

UNIVERSIDAD DE MURCIA



GRADO EN INGENIERÍA EN INFORMÁTICA

MENCIÓN: COMPUTACIÓN

TRABAJO FIN DE GRADO

**Herramienta web para el análisis,
procesamiento y visualización de las señales
del EEG**

Autor:

JOSÉ MARÍA SÁNCHEZ SALAS

Tutor:

JOSÉ TOMÁS PALMA MÉNDEZ

SEPTIEMBRE 2017

Una de las cosas más fascinantes de los programadores es que no puedes saber si están trabajando o no sólo con mirarlos. A menudo están sentados aparentemente tomando café, chismorreando o mirando a las nubes. Sin embargo, es posible que estén poniendo en orden todas las ideas individuales y sin relación que pululan por su mente.

Charles M. Strauss

*Como dirían en el cole:
no hay mejor consejo que el de ser tú mismo,
a pesar de que al resto no les mole.*

ZPU

Agradecimientos

Mi más sincero agradecimiento a mi tutor José Tomás Palma Méndez, por haberme alentado a empezar este proyecto, por todo su apoyo y dedicación durante todo este tiempo.

A todos mis compañeros de carrera, por haber hecho más fácil y llevadero el transcurso de todos estos años. Por tantos cafés, bocadillos y cervezas juntos.

Al kárate, por haber sido el eje central en el que basar mi vida. A toda la familia del Kárate Club Tensho. A mis compañeros y amigos de entrenamiento. A mi sensei, por habernos transmitido sus conocimientos y tener fe en mí.

A mis amigos Miki, Negro y Wawas, por haber recorrido conmigo todo este camino desde pequeños. Por haberme aguantado en mis peores momentos. Por haber entendido a la perfección todo el tiempo que no he podido estar con ellos. Por su amistad incondicional. Por los momentos vividos y los que nos quedan por vivir.

A Laura por haber crecido conmigo y regalarme todos estos años. Por haber sido el pilar en el que apoyarme. A su familia, por haberme acogido, querido y apoyado como uno más.

A toda mi familia y más concretamente, a mis padres, Juan y Carmen, y a mi hermano Juan Luis, por haber dedicado su tiempo, esfuerzo y amor a que yo esté aquí. Por haberme enseñado que si de verdad quieres algo, tienes que luchar por ello, cueste lo que cueste. Por su educación y empeño. Por ser un apoyo incondicional, tanto en los buenos como en los malos momentos. Por ser los referentes de mi vida.

Resumen

En este Trabajo Fin de Grado (TFG) se expone el diseño e implementación de una herramienta web para el análisis, procesamiento y visualización de señales del electroencefalograma (EEG). Este trabajo es la continuación del TFG: *Análisis de EEG para la evaluación de las emociones* [29], en el que se describen las técnicas y procedimientos para el procesamiento y análisis de señales del EEG, y que está implementado completamente en Matlab.

El objetivo principal subyacente a la realización de este trabajo es la obtención de una herramienta de *software libre* que permita el cálculo de medidas objetivas a partir del procesamiento del electroencefalograma. En este trabajo, las medidas objetivas son los siguientes cuatro índices: *Índice de atención*, *Índice de memorización*, *Índice de aproximación-rechazo* e *Índice de compromiso*.

Además, otro de los objetivos de este trabajo es que la herramienta proporcione la visualización de todas las señales y de todos los valores intermedios que se utilizan en todas las fases del cálculo de estos índices.

Esta herramienta web está implementada completamente en el entorno matemático R, tanto la parte del servidor (*backend*) como la parte de interfaz de usuario (*frontend*). Para ello se ha hecho uso del framework web Shiny.

Para la obtención de los índices, el primer problema planteado es el entendimiento e interiorización de todo el TFG [29], desde los conceptos médicos del EEG hasta el cálculo final de los índices. El segundo problema consiste en aprender el framework Shiny, sus características y cómo funciona para poder realizar la implementación de la herramienta.

El tercero es estudiar y analizar distintas técnicas de filtrado de señales, así como distintas herramientas para la visualización de las mismas. El cuarto problema es realizar la implementación de todo el procedimiento para el cálculo de los índices. Y por último, realizar una evaluación de los resultados obtenidos en comparación con los obtenidos en [29].

Extended Abstract

The human being has always experienced an ongoing concern to investigate and explain several questions related to the human's brain and how it functions. This matter is the main reason why, through the history, the human being has been developing different techniques and methodologies in order to be able to study the human brain more deeply.

Among these techniques, it is found the one called *electroencephalography*, whose task is to capture the electrical activity produced in the brain, through the synchronization of the neurons that comprise it. The record obtained is called *electroencephalogram* (EEG), from which a great source of information is obtained about the subject being studied.

The EEG stands out over other neuroimaging techniques thanks to its low cost, its good temporal resolution, the facilitation of the acquisition, analysis and storage of the signal and the possibility to modify, not only during the record but also after it, some parameters like filters, sensitivity, recording time and assemblies. In addition, this technique allows freedom of movement, so that the subject is able to move around the room where the test is taking place.

Regarding the medical scope, the study of EEG can provide a large amount of information clinically relevant as well in epileptic pathologies as in non-epileptic pathologies, and also in pathologies during the coma state and in the cerebral death, among others. Outside of the medical scope, it also has application in the sports area, computer sciences and neuromarketing. However, it is important to clarify that, regardless of its study field, the electroencephalogram does not provide information about the subject's intelligence.

The main hypothesis of this work discusses that through the signals provided by the EEG it is possible to obtain objective information about the state of the subject like, for example, the cognitive and emotive burden that the subject experiences while carrying out some particular tasks and also his reaction to some particular stimuli.

This work is the continuation of the final degree project titled "*Análisis de EEG para la evaluación de las emociones*" [29] whose main objective is the design, development and analysis of different techniques destined to extract objective measures from the EEG. This project was developed in the mathematical environment Matlab.

The main objective of the present work consists of developing the design and implementation of a free software web tool which allows the collecting of objective measures of the EEG's signal, in order to be able to quantify the cognitive effort of a subject in the development of certain tasks and his emotions regarding the presence of an specific stimuli. As well as being able to visualize the EEG signal and all the intermediate values necessary for collecting such objective measures. This objective can be decomposed in the following subobjectives:

- To analyse and study different signal filters for the pre-processing of the EEG signal.
- To analyse and study different tools used for the signal's visualization.
- To implement the procedures and algorithms necessary for the pre-processing of the EEG signal that allow the deletion of the noise and artefacts.
- To implement all processes for the calculation and the collection of the information and cognitive and emotive rates of the EEG signal.

- To design and develop a free software web platform that allows us, step by step and automatically, not only to collect the objective measures but also to visualize the signals and all the intermediate values.
- To perform an evaluation of the results obtained, comparing them with the ones obtained in [29].

This work starts with the study of the filtering of signals: any signal can be considered like a big set of sine and cosine waves of different frequencies, amplitudes and phases. A digital filter is a digital signal processor which attenuates the components of a signal of a determined frequency (band). The common reason, behind filtering in general, is found in attenuating the noise of the recordings, while the involved signal is preserved. However, it is impossible to design a filter which do not change the signal, consequently the ideal filter does not exist.

There are two categories of filters which are used to approximate an ideal behaviour, each one of them with its advantages and disadvantages. These filters are:

- The **Finite Impulse Response Filter (FIR)**, which uses a current input value and a finite set of last input values to calculate an output value.
- The **Infinite Impulse Response Filter (IIR)**, which uses the current input with a finite set of the previous outputs, for calculate current output. It is a cursive filter.

Due to the features of both filters and the requirements of this work regarding signal filtering of EEG, a FIR filter is going to be used to make the signals filtering.

The web tool developed in this work is completely implemented in the mathematical environment R, both the backend and frontend, and, for this, the web framework Shiny has been used allowing us to create web interactive applications without having previous knowledge of HTML, CSS or JavaScript.

Such as any other web application, a Shiny application has two main components:

- The user interface specification, which denotes how the user wants to structure the web page (layout), what title and what elements does it have, both input and output.
- The server script, which contains all actions required for obtaining the data that are going to be visualized in the interface.

However, in order to make this application interactive, both components must be connected. Therefore, Shiny provides a reactive programming library, which allows that the changes in the input values that provoke the execution of that parts of the code that provide the associated output values, updating the output values.

The elements that Shiny provides to use the reactive programming library are: reactive values, reactive expressions and observers. With these elements, Shiny supplies a simple and intuitive mechanism. In this way, input element (reactive value) changes, the associated output values are updated (reactive expressions/observers), necessary for any interactive application. It is important to point out that, it does not necessary explicitly indicate what relationships exist between what elements and what to do when one of the input elements changes, because it is Shiny who automatically handles all these details.

Besides, Shiny offers the option of modularizing the application code. A Shiny module cannot be executed in an independent way (such as a Shiny application) and it is included as part of a Shiny application or as part of other module (Shiny also allows the modules composition). A module can only be used for the interface, for the server or for both, and can have the desired complexity.

For the acquisition of the signals, the *NeXus-10* tool has been used, which consists of a measuring system able to capture both physiological signals (breathing, temperature, etc.) and neurological signals (EEG). Besides a software application has been used for storing the concrete time mark in which an image appears on the screen and the time mark until it disappears. In this way, it corresponds the recording of the EEG and EOG signals values with the duration of the image on the display.

During the experiments, the subject is submitted to the following phases:

- **Closed-eyes phase.** In this first phase, the rhythms and frequencies of the subject in a phase of minimal activity are known.
- **Baseline phase.** In this phase, the “normal” state of the subject is known, this means, being active without receiving any decisive stimuli for the test.
- **Exposure to the stimulus phase.** In this phase, the subject is submitted to the main stimulus from which results are supposed to be obtained.

The experiments which have been made are the following:

- A study on the influence of smells in response to visual stimuli related to art images as well as the efficiency in CANTAB tests.
- A study of an objective way on quantifying people’s welfare in process or state of retirement from visual stimuli related to this process (family, health, etc.) as well as efficiency in CANTAB tests.
- A study in high school as the search for relationships between CANTAB performance tests, the activation level to visual stimuli related to healthcare environments and the interest in their university education.

From the tests that the CANTAB software provides, the following tests have been made in the experiments:

- *Reaction Time (RTI)*. Test which measures the mental response towards a stimulus which consists in pressing a point when it is illuminated.
- *Spatial Working Memory (SWM)*. Test in which the subject must find a colour box between all boxes shown in the display, measuring in this way the capacity of retaining information of the subject.
- *One Touch Stockings (OTS)*. In this test, a final configuration is suggested to the subject and he must achieve this configuration using as few movements as possible from a specified initial configuration.
- *Emotion Recognition Task (ERT)*. It is a test in which the subject must recognize the emotion in the faces shown in a fast images presentation.

Once the signals have been obtained, the procedure used for its processing is the following:

1. **Adequacy of registered signals.** It consists of adjusting the frequencies of information in the EEG and the EOG to remove the artefacts of the EEG caused by the ocular movements and blinks. To do so, the following steps are made:
 - 1.1. **Adjustment of the frequencies of information in EEG and EOG.** It tries to filter the signals to obtain those frequencies carrying important information. For it, a FIR filter is applied in pass-band mode for the ranges 2-30Hz y 0,1-5Hz, for the EEG and EOG, respectively.
 - 1.2. **Elimination of the artefacts caused by ocular movements and blinks.** It tries to remove from the EEG signal all the information originated in the ocular surface. For it, the EOG signal and the Gratton algorithm are used. This ocular correction provides a signal from which the subject's blinks can be inferred.
2. **Segmentation of the signal.** Until this point, the complete signal of all experiment has been used. In this step, it tries to separate each phase and tasks made to the subject.
3. **Calculation of the IAF (Individual Alpha Frequency).** For setting the bands of the subject's frequencies correctly, since these may vary from subject to subject.
4. **Filtering by the bands α , β and θ of the signal.** Once IAF has been calculated and the bands of frequencies have been settled, the tasks are filtered by each band to obtain the GFP (Global Field Power) for each band. These bands have been settled the following way:
 - **Theta band (θ):** $[IAF - 6, IAF - 2]$.
 - **Alpha band (α):** $[IAF - 2, IAF + 2]$.
 - **Beta band (β):** $[> IAF + 2]$.
5. **Obtainment of the indices.** The main objective of the all signals processing consists of calculating the following four index:
 - **Attention Index (AI).** Thanks to it, it is possible to obtain the interest which a subject has dedicated to the completion of a task or to reception of a stimuli, performing finding and recovery processes of the brain memory's long-term structures. It is based in the α band study.
 - **Approach-Withdrawal Index (AW).** Allows the obtaining of the degree of pleasure that the subject experiences when he performs a task o is submitted to a stimulus. It is based in the α band study.
 - **Engagement Index (EI).** It measures the global attention which a subject pays in a determined task. It differentiates from the AI index in the fact that the EI index measures the response to the received stimuli variation. It is based in the α , β and θ bands study.
 - **Memorization Index (MI).** This index measures the degree in which a subject has been able to create new partnerships in the brain memory's structures. It is based in the θ band study.

To do so, it necessary to carry out the following steps:

- 5.1. **Calculation of the reference values.** To be able to obtain the indices values, first it is necessary to calculate the reference values of each index, obtained by the base line signal (obtained in the base line phase). This reference values are the mean (μ) and the standard deviation (σ).
- 5.2. **Normalization and calculation of the index.** With the reference values obtained for each index, a normalization of the index *point-to-point* is made for each task. Finally, the concrete value of the index is obtained performing the mean of its normalized value.

The functionality which provides the tool developed in this work is linked to the mentioned procedure and it is the following:

- It allows you to load as many data as you want to process.
- It allows you to select the data with which you want to work.
- It allows you to display any of obtained signal to the subject.
- It allows you to filter the EEG and EOG signals to remove all non-relevant information for the study.
- It allows you to perform the ocular correction, applying the Gratton algorithm. After the correction, it allows you to display its result.
- It allows you to apply the signal segmentation in bands, and to obtain in this way, the α , β and θ bands for each one of the tasks. In addition, it allows you to display the GPS signal, the IAF value in this signal and the bands of the each one of the tasks.
- It allows you to calculate the indices, the visualization of the reference values used for its calculation and the indices values for each one of the tasks.
- It allows you to execute automatically all processes for the index calculation and the exportation of the calculated data: index and signals used in its calculation.

To finish this work, it is necessary to perform an evaluation in order to verify that the index values obtained by the tool are correct. In this way, the obtained results have been compared in:

- The signals filtering, where there is a mistake in the $10^{-3}V$ order.
- The ocular correction application, where an error is perpetrated $10^{-3}V$ order.
- The IAF calculation, where an error is found in the $10^{-1}Hz$ order.
- The index calculation, where an error is committed in the 10^{-3} order (the index have no units).

These results have been obtained not only with the help of the tool (developed in R) of this work and but also with the help of the ones calculated in [29] (developed in Matlab), in order to calculate the error that exists between them. The error has been obtained using the Root Mean Square Error (RMSE) for the signals filtering and the ocular correction, while the absolute value of the difference has been used for the IAF and the index.

From this evaluation, it can be concluded that the error committed has to do with the differences that exist between the implementations of R and Matlab. For example, to perform the application of the FIR filter or to apply the lineal regression in the Gratton algorithm. Therefore, the index calculation performed by this work is correct.

This work is located within the research lines of the research groups *Neurociencia clínica y experimental (NiCE)* of the *Facultad de Medicina* at the University of Murcia, operated by the professor Mrs. María Trinidad Herrero Ezquerro, and *Artificial Intelligence Knowledge Engineering (AIKE)* of the *Facultad de Informática* at the University of Murcia, operated by the professor Mr. José Tomás Palma Méndez.

Índice general

Agradecimientos	VII
Resumen	IX
Extended Abstract	XI
Índice de figuras	XIX
Índice de tablas	XXI
1. Introducción al trabajo	1
1.1. Introducción	1
1.2. Hipótesis del trabajo y objetivo	2
1.3. Contexto del trabajo	2
1.4. Estructura de la memoria	3
2. Fundamentos y Estado del Arte	5
2.1. Introducción	5
2.2. Fundamentos del electroencefalograma (EEG)	5
2.3. Relación de índices de estudio	7
2.4. Filtrado de señales	8
2.4.1. Tipos de filtrado	8
2.4.2. Filtros digitales	9
2.4.3. Elección del filtro adecuado para este trabajo	11
2.5. Tecnologías existentes	11
2.5.1. Procesamiento de señales	11
2.5.2. Desarrollo web	12
2.5.3. Tecnologías escogidas	12
2.6. Herramientas de visualización de señales	13
2.6.1. Herramienta de visualización escogida	13
3. Metodología y diseño de la solución	15
3.1. Introducción	15
3.2. Metodología seguida	15
3.3. Desarrollo web	16
3.3.1. Estructura de una aplicación Shiny	16
3.3.2. Reactividad en Shiny	17
3.3.3. Modularización de código en una aplicación Shiny	20
3.3.4. Uso del framework Shiny para el desarrollo web del proyecto	21
3.4. Funcionalidad de la aplicación	23
3.5. Funciones implementadas	24
3.5.1. Aplicación de un filtro FIR para el filtrado de las señales	24
3.5.2. Aplicación del algoritmo de Gratton para la corrección ocular	25

4. Acciones y procedimientos para el cálculo de los índices	27
4.1. Introducción	27
4.2. Adecuación de las señales registradas	30
4.2.1. Ajuste a las frecuencias de información en EEG y EOG	30
4.2.2. Eliminación de los artefactos provocados por movimientos oculares y parpadeos	32
4.3. Segmentación de la señal	33
4.4. Cálculo de la IAF (<i>Individual Alpha Frequency</i>)	33
4.5. Filtrado por bandas α , β y θ de la señal	34
4.6. Obtención de los índices	35
4.6.1. Definición de los índices	35
4.6.2. Valores de referencia	36
4.6.3. Normalización y cálculo de los índices	36
5. Evaluación de resultados	39
5.1. Introducción	39
5.2. Evaluación de las funciones implementadas	39
5.2.1. Procedimiento para la obtención de los datos a evaluar	39
5.2.2. Evaluación del filtrado	40
5.2.3. Evaluación de la corrección ocular	46
5.3. Evaluación del IAF	53
5.4. Evaluación de los índices	55
5.4.1. Procedimiento para la obtención de los datos a evaluar	55
5.4.2. Evaluación de los índices	55
5.5. Conclusiones obtenidas	57
6. Conclusiones y vías futuras	59
6.1. Conclusiones	59
6.1.1. Conclusiones personales	59
6.2. Vías futuras	60
A. Formato de los ficheros de entrada	61
B. Manual de Usuario	63
Glosario	65
Bibliografía	67

Índice de figuras

2.1. Resumen de las bandas $\delta, \theta, \alpha, \beta$ en las que se divide el EEG.	7
2.2. Efecto de un filtro paso-bajo aplicado a una señal.	8
2.3. Efecto de un filtro paso-alto aplicado a una señal.	8
2.4. Efecto de un filtro pasa-banda aplicado a una señal.	8
2.5. Efecto de un filtro banda-rechazada aplicado a una señal.	9
 3.1. Estructura general de una aplicación Shiny, con algunos elementos.	17
3.2. Conexión entre valores de entrada, código en R y valores de salida en una aplicación Shiny.	17
3.3. Símbolos de los tipos de objetos en la programación reactiva de Shiny.	18
3.4. Estructura reactiva con una fuente y un punto final.	19
3.5. Estructura reactiva con una fuente, un conductor y dos puntos finales.	19
3.6. Estructura de la aplicación web en módulos.	23
 4.1. Flujo de trabajo para el cálculo de los índices AI, AW, EI y MI. Obtención de los valores de referencia.	29
4.2. Flujo de trabajo para el cálculo de los índices AI, AW, EI y MI. Obtención de los índices.	30
4.3. Señal EEG (superior) y EOG (inferior) sin filtrar.	31
4.4. Señal EEG (superior) y EOG (inferior) filtradas para los rangos 2-30Hz y 0.1-5Hz, respectivamente.	32
4.5. Señal EEG sin artefactos oculares.	33
4.6. Señal GPS y valor de la IAF.	34
4.7. Banda <i>alpha</i> para la fase basal.	35
4.8. Valor de referencia <i>punto-a-punto</i> para el índice AW.	37
4.9. Índice AW <i>punto-a-punto</i> para la tarea Spa_10g.	37
 5.1. Diferencias en el dominio del tiempo de las señales filtradas, del sujeto ANDGAR.	40
5.2. Diferencias en el dominio del tiempo de las señales filtradas, del sujeto JIMTOM.	41
5.3. Diferencias en el dominio del tiempo de las señales filtradas, del sujeto NICANA.	41
5.4. Distribución de las diferencias de las señales filtradas, en el dominio del tiempo, para los sujetos ANDGAR, JIMTOM y NICANA.	44
5.5. Diferencias en el dominio de la frecuencia de las señales filtradas, del sujeto ANDGAR.	45
5.6. Diferencias en el dominio de la frecuencia de las señales filtradas, del sujeto JIMTOM.	45
5.7. Diferencias en el dominio de la frecuencia de las señales filtradas, del sujeto NICANA.	46
5.8. Diferencias de las señales corregidas, del sujeto ANDGAR.	46
5.9. Diferencias de las señales corregidas, del sujeto JIMTOM.	47
5.10. Diferencias de las señales corregidas, del sujeto NICANA.	47
5.11. Distribución de las diferencia de las señales corregidas, para los sujetos ANDGAR, JIMTOM y NICANA.	49

5.12. Diferencias de las señales de parpadeos, del sujeto ANDGAR.	49
5.13. Diferencias de las señales de parpadeos, del sujeto JIMTOM.	50
5.14. Diferencias de las señales de parpadeos, del sujeto NICANA.	50
5.15. Distribución de las diferencia de las señales de parpadeos, para los sujetos ANDGAR, JIMTOM y NICANA.	52
5.16. Superposición de las PSD obtenidas por R y Matlab, para todas las señales filtradas del sujeto ANDGAR.	53
5.17. Superposición de las PSD obtenidas por R y Matlab, para todas las señales filtradas del sujeto JIMTOM.	54
5.18. Superposición de las PSD obtenidas por R y Matlab, para todas las señales filtradas del sujeto del sujeto NICANA.	54
5.19. Distribución de las diferencias de los índices AI, AW, EI y MI, para los sujetos ANDGAR, JIMTOM y NICANA.	58

Índice de tablas

5.1. Error obtenido en el dominio del tiempo de las señales filtradas, de algunas tareas del sujeto ANDGAR.	42
5.2. Error obtenido en el dominio del tiempo de las señales filtradas, de algunas tareas del sujeto JIMTOM.	42
5.3. Error obtenido en el dominio del tiempo de las señales filtradas, de algunas tareas del sujeto NICANA.	43
5.4. Error obtenido en el dominio de la frecuencia, del sujeto ANDGAR.	43
5.5. Error obtenido en el dominio de la frecuencia, del sujeto JIMTOM.	43
5.6. Error obtenido en el dominio de la frecuencia, del sujeto NICANA.	44
5.7. Error obtenido de las señales corregidas, para algunas tareas del sujeto ANDGAR.	47
5.8. Error obtenido de las señales corregidas, para algunas tareas del sujeto JIMTOM.	48
5.9. Error obtenido de las señales corregidas, para algunas tareas del sujeto NICANA.	48
5.10. Error obtenido de las señales de parpadeos, para algunas tareas del sujeto ANDGAR.	50
5.11. Error obtenido de las señales de parpadeos, para algunas tareas del sujeto JIMTOM.	51
5.12. Error obtenido de las señales de parpadeos, para algunas tareas del sujeto NICANA.	51
5.13. Comparación del IAF obtenido en R y Matlab, para los tres sujetos.	55
5.14. Valor absoluto de la diferencia entre los índices calculados por R y Matlab, para algunas tareas del sujeto ANDGAR.	56
5.15. Valor absoluto de la diferencia entre los índices calculados por R y Matlab, para algunas tareas del sujeto JIMTOM.	56
5.16. Valor absoluto de la diferencia entre los índices calculados por R y Matlab, para algunas tareas del sujeto NICANA.	57

Capítulo 1

Introducción al trabajo

1.1. Introducción

El ser humano siempre ha experimentado una preocupación constante por indagar y dar explicación a muchos interrogantes relacionados con el cerebro humano y su funcionamiento. El cerebro humano es el centro del sistema nervioso y gran parte de sus funciones fisiológicas implica recibir información del entorno o del propio cuerpo, interpretarla y guiar la respuesta del cuerpo. También desempeña funciones vitales como respirar o el ritmo cardíaco, pasando por las funciones más básicas como dormir o tener hambre, hasta las funciones superiores como pensar, recordar o hablar.

Es por esta preocupación por la que el ser humano ha ido, a lo largo de la historia, desarrollando técnicas y metodologías para poder estudiar el cerebro cada vez con mayor profundidad. Entre estas técnicas se encuentra la *electroencefalografía*, que se encarga de capturar la actividad eléctrica que se produce en el cerebro, mediante la sincronización de las neuronas que lo conforman. Al registro obtenido se le denomina *electroencefalograma* (EEG), y de él se obtiene una gran fuente de información acerca del sujeto sobre el que se realiza, pues es dependiente del estado y salud del sujeto en el momento de su captación. Por este motivo, uno de los campos de estudio más relevantes en los que se aplica esta técnica es en el diagnóstico de enfermedades. Aunque existen otros campos de estudio, como son el deporte o el *marketing*.

