | 406,64 | CUATROCIENTOS SEIS EUROS CON SESENTA Y CUATRO CÉNTIMOS |

4. Cuadro de precios descompuestos

| Nº | Designación | Importe | |
|----------------------|---|-----------------|---------------|
| | | Parcial (Euros) | Total (Euros) |
| 1.1 | 1 Definición del proyecto | | |
| | h Reunión inicial con el tutor del TFG | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 2,000 h | 16,44 |
| | Ingeniero Biomédico Senior | 2,000 h | 30,70 |
| 1.2 | (Maquinaria) | | |
| | Ordenador portátil | 0,083 | 1.000,00 |
| | 4% Costes indirectos | | 7,09 |
| | | | 184,37 |
| 2.1 | h Reunión en el Hospital la Pedrera para definir especificaciones | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 10,000 h | 16,44 |
| | Ingeniero Biomédico Senior | 10,000 h | 30,70 |
| | (Maquinaria) | | |
| 2.2 | Ordenador portátil | 0,083 | 1.000,00 |
| | 4% Costes indirectos | | 22,18 |
| | | | 576,58 |
| | 2 Investigación del estado del arte | | |
| 2.1 | h Investigación BIS | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 30,000 h | 16,44 |
| | Ingeniero Biomédico Senior | 2,000 h | 30,70 |
| | (Maquinaria) | | |
| 2.2 | Ordenador portátil | 0,083 | 1.000,00 |
| | 4% Costes indirectos | | 25,50 |
| | | | 663,10 |
| | h Apendizaje manejo de software | | |
| 3.1 | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 15,000 h | 16,44 |
| | (Maquinaria) | | |
| | Licencia Matlab 2019b | 0,167 | 2.000,00 |
| | Ordenador portátil | 0,083 | 1.000,00 |
| 3.1 | 4% Costes indirectos | | 26,54 |
| | | | 690,14 |
| | 3 Implementación del sistema hardware | | |
| | u Dispositivo | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 2,000 h | 16,44 |
| | (Maquinaria) | | |
| | Arduino Uno | 0,056 | 24,00 |
| | Cable conexión electrodos | 0,056 | 10,00 |
| | Cable de puente macho-hembra | 0,056 | 0,10 |
| AD8232 | 0,056 | 10,00 | |
| EEG Click | 0,056 | 40,00 | |
| Electrodos de medida | 0,056 u | 1,00 | |
| 4% Costes indirectos | | 1,51 | |
| | | 39,16 | |

| Nº | Designación | Importe | | |
|----------------------------------|--|---|--|---|
| | | Parcial (Euros) | Total (Euros) | |
| 3.2 | h Programación placa Arduino Uno (Mano de obra) Ingeniero Biomédico Junior | 3,000 h | 16,44 | 49,32 |
| | (Maquinaria) Ordenador portátil 4% Costes indirectos | 0,083 | 1.000,00 | 83,00 5,29 |
| | | | | 137,61 |
| 4 Implementación software | | | | |
| 4.1 | h Preprocesado de señal (Mano de obra) Ingeniero Biomédico Junior | 5,000 h | 16,44 | 82,20 |
| | (Maquinaria) Licencia Matlab 2019b Ordenador portátil 4% Costes indirectos | 0,167 0,083 | 2.000,00 1.000,00 | 334,00 83,00 19,97 |
| | | | | 519,17 |
| 4.2 | h Cálculo parámetros (Mano de obra) Ingeniero Biomédico Junior | 10,000 h | 16,44 | 164,40 |
| | (Maquinaria) Licencia Matlab 2019b Ordenador portátil 4% Costes indirectos | 0,167 0,083 | 2.000,00 1.000,00 | 334,00 83,00 23,26 |
| | | | | 604,66 |
| 4.3 | h Diseño y desarrollo de la interfaz de usuario (Mano de obra) Ingeniero Biomédico Junior | 15,000 h | 16,44 | 246,60 |
| | (Maquinaria) Licencia Matlab 2019b Ordenador portátil 4% Costes indirectos | 0,167 0,083 | 2.000,00 1.000,00 | 334,00 83,00 26,54 |
| | | | | 690,14 |
| 4.4 | h Comprobación de funcionamiento (Mano de obra) Ingeniero Biomédico Junior | 2,000 h | 16,44 | 32,88 |
| | (Maquinaria) Arduino Uno Cable conexión electrodos Cable de puente macho-hembra AD8232 EEG Click Electrodos de medida Licencia Matlab 2019b Ordenador portátil 4% Costes indirectos | 0,056 0,056 0,056 0,056 0,056 12,000 u 0,167 0,083 | 24,00 10,00 0,10 10,00 40,00 1,00 2.000,00 1.000,00 | 1,34 0,56 0,01 0,56 2,24 12,00 334,00 83,00 18,66 |
| | | | | 485,25 |
| 5 Redacción y defensa TFG | | | | |