El proceso de captación del electroencefalograma ha ido evolucionando y mejorando a lo largo de los años, y es muy importante que se realice de la manera más correcta posible. Sin embargo, no hay que olvidar que para obtener la información que proporciona, es necesario contar con un proceso de análisis con el cual se puedan realizar los procedimientos adecuados y necesarios para la obtención de dicha información. Durante los años han ido apareciendo distintos procesos de análisis que evalúan, de manera objetiva e intuitiva, qué le sucede a un determinado sujeto cuando se le somete a un determinado estímulo o qué grado de atención pone el sujeto cuando está realizando una determinada prueba, por ejemplo.

Por esto, resulta interesante contar con herramientas que sirvan de apoyo y permitan la automatización de procedimientos y acciones que obtengan la información que proporciona el electroencefalograma, para que un experto en la materia pueda analizar los resultados obtenidos.

1.2. Hipótesis del trabajo y objetivo

La hipótesis principal de este trabajo es que a través de las señales proporcionadas por el EEG se puede obtener información objetiva sobre el estado de un sujeto, como puede ser la carga cognitiva y la carga emotiva que experimenta en la realización de determinadas tareas y ante la presencia de determinados estímulos.

Se trata de la continuación del Trabajo Fin de Grado (TFG) titulado *Análisis de EEG para la evaluación de las emociones* [29] cuyo objetivo principal es el diseño, desarrollo y análisis de diferentes técnicas que permitieran extraer medidas objetivas de la señal del EEG. En este trabajo se hizo uso de la herramienta EEGLab [11] desarrollada para el procesamiento de las señales del EEG para el entorno matemático Matlab.

El objetivo principal de este Trabajo Fin de Grado consiste en realizar el diseño e implementación de una **herramienta web de software libre** que permita la obtención de medidas objetivas de la señal del EEG, para poder cuantificar el esfuerzo cognitivo de un sujeto en el desarrollo de determinadas tareas y las emociones que este expresa ante la presencia de ciertos estímulos. Así como poder visualizar la señal del EEG y todos los valores intermedios necesarios para la obtención de dichas medidas objetivas.

Dicho objetivo se puede dividir en los siguientes subobjetivos:

- Analizar y estudiar distintos filtros de señales para el preprocesamiento de la señal EEG.
- Analizar y estudiar distintas herramientas para la visualización de las señales.
- Implementar los procedimientos y algoritmos para el preprocesamiento de la señal EEG que permitan la eliminación de ruido y artefactos.
- Implementar todo el proceso para el cálculo y obtención de la información e índices cognitivos y emotivos de la señal procedente del EEG.
- Diseñar y desarrollar una plataforma web de software libre que permita realizar, paso por paso y de manera automática, la obtención de las medidas objetivas. Así como la visualización de las señales y valores intermedios.
- Realizar una evaluación de los resultados obtenidos, en comparación con los obtenidos en el TFG [29].

1.3. Contexto del trabajo

El contexto de este TFG se encuentra dentro de las líneas de investigación de los grupos de investigación *Neurociencia clínica y experimental (NiCE)* de la Facultad de Medicina de la Universidad de Murcia, dirigido por la profesora Dra. Dña María Trinidad Herrero Ezquerro y *Artificial Intelligence Knowledge Engineering (AIKE)* de la Facultad de Informática de la Universidad de Murcia, dirigido por el profesor Dr. D. José Tomás Palma Méndez.

1.4. Estructura de la memoria

Esta memoria se encuentra estructurada en seis capítulos, en los que se detallan lo siguiente:

- **Capítulo 1.** El primer capítulo es el presente capítulo y en él se realiza la introducción al trabajo, su hipótesis y objetivo, el contexto en el que se encuentra y la estructura que tiene la memoria.
- **Capítulo 2.** En el segundo capítulo se hace una breve introducción al EEG y a la relación de índices que se pueden obtener a partir de la señal EEG. También se realiza el estudio sobre el estado del arte del filtrado de las señales, las tecnologías existentes para el procesamiento de señales y el desarrollo web, y las distintas herramientas de visualización de señales.
- **Capítulo 3.** En el tercer capítulo se expone la metodología que se ha seguido durante el desarrollo del trabajo, el desarrollo web realizado (desde las características del framework Shiny hasta su uso para el desarrollo de la herramienta web), la funcionalidad con la que cuenta la herramienta y las funciones que se han tenido que implementar para el preprocesamiento de las señales del EEG.
- **Capítulo 4.** El cuarto capítulo describe todas las acciones y procedimientos para la obtención de la información que proporciona la señal del EEG
- **Capítulo 5.** En el quinto capítulo se realiza la evaluación de los resultados obtenidos, describiendo los procedimientos utilizados, analizando los resultados en comparación con los obtenidos en [29] y exponiendo las conclusiones a las que se llega tras la evaluación.
- **Capítulo 6.** En el último y sexto capítulo se exponen las conclusiones a las que se ha llegado tras la resolución de este trabajo y las posibles vías futuras.

Capítulo 2

Fundamentos y Estado del Arte

2.1. Introducción

En este capítulo se va a realizar una pequeña introducción a los fundamentos del electroencefalograma, así como una breve explicación de los índices de estudio para este trabajo. Además, también se realiza un estudio sobre el estado del arte en diferentes campos: primero se hace un estudio sobre el filtrado de las señales; seguido del estudio sobre las distintas tecnologías existentes para el desarrollo del proyecto; y por último, se hace un estudio sobre las herramientas existentes para la visualización de señales.

2.2. Fundamentos del electroencefalograma (EEG)

El cerebro humano produce impulsos eléctricos que viajan a través de las neuronas. Estos impulsos producen ritmos que se conocen como ondas cerebrales, y que contienen información que viaja de neurona a neurona para poder ejecutar una función determinada.

La *electroencefalografía* es una técnica de exploración funcional del Sistema Nervioso Central (SNC), que se encarga de obtener un registro de la actividad eléctrica producida por los impulsos eléctricos del cerebro [31]. Este registro de la actividad cerebral a lo largo del tiempo se denomina *electroencefalograma* (EEG). Para obtener el EEG se hace uso de electrodos que se sitúan en el cuero cabelludo (en su versión no invasiva) y que miden la actividad eléctrica de la corteza cerebral. La señal que se obtiene es tan pequeña que se requiere de sistemas de ampliación diferenciales, es decir, sistemas que reciben el impulso eléctrico de dos puntos y magnifican la diferencia de potencial entre ellos [33]. Existen diversas maneras de colocación de los electrodos, y de la colocación depende la correcta medición del EEG; sin embargo, el método más utilizado en la actualidad consiste en colocar los electrodos siguiendo el sistema internacional 10-20 establecido por la *International Federation of EEG*.

El EEG destaca sobre otras técnicas de neuroimagen, como pueden ser la Tomografía Axial Computerizada (TAC), la Resonancia Magnética Nuclear (RMN) o la Resonancia Magnética Funcional (RMf), por su bajo coste, su buena resolución temporal (la señal se toma cada milisegundo), la facilitación de la adquisición, el análisis y almacenamiento de la señal y la posibilidad de modificar, tanto durante la grabación como *a posteriori*, parámetros como filtros, sensibilidad, tiempo de registro y montajes. Además, permite libertad de movimientos al sujeto, ya que este puede moverse en la estancia donde se realiza la prueba [31].

Una de las partes más importantes en el estudio del electroencefalograma es el referido a la eliminación de los *artefactos*. Los artefactos son señales que se obtienen en el período de captación del electroencefalograma pero no pertenece a él. Existen dos tipos de artefactos [33]: los artefactos externos, que son los derivados por causas no pertenecientes al sujeto,

como pueden ser la mala colocación de los electrodos, la alteración de los parámetros de calibración o los propios electrodos; y los artefactos internos, que son los derivados por causas referentes al sujeto, como pueden ser el pulso, la respiración, la contracción de los músculos de la cara o los movimientos oculares. Para que el análisis del electroencefalograma sea lo más correcto posible, es importante que este sea lo más “puro” posible, es decir, que en él haya el mínimo número de artefactos posibles.

Dentro del ámbito médico, el estudio del EEG puede proporcionar gran cantidad de información clínicamente relevante tanto en patologías epilépticas como no epilépticas, así como en patologías en el coma y en la muerte cerebral, entre otras. Fuera del ámbito médico, también tiene aplicación en el ámbito del deporte, la computación y el *neuromarketing*. Sin embargo, es importante dejar claro que, independientemente del ámbito de estudio, el electroencefalograma no aporta información acerca de la inteligencia del sujeto.

La señal EEG está constituida por un gran conjunto de ondas de diferentes frecuencias, donde cada frecuencia proporciona una información determinada. Los rangos de frecuencia de la señal EEG se pueden dividir en las siguientes cuatro bandas [40, 30]:

- **Banda delta (δ)**: de 0,5 a 4 Hz. Se trata de la banda de frecuencias más baja (no llega a cero, pues significaría la muerte cerebral) que aparece cuando el sujeto se encuentra en un estado de sueño profundo.
- **Banda theta (θ)**: de 4 a 7,5 Hz. La banda *theta* aparece cuando el sujeto se encuentra en un estado de calma profunda o sueño ligero. También aparece cuando realiza tareas que el sujeto tiene automatizadas y no necesita tener un control consciente de su ejecución.
- **Banda alpha (α)**: de 7,5 a 12,5 Hz. Se trata de la banda más predominante cuando el sujeto, una persona adulta y mentalmente sana, se encuentra despierto, en reposo, relajado, sin actividad mental alguna y con los ojos cerrados. La banda *alpha* se atenúa conforme se realiza actividad mental o se abre los ojos.
- **Banda beta (β)**: de 12,5 a 30 Hz. A medida que el sujeto realiza actividad mental intensa (como puede ser estudiar o dar un discurso) o este se encuentra en un estado de alerta, esta banda es la que más predomina. Es decir, al contrario que la banda *alpha*, la banda *beta* predomina cuando la persona se encuentra despierta y en estado de excitación. Tiene relación con la función senso-motora y ciertas frecuencias predominantes tienen relación con algunas patologías o efectos de fármacos. Además, la presencia de esta banda en pacientes en coma es signo de buen pronóstico.

La Figura 2.1 muestra un resumen de las bandas mencionadas.

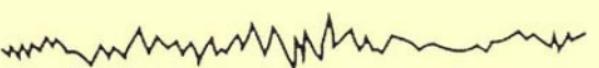
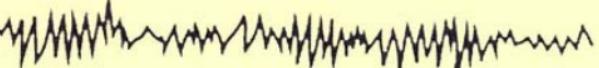
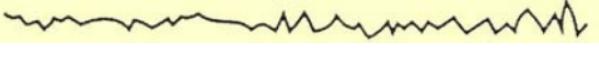
Ondas cerebrales	Frecuencia	Estado mental
Onda delta	0,5 - 3 Hz 	sueño profundo
Onda theta	4 - 7 Hz 	sueño ligero
Onda alfa	8 - 13 Hz 	despierto, relajado
Onda beta	14 Hz 	despierto, excitado

FIGURA 2.1: Resumen de las bandas $\delta, \theta, \alpha, \beta$ en las que se divide el EEG.

2.3. Relación de índices de estudio

Del conjunto de bandas mencionadas en la sección 2.2 se puede extraer información sobre el sujeto de estudio. Esta información se obtiene a través de una serie de índices, obtenidos a partir de una determinada banda o a partir de un conjunto de ellas. Los índices de estudio en los que se centrará este trabajo son:

- **Índice de atención** [35, 30, 36]. El interés que un sujeto ha dedicado a la realización de una tarea o a la recepción de un estímulo, realizando procesos de búsqueda y recuperación de las estructuras de memoria del cerebro a largo plazo, se obtiene a partir del Índice de Atención (*AI, Attention Index*), que está basado en el estudio de la banda *alpha*.
- **Índice de aproximación-rechazo** [34, 37]. Cuando un sujeto realiza una tarea o se somete a un estímulo, esto produce en él un determinado grado de placer, que se mide con el Índice de Aproximación-Rechazo (*AW, Approach-Withdrawal Index*), que, al igual que el índice *AI*, está basado en el estudio de la banda *alpha*.
- **Índice de compromiso** [39, 38]. La atención global que un sujeto presta a una determinada tarea se mide con el Índice de Compromiso (*EI, Engagement Index*). Difiere del índice *AI* en que el *EI* mide la capacidad de respuesta a la variación del estímulo recibido, mientras que el *AI* lo que mide es la capacidad de recuperación de información previamente memorizada. Está basado en el estudio de las bandas *alpha, beta y theta*.
- **Índice de memorización** [35, 30, 36]. El grado en el que un sujeto ha sido capaz de crear nuevas asociaciones en las estructuras de memoria del cerebro, se mide con el Índice de Memorización (*MI, Memorization Index*), que se basa en el estudio de la banda *theta*.

2.4. Filtrado de señales

Una vez que la señal del EEG se ha obtenido y almacenado, podemos usar un filtrado digital sobre ella. En esta sección, se va a hablar sobre los diferentes tipos de filtrado que se le puede aplicar a una señal, así como de los distintos filtros que existen y cuál de ellos se ha elegido para la realización del filtrado de las señales en este trabajo.

2.4.1. Tipos de filtrado

Filtrar, en su sentido más general, significa eliminar las partes del espectro de una señal. Dependiendo de la parte que se quiera eliminar, existen diversos tipos básicos de filtros [3, 2]:

- **Filtro paso-bajo.** Con este tipo de filtro, se permite que las frecuencias más bajas puedan pasar, lo que implica la eliminación de las frecuencias más altas. La Figura 2.2 muestra el efecto que provoca este tipo de filtro.

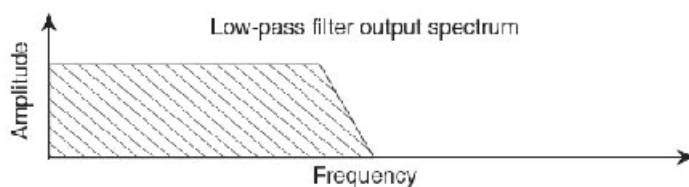


FIGURA 2.2: Efecto de un filtro paso-bajo aplicado a una señal.

- **Filtro paso-alto.** Al contrario del filtro anterior, con este se permite que las frecuencias más altas pasen, eliminando así las frecuencias más bajas. La Figura 2.3 muestra el efecto que provoca este tipo de filtro.

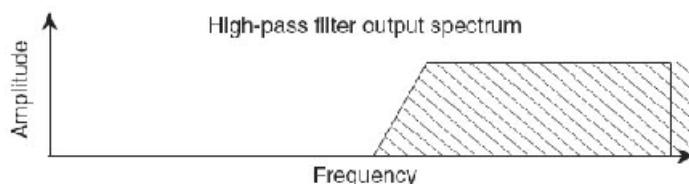


FIGURA 2.3: Efecto de un filtro paso-alto aplicado a una señal.

- **Filtro pasa-banda.** Este tipo de filtro es una combinación de los dos filtros anteriores. Esto implica que solamente las frecuencias que estén dentro de cierto rango (banda) puedan pasar el filtro. La Figura 2.4 muestra el efecto que provoca este tipo de filtro.

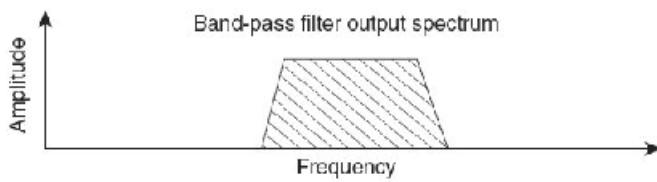


FIGURA 2.4: Efecto de un filtro pasa-banda aplicado a una señal.

- **Filtro banda-rechazada.** Este filtro es el contrario al filtro pasa-banda, deja pasar todas las frecuencias que se encuentren fuera de un cierto rango (banda). Si la banda de frecuencias es muy estrecha y muy atenuada sobre unos pocos hertzios, entonces el filtro se denomina más comúnmente filtro de corte (*notch-filter*). La Figura 2.5 muestra el efecto que provoca este tipo de filtro.

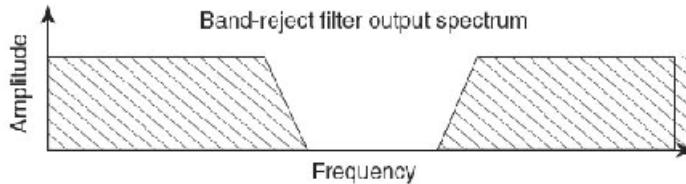


FIGURA 2.5: Efecto de un filtro banda-rechazada aplicado a una señal.

2.4.2. Filtros digitales

Cualquier señal puede considerarse como un gran conjunto de ondas seno y coseno de diferentes frecuencias, amplitudes y fases. Un filtro digital es un procesador de señal digital que atenúa las componentes de una señal de una determinada frecuencia (banda). La razón común, detrás del filtrado en general, está en atenuar el ruido de las grabaciones, mientras se preserva la señal de interés. Sin embargo, es imposible diseñar un filtro que no modifique la señal, por lo que no existe un filtro ideal.

Existen dos categorías de filtros que son usados para aproximar un comportamiento ideal, cada uno con sus ventajas e inconvenientes. Estos son:

2.4.2.1. Filtro de Respuesta Finita al Impulso (FIR, *Finite Impulse Response*)

El filtro de respuesta finita al impulso (FIR) es un filtro que, para calcular un valor de salida, hace uso del valor de entrada actual y de un conjunto finito de valores de entrada anteriores. Esto implica que el filtro FIR sea un filtro no recursivo [41].

Debido a que la salida es calculada usando un número limitado de entradas, la respuesta al impulso es multiplicada por una ventana de longitud finita (por ejemplo, Hamming). Esto proporciona un número finito de coeficientes que son necesarios para el cálculo de la salida. El orden del filtro lo determina dicho número de coeficientes, o lo que es lo mismo, el tamaño de la ventana [9]. Así pues, para un tamaño de ventana N , un filtro FIR viene dado por la ecuación 2.1 [7]:

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + \dots + b_{N-1} \cdot x(n-N+1) = \sum_{k=0}^{N-1} b_k \cdot x(n-k) \quad (2.1)$$

donde los $\{b_k\}$ son los coeficientes del filtro.

El tamaño de la ventana también determina el retardo. Cuando una señal se introduce en un filtro FIR, la salida tendrá desplazada su fase en comparación con la entrada. El desplazamiento de fase es igual a la mitad de la longitud de la ventana, multiplicado por el tiempo de muestreo; y cada frecuencia de la señal será desplazada con la misma constante de desplazamiento de fase.

Las **ventajas** que tienen los filtros FIR son las siguientes [9, 5, 1]:

- La salida de un filtro FIR siempre es estable, ya que son filtros no recursivos.
- Tienen un retardo que es constante en todas las frecuencias.
- El tamaño de la señal no se verá influenciado por el desplazamiento de fase.

- Son simples de implementar. La mayoría de los microprocesadores de procesamiento de señales digitales pueden hacer el cálculo del filtro en una simple instrucción.
- No producen distorsiones de fase lineal.

Por contra, las **desventajas** que presentan son [9, 5, 1]:

- Son ineficientes, pues usan muchas entradas para calcular la salida.
- Lo anterior conlleva a que tengan un gran retardo.
- Requieren un mayor consumo de memoria y coste computacional.

En el contexto de las mediciones del EEG, este filtro podría utilizarse como filtros de paso alto o paso bajo. Además, son especialmente adecuados si la característica de fase de la señal es muy importante y, por lo tanto, puede utilizarse para medidas de coherencia y sincronía.

2.4.2.2. Filtro de Respuesta Infinita al Impulso (IIR, Infinite Impulse Response)

Un filtro de respuesta infinita al impulso (IIR) emplea un enfoque distinto al FIR en el cálculo de la salida. El filtro IIR es recursivo: utiliza la entrada actual junto con un conjunto de salidas previas (además del conjunto de entradas previas), para calcular la salida actual. El orden del filtro está determinado por el máximo entre el tamaño de ventana (igual que el filtro FIR) y el número de salidas previas utilizadas en el cálculo de la salida. Naturalmente, hay una desventaja en el incremento del orden del filtro: mayor número de salidas previas para el cálculo de la salida implica que el filtro sea menos eficiente. Esto se ve en una degradación de la característica de fase [9].

Así pues, para un tamaño de ventana M y un número N de salidas previas, un filtro IIR viene dado por la ecuación 2.2 [8]:

$$y(n) = \sum_{k=0}^M b_k \cdot x(n-k) - \sum_{k=1}^N a_k \cdot y(n-k) \quad (2.2)$$

donde el orden es igual al máximo de M y N .

Las **ventajas** que tienen los filtros IIR son las siguientes [9, 5, 1]:

- Son muy eficientes, pues usan pocos valores de salida previos para el cálculo de la salida actual.
- Lo anterior conlleva que tengan un retardo pequeño.
- Debido a su eficiencia son utilizados para el filtrado de señales en tiempo real.

Por contra, las **desventajas** que presentan son [9, 5, 1]:

- Pueden convertirse en inestables si los coeficientes no son bien elegidos.
- La respuesta al impulso puede no ser simétrica. Esto implica que la característica de fase no sea constante en todas las frecuencias. Por lo que, diferentes frecuencias aparecerán en la salida con diferentes desplazamientos de fase. La forma de la señal de salida cambiará en consecuencia a los distintos desplazamientos de fase.

El uso de filtros IIR es apropiado cuando se permite un pequeño retardo o cuando son necesarios fuertes requerimientos en la característica de amplitud. Mientras que solo hay un único filtro FIR, existen varios tipos de filtros IIR. Entre ellos, destacan: Butterworth, Chebyshev y Elliptic [9].

2.4.3. Elección del filtro adecuado para este trabajo

Los filtros IIR son filtros que presentan respuestas al impulso asimétricas y de fase no lineal. Estas respuestas al impulso se encuentran implementadas de manera implícita, en una forma funcional definida por los coeficientes y el estado de las variables del filtro [10]. Además, la inestabilidad que pueden presentar se debe a que, durante la ejecución del filtro se van acumulando errores de redondeo en las variables del filtro, y que hacen que la respuesta del filtro se vea afectada. Por este motivo, es necesario que se esté comprobando la estabilidad en su ejecución, sobre todo si se necesita un corte de frecuencias estricto.

En cuanto a los filtros FIR, estos pueden presentar tanto respuestas al impulso simétricas y de fase lineal, como asimétricas y de fase no lineal. Y su implementación se realiza como una convolución de la entrada del filtro con la respuesta al impulso; los coeficientes del filtro FIR son idénticos a la respuesta al impulso, por lo que se puede decir que la longitud de un filtro FIR es la longitud de la respuesta al impulso [10]. En el ámbito del EEG, la gran mayoría de los filtros FIR son filtros simétricos de longitud impar, ya que solo a los filtros FIR de longitud impar, se les puede corregir el desplazamiento de fase debido a que el grupo de desplazamientos es un número entero de muestras.

El orden de los filtros IIR y FIR no se pueden comparar, ya que los filtros IIR son recursivos; sin embargo, es importante destacar que el orden de los filtros IIR suele ser siempre menor que el de los filtros FIR y que para un determinado filtrado, si el orden del filtro IIR es elevado, si se quiere realizar con un filtro FIR, el orden será más elevado todavía.

Sin embargo, el resultado de la longitud de la respuesta al impulso sí puede ser comparada. A pesar de que los filtros IIR están normalmente considerados computacionalmente más eficientes, solo son recomendados cuando se necesita un alto rendimiento y cortes definidos [10]. En el ámbito del EEG, el rendimiento solo es relevante durante la grabación. Para una análisis *off-line*, sin embargo, el rendimiento y el tiempo computacional no importan con los componentes hardware modernos.

Los filtros FIR son más fáciles de controlar, más estables, tienen una bien definida pasa banda y se puede corregir el desplazamiento de fase sin cómputo adicional. Gracias a los componentes hardware modernos, su consumo computacional y de memoria no son significativos [10]. Por lo tanto, para el propósito de este trabajo, se ha elegido un filtro FIR para el filtrado de las señales.

2.5. Tecnologías existentes

En esta sección se van a exponer las distintas herramientas y opciones para el procesamiento de señales del EEG y su posterior análisis, así como para el desarrollo de la aplicación web. Se terminará esta sección con la elección de las tecnologías para este trabajo.

2.5.1. Procesamiento de señales

Para el procesamiento de las señales, las herramientas destacadas son las siguientes:

- **Matlab** [13]. Herramienta diseñada con un lenguaje de programación propio, de alto nivel, para cálculos de ingeniería y científicos, y que cuenta con un entorno de desarrollo integrado. Cuenta con numerosos plugins (integrados y complementarios), como el plugin EEGLAB [11] para el procesamiento de señales del EEG. Se trata de una herramienta con licencia privativa.

- **R** [14]. Entorno matemático con un lenguaje de programación propio y cuya implementación es de software libre. En este entorno no se puede utilizar EEGLAB, aunque cuenta con herramientas para procesamiento de señales, como *eegkit* [16].
- **Octave** [15]. Esta herramienta se considera la alternativa libre a Matlab, se trata de un entorno matemático cuya implementación es de software libre, perteneciente al proyecto GNU. Puede ejecutar funciones de Matlab, aunque no todas son compatibles. EEGLAB puede ejecutarse parcialmente en este entorno.

2.5.2. Desarrollo web

Las distintas opciones que destacan en el desarrollo web se dividen en dos grupos (todas de software libre):

- **Tecnologías Frontend.** Para la parte *frontend*, se encuentran las siguientes tecnologías:
 - **HTML, CSS y JavaScript** [17, 18, 19]. Estas tres tecnologías permiten crear la estructura, el diseño y la interactividad de una página web, respectivamente.
 - **Framework Shiny** [42]. Se trata de un framework para el desarrollo de aplicaciones web interactivas, implementado por RStudio [21] para el entorno matemático R. Este framework permite crear aplicaciones web sin tener conocimientos previos de HTML, CSS o JavaScript, solamente con código en R.
- **Tecnologías Backend.** Para la parte *backend*, se encuentran las siguientes tecnologías:
 - **PHP** [22]. Popular lenguaje de scripting de uso general, especialmente adecuado para el desarrollo web. No cuenta con las características propias de un entorno matemático.
 - **Framework Shiny** [42]. Además de crear aplicaciones web en la parte *frontend*, también integra la parte *backend* de las aplicaciones.

2.5.3. Tecnologías escogidas

Para este trabajo, de las tecnologías mencionadas en las secciones 2.5.1 y 2.5.2, se han escogido las siguientes:

- Para el desarrollo web: **framework Shiny**.
- Para el procesamiento de señales: **entorno matemático R**.

Shiny proporciona una manera sencilla e intuitiva con la que crear el *frontend* de una aplicación web, sin que se tenga conocimientos en herramientas como HTML, CSS o JavaScript, así como la creación del *backend* de la misma aplicación. Teniendo en cuenta que todo el procesamiento de las señales se hará en la parte del servidor (*backend*), y teniendo que Shiny está implementado para R, como herramienta para el procesamiento de las señales se ha elegido R.

Otro motivo por el que el procesamiento de señales se va a realizar en R es porque permite aprovechar toda la potencia de la herramienta, en lo relativo a la parte de interpretación de los análisis estadísticos de los resultados de los experimentos, así como la facilidad que ofrece al aplicar técnicas de aprendizaje computacional. De hecho, dentro de los trabajos de investigación realizados, los datos generados por Matlab han sido analizados en R.

De esta manera, se tiene que la aplicación está implementada sobre el mismo lenguaje, tanto *frontend* como *backend*. Evitando así posibles conflictos que puedan surgir entre distintos lenguajes.

2.6. Herramientas de visualización de señales

La visualización de las señales es una de las partes más importantes del proyecto, por este motivo es importante hacer un estudio sobre las herramientas que hay para dicha visualización. Dado que el proyecto se va a desarrollar en R usando Shiny (como se ha comentado en 2.5.3), las distintas herramientas de las que dispone para la visualización de señales son:

- `plot` [23]. Es la función de visualización de datos genérica que R proporciona por defecto. Esta función requiere que se le pasen las coordenadas X e Y de los puntos a mostrar. Proporciona distintas maneras de visualizar los datos: solo los puntos, solo las líneas que unen los puntos, en forma de histograma, etc.
- `ggplot2` [24]. Se trata de una herramienta para crear gráficos de manera declarativa, siguiendo las indicaciones de The Grammar of Graphics [27]. A esta herramienta hay que proporcionarle los datos en forma de tabla, cómo mapear las variables de los datos para su visualización y qué tipo de gráfica se quiere obtener. Con esto, ella se encarga de todos los detalles de visualización. Esta herramienta tiene un gran abanico de personalización de las gráficas y permite que se pueda implementar interactividad con las gráficas (selección de datos, zoom, etc), a diferencia de la herramienta anterior.
- `highcharter` [26]. Es una herramienta implementada en JavaScript y proporciona gran cantidad de gráficas interactivas: gráficas de puntos, gráficas de series temporales, gráficas en forma de árbol, gráficas de mapas, grafos, etc. Proporciona una función genérica que es capaz de visualizar varios tipos de objetos R: tablas, vectores numéricos, histogramas, vectores de cadenas, factores, etc. Al igual que `ggplot2` permite un gran abanico de personalización de las gráficas, la diferencia entre ambas, es que esta herramienta ya proporciona la interactividad con las gráficas.
- `dygraphs` [25]. Se trata de una herramienta implementada en JavaScript para la visualización interactiva de series temporales. Proporciona una gran interactividad (zoom, realce de series/puntos, etc) y una visualización de los datos y de los ejes muy configurable, entre otras características. A esta herramienta se le pueden proporcionar los datos de dos maneras: como serie temporal o como datos numéricos. En el caso de datos numéricos, el primer campo debe contener los valores del eje X, mientras que el resto de campos corresponden a los valores de la serie.

2.6.1. Herramienta de visualización escogida

Tanto `plot` como `ggplot2` son dos herramientas potentes para la generación de gráficas, sin embargo, no están pensadas para tener interacción con ellas, aunque `ggplot2` permita implementar cierta interactividad. Es por ello que estas dos herramientas no son muy idóneas para la visualización en este proyecto.

En cambio, `highcharter` y `dygraphs` sí proporcionan gráficas interactivas. El motivo principal para la selección de una de las dos herramientas es que las señales del EEG son series temporales. Aunque `highcharter` disponga de gráficas para series temporales, tras varias pruebas en las que se visualizan datos del EEG, esta herramienta no está optimizada para series temporales con gran cantidad de datos, pues tarda en generar la gráfica cuando

se trata de gran cantidad de datos. Con los mismos datos, se probó dygraphs (que está implementada para series temporales) y se vio que esta herramienta generaba la gráfica más rápidamente, además de que su interactividad es más fluida.

Es por ello que, para la visualización de las señales, se ha escogido **dygraphs** como herramienta.

Capítulo 3

Metodología y diseño de la solución

3.1. Introducción

En este capítulo se describe la metodología seguida para el diseño de este trabajo; el desarrollo web realizado, describiendo el framework Shiny, sus características y cómo estas se han utilizado para implementar la aplicación web; la descripción de toda la funcionalidad que proporciona la aplicación; y por último, la explicación de las funciones que se han tenido que implementar para el filtrado de las señales y la corrección ocular.

3.2. Metodología seguida

La metodología que se ha seguido en el desarrollo del trabajo ha sido la siguiente:

1. **Creación de la interfaz gráfica para la carga, selección y visualización de los datos.** Previo al filtrado de las señales, es necesario que los datos puedan ser cargados, seleccionados y visualizados, para estar seguro de que todo lo anterior se hace de manera similar que en la aplicación EEGLab.
2. **Implementación del filtrado de las señales en R.** Una vez realizado lo anterior, el siguiente paso a seguir ha sido la implementación de la función `eegfilt()`, basándome en la implementación de dicha función empleada en EEGLab [6] (véase sección 3.5.1).
3. **Comparación de las señales filtradas en R con las filtradas en Matlab, así como de su respuesta en frecuencia.** Para comprobar que la implementación de la función `eegfilt()` es correcta, se ha realizado una comparación entre los resultados obtenidos tras filtrar las señales usando la implementación de EEGLab y usando la implementación en R. Además, también se ha realizado una comparación de la respuesta en frecuencia de las señales filtradas.
4. **Visualización de las señales filtradas y su respuesta en frecuencia.** Una vez que se ha realizado la comparación de las señales filtradas, se ha incluido en la aplicación tanto la visualización de las señales filtradas como su respuesta en frecuencia.
5. **Implementación de la corrección ocular en R.** Tras la comparación del filtrado, se ha proseguido con la implementación de la función `gratton()`, basándome en la implementación de dicha función empleada en EEGLab [12] y que fue modificada en el TFG [29] (véase sección 3.5.2).
6. **Comparación de la señal corregida en R con la corregida en Matlab.** Con el mismo objetivo que en el filtrado de las señales, se ha realizado una comparación entre los resultados obtenidos tras aplicar la corrección ocular usando la implementación modificada en el TFG [29] y usando la implementación en R.

7. **Visualización de las señales corregidas.** Una vez que se ha realizado la comparación de las señales corregidas, se ha incluido en la aplicación su visualización.
8. **Segmentación de la señal por tareas, cálculo de la IAF y segmentación de bandas para cada tarea.** Tras la comparación y la visualización de las señales corregidas, el siguiente paso a realizar ha sido implementar la segmentación de la señal por tareas, el cálculo de la IAF y la segmentación de las bandas α , β y θ para cada una de las tareas.
9. **Visualización de la IAF y las bandas para cada tarea.** Tras el paso anterior, se ha añadido a la aplicación la visualización de la IAF así como la visualización de las bandas α , β y θ para cada tarea.
10. **Obtención de los valores de referencia para el cálculo de los índices.** Previo al cálculo de los índices para cada tarea, es necesario obtener los valores de referencia de los índices de la fase basal.
11. **Cálculo de los índices para cada tarea.** Con los valores de referencia de los índices, se calculan todos los índices para cada una de las tareas.
12. **Visualización de los valores de referencia y los índices para cada tarea.** Tras la obtención de los índices para cada tarea, se ha incluido su visualización en la aplicación, así como la posibilidad de visualizar los valores de referencia obtenidos.
13. **Exportación de los índices obtenidos y señales empleadas en su cálculo.** Una vez calculados los índices y realizada su visualización, el penúltimo paso ha sido incluir la exportación de los valores de los índices para cada tarea, así como de las señales que se han empleado para su cálculo (normales, filtradas y corregidas).
14. **Comparación de los resultados obtenidos.** Por último, se ha hecho una comparación de los resultados obtenidos con la implementación en R y los obtenidos con la implementación en Matlab por el TFG [29].

3.3. Desarrollo web

En esta sección se va a describir el desarrollo web que se ha empleado en este proyecto, haciendo uso del framework Shiny [42]. Primero se hablará sobre la estructura de una aplicación Shiny; tras esto, se profundizará en el concepto de reactividad de Shiny y cómo Shiny proporciona la modularización del código. Por último, se detallará cómo se ha hecho uso del framework, sabiendo todo lo anterior, para desarrollar la parte web del proyecto.

3.3.1. Estructura de una aplicación Shiny

Una aplicación Shiny tiene dos componentes principales: la definición de la interfaz de usuario y el script del servidor. Estos dos componentes se tienen que definir dentro de los dos siguientes ficheros: `ui.R` y `server.R`, respectivamente [20].

El fichero `ui.R` (de aquí en adelante, *la interfaz*) es el que contiene las directrices para la definición de la interfaz de usuario: en ella se indica cómo se quiere estructurar la página web (*layout*), qué título y qué elementos tiene dicha página, tanto de entrada como de salida. Es decir, en este fichero se declaran tanto aquellos elementos que sirven para que el usuario introduzca datos, realice acciones, etc, como aquellos elementos que sirven para la modificación, visualización, etc, de los datos según las acciones del usuario. Cada uno de los elementos de la página tiene un identificador que hay que indicar en la creación del mismo.

Este identificador servirá para que el servidor pueda obtener información sobre dicho elemento.

El fichero `server.R` (de aquí en adelante, *el servidor*) es el que contiene todas las acciones para obtener los datos que después se van a visualizar en la interfaz. Al servidor, se le pasan tres objetos muy importantes: `input`, `output` y `session` (opcional). En el primero de ellos, se encuentran los valores de todos los (identificadores) elementos de entrada que se han definido en la interfaz; el segundo de ellos, contiene todos los (identificadores) elementos de salida, en ellos se almacenarán los resultados esperados para cada elemento de salida; el último de los objetos contiene la información de toda la sesión que el servidor tiene con el cliente.

La Figura 3.1 muestra una gráfica de la estructura general de una aplicación Shiny, con algunos de los elementos que se pueden emplear.

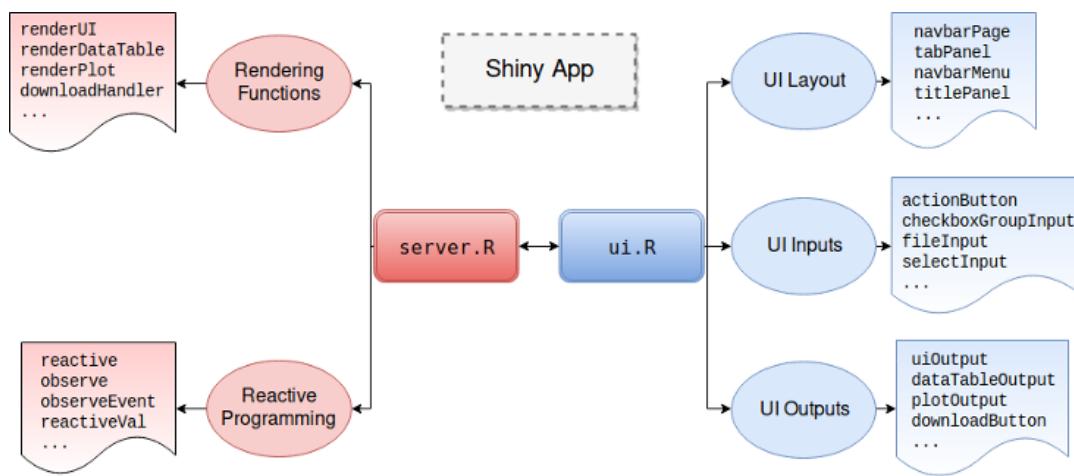


FIGURA 3.1: Estructura general de una aplicación Shiny, con algunos elementos.

Una vez que se han descrito los componentes principales de una aplicación Shiny, cabe preguntarse lo siguiente: ¿cómo se realiza la interacción entre la interfaz y el servidor para que cuando se modifique un valor de entrada de la interfaz, automáticamente se actualicen todos los elementos de salida asociados a ese valor? Para ello, Shiny proporciona lo que se denomina **reactividad**.

3.3.2. Reactividad en Shiny

Una de las grandes características del framework Shiny es que facilita la conexión de los valores de entrada (`input`) de una página web, haciéndolos fácilmente accesibles en el código del servidor (código en R) y que los resultados obtenidos en el código puedan ser escritos como valores de salida (`output`) hacia la página web [43]. Esta conexión se ve en la Figura 3.2.



FIGURA 3.2: Conexión entre valores de entrada, código en R y valores de salida en una aplicación Shiny.

Dado que las aplicaciones Shiny son aplicaciones interactivas, los valores de entrada pueden cambiar en cualquier momento. Por lo tanto, los valores de salida deben ser actualizados

para reflejar esos cambios. Para ello, Shiny cuenta con una librería de programación reactiva, que puede ser utilizada para la estructuración de la lógica de la aplicación. Usando esta librería, los cambios de los valores de entrada provocarán que las partes del código en R que proporcionan los valores de salida asociados, se vuelvan a ejecutar, haciendo que las salidas sean actualizadas.

En Shiny, existen tres tipos de elementos en la programación reactiva: fuentes reactivas, conductores reactivos y puntos finales reactivos, cuyos símbolos se encuentran en la Figura 3.3.

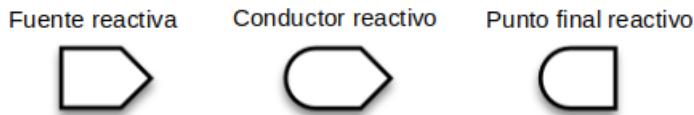


FIGURA 3.3: Símbolos de los tipos de objetos en la programación reactiva de Shiny.

- **Fuentes reactivas.** Las fuentes reactivas hacen referencia a todas las entradas del usuario a través de la interfaz del navegador. Por ejemplo, seleccionar un ítem o clickar un botón son acciones que establecerán valores que son fuentes reactivas. Estos elementos tienen elementos dependientes de ellos, como puntos finales o conductores.
- **Puntos finales reactivos.** Los puntos finales reactivos, por el contrario, son cualquiera de los elementos de salida que aparece en el navegador, como una tabla o una gráfica. Estos elementos dependen de otros objetos, como fuentes o conductores.
- **Conductores reactivos.** Los conductores reactivos son elementos que se encuentran entre los dos anteriores. Un conductor puede ser tanto dependiente de algún elemento, como tener elementos que dependan de él. Es decir (viendo la estructura reactiva como un árbol), mientras que las fuentes solamente pueden ser los padres de la estructura reactiva y los puntos finales solo pueden ser los hijos de dicha estructura, los conductores pueden ser tanto padres como hijos.

Son elementos muy importantes para encapsular operaciones lentas o computacionalmente costosas y que estas se ejecuten solamente una vez, aunque de ellas dependan varios elementos. Esto es: supongamos que tenemos una fuente de la que depende un cálculo computacionalmente costoso y que de su resultado dependen varios puntos finales. Si no utilizamos conductores, cada vez que cambie la fuente el cálculo se realizará para cada uno de los puntos finales. En cambio, si utilizamos conductores, podemos hacer que un conductor dependa de la fuente y realice el cálculo, y que los puntos finales dependan del conductor. De esta manera, cuando la fuente cambie se ejecutará el cálculo en el conductor **una vez** y su resultado irá a todos los puntos finales por igual.

Con esto se consigue que la aplicación sea más eficiente.

Las Figuras 3.4 y 3.5 muestran las estructuras reactivas cuando se utiliza solamente una fuente y un punto final, y cuando se utiliza una fuente, un conductor y varios puntos finales (nótese la importancia de la orientación de las flechas: *fuente → puntofinal*, ya que estas indican que el punto final depende de la fuente).



FIGURA 3.4: Estructura reactiva con una fuente y un punto final.

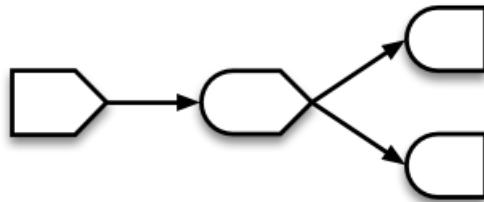


FIGURA 3.5: Estructura reactiva con una fuente, un conductor y dos puntos finales.

Las fuentes no solo pueden estar conectadas a un punto final (o conector), sino que pueden estar conectadas a múltiples puntos finales (o conectores) y viceversa (lo mismo pasa con los conectores).

Los elementos mencionados anteriormente son términos generales que desempeñan una funcionalidad particular dentro de un programa reactivo. Concretamente, Shiny tiene una clase de objetos que actúan como fuentes reactivas, una clase de objetos que actúan como conductores reactivos y otra clase de objetos que actúan como puntos finales reactivos. Estas clases concretas son:

- **Valores reactivos.** Los valores reactivos son la implementación de las fuentes reactivas. Contienen los valores que pueden ser leídos por otros objetos reactivos (como expresiones u observadores). Concretamente, el objeto `input` que recibe el servidor (3.3.1) contiene a todos los valores reactivos de la aplicación.
- **Expresiones reactivas.** Las expresiones reactivas son la implementación de los conductores reactivos. Pueden acceder a los valores reactivos o a los valores de otras expresiones reactivas, y devuelven el valor calculado en la expresión. Concretamente, el objeto `output` que recibe el servidor (3.3.1) contiene todas las expresiones reactivas de la aplicación.

Para ser más eficientes, las expresiones reactivas cachean su resultado. La diferencia que hay entre los conductores reactivos y esta implementación, es que los primeros no cachean los resultados necesariamente, mientras que las expresiones reactivas sí lo hacen.

Una expresión reactiva está pensada para cachear los resultados de cualquier operación que ocurre cuando los valores de entrada cambian. Algunas de estas operaciones pueden ser: leer datos de un fichero, cargar los datos para una gráfica, realizar el filtrado de una señal, etc.

- **Observadores.** Los observadores son la implementación de los puntos finales reactivos. Tienen acceso tanto a valores como expresiones reactivas. No devuelven ningún resultado y son utilizados para realizar efectos secundarios ante cambios en los valores reactivos, como puede ser enviar los datos al navegador.

Con estos elementos, Shiny proporciona un mecanismo sencillo e intuitivo para que cuando un elemento de entrada (valor reactivo) varíe, se actualicen los valores de salida asociados (expresiones reactivas/observadores), necesario para cualquier aplicación interactiva. Es importante destacar que no es necesario indicar explícitamente en R qué relaciones existen entre qué elementos y qué hacer cuando uno de los elementos de entrada cambia, pues es Shiny quien maneja automáticamente todos estos detalles.

3.3.3. Modularización de código en una aplicación Shiny

En la sección 3.3.1 se ha comentado que una aplicación Shiny se encuentra dividida en dos ficheros: `ui.R` y `server.R` que representan la interfaz y el servidor de la aplicación. Cuando se trata de aplicaciones simples, con poca funcionalidad, suele ser suficiente tener solamente esos dos ficheros. Sin embargo, cuando se trata de aplicaciones más grandes y complejas, con varias ventanas (cada una con una función distinta), no suele ser una buena técnica de programación tenerlo todo en un único fichero. Esto implica, tener que modularizar el código de la aplicación.

Shiny da la opción de poder modularizar el código de una aplicación [44]. Un módulo Shiny no puede ser ejecutado de manera independiente (como una aplicación Shiny) y se incluye como parte de una aplicación Shiny o como parte de otro módulo (también permite la composición de módulos). Un módulo puede ser solamente para la interfaz, para el servidor o para ambas, y puede tener la complejidad que se desee: desde la ejecución y visualización de una tabla, a una interfaz con varias pestañas, cada una con varios elementos de entrada y salida, y su correspondiente código de obtención de los elementos de salida en el servidor.

De manera similar a como se crea una aplicación Shiny, un módulo consta de dos partes fundamentales (dos funciones): la interfaz y la lógica del servidor que utiliza dicha interfaz. Si se quiere realizar un módulo que solo forme parte de la interfaz, la lógica del servidor debe estar vacía (y viceversa si se quiere utilizar como módulo para la lógica del servidor). Debido a que los diferentes elementos de la interfaz tienen un identificador, para evitar conflictos de nombres entre módulos, cada módulo tiene su propio *namespace*, que permite que todos los identificadores de ese módulo sean únicos.

Shiny también controla automáticamente la reactividad entre los valores reactivos, las expresiones reactivas y observadores para los módulos, haciendo uso del *namespace* del módulo.

Para poder crear la interfaz de un módulo, necesita que, a la función, se le pase como argumento el identificador que tendrá el módulo para que pueda crear su *namespace*. Este identificador pertenece al *namespace* de la aplicación, por lo que debe ser único para cada módulo. Esta función debe tener como nombre, el nombre que se quiera dar al módulo seguido de cualquiera de los siguientes sufijos: `Input`, `Output` o `UI`. La creación de la interfaz de este módulo se hace de manera similar a como se hace en la interfaz `ui.R`, aunque con la diferencia de que los identificadores tienen que indicarse haciendo uso del *namespace* del módulo.

En cuanto al servidor, su función debe tener el mismo nombre que su correspondiente interfaz, pero sin el sufijo. Es decir, si tenemos, por ejemplo, que el módulo se llama `module`, la interfaz tendrá el nombre de `moduleUI`, mientras que el servidor tendrá el nombre `module`. La función el servidor recibe como parámetros los mismos parámetros que al servidor `server.R`: `input`, `output` y `session`. Aunque se puede ampliar para que reciba

como parámetros los datos que necesita el módulo para su ejecución. Igual que en el servidor `server.R`, el objeto `input` contiene todos los valores reactivos, mientras que el objeto `output` contiene todas las expresiones reactivas para **ese módulo**.

Para hacer uso de los módulos en la aplicación, se hace de manera diferente en la interfaz y en el servidor de la aplicación: en la interfaz solamente es necesario llamar a la función que contiene la interfaz del módulo (en el ejemplo anterior, se llamaría a la función `moduleUI`), pasándole como argumento el identificador del módulo. En el servidor, en cambio, Shiny proporciona la función `callModule()`, en el que se indica el nombre del módulo (en el ejemplo, `module`) seguido del mismo identificador indicado en la interfaz (si un módulo requiere parámetros extra para su funcionalidad, es en esta función donde se han de proporcionar dichos parámetros, detrás del identificador).

Al tratarse de una función, el servidor de un módulo puede devolver aquello que ha calculado para que disponga de él el servidor central. Este valor será devuelto por la función `callModule()`.

Es importante destacar que haciendo uso de los módulos, si quiere realizar un módulo que calcula de manera dinámica el número de gráficas a visualizar, según la elección del usuario, y devuelve las gráficas para que se visualicen en la interfaz, la visualización de estas gráficas no puede hacerse en la interfaz del módulo concreto, si no en la interfaz de la aplicación. Es por este motivo por el que, para aquellos módulos que desempeñen esta funcionalidad, deben devolver los valores de las gráficas al servidor central y este lo recibirá tras la función `callModule()`.

3.3.4. Uso del framework Shiny para el desarrollo web del proyecto

Para la realización de la aplicación web de este proyecto, se ha hecho uso de todas las características que se han comentado en las anteriores secciones. En cuanto a la estructura de la aplicación, esta se encuentra modularizada en los siguientes módulos:

- **Módulo central.** Es el que contiene los ficheros `ui.R` y `server.R` para el lanzamiento de la aplicación. Este módulo se encarga de llamar a todos los módulos posteriores y visualizar las gráficas que algunos de ellos producen, tal y como se ha comentado al final de la sección 3.3.3.
- **Módulo para la carga de datos.** Se encarga de proporcionar una interfaz para que el usuario pueda introducir los nombres de los ficheros de datos y de cargar dichos datos en la aplicación.
- **Módulo para la selección de datos.** Este módulo tiene como finalidad seleccionar los datos con los que se quiera trabajar en la aplicación, de entre todos los que se han cargado.
- **Módulo para la visualización de las señales.** Proporciona la interfaz para poder seleccionar las señales que se quieren visualizar, y realiza los cálculos necesarios para obtener las gráficas y pasárselas al módulo central para que él las visualice.
- **Módulo para el filtrado de las señales.** Se encarga de proporcionar la interfaz para que el usuario pueda filtrar las señales de manera adecuada. La parte servidor de este módulo es el que se encarga, por tanto, de realizar todas las acciones necesarias para el filtrado de las señales. Tras el filtrado de las señales, permite visualizar las gráficas de la respuesta en frecuencia de las señales filtradas, para comprobar que el filtrado ha

sido correcto. Por lo que, al igual que el módulo anterior, obtendrá las gráficas y se las pasará al módulo central.