| Nº | Designación | Importe | |
|----------------------------------|--------------------------------------|-----------------|---------------|
| | | Parcial (Euros) | Total (Euros) |
| 5.1 | h Redacción de los documentos | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 80,000 h | 16,44 |
| | | | 1.315,20 |
| | (Maquinaria) | | |
| Licencia de Microsoft Office 365 | 0,083 | 69,00 | 5,73 |
| Ordenador portátil | 0,083 | 1.000,00 | 83,00 |
| 4% Costes indirectos | | | 56,16 |
| | | | 1.460,09 |
| 5.2 | h Revisión de los documentos | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 4,000 h | 16,44 |
| | Ingeniero Biomédico Senior | 2,000 h | 30,70 |
| | | | 65,76 |
| (Maquinaria) | | | |
| Ordenador portátil | 0,083 | 1.000,00 | 83,00 |
| 4% Costes indirectos | | | 8,41 |
| | | | 218,57 |
| 5.3 | h Preparación defensa | | |
| | (Mano de obra) | | |
| | Ingeniero Biomédico Junior | 15,000 h | 16,44 |
| | Ingeniero Biomédico Senior | 2,000 h | 30,70 |
| | | | 246,60 |
| (Maquinaria) | | | |
| Ordenador portátil | 0,083 | 1.000,00 | 83,00 |
| 4% Costes indirectos | | | 15,64 |
| | | | 406,64 |

5. Cuadro de presupuestos parciales

Presupuesto parcial nº 1 Definición del proyecto

| Num. | Ud | Descripción | Medición | Precio (€) | Importe (€) |
|--|----|---|----------|------------|---------------|
| 1.1 | h | Reunión inicial con el tutor del TFG | 1,000 | 184,37 | 184,37 |
| 1.2 | h | Reunión en el Hospital la Pedrera para definir especificaciones | 1,000 | 576,58 | 576,58 |
| Total presupuesto parcial nº 1 Definición del proyecto: | | | | | 760,95 |

Presupuesto parcial nº 2 Investigación del estado del arte

| Num. | Ud | Descripción | Medición | Precio (€) | Importe (€) |
|--|----|-------------------------------|----------|------------|-----------------|
| 2.1 | h | Investigación BIS | 1,000 | 663,10 | 663,10 |
| 2.2 | h | Apendizaje manejo de software | 1,000 | 690,14 | 690,14 |
| Total presupuesto parcial nº 2 Investigación del estado del arte: | | | | | 1.353,24 |

Presupuesto parcial nº 3 Implementación del sistema hardware

| Num. | Ud | Descripción | Medición | Precio (€) | Importe (€) |
|--|----|--------------------------------|----------|------------|---------------|
| 3.1 | u | Dispositivo | 1,000 | 39,16 | 39,16 |
| 3.2 | h | Programación placa Arduino Uno | 1,000 | 137,61 | 137,61 |
| Total presupuesto parcial nº 3 Implementación del sistema hardware: | | | | | 176,77 |

Presupuesto parcial nº 4 Implementación software

| Num. | Ud | Descripción | Medición | Precio (€) | Importe (€) |
|--|----|---|----------|------------|-----------------|
| 4.1 | h | Preprocesado de señal | 1,000 | 519,17 | 519,17 |
| 4.2 | h | Cálculo parámetros | 1,000 | 604,66 | 604,66 |
| 4.3 | h | Diseño y desarrollo de la interfaz de usuario | 1,000 | 690,14 | 690,14 |
| 4.4 | h | Comprobación de funcionamiento | 1,000 | 485,25 | 485,25 |
| Total presupuesto parcial nº 4 Implementación software: | | | | | 2.299,22 |

Presupuesto parcial nº 5 Redacción y defensa TFG

| Num. | Ud | Descripción | Medición | Precio (€) | Importe (€) |
|--|----|-----------------------------|----------|------------|-----------------|
| 5.1 | h | Redacción de los documentos | 1,000 | 1.460,09 | 1.460,09 |
| 5.2 | h | Revisión de los documentos | 1,000 | 218,57 | 218,57 |
| 5.3 | h | Preparación defensa | 1,000 | 406,64 | 406,64 |
| Total presupuesto parcial nº 5 Redacción y defensa TFG: | | | | | 2.085,30 |

6. Presupuesto de ejecución material

| Presupuesto de ejecución material | Importe (€) |
|---------------------------------------|-----------------|
| 1 Definición del proyecto | 760,95 |
| 2 Investigación del estado del arte | 1.353,24 |
| 3 Implementación del sistema hardware | 176,77 |
| 4 Implementación software | 2.299,22 |
| 5 Redacción y defensa TFG | 2.085,30 |
| Total | 6.675,48 |

Asciende el presupuesto de ejecución material a la expresada cantidad de SEIS MIL SEISCIENTOS SETENTA Y CINCO EUROS CON CUARENTA Y OCHO CÉNTIMOS.