- **Módulo para la corrección ocular.** Proporciona una interfaz en la que se visualizan las señales que se van a tener en cuenta para la aplicación de la corrección ocular y su correspondiente aplicación. Por lo que la parte del servidor de este módulo es el que se encarga de realizar todas las acciones necesarias para la corrección ocular. Una vez que se haya realizado la corrección ocular, permite visualizar su resultado, seleccionando la señal que se quiera visualizar. Este módulo proporciona los datos de las gráficas al módulo central.
- **Módulo para la segmentación de las bandas.** Este módulo proporciona una interfaz sencilla en la que permite al usuario aplicar la segmentación de las bandas. La parte del servidor es la que se encarga de realizar todas las acciones correspondientes para su cálculo.
- **Módulo para la visualización de la señal GPS y el valor IAF.** Este módulo se encarga de visualizar la señal GPS y señalar en dicha señal el valor IAF. El servidor se encarga de obtener los valores de la gráfica y proporcionárselos al módulo central.
- **Módulo para la visualización de las bandas para cada tarea.** Se encarga de proporcionar la interfaz para que el usuario pueda seleccionar una tarea y pueda visualizar tanto la señal normal, como cada una de las bandas. El servidor de este módulo se encarga de obtener las gráficas y proporcionárselas al módulo central para su visualización.
- **Módulo para el cálculo de los índices.** Al igual que el módulo de la segmentación de las bandas, este módulo proporciona una interfaz sencilla en la que permite al usuario realizar el cálculo de los índices. Es el servidor de este módulo el que se encarga de realizar los procedimientos y acciones para la obtención del cálculo de los índices.
- **Módulo para la visualización de los valores de referencia para la obtención de los índices.** Ya que para la obtención de los índices se han tenido que calcular ciertos valores de referencia, este módulo proporciona una interfaz para la visualización de dichos valores. Como en los anteriores módulos, el servidor se encarga de obtener los datos y pasarle las gráficas al módulo central.
- **Módulo para la visualización de los índices por tareas.** Este módulo se encarga de mostrar una interfaz en la que se permite al usuario seleccionar una tarea, ver el valor de los índices obtenidos para dicha tarea y la visualización de cada índice en el transcurso de la tarea. Es por esto por lo que el servidor se encarga de obtener todos los datos y proporcionar al módulo central las gráficas para su visualización.
- **Módulo para la ejecución automática de todo el proceso.** Los anteriores módulos permitían realizar paso por paso la obtención de los índices. Este módulo, en cambio, permite realizar todo el proceso de manera automática. Proporciona una interfaz sencilla en la que permite al usuario lanzar la ejecución de todo el proceso y el servidor de este módulo se encarga de realizar todos los pasos para la obtención de los índices.
- **Módulos para la exportación de los valores calculados.** Estos módulos se encargan de proporcionar al usuario la posibilidad de exportar los datos calculados: todos los índices para cada una de las tareas y las señales empleadas en su cálculo (señales normales, filtradas y corregidas).

Todos los módulos hacen uso de la reactividad, en menor o mayor medida. Pues es gracias a ella por la que la aplicación es interactiva. La Figura 3.6 muestra la estructura de la aplicación web con todos los módulos mencionados.

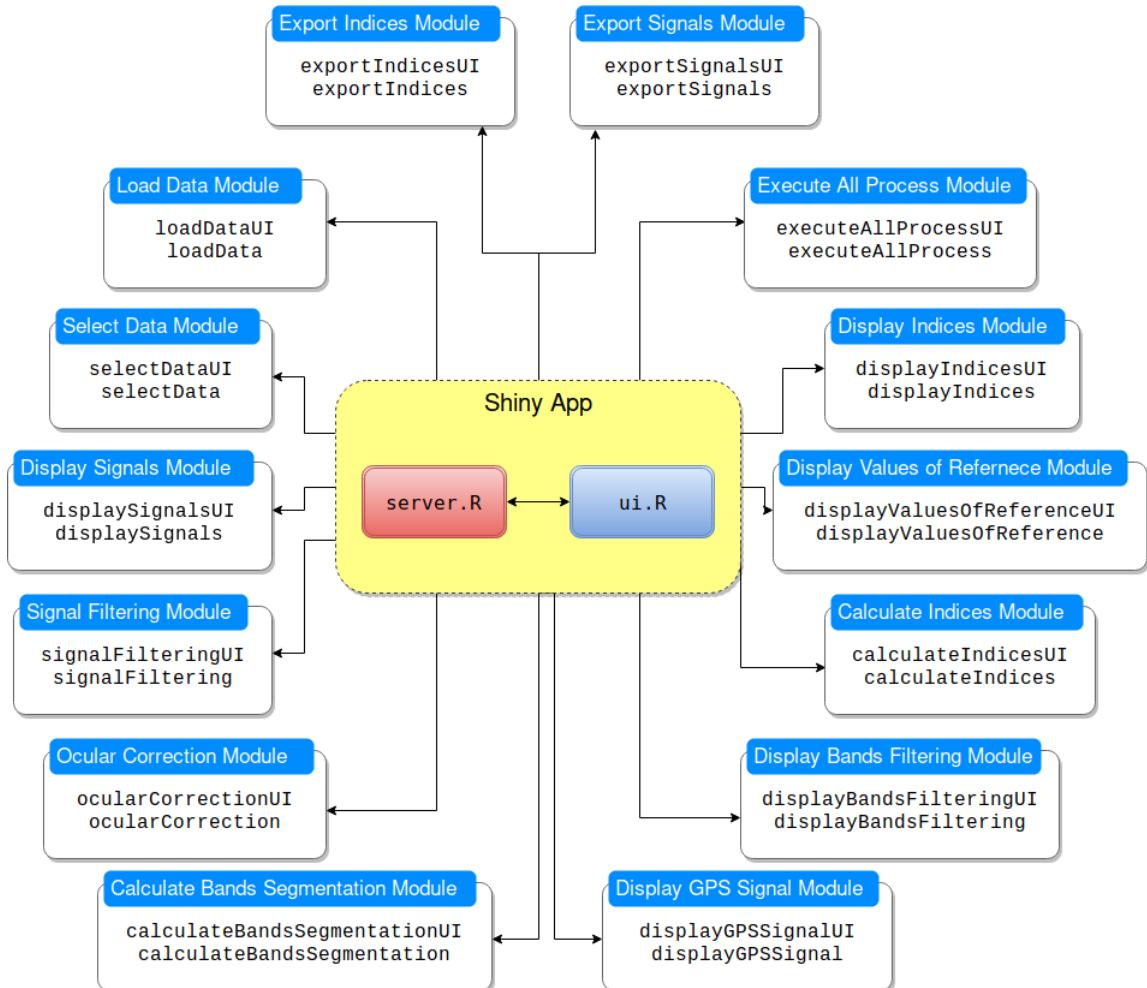


FIGURA 3.6: Estructura de la aplicación web en módulos.

3.4. Funcionalidad de la aplicación

En esta sección se va a exponer la funcionalidad con la que cuenta la aplicación. Esta funcionalidad está ligada directamente con los módulos expuestos en la sección 3.3.4, pues cada uno de los módulos de la aplicación desempeña una parte de la funcionalidad de la aplicación. Así pues, la funcionalidad de la aplicación es la siguiente:

- Permite **cargar** tantos **datos** como se quieran procesar, proporcionando el fichero de las señales junto con el fichero de marcas de tiempo correspondiente. Estos ficheros, internamente, tienen que tener el formato expuesto en el apéndice A. Además, para que los datos puedan ser cargados, los nombres de los ficheros deben ser coherentes, de manera que si el fichero de señales se llama `Data.txt`, el fichero de marcas de tiempo se ha de llamar `Test-Data.txt`.
- De todos los datos cargados, permite **seleccionar** uno de los **datos** con los que se quiera trabajar en ese momento.
- Una vez seleccionados los datos, permite **visualizar** cualquiera de las **señales** que se obtuvieron del sujeto. Ya sea visualizar dicha señal para todo el proceso de captación o para una determinada tarea. En el caso de las señales que puedan ser filtradas (electroencefalograma (EEG) y electrooculograma (EOG)), además de la visualización de la señal normal, también se visualizará la señal filtrada (si está filtrada).

- Permite **filtrar** las **señales** del EEG y del EOG para eliminar toda la información que no es relevante para el estudio. Por defecto, la aplicación establece los límites superior e inferior del filtrado, aunque estos pueden ser modificados libremente por el usuario. Tras el filtrado de las señales, permite visualizar la respuesta en frecuencia de cada una de las señales filtradas, para comprobar que el filtrado se ha realizado correctamente. Para realizar este filtrado, se hace uso de un filtro FIR en modo pasa-banda (véase sección 3.5.1).
- Con todas las señales filtradas, permite realizar la **corrección ocular**, aplicando el algoritmo de Gratton [28] (véase sección 3.5.2), que elimina el efecto de los parpadeos en la señal EEG haciendo uso de la señal EOG. Tras la corrección ocular, permite visualizar su resultado. Debido a que el algoritmo de Gratton [28] hace uso de una única señal EOG, si en los datos hay más de una señal EOG a la hora de aplicar la corrección ocular, la aplicación mostrará un mensaje de error indicando que hay más de una señal y no podrá aplicar la corrección ocular.
- Una vez que se ha realizado la adecuación de las señales (filtrado y corrección ocular), permite aplicar la **segmentación** de la señal **en bandas**, y obtener así, las bandas α , β y θ (2.2) para cada una de las tareas. Además, permite la visualización de la señal GPS, el valor de la IAF en dicha señal y las bandas para cada una de las tareas.
- Tras haber obtenido la segmentación de la señal en bandas, permite **calcular los índices**, la visualización de los valores de referencia utilizados para su cálculo y los valores de los índices para cada una de las tareas.
- También permite la **ejecución de manera automática** todo el proceso para **el cálculo de los índices** y la **exportación de los datos** calculados: índices y señales empleadas en su cálculo.

3.5. Funciones implementadas

Para poder realizar ciertas operaciones en el proceso del cálculo de los índices, ha sido necesario implementar algunas funciones importantes. Estas han sido:

3.5.1. Aplicación de un filtro FIR para el filtrado de las señales

`eegfilt()`. Esta función ha sido implementada para poder realizar el filtrado de las señales y poder así ajustar las frecuencias de información tanto del EEG como del EOG. La implementación de esta función se basa en la realizada para la herramienta EEGLab [6] y que se encarga de realizar un filtrado (paso-alto/bajo/banda) FIR. Debido al retardo constante en todas las frecuencias provocado por el filtro FIR (como se ha comentado en la sección 2.4.2), esta función lo solventa realizando el filtrado dos veces: uno hacia delante y otro hacia atrás.

A esta función es necesario pasarle los siguientes argumentos:

- Los datos a filtrar (en forma de vector).
- La frecuencia de muestreo de los datos (en Hz).
- La frecuencia de corte inferior (en Hz) para el filtrado *pasa-banda* (si se quiere realizar un filtrado *paso-bajo* este debe ser 0).
- La frecuencia de corte superior (en Hz) para el filtrado *pasa-banda* (si se quiere realizar un filtrado *paso-alto* este debe ser 0).

Tras el filtrado, la función devuelve los siguientes valores:

- Si los parámetros son correctos y se ha aplicado el filtro, devuelve una lista con dos campos:
 - *Smooth Data*. Contiene los datos filtrados, en el mismo formato que la entrada.
 - *Coefficients*. Contiene los coeficientes obtenidos en la aplicación del filtro FIR, en forma de vector.
- En el caso de que los parámetros no sean correctos, devolverá una cadena indicando el error producido.

El algoritmo 1 muestra el pseudocódigo de esta función.

Algoritmo 1: Pseudocódigo de la función `eegfilt()`

Entrada:

data: datos a filtrar.
srate: frecuencia de muestreo.
locutoff: frecuencia de corte inferior.
hicutoff: frecuencia de corte superior.

Salida :

Si ha habido algún error, una cadena indicando el error. Si no:

Smooth Data: datos filtrados.

Coefficients: coeficientes del filtro.

type \leftarrow Establecer tipo de filtrado, según *locutoff* y *hicutoff*.

bandEdges \leftarrow Establecer bordes de banda según *type*, *locutoff* y *hicutoff*.

Coefficients \leftarrow Obtener coeficientes del filtro FIR, con *filtorder*, *bandEdges*, *type*.

Smooth Data \leftarrow Filtrar *data* con *Coefficients*.

return *Smooth Data*, *Coefficients*.

3.5.2. Aplicación del algoritmo de Gratton para la corrección ocular

`gratton()`. Esta función ha sido implementada para poder aplicar la eliminación de los artefactos oculares y parpadeos de la señal EEG, haciendo uso de la señal EOG. La implementación de esta función se basa en la realizada para la herramienta EEGLab [12] y que se encarga de realizar la corrección ocular como especifica el algoritmo de Gratton [28]: calcula a partir de las señales EEG y EOG mediante *regresión lineal múltiple* la señal provocada exclusivamente por los movimientos oculares. En esta regresión lineal, la variable objetivo es la señal EOG y como predictores se usarán los distintos canales del EEG. De esta manera, la señal EEG sin artefactos oculares es el resultado de restar la señal obtenida por la regresión lineal a la señal original. Es más, este algoritmo proporciona también una señal en la que se pueden inferir los parpadeos del sujeto.

A esta función es necesario pasarle los siguientes argumentos:

- Los datos de la señal EEG (en forma de matriz, donde cada columna corresponde a cada uno de los canales de la señal y las filas contienen los valores de la señal para cada canal).
- Los datos de la señal EOG (en forma de vector).

Tras la aplicación del algoritmo de Gratton, la función devuelve los siguientes valores:

- Si los parámetros son correctos y se ha aplicado el algoritmo, devuelve una lista con dos campos:
 - *ceeg*. Señal EEG corregida (sin artefactos oculares), en el mismo formato que la entrada.
 - *neog*. Señal donde se pueden inferir los parpadeos del sujeto, en el mismo formato que el anterior.
- En el caso de que los parámetros no sean correctos, devolverá una cadena indicando el error producido.

El algoritmo 2 muestra el pseudocódigo de esta función.

Algoritmo 2: Pseudocódigo de la función *gratton()*

Entrada:

eeg: Señal EEG, con una columna para cada canal.
eog: Señal EOG.

Salida :

Si ha habido algún error, una cadena indicando el error. Si no:

ceeg: Señal EEG corregida.

neog: Señal para inferir parpadeos.

```

 $el \leftarrow$  Número de canales de la señal eeg.
 $Blinks \leftarrow$  Array vacío de tamaño el.
 $x \leftarrow eog$ 
for i  $\leftarrow 1$  to el do
|    $y \leftarrow$  Canal i de la señal eeg.
|   coefficients  $\leftarrow$  Regresión lineal múltiple con y, x.
|    $Blinks[i] \leftarrow$  coefficients.
end
 $neog \leftarrow Blinks \cdot eog$ .
 $ceeg \leftarrow eeg - neog$ .
return ceeg, neog

```

Capítulo 4

Acciones y procedimientos para el cálculo de los índices

4.1. Introducción

Una vez que se ha obtenido la señal EEG, para el cálculo de los índices (2.3) es necesario realizar ciertas acciones y procedimientos. En este capítulo, se va detallar todo el proceso realizado para dicho cálculo.

Antes de empezar con la explicación de todo el proceso es importante detallar a qué fases son sometidos los sujetos en la captación del EEG, los experimentos que se han realizado y la herramienta que se ha utilizado para la adquisición de las señales [29].

Independientemente del procedimiento llevado para su captación, **las fases** se dividen en tres tipos:

- **Fase de reposo.** Se trata de la primera fase que se realiza al sujeto. En ella es necesario que el sujeto se encuentre en una fase de actividad mínima, para poder conocer los ritmos y frecuencias propias del sujeto cuando se encuentra en este estado. Es por ello que, para la realización de esta fase, el sujeto debe cerrar los ojos y relajarse. A esta fase también se le conoce como **fase de ojos cerrados (OC)**.
- **Fase de medición basal.** Con esta fase se da a conocer cuál es el estado “normal” del sujeto cuando está activo, sin recibir ningún estímulo determinante para la prueba. Para la captación de esta fase, se utiliza un estímulo neutro que active los órganos sensoriales para recibir el estímulo principal. A esta fase se le conoce como **basal o baseline (BS)**.
- **Fase de exposición al estímulo.** Se trata de la fase en la que se somete al sujeto a los estímulos principales de los que se quiere obtener sus resultados. A esta fase se le denomina **tarea (TASK)**.

Dentro de la fase de exposición al estímulo, **los experimentos** que han sido realizados se encuentran ubicados en el marco del grupo de investigación NiCE, y son los siguientes:

- Estudio sobre la influencia de los olores en respuesta a estímulos visuales relacionados con las obras de arte, así como el rendimiento en los tests CANTAB.
- Estudio de una forma objetiva de cuantificar el bienestar de personas en proceso o estado de jubilación a partir de estímulos visuales relacionados con dicho proceso (familia, médicos, tiempo libre, etc), así como el rendimiento en los tests CANTAB.
- Estudio en el instituto como búsqueda de relaciones entre el rendimiento en los tests CANTAB, el nivel de activación ante estímulos visuales relacionados con entornos sanitarios y el interés por su formación universitaria.

La metodología utilizada para realizar los estímulos visuales se ha llevado a cabo mostrando al sujeto de estudio una secuencia/serie de imágenes digitales, mientras que el *Cambridge Neuropsychological Test Automated Battery* (CANTAB) es un software desarrollado por la Universidad de Cambridge para la realización de distintas pruebas neuropsicológicas que permiten evaluar al sujeto en distintas áreas, como pueden ser la memoria, la atención o la toma de decisiones. Las pruebas seleccionadas para los experimentos han sido las siguientes (son aquellas pruebas que se ajustan más a los índices que se quieren calcular):

- *Reaction Time (RTI)*. Se trata de una prueba que mide la respuesta mental ante un estímulo que consiste en presionar un punto cuando este se ilumina.
- *Spatial Working Memory (SWM)*. Se trata de una prueba en la que el sujeto debe buscar la casilla que tenga color de entre todas las que aparecen en pantalla, midiendo así la capacidad de retener información que posee el sujeto.
- *One Touch Stockings (OTS)*. En esta prueba al sujeto se le plantea una configuración final a la que tiene que llegar en el menor número de movimientos posibles, a partir de una configuración inicial especificada.
- *Emotion Recognition Task (ERT)*. Se trata de una prueba en la que el sujeto debe reconocer la emoción de los rostros que se muestran en una presentación de imágenes fugaces.

Para la **adquisición de las señales** se ha hecho uso de la herramienta *NeXus-10*, que consiste en un sistema de medición capaz de captar tanto señales fisiológicas (respiración, temperatura, etc) como neurológicas (EEG). Para abarcar todas las funcionalidades cuenta con distintos sensores que deben ser colocados en sus respectivas entradas. Para realizar las grabaciones se ha utilizado el sistema internacional 10-20 en la colocación de los electrodos. Este dispositivo se conecta al ordenador mediante bluetooth o puerto USB para que los registros puedan ser centralizados y se procesen para su visualización y almacenamiento. Para ello, cuenta con el software *BioTrace+*. También cuenta con la opción de exportar los valores registrados a una tarjeta SD sin uso de *BioTrace+*.

Existen tres tipos de canales de registros distintos: los *canales EXG* (hasta un máximo de cuatro) que se encargan de registrar las señales eléctricas: electroenfecalograma (EEG), electrocardiograma (ECG), electromiograma (EMG) y electrooculograma (EOG); los *canales AUX* (hasta un máximo de cuatro) en los que se conectan el resto de posibles sensores (excepto oximetría); y otros *canales* que posee un canal para la oximetría y otro para una entrada digital adicional.

En compañía con el *NeXus-10*, para el registro de los estímulos visuales se ha hecho uso de una aplicación software implementada en Java y desarrollada por [29], cuyo objetivo es almacenar la marca de tiempo concreta en la que una imagen aparece en pantalla y la marca de tiempo en la que desaparece. De tal manera que se pueda corresponder el registro de los valores de las señales EEG y EOG con la duración de la imagen en pantalla.

De entre las señales que se toman al sujeto, en los experimentos realizados se han medido: el EEG y el EOG (ambos a una frecuencia de 256 Hz), el pulso de volumen sanguíneo (*BVP, Blood Volume Pulse*), la frecuencia cardíaca (*HR, Heart Rate*), la conductancia de la piel (*SC, Skin Conductance*) y el seguimiento del recorrido de visualización de la pantalla (para el que se ha usado la herramienta *EyeTracker*). Para el proceso de obtención de los índices solamente se van a hacer uso del EEG y EOG, mientras que el resto son utilizados en otros trabajos dentro del grupo de investigación NiCE.

Las Figuras 4.1 y 4.2 muestran el flujo de trabajo realizado para el cálculo de los índices. En la Figura 4.1 se muestra desde la adecuación de las señales hasta el cálculo de los valores de referencia; mientras que, en la Figura 4.2 se muestra el cálculo de los índices para las tareas (se simplifica a TASKS, pues es el mismo procedimiento para todas).

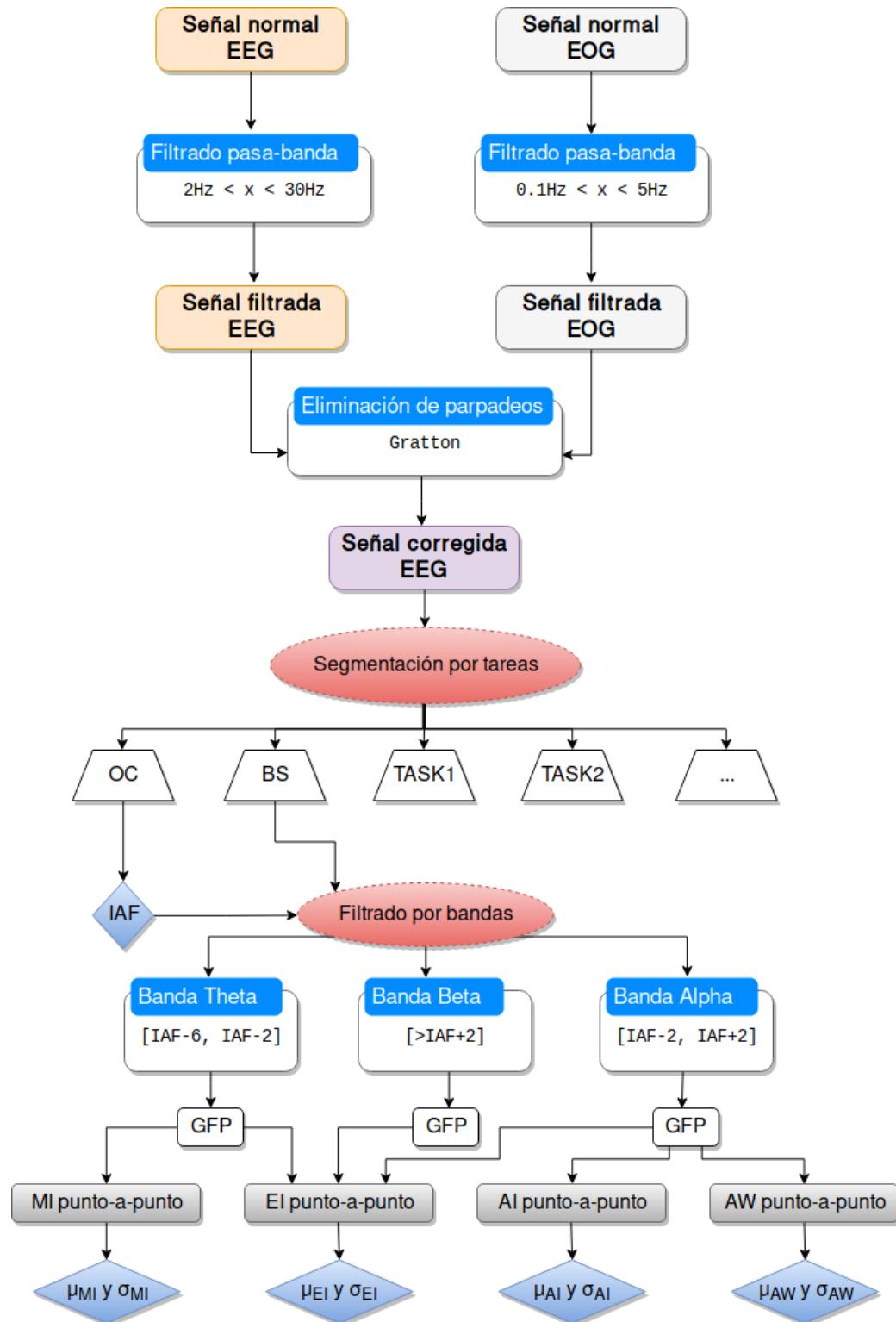


FIGURA 4.1: Flujo de trabajo para el cálculo de los índices AI, AW, EI y MI.
Obtención de los valores de referencia.

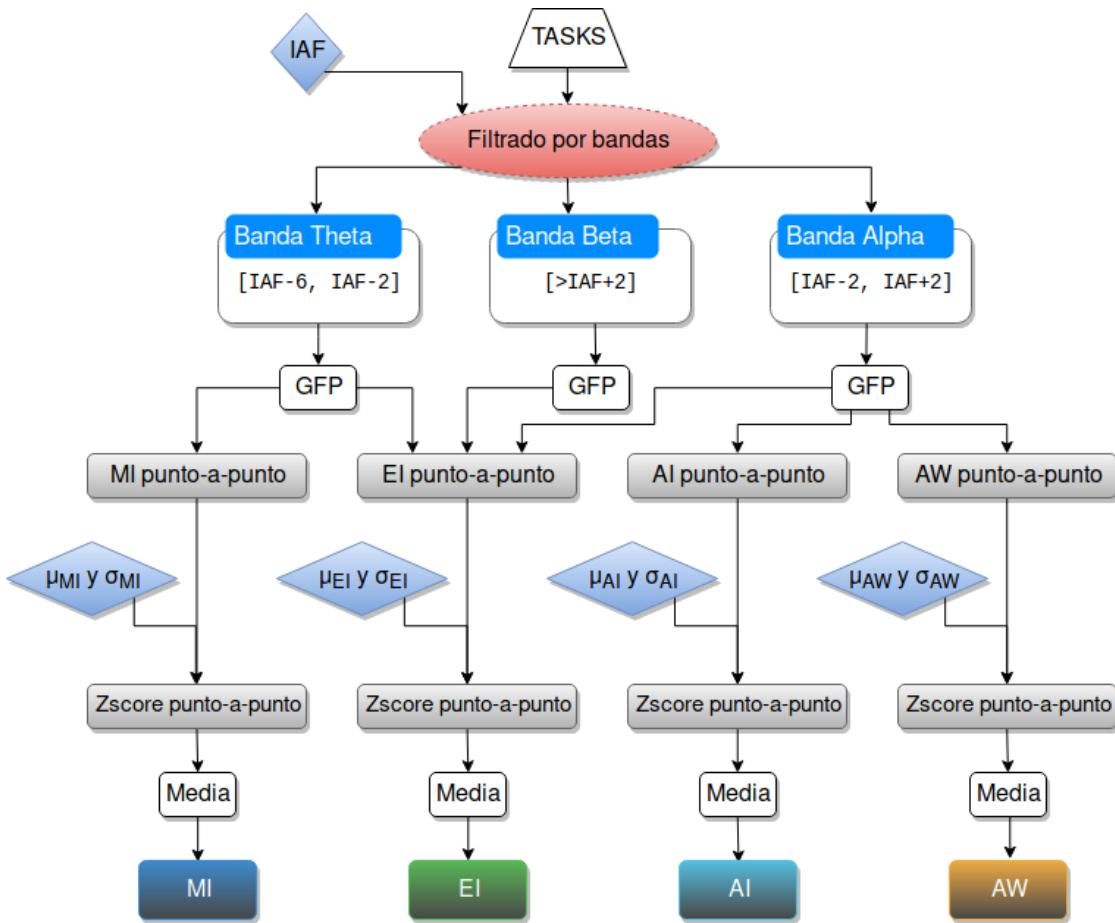


FIGURA 4.2: Flujo de trabajo para el cálculo de los índices AI, AW, EI y MI.
Obtención de los índices.

4.2. Adecuación de las señales registradas

Tras la adquisición de las señales, se continúa con la adecuación de las mismas para poder obtener los índices AI, AW, EI y MI (véase sección 2.3). Esta adecuación consiste en ajustar las frecuencias de información en EEG y EOG para poder eliminar los artefactos del EEG provocados por los movimientos oculares y parpadeos.

4.2.1. Ajuste a las frecuencias de información en EEG y EOG

Tanto la señal del EEG como la del EOG contienen mucha información que está producida por diversos parámetros. Sin embargo, para poder realizar un cálculo de índices que sea lo más preciso posible, hay que eliminar la información que no se necesita de ambas señales. Como se ha comentado en la sección 2.2, las cuatro bandas principales en las que se puede descomponer la señal EEG son: *alpha* (α), *beta* (β), *theta* (θ) y *delta* (δ), por lo que el rango de frecuencias que nos interesa es de 2Hz a 30Hz . Nótese que las frecuencias más bajas de la señal *delta* se atenúan debido a que los experimentos que se llevan a cabo son realizados con sujetos despiertos y activos. En cambio, el rango de frecuencias que nos interesan para el EOG es de 0.1Hz a 5Hz .

En ambos casos, para este ajuste se aplica el algoritmo de filtrado implementado en la sección 3.5.1. Se utiliza este filtro en modo *pasa-banda*, pasándole la frecuencia de muestreo

de la señal (recuérdese que ambas están medidas a 256 Hz) y los rangos de frecuencias indicados para cada una de las señales.

La Figura 4.3 muestra las señales EEG y EOG sin la aplicación de ningún filtro, mientras que la Figura 4.4 muestra las mismas señales tras haber realizado el filtrado anteriormente mencionado.

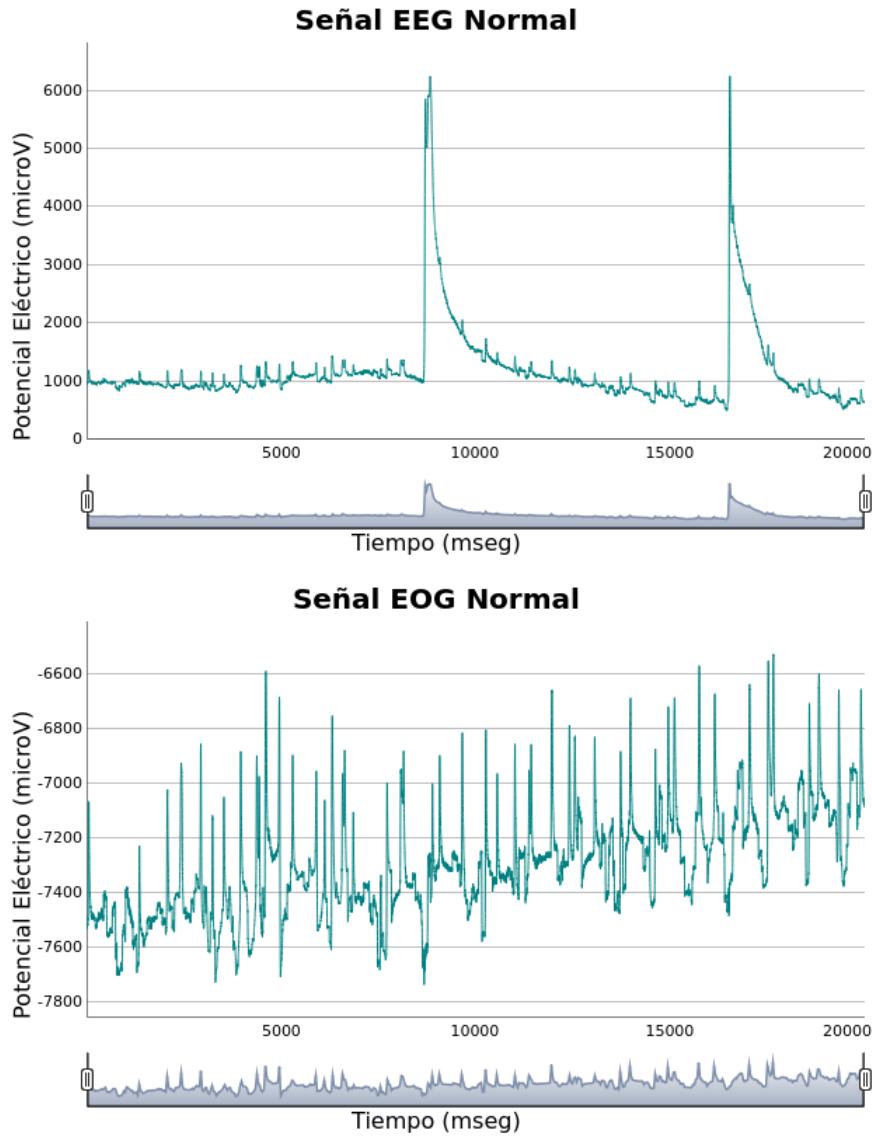


FIGURA 4.3: Señal EEG (superior) y EOG (inferior) sin filtrar.

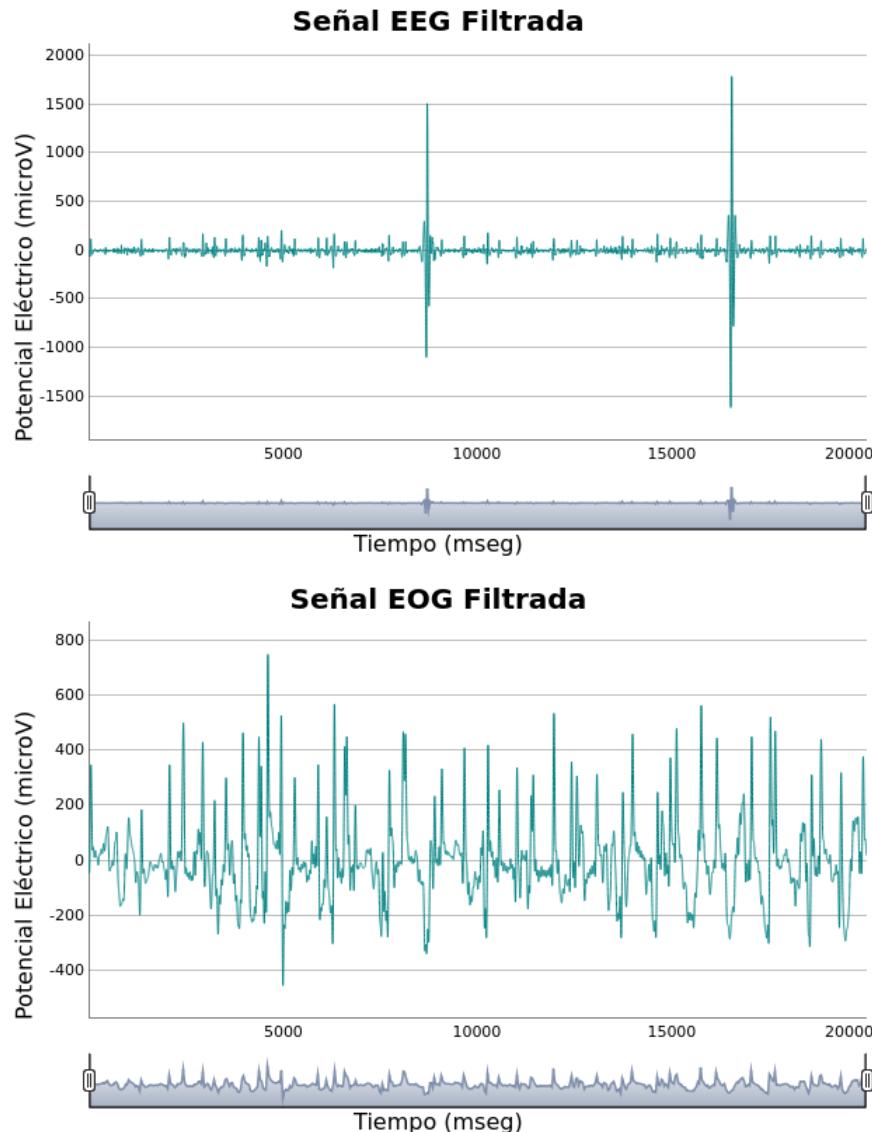


FIGURA 4.4: Señal EEG (superior) y EOG (inferior) filtradas para los rangos 2-30Hz y 0.1-5Hz, respectivamente.

4.2.2. Eliminación de los artefactos provocados por movimientos oculares y párpadeos

El paso que sigue al ajuste de las frecuencias de información de las señales es eliminar de la señal EEG toda aquella información que proviene de la superficie ocular, ya que esta influye en el EEG de dos maneras: por la actividad eléctrica que se produce por el movimiento entre la córnea y la retina, y por el campo eléctrico que se produce por el movimiento de los párpados, ya que ambos se propagan por el cráneo.

Para realizar la eliminación de los artefactos oculares, se hace uso de la señal EOG y del algoritmo de Gratton [28] que se ha implementado y explicado en la sección 3.5.2. Tras la aplicación de este algoritmo, se obtiene una señal EEG corregida.

La Figura 4.5 muestra la señal EEG tras haber realizado la eliminación de artefactos oculares.

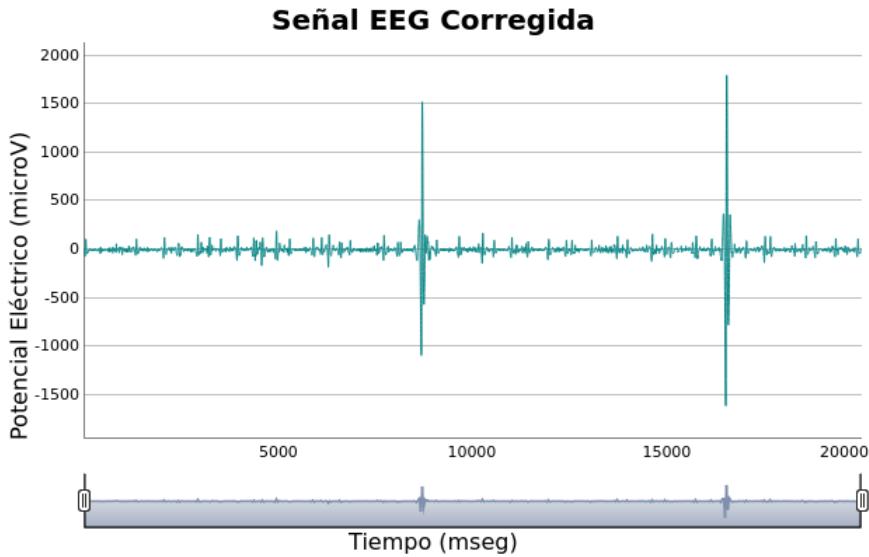


FIGURA 4.5: Señal EEG sin artefactos oculares.

4.3. Segmentación de la señal

Debido a que no es práctico estar parando la medición de la señal para cada tarea o prueba que se le realiza al sujeto, la señal que se tiene es la global a todo el experimento. Puesto que el cálculo de los índices es para cada una de las tareas realizadas, es necesario segmentar toda la señal por tareas, teniendo así la señal capturada para cada tarea.

Para esta segmentación se hace uso de las marcas de tiempo asociadas al experimento (obtenidas por la aplicación software mencionada en 4.1), en las que se indica el inicio y fin de cada tarea en la señal global. Con estas marcas de tiempo, se consigue una señal segmentada que contiene los tres conjuntos de tareas siguientes (correspondientes a las fases mencionadas en la sección 4.1): ojos cerrados, basales y tareas a evaluar.

4.4. Cálculo de la IAF (*Individual Alpha Frequency*)

Las bandas de frecuencias en las que puede dividir la señal EEG, cuyos valores concretos se han detallado en la sección 2.2, pueden variar de un sujeto a otro de manera sutil. De todas las bandas, la banda *alpha* es la que sufre mayor variación, por lo que se hace uso de dicha banda para determinar correctamente los límites entre las bandas. La IAF (*Individual Alpha Frequency*) es el valor de referencia que describe cuál es la frecuencia en la que se detecta un pico en el espectro de potencia cuando el sujeto está en reposo [30]. Para el cálculo de la IAF se sigue el siguiente procedimiento:

1. Se selecciona la señal correspondiente a la tarea de *ojos cerrados*, en la que el sujeto está en reposo.

2. Se calcula la Densidad espectral de potencia (*PSD, Power Spectral Density*) para cada uno de los canales de la señal EEG, utilizando una ventana *Hanning* con un solapamiento del 10 %. Convirtiendo así la señal del dominio del tiempo, al dominio de la frecuencia.
3. De la PSD, se extraen las frecuencias correspondientes a la banda *alpha* (véase sección 2.2), para obtener el intervalo donde se encuentra el valor IAF.
4. Se calcula el Espectro de potencia global (*GPS, Global Power Spectrum*) realizando la media de las PSD para cada canal.
5. El valor IAF que buscamos se obtiene como el máximo valor de la señal GPS. En el caso de que la señal GPS sea descendente o ascendente, es decir, no haya ningún pico, es necesario realizar el siguiente paso:
 - 5.1. El valor IAF se obtiene aplicando la técnica del *centro de gravedad*.

La Figura 4.6 muestra la señal GPS calculada y en la que aparece señalado el valor de la IAF.

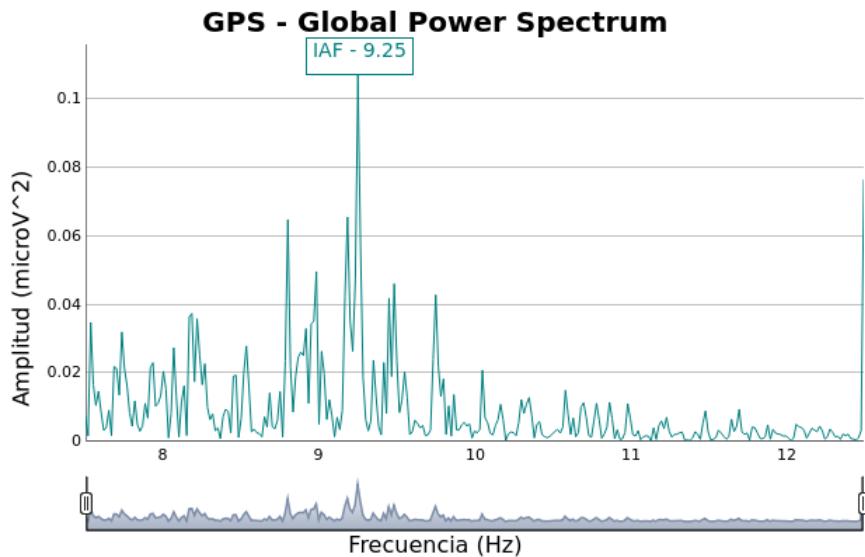


FIGURA 4.6: Señal GPS y valor de la IAF.

4.5. Filtrado por bandas α , β y θ de la señal

La IAF permite establecer las bandas de frecuencia del sujeto, por lo que tras haber calculado su valor, es necesario establecer los rangos de cada una de las bandas. De esta manera, las bandas son ajustadas con los siguientes rangos [30, 36]:

- La **banda theta** (θ) se ajusta al rango: $[IAF - 6, IAF - 2]$.
- La **banda alpha** (α) se ajusta al rango: $[IAF - 2, IAF + 2]$.
- La **banda beta** (β) se ajusta al rango: $[> IAF + 2]$.

Una vez que se tienen bien definidos los rangos de cada una de las bandas de frecuencia para el sujeto concreto, hay que filtrar la señal para cada tarea. Para ello, se hace uso del mismo filtro que en la sección 4.2.1, pero usándolo de manera distinta: debido a que puede haber

tareas con poca duración, la aplicación de un filtro *pasa-banda* puede producir errores en el resultado debido a un posible solapamiento de las zonas de filtrado (*paso-alto* y *paso-bajo*). Es por esto por lo que, primero se realiza un filtrado *paso-alto* y luego un filtrado *paso-bajo*.

La Figura 4.7 muestra la banda *alpha* para la fase basal.

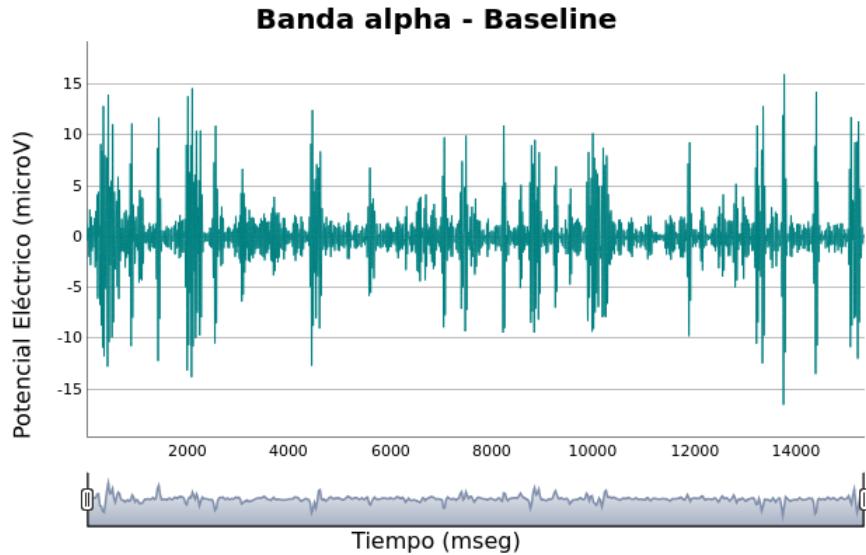


FIGURA 4.7: Banda *alpha* para la fase basal.

4.6. Obtención de los índices

Llegados a este punto, se tienen todos los elementos disponibles para la obtención de los índices. Para su cálculo, es necesario que primero se calculen los valores de referencia de cada uno de los índices, obtenidos a partir de la señal basal (BS). Con estos valores de referencia se calculan los índices para cada tarea y se hace una normalización *Zscore* de todos los índices para obtener el valor del índice relativo al sujeto de estudio. Previo a lo anterior, es necesario exponer cómo se calculan dichos índices según su definición.

4.6.1. Definición de los índices

Para el cálculo de los índices (2.3), es necesario calcular el Campo global de potencia (*GFP*, *Global Field Power*) para cada una de las bandas, que se obtiene a partir del cuadrado de los valores que componen cada banda. El cálculo de cada uno de los índices se hace según su definición:

- **Índice de atención (AI)** [35, 30, 36]. Haciendo uso de la banda *alpha*, se calcula su *GFP* y se invierte su resultado. Para obtener el valor del índice en un punto, se realiza la media de los canales. La fórmula que define este índice es la ecuación 4.1:

$$AI = -\frac{GFP_{Der}^{\alpha} + GFP_{Izq}^{\alpha}}{2} \quad (4.1)$$

- **Índice de aproximación/rechazo (AW)** [34, 37]. Haciendo uso de la banda *alpha*, se calcula la GFP y se obtiene su valor como la diferencia entre los canales. La fórmula que define este índice es la ecuación 4.2:

$$AW = GFP_{Der}^{\alpha} - GFP_{Izq}^{\alpha} \quad (4.2)$$

- **Índice de compromiso (EI)** [39, 38]. Haciendo uso de las tres bandas *alpha*, *beta* y *theta*, se calcula la GFP para cada una y se realiza la media de los canales por cada una. Con las GFP calculadas, el valor de este índice se calcula mediante la ecuación 4.3:

$$EI = \frac{GFP^{\beta}}{GFP^{\alpha} + GFP^{\theta}} \quad (4.3)$$

- **Índice de memorización (MI)** [35, 30, 36]. Haciendo uso de la banda *theta*, directamente se obtiene calculando su GFP. Para obtener el valor del índice en un punto, se realiza la media de los canales. La fórmula que define este índice es la ecuación 4.4:

$$MI = \frac{GFP_{Der}^{\theta} + GFP_{Izq}^{\theta}}{2} \quad (4.4)$$

4.6.2. Valores de referencia

Para poder realizar la normalización *Zscore* de los índices, es necesario calcular la media (μ) y desviación típica (σ) para cada índice de la señal basal (BS). Por lo tanto, primero se calcula el valor de cada índice (siguiendo las ecuaciones expuestas en la sección 4.6.1) por cada punto de la señal basal para, posteriormente, calcular la media y desviación típica de todo el conjunto de puntos. A este conjunto de puntos para cada índice, se le denomina *Índice punto-a-punto*.

Es importante destacar que pueden realizarse varias fases basales al sujeto durante el experimento. Por este motivo, es necesario calcular para cada una de las fases basales los valores de referencia. De tal manera que, en el posterior cálculo de los índices para cada tarea, los **valores de referencia** que se van a utilizar van a ser los **correspondientes a la última fase basal** realizada.

En la Figura 4.8 se muestra el valor de referencia *punto-a-punto* para el índice AW.

4.6.3. Normalización y cálculo de los índices

Para finalizar, primero se ha de calcular (para cada una de las tareas) los valores de cada índice punto a punto (haciendo uso de las ecuaciones expuestas en la sección 4.6.1). Una vez calculados todos los índices punto a punto para cada tarea, se realiza una normalización *Zscore* para obtener el índice relativo al sujeto de estudio.

Haciendo uso de la media (μ_{basal}) y la desviación típica (σ_{basal}) calculados en la sección 4.6.2, la normalización *Zscore* para **cada punto** X_i del índice se obtiene según la ecuación 4.5 [34]:

$$Z_i = \frac{X_i - \mu_{basal}}{\sigma_{basal}} \quad (4.5)$$

Finalmente, para cada tarea, el valor de los índices AI, AW, EI y MI relativos al sujeto de estudio se obtienen realizando la **media de todos los valores punto a punto normalizados**.

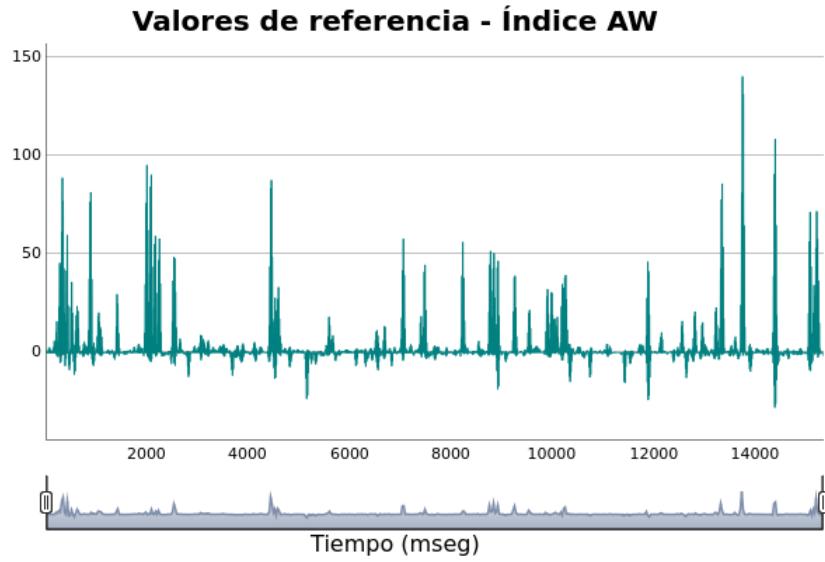


FIGURA 4.8: Valor de referencia *punto-a-punto* para el índice AW.

En la Figura 4.9 se muestra el valor *punto-a-punto* del índice AW para la tarea Spa_10g.

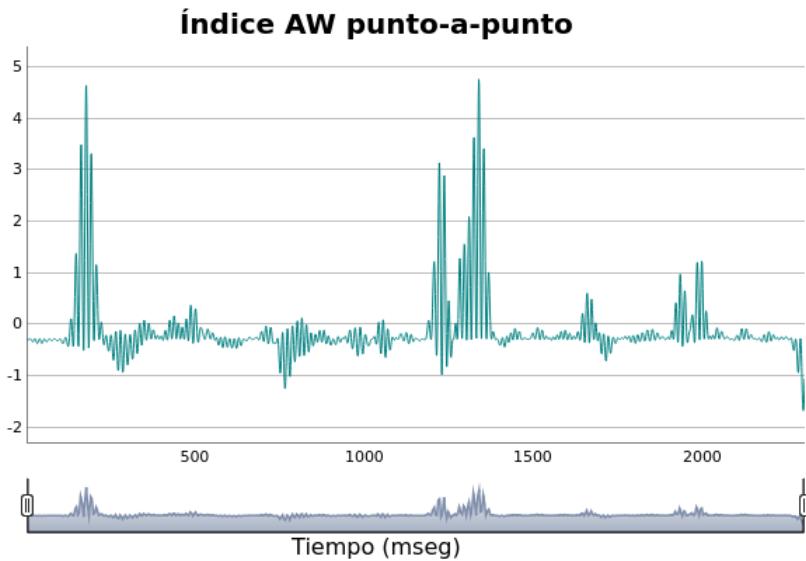


FIGURA 4.9: Índice AW *punto-a-punto* para la tarea Spa_10g.

Capítulo 5

Evaluación de resultados

5.1. Introducción

En este capítulo se van a realizar las evaluaciones de las funciones que se han implementado (descritas en la sección 3.5), así como de los resultados que se han obtenido en el cálculo de los índices mediante el procedimiento descrito en la sección 4.

Para ello, se ha hecho uso de los datos de tres sujetos distintos, a los que se le han realizado varios experimentos. Durante toda esta sección, para distinguir los resultados de cada uno de ellos, estos tres sujetos reciben los siguientes nombres: ANDGAR, JIMTOM y NICANA.

Tanto en las funciones como en los índices, la evaluación se ha llevado a cabo obteniendo el error que hay entre la implementación en R (proporcionada por este trabajo) y la implementación en Matlab (proporcionada por el TFG [29]). Para ello, se ha tomado la siguiente convención:

$$\text{Error} = R - \text{Matlab}$$

de manera que, **en todos los casos**, cuando se mencione algún cálculo de error o diferencias entre señales, siempre se calcula haciendo la diferencia de los datos de R *menos* los datos de Matlab.

En el caso de las funciones, dado que los resultados son señales con gran cantidad de puntos, se ha hecho uso de la Raíz cuadrada del Error Cuadrático Medio (*RMSE, Root Mean Square Error*) (la ecuación empleada se muestra en la sección 5.2); mientras que para los índices, dado que se trata de un solo valor, se ha hecho uso del valor absoluto de la diferencia (la ecuación empleada se muestra en la sección 5.4).

5.2. Evaluación de las funciones implementadas

5.2.1. Procedimiento para la obtención de los datos a evaluar

Para realizar la evaluación de las funciones que se han descrito en la sección 3.5, se ha realizado el siguiente procedimiento:

1. Realizar el filtrado *pasa-banda* y la corrección ocular en R y en Matlab, tal y como se describe en la sección 4.2. Esto proporciona tres tipos de señales: filtrada, corregida y parpadeos, cada una con sus canales correspondientes.
2. Para cada señal, obtener la diferencia de sus canales de manera gráfica.
3. Para cada señal filtrada, obtener la diferencia del espectro de frecuencia de sus canales, de manera gráfica.

4. Segmentar por tareas todas las señales obtenidas, tal y como se describe en la sección 4.3.
5. Para cada una de las señales (filtrada, corregida, parpadeos), se calcula el error mediante la ecuación 5.1, siendo ch el canal de la señal, t la tarea a evaluar, N el tamaño del canal ch para la tarea t y chR, chM los valores R y Matlab para el canal ch , respectivamente:

$$\text{Error}(ch)^t = \sqrt{\frac{\sum_{i=1}^N (chR_i^t - chM_i^t)^2}{N}} \quad (5.1)$$

5.2.2. Evaluación del filtrado

Existen dos maneras para evaluar el filtrado de las señales: en el dominio del tiempo y en el dominio de la frecuencia. Por lo tanto, en esta sección se va realizar la evaluación de las señales filtradas en ambos dominios.

En el **dominio del tiempo**, las Figuras 5.1, 5.2 y 5.3 muestran las diferencias de las señales filtradas de manera gráfica, para los sujetos ANDGAR, JIMTOM y NICANA, respectivamente.

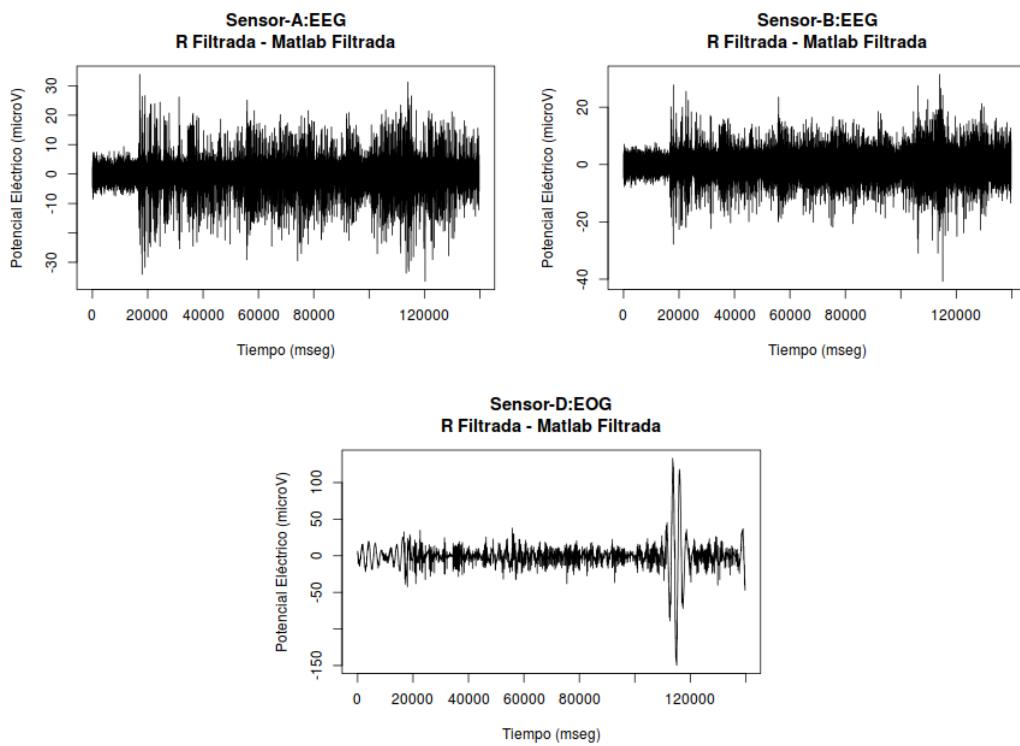


FIGURA 5.1: Diferencias en el dominio del tiempo de las señales filtradas, del sujeto ANDGAR.

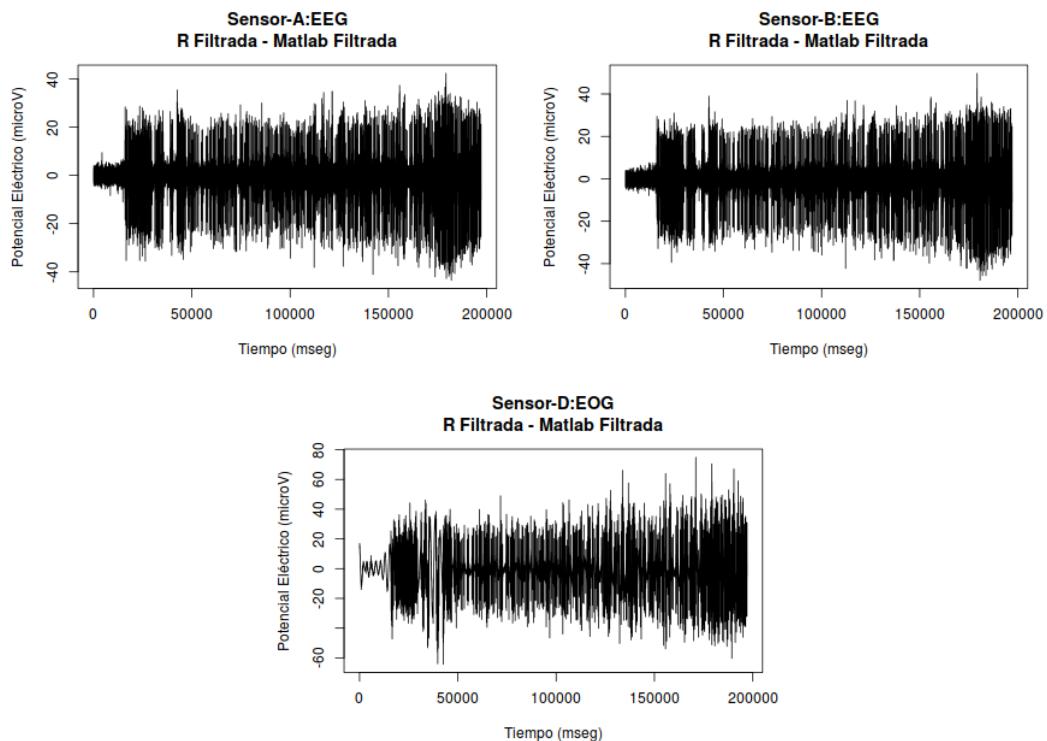


FIGURA 5.2: Diferencias en el dominio del tiempo de las señales filtradas, del sujeto JIMTOM.

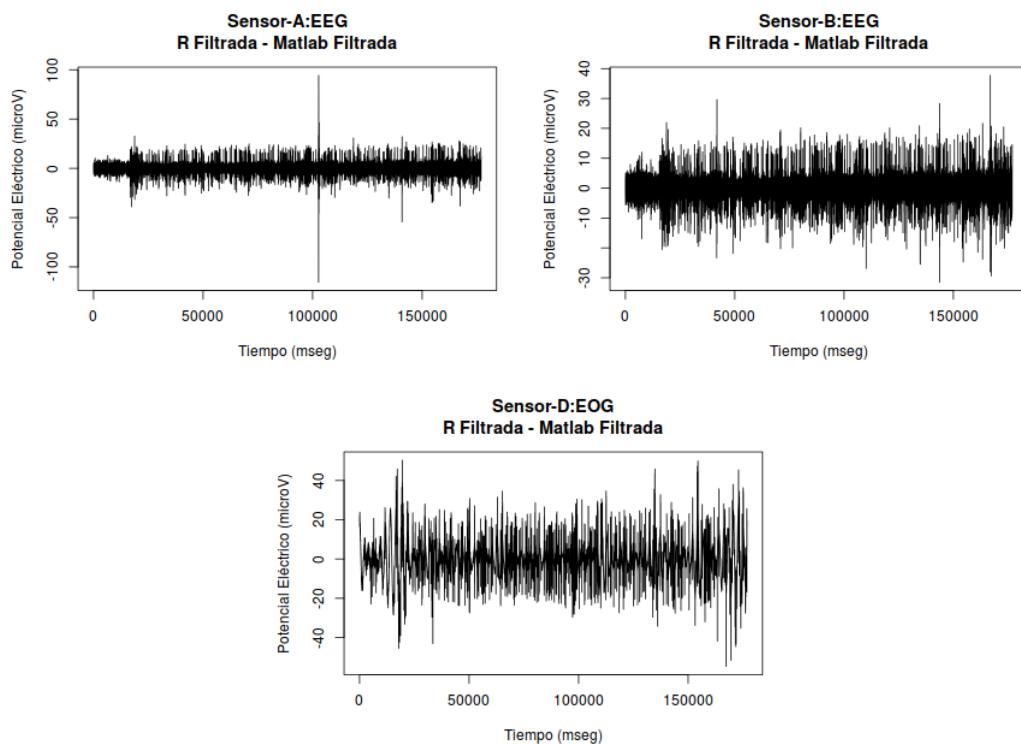


FIGURA 5.3: Diferencias en el dominio del tiempo de las señales filtradas, del sujeto NICANA.

Las Tablas 5.1, 5.2 y 5.3 muestran el error obtenido mediante la ecuación 5.1 para algunas de las tareas realizadas a los sujetos. Nótese que, aunque se enumeren igual, las tareas pueden no ser las mismas. En la parte inferior de las tablas aparece la media y la desviación típica del error obtenido para cada canal.

	Sensor-A:EEG	Sensor-B:EEG	Sensor-D:EOG
Tarea 1	4.072299	3.485974	4.676781
Tarea 2	2.997959	2.988413	3.008327
Tarea 3	3.333875	3.104637	3.127869
Tarea 4	4.353203	3.758474	6.714817
Tarea 5	3.094663	2.98134	6.077258
Tarea 6	3.440801	3.171713	7.282244
Tarea 7	3.615699	3.113885	8.828839
Tarea 8	3.854718	3.454712	9.321651
Tarea 9	7.301925	6.103906	12.799772
Tarea 10	5.129666	4.25147	10.525547
—	—	—	—
media	4.421575	3.922261	11.424315
desv. típica	1.148959	0.993618	18.232443

TABLA 5.1: Error obtenido en el dominio del tiempo de las señales filtradas, de algunas tareas del sujeto ANDGAR.

	Sensor-A:EEG	Sensor-B:EEG	Sensor-D:EOG
Tarea 1	4.445821	4.54711	7.2504
Tarea 2	5.84844	5.784408	8.691971
Tarea 3	4.30408	4.699616	7.077917
Tarea 4	3.936296	4.294852	6.135915
Tarea 5	3.175059	3.520278	4.882929
Tarea 6	5.225289	5.530017	8.644041
Tarea 7	6.217608	6.425467	9.315333
Tarea 8	5.328717	5.411942	8.769106
Tarea 9	7.103721	7.577233	12.066023
Tarea 10	4.522886	4.714267	6.570992
—	—	—	—
media	6.00308	6.307818	10.487085
desv. típica	1.556251	1.640028	3.266943

TABLA 5.2: Error obtenido en el dominio del tiempo de las señales filtradas, de algunas tareas del sujeto JIMTOM.

	Sensor-A:EEG	Sensor-B:EEG	Sensor-D:EOG
Tarea 1	3.948233	2.953281	5.630233
Tarea 2	3.977975	3.031673	5.547536
Tarea 3	4.431315	3.229801	5.843656
Tarea 4	5.205466	4.256191	7.313524
Tarea 5	3.358773	2.381653	5.849156
Tarea 6	4.439075	3.220967	6.025417
Tarea 7	5.559762	4.152889	7.522856
Tarea 8	4.648702	3.367659	7.717808
Tarea 9	3.682312	2.515246	5.499425
Tarea 10	4.441157	3.487174	5.909551
—	—	—	—
media	4.856858	3.580301	7.173677
desv. típica	1.559981	0.665472	2.419813

TABLA 5.3: Error obtenido en el dominio del tiempo de las señales filtradas, de algunas tareas del sujeto NICANA.

Hay que tener en cuenta que las señales están medidas en *micro-voltios* (μV), o lo que es lo mismo $10^{-3}V$. Esto quiere decir que, como se puede observar, el error producido está en el orden de $10^{-3}V$.

Para los canales *Sensor-A:EEG*, *Sensor-B:EEG* y *Sensor-D:EOG*, las **medias** de los errores obtenidos para el sujeto ANDGAR son: $4.421575 \mu V$, $3.922261 \mu V$ y $11.424315 \mu V$; para el sujeto JIMTOM son: $6.00308 \mu V$, $6.307818 \mu V$ y $10.487085 \mu V$; y para el sujeto NICANA son: $4.856858 \mu V$, $3.580301 \mu V$ y $7.173677 \mu V$. Mientras que las **desviaciones típicas** para el sujeto ANDGAR son: $1.148959 \mu V$, $0.993618 \mu V$ y $18.232443 \mu V$; para el sujeto JIMTOM son: $1.556251 \mu V$, $1.640028 \mu V$ y $3.266943 \mu V$; y para el sujeto NICANA son: $1.559981 \mu V$, $0.665472 \mu V$ y $2.419813 \mu V$.

La Figura 5.4 muestra las distribuciones de los resultados mostrados en las tablas anteriores para todos los sujetos.

En el **dominio de la frecuencia**, las Figuras 5.5, 5.6 y 5.7 muestran las diferencias entre las señales filtradas de manera gráfica, para los tres sujetos. Las Tablas 5.4, 5.5 y 5.6, muestran el error obtenido mediante la ecuación 5.1, en el dominio de la frecuencia, para cada uno de los sujetos.

	Sensor-A:EEG	Sensor-B:EEG	Sensor-D:EOG
Error	0.009296	0.009455	0.008976

TABLA 5.4: Error obtenido en el dominio de la frecuencia, del sujeto ANDGAR.

	Sensor-A:EEG	Sensor-B:EEG	Sensor-D:EOG
Error	0.008489	0.007953	0.005659

TABLA 5.5: Error obtenido en el dominio de la frecuencia, del sujeto JIMTOM.

	Sensor-A:EEG	Sensor-B:EEG	Sensor-D:EOG
Error	0.009156	0.009230	0.007690

TABLA 5.6: Error obtenido en el dominio de la frecuencia, del sujeto NICANA.

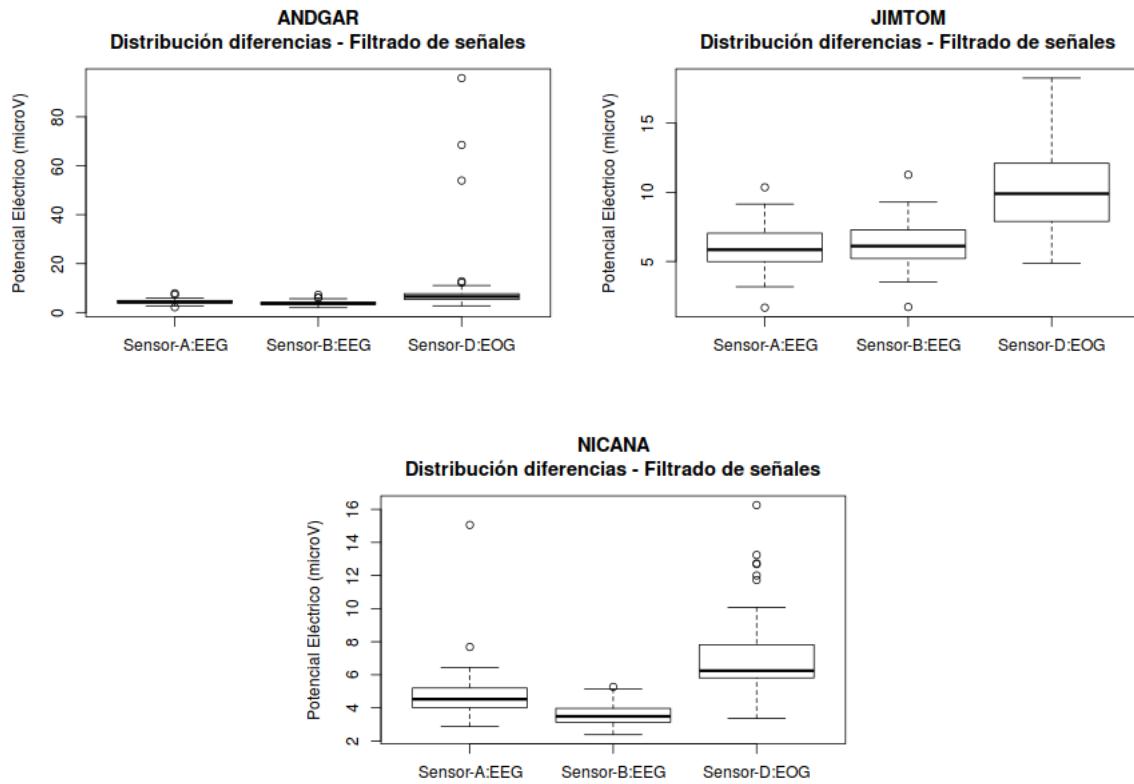


FIGURA 5.4: Distribución de las diferencias de las señales filtradas, en el dominio del tiempo, para los sujetos ANDGAR, JIMTOM y NICANA.

Al igual que pasa en el dominio del tiempo, el error cometido en el dominio de la frecuencia sigue estando en el orden de $10^{-3}V$.

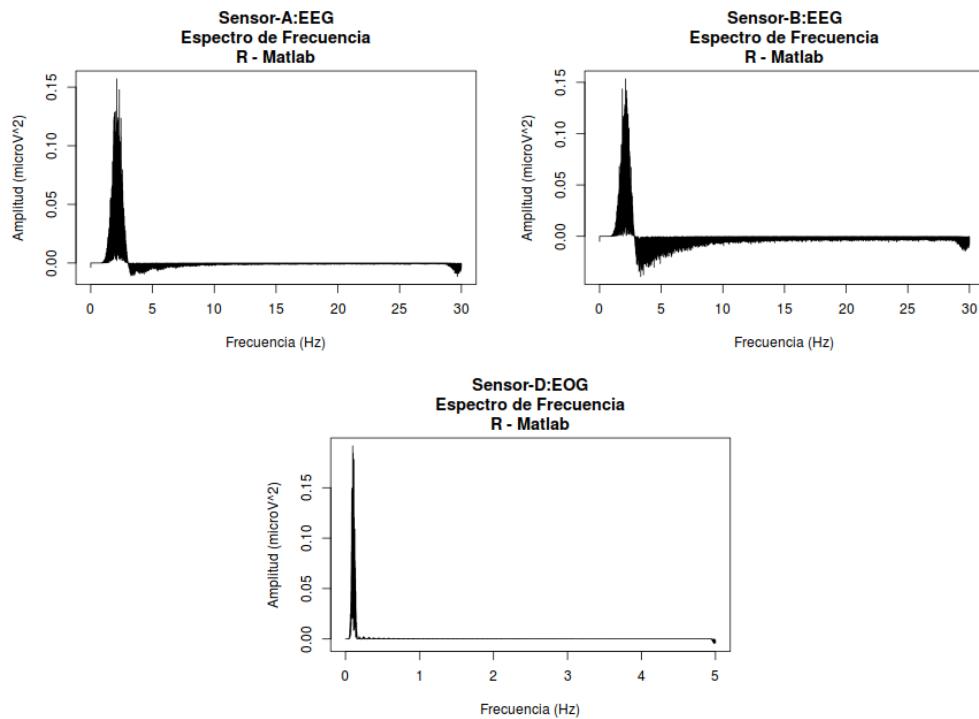


FIGURA 5.5: Diferencias en el dominio de la frecuencia de las señales filtradas, del sujeto ANDGAR.

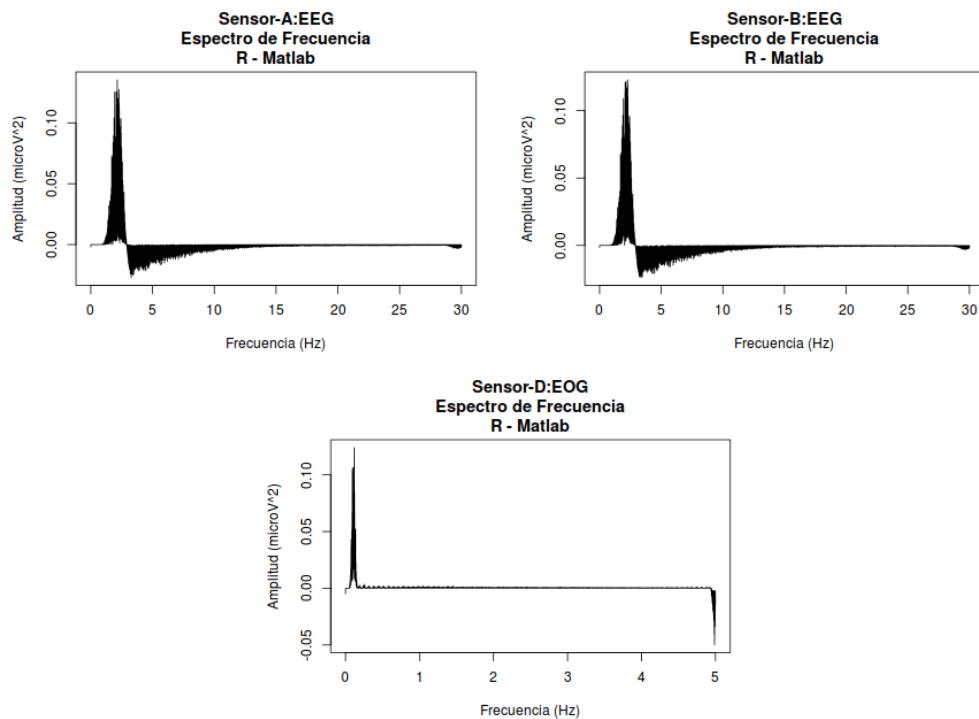


FIGURA 5.6: Diferencias en el dominio de la frecuencia de las señales filtradas, del sujeto JIMTOM.

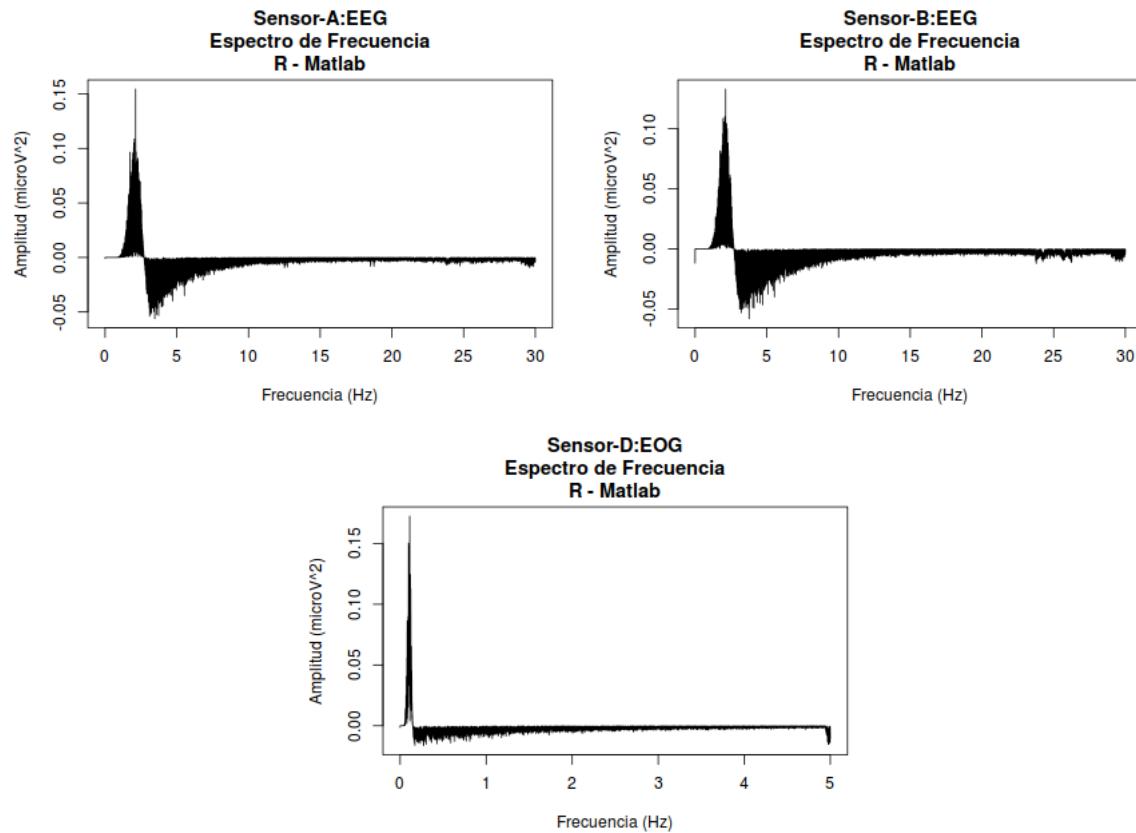


FIGURA 5.7: Diferencias en el dominio de la frecuencia de las señales filtradas, del sujeto NICANA.

5.2.3. Evaluación de la corrección ocular

Como se ha comentado en la sección 3.5.2, la corrección ocular además de proporcionar la señal EEG sin artefactos oculares, también proporciona una señal de la que se pueden inferir los parpadeos del sujeto. En esta sección se va a hacer una evaluación de ambas señales.

En lo referente a la **señal EEG corregida**, las Figuras 5.8, 5.9 y 5.10 muestran las diferencias de las señales corregidas de manera gráfica, para los tres sujetos.

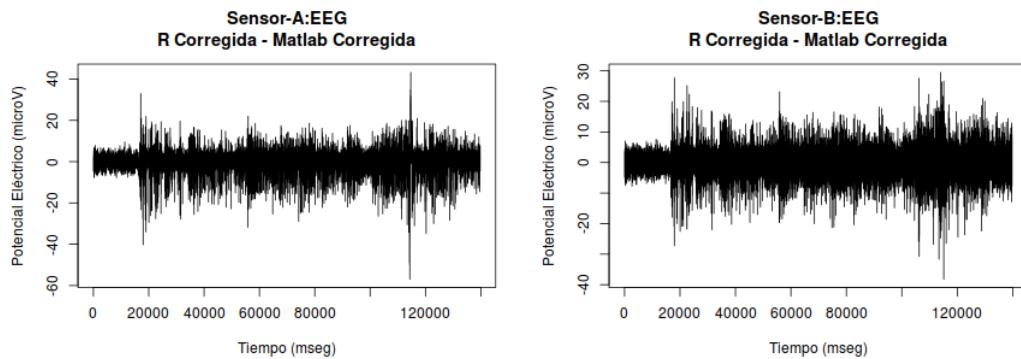


FIGURA 5.8: Diferencias de las señales corregidas, del sujeto ANDGAR.

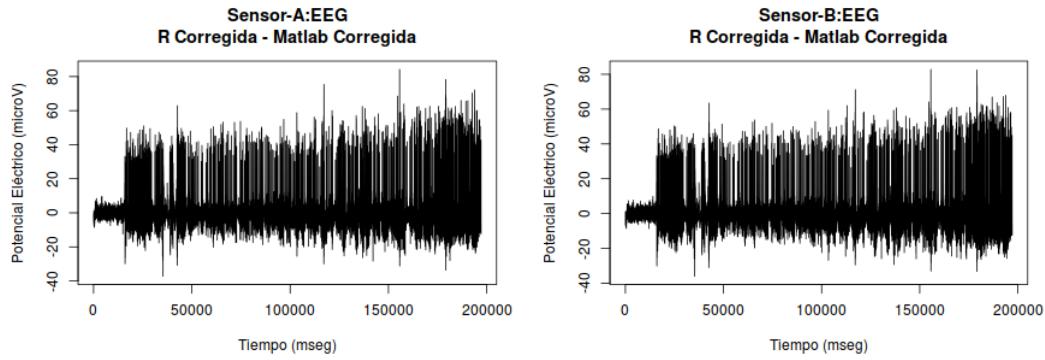


FIGURA 5.9: Diferencias de las señales corregidas, del sujeto JIMTOM.

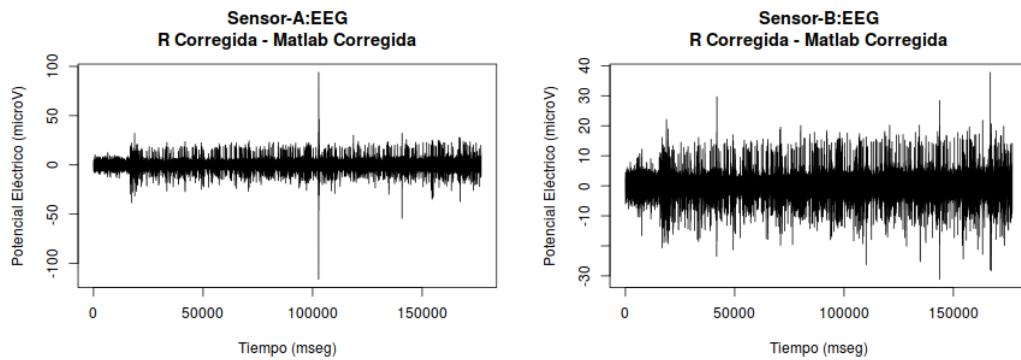


FIGURA 5.10: Diferencias de las señales corregidas, del sujeto NICANA.

Las Tablas 5.7, 5.8 y 5.9 muestran el error obtenido mediante la ecuación 5.1 para algunas de las tareas realizadas a los sujetos (son las mismas tareas que las mostradas en las Tablas 5.1, 5.2 y 5.3). En la parte inferior de las tablas aparece la media y la desviación típica del error obtenido para cada canal.

	Sensor-A:EEG	Sensor-B:EEG
Tarea 1	3.955406	3.468608
Tarea 2	2.961691	2.97647
Tarea 3	3.273036	3.092396
Tarea 4	4.217119	3.725877
Tarea 5	3.084533	2.971354
Tarea 6	3.270782	3.155958
Tarea 7	3.498713	3.094015
Tarea 8	3.714162	3.433555
Tarea 9	6.764311	6.052837
Tarea 10	4.949482	4.217963
—	—	—
media	4.487208	3.908225
desv. típica	1.953751	1.008255

TABLA 5.7: Error obtenido de las señales corregidas, para algunas tareas del sujeto ANDGAR.

	Sensor-A:EEG	Sensor-B:EEG
Tarea 1	6.965191	6.797005
Tarea 2	8.729696	8.420772
Tarea 3	6.812574	6.86022
Tarea 4	6.057019	6.132874
Tarea 5	4.961577	5.050222
Tarea 6	8.45808	8.385914
Tarea 7	11.459431	11.058017
Tarea 8	8.209605	7.952497
Tarea 9	13.105566	12.863604
Tarea 10	7.111303	6.987
—	—	—
media	10.478975	10.271844
desv. típica	3.576495	3.430086

TABLA 5.8: Error obtenido de las señales corregidas, para algunas tareas del sujeto JIMTOM.

	Sensor-A:EEG	Sensor-B:EEG
Tarea 1	3.870429	2.905557
Tarea 2	3.887054	2.981084
Tarea 3	4.354351	3.197116
Tarea 4	5.131081	4.22997
Tarea 5	3.297593	2.351899
Tarea 6	4.375944	3.197897
Tarea 7	5.434454	4.093114
Tarea 8	4.488391	3.263963
Tarea 9	3.605552	2.469776
Tarea 10	4.353353	3.451496
—	—	—
media	4.771514	3.540287
desv. típica	1.556003	0.656775

TABLA 5.9: Error obtenido de las señales corregidas, para algunas tareas del sujeto NICANA.

La señal sin artefactos oculares sigue estando en μV , por lo que el error cometido sigue estando en el orden de $10^{-3}V$.

Para los canales *Sensor-A:EEG* y *Sensor-B:EEG*, las **medias** de los errores obtenidos para el sujeto ANDGAR son: $4.487208 \mu V$ y $3.908225 \mu V$; para el sujeto JIMTOM son: $10.478975 \mu V$ y $10.271844 \mu V$; y para el sujeto NICANA son: $4.771514 \mu V$ y $3.540287 \mu V$. Mientras que las **desviaciones típicas** para el sujeto ANDGAR son: $1.953751 \mu V$ y $1.008255 \mu V$; para el sujeto JIMTOM son: $3.576495 \mu V$ y $3.430086 \mu V$; y para el sujeto NICANA son: $1.556003 \mu V$ y $0.656775 \mu V$.

La Figura 5.11 muestra las distribuciones de los resultados mostrados en las tablas anteriores para todos los sujetos.

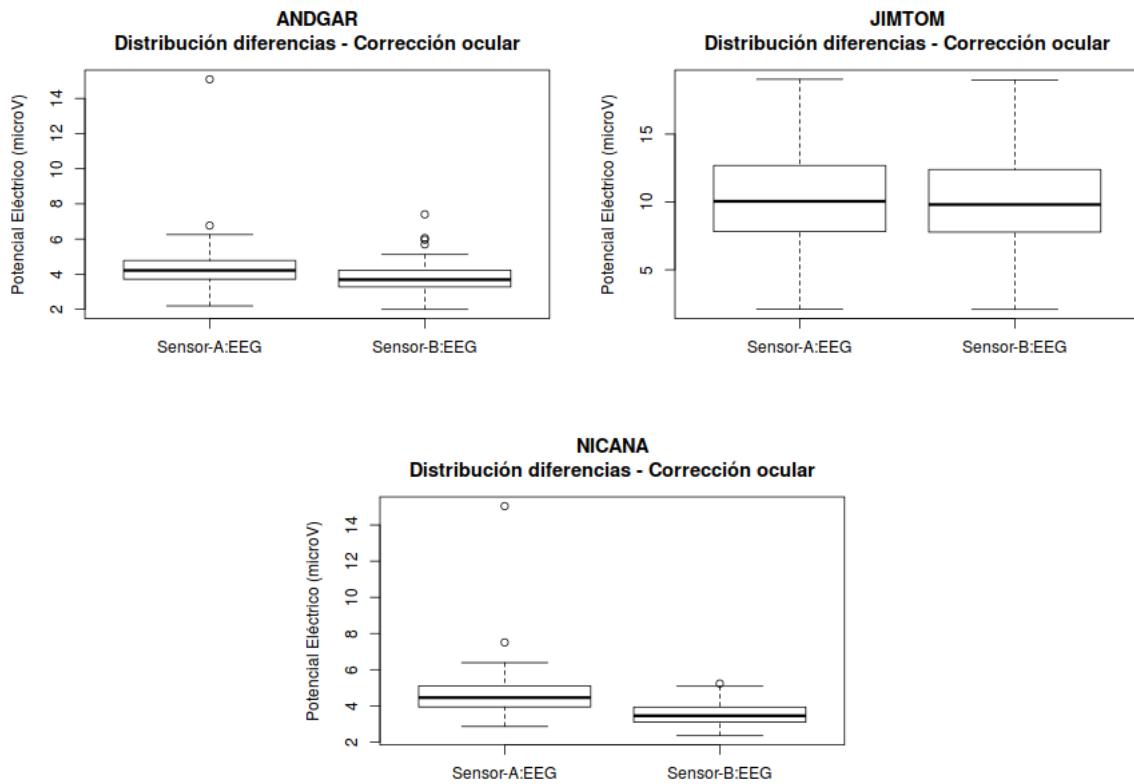


FIGURA 5.11: Distribución de las diferencia de las señales corregidas, para los sujetos ANDGAR, JIMTOM y NICANA.

En lo referente a la **señal de parpadeos**, las Figuras 5.12, 5.13 y 5.14 muestran las diferencias de las señales corregidas de manera gráfica, para los tres sujetos.

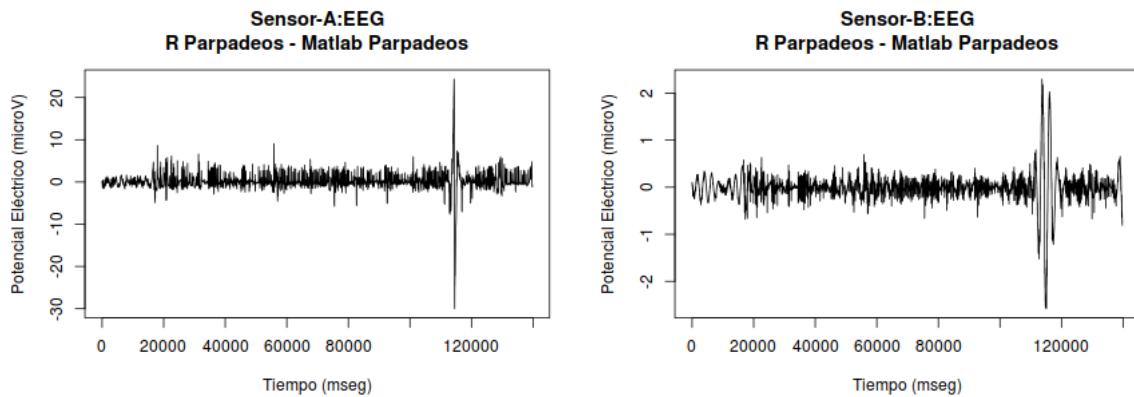


FIGURA 5.12: Diferencias de las señales de parpadeos, del sujeto ANDGAR.

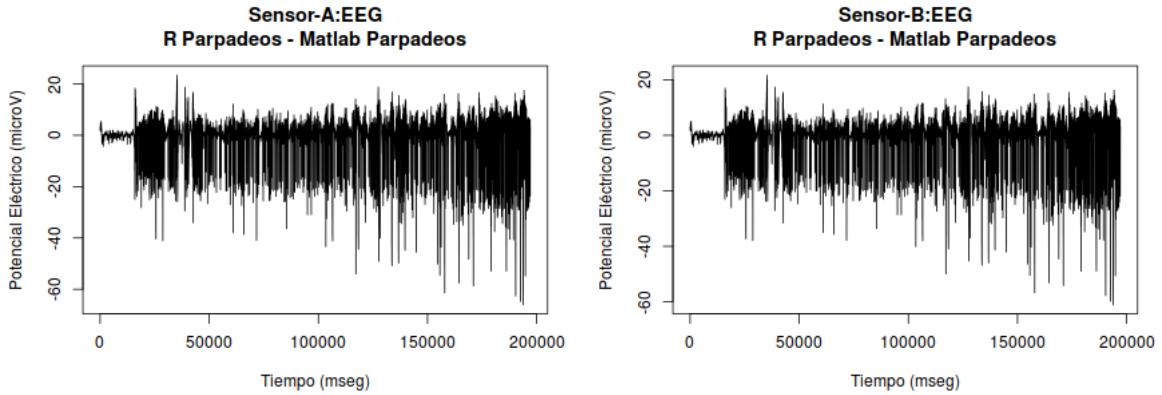


FIGURA 5.13: Diferencias de las señales de parpadeos, del sujeto JIMTOM.

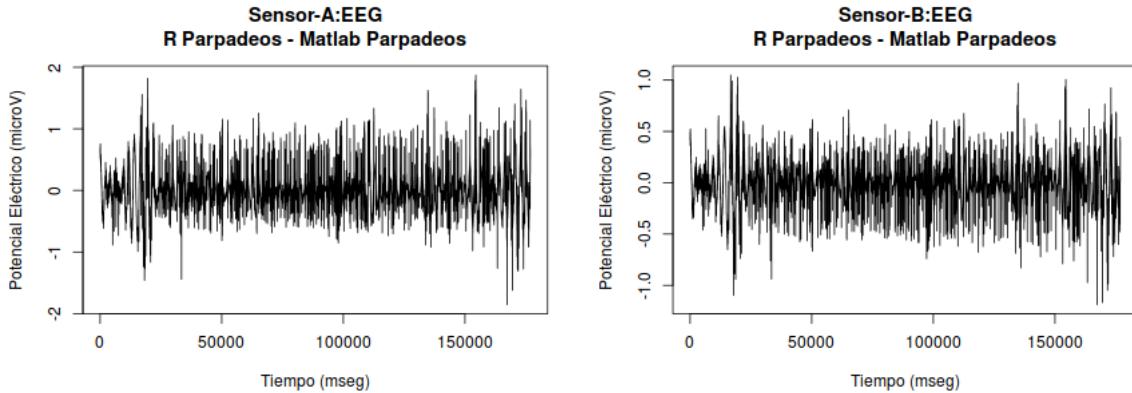


FIGURA 5.14: Diferencias de las señales de parpadeos, del sujeto NICANA.

Las Tablas 5.10, 5.11 y 5.12, muestran el error obtenido mediante la ecuación 5.1 para algunas de las tareas realizadas a los sujetos. En la parte inferior de las tablas aparece la media y la desviación típica del error obtenido para cada canal.

	Sensor-A:EEG	Sensor-B:EEG
Tarea 1	0.826173	0.080628
Tarea 2	0.698936	0.052039
Tarea 3	0.625833	0.053724
Tarea 4	1.247695	0.115864
Tarea 5	0.734201	0.105187
Tarea 6	0.898399	0.127848
Tarea 7	1.041569	0.152163
Tarea 8	0.849086	0.158693
Tarea 9	2.026909	0.224667
Tarea 10	1.306866	0.1828
—	—	—
media	1.447808	0.198453
desv. típica	1.920519	0.320161

TABLA 5.10: Error obtenido de las señales de parpadeos, para algunas tareas del sujeto ANDGAR.

	Sensor-A:EEG	Sensor-B:EEG
Tarea 1	4.740593	4.384114
Tarea 2	5.185884	4.799569
Tarea 3	4.405264	4.075093
Tarea 4	3.888635	3.596374
Tarea 5	3.305575	3.058846
Tarea 6	5.765759	5.336499
Tarea 7	8.022036	7.417091
Tarea 8	5.147192	4.764
Tarea 9	9.121283	8.434044
Tarea 10	4.472078	4.139116
—	—	—
media	7.270981	6.724651
desv. típica	3.030735	2.800255

TABLA 5.11: Error obtenido de las señales de parpadeos, para algunas tareas del sujeto JIMTOM.

	Sensor-A:EEG	Sensor-B:EEG
Tarea 1	0.198985	0.125651
Tarea 2	0.195663	0.123596
Tarea 3	0.200533	0.130793
Tarea 4	0.255196	0.16344
Tarea 5	0.200887	0.130093
Tarea 6	0.206601	0.130601
Tarea 7	0.25981	0.165496
Tarea 8	0.256496	0.175674
Tarea 9	0.196984	0.123079
Tarea 10	0.201644	0.133013
—	—	—
media	0.251661	0.157863
desv. típica	0.087165	0.049625

TABLA 5.12: Error obtenido de las señales de parpadeos, para algunas tareas del sujeto NICANA.

La señal de parpadeos está medida también en μV , por lo que el error cometido sigue estando en el orden de $10^{-3}V$.

Para los canales *Sensor-A:EEG* y *Sensor-B:EEG*, las **medias** de los errores obtenidos para el sujeto ANDGAR son: $1.447808 \mu V$ y $0.198453 \mu V$; para el sujeto JIMTOM son: $7.270981 \mu V$ y $6.724651 \mu V$; y para el sujeto NICANA son: $0.251661 \mu V$ y $0.157863 \mu V$. Mientras que las **desviaciones típicas** para el sujeto ANDGAR son: $1.920519 \mu V$ y $0.320161 \mu V$; para el sujeto JIMTOM son: $3.030735 \mu V$ y $2.800255 \mu V$; y para el sujeto NICANA son: $0.087165 \mu V$ y $0.049625 \mu V$.

La Figura 5.15 muestra las distribuciones de los resultados mostrados en las tablas anteriores para todos los sujetos.

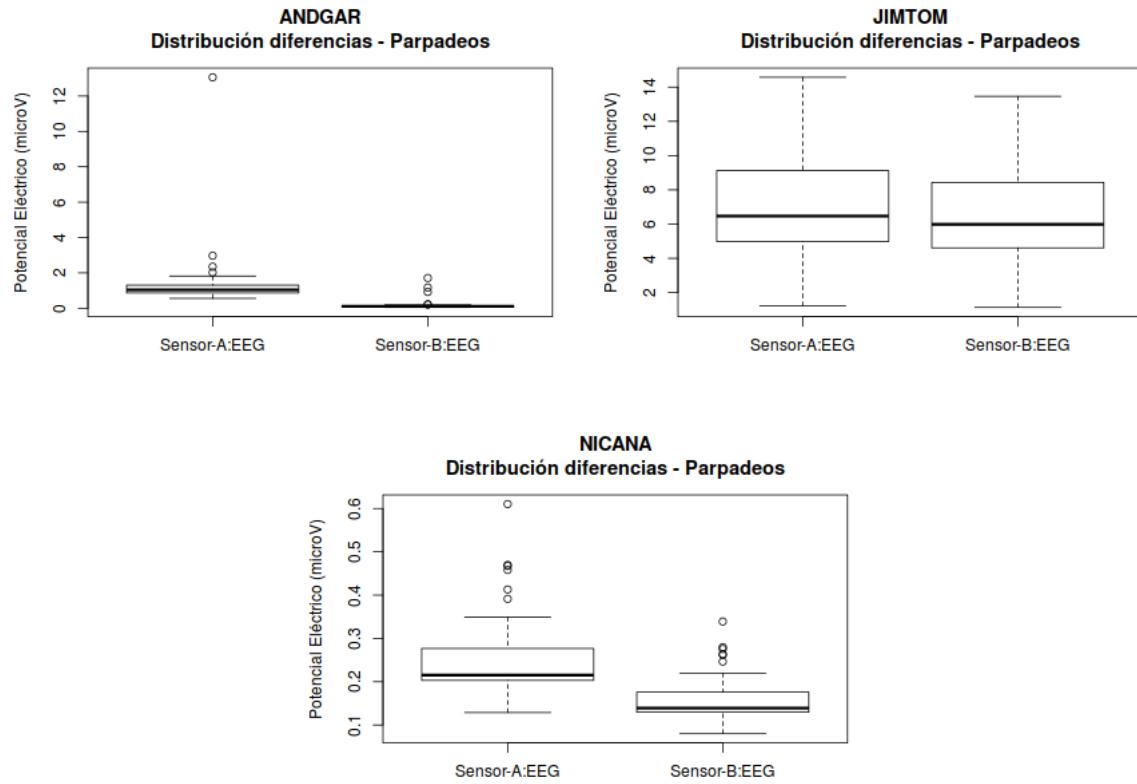


FIGURA 5.15: Distribución de las diferencia de las señales de parpadeos, para los sujetos ANDGAR, JIMTOM y NICANA.

5.3. Evaluación del IAF

Uno de los pasos más importantes en el cálculo de los índices es el cálculo de la IAF (*Individual Alpha Frequency*). Para su cálculo, como se expone en la sección 4.4, es necesario calcular la PSD (*Power Spectral Density*) para la señal correspondiente a la tarea de *ojos cerrados*.

En esta sección se va a hacer una evaluación sobre la IAF, basándonos en las implementaciones que tienen R y Matlab para el cálculo de la PSD. A diferencia del resto de evaluaciones, en esta sección se va a hacer una evaluación visualizando las PSD obtenidas por R y Matlab, ya que los resultados no tienen las mismas dimensiones, y realizar una evaluación hallando el error como se ha hecho en la sección anterior, no es posible. Por ello, las Figuras 5.16, 5.17 y 5.18 muestran las gráficas de las PSD obtenidas para todas las señales filtradas, de los sujetos ANDGAR, JIMTOM y NICANA.

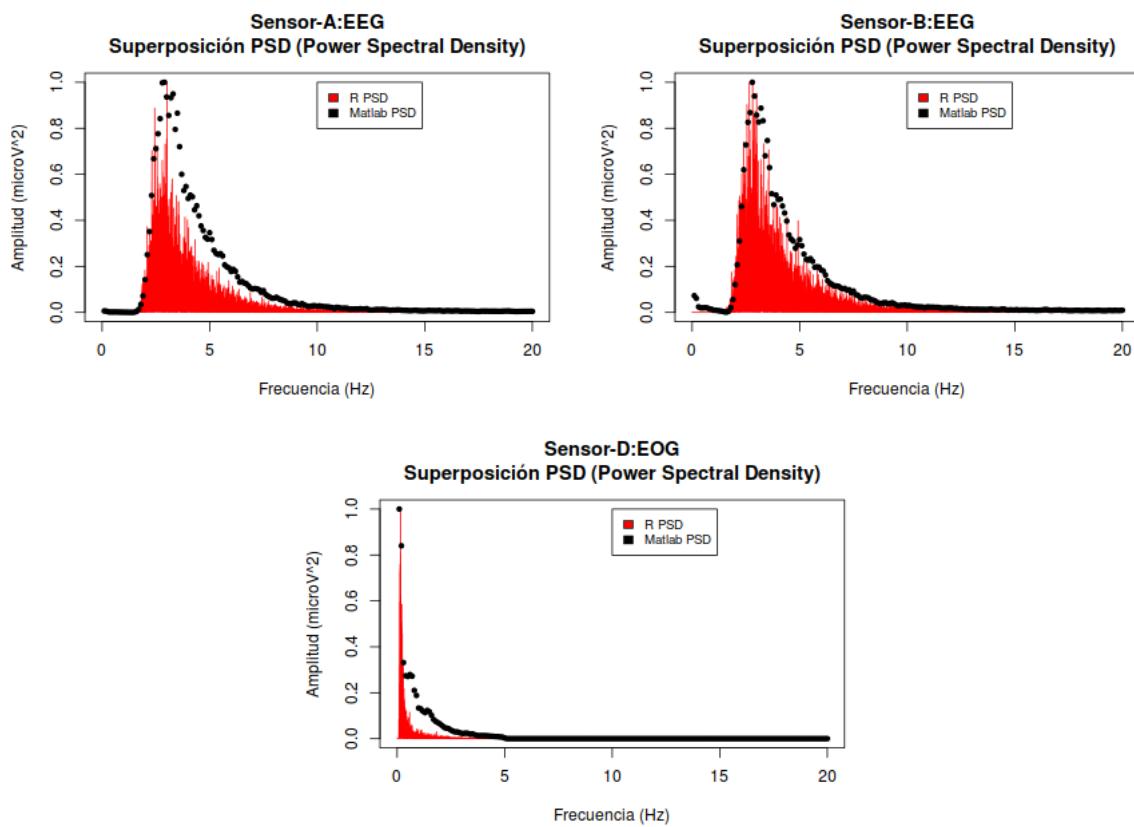


FIGURA 5.16: Superposición de las PSD obtenidas por R y Matlab, para todas las señales filtradas del sujeto ANDGAR.

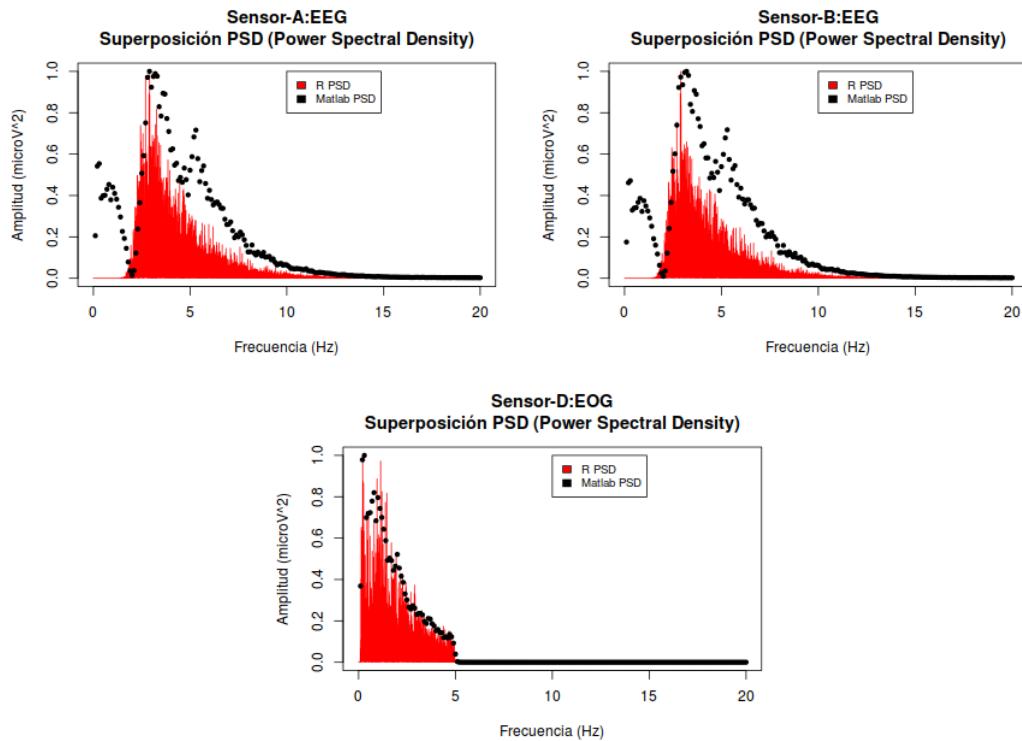


FIGURA 5.17: Superposición de las PSD obtenidas por R y Matlab, para todas las señales filtradas del sujeto JIMTOM.

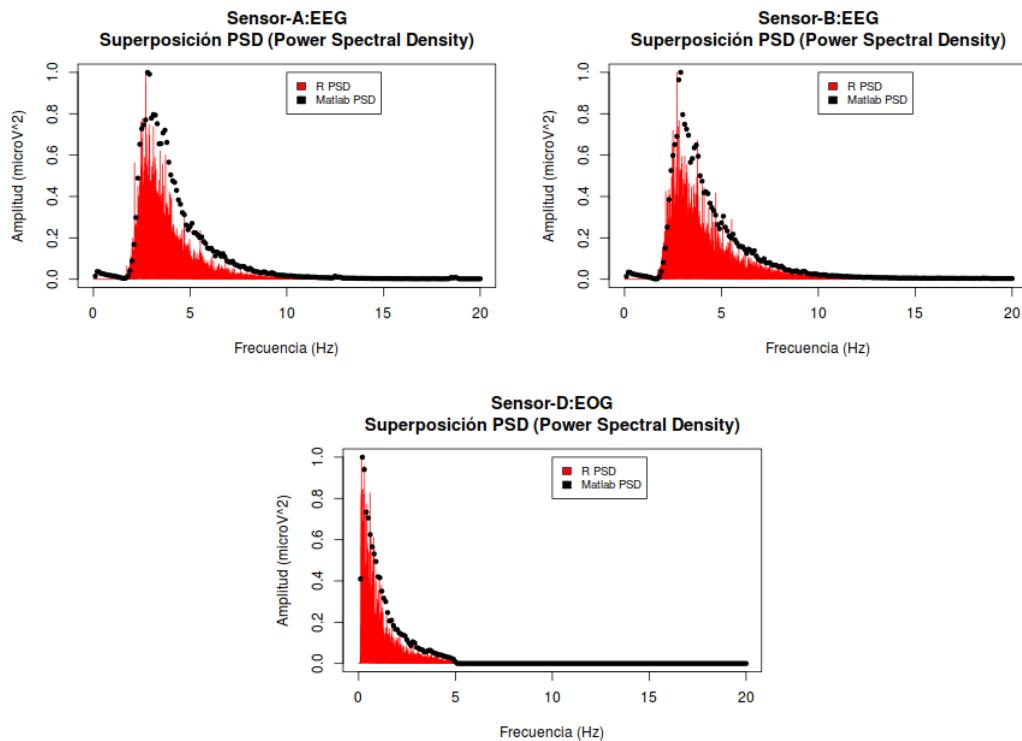


FIGURA 5.18: Superposición de las PSD obtenidas por R y Matlab, para todas las señales filtradas del sujeto del sujeto NICANA.

Como se puede observar, la forma de las funciones es prácticamente la misma. Las diferencias entre ambas se deben a la distinta implementación que R y Matlab tienen para el cálculo de la PSD. Debida a esta diferencia entre las implementaciones, la IAF obtenida por R va a tener cierto error con el obtenido por Matlab. Para ver estas diferencias, la Tabla 5.13 muestra los valores de las IAF obtenidas para los tres sujetos anteriores, tanto en R como en Matlab (recuérdese que la IAF se mide en Hz).

	ANDGAR	JIMTOM	NICANA
R	10.41	9.28	9.25
Matlab	10.4	9.1	9.2

TABLA 5.13: Comparación del IAF obtenido en R y Matlab, para los tres sujetos.

Observando los resultados y teniendo en cuenta que en el cálculo de la PSD hay error debido a las distintas implementaciones de R y Matlab, se puede decir que el valor de la IAF es prácticamente el mismo. De hecho, como es de esperar, cuanto mayor sea el error en el cálculo de la PSD, mayor será el error en los valores de la IAF, como se puede observar con el sujeto JIMTOM (Figura 5.17).

5.4. Evaluación de los índices

5.4.1. Procedimiento para la obtención de los datos a evaluar

Para realizar la evaluación de los índices AI, AW, EI y MI calculados mediante el procedimiento descrito en la sección 4, se ha realizado el siguiente procedimiento:

1. Realizar el filtrado *pasa-banda* y la corrección ocular en R y en Matlab, tal y como se ha descrito en la sección 4.2.
2. Calcular los índices con los datos de R y de Matlab, siguiendo el procedimiento descrito desde la sección 4.3 a la sección 4.6.
3. Para cada tarea t realizada al sujeto, se calcula, para cada índice $indice$, su error mediante la ecuación 5.2, siendo $indice_R^t$, $indice_M^t$ los valores de R y Matlab para el índice $indice$:

$$Error(indice)^t = |indice_R^t - indice_M^t| \quad (5.2)$$

5.4.2. Evaluación de los índices

A diferencia de la evaluación realizada en la sección 5.2, en la que se trata con señales en tiempo (y en frecuencia), en esta sección se trata con valores concretos de los índices AI, AW, EI y MI para cada tarea. Por esta razón, se van a mostrar tablas que muestran el error obtenido para cada índice de cada una de las tareas realizadas a los sujetos ANDGAR, JIMTOM y NICANA, además de las gráficas en la que se visualizan la distribución de estos errores para cada sujeto.

Las Tablas 5.14, 5.15 y 5.16, muestran el error obtenido para algunas de las tareas realizadas a los sujetos. Estas tareas son las mismas que las mostradas en la sección 5.2. En la parte inferior de las tablas aparece la media y la desviación típica del error obtenido para cada índice.

	AI	MI	AW	EI
Tarea 1	2e-04	0.000149	0.001029	3.6e-05
Tarea 2	0.000251	0.000266	0.001212	0.002531
Tarea 3	0.000694	6e-06	0.000227	0.005197
Tarea 4	0.000234	0.000477	0.000821	0.003529
Tarea 5	0.002056	0.000214	0.00423	0.003702
Tarea 6	0.001183	0.001049	0.001194	0.001157
Tarea 7	0.003071	0.00011	0.005935	0.000944
Tarea 8	0.008838	0.001865	0.010771	0.004591
Tarea 9	0.000377	0.000232	0.001777	0.002067
Tarea 10	0.004687	0.000679	0.004602	0.002353
—	—	—	—	—
media	0.004286	0.000907	0.006211	0.003347
desv. típica	0.008505	0.001135	0.013068	0.003741

TABLA 5.14: Valor absoluto de la diferencia entre los índices calculados por R y Matlab, para algunas tareas del sujeto ANDGAR.

	AI	MI	AW	EI
Tarea 1	0.000302	0.005234	0.000307	0.001298
Tarea 2	0.000685	0.006266	0.000704	0.021058
Tarea 3	2.9e-05	0.002676	3.8e-05	0.016334
Tarea 4	0.000271	0.005497	0.000261	0.000212
Tarea 5	0.000161	0.002558	0.000543	0.012447
Tarea 6	0.001249	0.003304	0.000513	0.036289
Tarea 7	0.000914	0.00537	0.003618	0.01037
Tarea 8	0.00142	0.001215	0.008234	0.008744
Tarea 9	0.000271	0.003349	0.006864	0.0021
Tarea 10	0.000123	0.003814	0.00049	0.054608
—	—	—	—	—
media	0.002538	0.00464	0.002341	0.029328
desv. típica	0.004816	0.003578	0.003773	0.038349

TABLA 5.15: Valor absoluto de la diferencia entre los índices calculados por R y Matlab, para algunas tareas del sujeto JIMTOM.

	AI	MI	AW	EI
Tarea 1	0.000345	0.000768	0.000976	0.000128
Tarea 2	0.001534	0.000385	0.002023	0.001458
Tarea 3	0.010153	0.002915	0.011372	0.001115
Tarea 4	0.000203	0.00027	0.001729	0.001524
Tarea 5	0.008721	0.0025	0.015029	0.003323
Tarea 6	0.016628	0.003696	0.014637	0.003589
Tarea 7	0.000104	0.000704	5.2e-05	0.000151
Tarea 8	0.000157	0.003589	0.000133	0.000727
Tarea 9	0.000105	0.000332	0.000258	0.004948
Tarea 10	0.000952	0.000404	1.8e-05	0.00179
—	—	—	—	—
media	0.003546	0.001193	0.00388	0.004482
desv. típica	0.007378	0.001395	0.007389	0.007594

TABLA 5.16: Valor absoluto de la diferencia entre los índices calculados por R y Matlab, para algunas tareas del sujeto NICANA.

Los índices no tienen unidad de medida, sino que simplemente es un valor que es significativo para el experto que analiza los resultados. Por eso mismo se puede observar que el error cometido está en el orden de 10^{-3} , al igual que en la sección 5.2.

Para los índices AI, MI, AW y EI, las **medias** de los errores obtenidos para el sujeto ANDGAR son: 0.004286, 0.000907, 0.006211 y 0.003347; para el sujeto JIMTOM son: 0.002538, 0.00464, 0.002341 y 0.029328; y para el sujeto NICANA son: 0.003546, 0.001193, 0.00388 y 0.004482. Mientras que las **desviaciones típicas** para el sujeto ANDGAR son: 0.008505, 0.001135, 0.013068 y 0.003741; para el sujeto JIMTOM son: 0.004816, 0.003578, 0.003773 y 0.038349; y para el sujeto NICANA son: 0.007378, 0.001395, 0.007389 y 0.007594.

La Figura 5.19 muestra las distribuciones de los resultados mostrados en las tablas anteriores para todos los sujetos.

5.5. Conclusiones obtenidas

Tras la realización de todas las evaluaciones anteriores, el resumen de los valores obtenidos es el siguiente:

- Para el filtrado de las señales: el error cometido está en el orden de $10^{-3}V$.
- Para la corrección ocular: el error cometido está en el orden de $10^{-3}V$.
- Para la IAF: el error cometido está en el orden de $10^{-1}Hz$.
- Para los índices: el error cometido está en el orden de 10^{-3} .

Para obtener las conclusiones de estas evaluaciones, hay que tener en cuenta que entre R y Matlab hay cierto error, debido a las distintas implementaciones de la aplicación del filtro con los coeficientes FIR, en la función `eegfilt()` (3.5.1); de la regresión lineal, usada en la función `gratton()` (3.5.2); y del cálculo de la PSD, usada para el cálculo de la IAF (4.4).

Por ello y debido a que el error obtenido en los valores de los índices es pequeño (del orden de 10^{-3}), se puede decir que este error se debe a las diferencias entre las implementaciones que tiene R y Matlab. Además, hay que tener en cuenta que en el proceso de captación

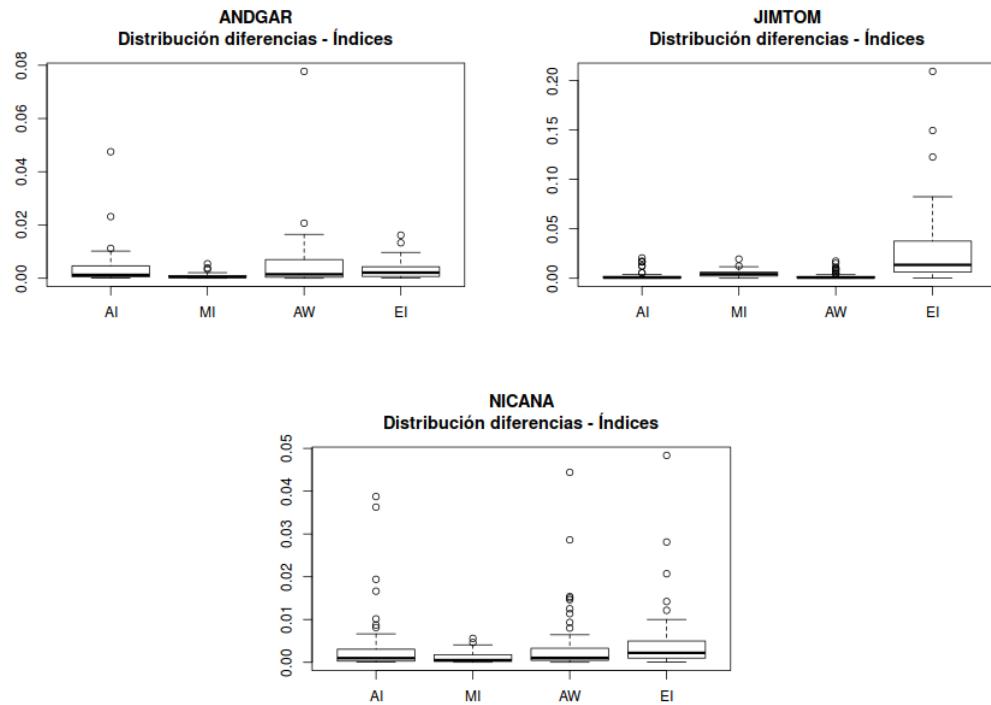


FIGURA 5.19: Distribución de las diferencias de los índices AI, AW, EI y MI, para los sujetos ANDGAR, JIMTOM y NICANA.

de las señales puede haber errores debido, por ejemplo, a que durante la realización de una prueba uno de los sensores se mueva. Es por este motivo por lo que, en algunos de los resultados anteriores (como se puede observar en la Figura 5.15), hayan valores atípicos (*outliers*).

Por lo tanto, teniendo en cuenta lo anterior y que el error cometido en el cálculo de los índices es pequeño, se puede concluir con que el cálculo realizado por este proyecto es correcto.

Capítulo 6

Conclusiones y vías futuras

6.1. Conclusiones

La realización de este Trabajo Fin de Grado ha proporcionado una herramienta web de software libre, capaz de realizar todo el procedimiento para la obtención de medidas objetivas de la señal del electroencefalograma, a partir del estudio de las bandas α , β y θ , además de proporcionar la visualización de señales y valores intermedios en todas las etapas del procedimiento. Dichas medidas objetivas son los siguientes cuatro índices: *Índice de Atención* (AI) (banda α), *Índice de Aproximación-Rechazo* (AW) (banda α), *Índice de Memorización* (MI) (banda θ) e *Índice de Compromiso* (EI) (todas las bandas).

Se han analizado y estudiado los siguientes tipos de filtros de señales: filtros FIR o filtros de Respuesta Finita al Impulso y filtros IIR o filtros de Respuesta Infinita al Impulso. Debido a las ventajas que proporcionan los filtros FIR para la realización del filtrado de señales *off-line*, se ha decidido por implementar una función de filtrado de señales que hace uso de un filtro FIR. También se han analizado y estudiado distintas herramientas para la visualización de señales en la herramienta. Estas herramientas son: `plot`, `ggplot2`, `highcharter` y `dygraphs`. La herramienta `dygraphs` proporciona una manera simple, cómoda y eficiente de poder realizar gráficas interactivas de series temporales, por lo que para la visualización de las señales (que son series temporales) se realiza con esta herramienta.

Se han implementado las funciones necesarias para la realización del filtrado de las señales, así como de la eliminación de los artefactos oculares de la señal EEG. También se ha implementado todo el proceso para la obtención de los índices AI, AW, EI y MI.

Se ha hecho uso del framework web Shiny para el desarrollo e implementación de la herramienta web, que proporciona de manera muy simple y cómoda, la creación de páginas web interactivas, solamente con código en R, sin tener casi conocimientos de HTML, CSS y JavaScript.

Por último, se ha realizado una evaluación sobre los resultados que proporciona la herramienta, en comparación con los resultados obtenidos por el TFG [29]. De esta evaluación podemos concluir que la herramienta calcula de manera correcta los índices, con un error inherente a las distintas implementaciones que R y Matlab tienen para funciones que realizan el mismo cálculo. Las medias y desviaciones típicas de los errores obtenidos se pueden ver en las secciones 5.2.2, 5.2.3 y 5.4.

6.1.1. Conclusiones personales

Uno de los principales motivos por los que decidí estudiar informática fue idea de aplicar la informática al campo de la medicina, creando algo que pudiera ayudar en la labor del

personal sanitario. Conforme fui avanzando en la carrera, me fui dando cuenta de que podía llevar a cabo aquella idea, llamémoslo sueño, con la que empecé.

Este trabajo me ha proporcionado la oportunidad de acercarme al campo de la medicina, más concretamente al estudio del electroencefalograma y la información que proporciona del sujeto del que se obtiene. También me ha permitido darme cuenta de que por muy frustrante que pueda parecer al principio, todos los trabajos de investigación tienen su fruto, y se podría decir que la herramienta web desarrollada en este trabajo es uno de los frutos que se ven después de muchos años de investigación por parte de los grupos de investigación AIKE y NiCE. Además, estoy muy seguro de que esta herramienta servirá de ayuda en posteriores estudios y trabajos en este campo.

Además de la complejidad que supuso introducirme en el campo del electroencefalograma, en este trabajo se encuentra una dificultad intrínseca que se aprecia apenas, y es la dificultad en lo relativo al procesamiento de señales, algo que no se ve prácticamente durante el grado.

El hecho de que toda la herramienta esté implementada y diseñada en R, me ha permitido disfrutar de este gran lenguaje y aprender muchísimo sobre su potencial. También me ha ayudado a ser más autodidacta, pues he tenido que enfrentarme al aprendizaje (desde cero) del framework web Shiny, y estoy seguro de que sin él, la implementación web habría sido todo un reto, pues el desarrollo *frontend* no es uno de mis puntos fuertes.

Por fin, cuatro años después, con la realización de este Trabajo Fin de Grado puedo decir que el sueño con el que empecé la carrera ha sido cumplido y me siento bastante orgulloso de haber aportado mi granito de arena.

6.2. Vías futuras

En cuanto a las líneas futuras de este TFG podemos destacar:

- Mejorar la interfaz de usuario de la herramienta, para que sea más intuitiva y atractiva. Además de proporcionar una mayor configuración en la visualización de las señales.
- Realizar más experimentos para confirmar los resultados obtenidos. Se trataría de variar el tipo de tarea realizada y la tipología de sujetos utilizados.
- Análisis de otros tipos de filtros y estudiar su eficacia. Esto nos llevaría también al análisis de diferentes librerías de procesados de señales en diferentes lenguajes.
- Generalización del proceso de obtención de índices para el caso de poder disponer de más electrodos.
- La utilización de más electrodos nos permitiría la utilización de técnicas basadas en el Análisis de Componentes Independientes (ACI) para la eliminación de artefactos en las señales, especialmente la influencia del movimiento de los ojos y músculos.
- Incluir técnicas de visualización de la información en la dimensión espacial, para poder ver qué partes del cerebro están siendo activadas en cada una de las tareas. Esto sería posible gracias a la aplicación del ACI.
- Tampoco descartamos, gracias a los conocimientos adquiridos, plantear el cálculo de los índices en tiempo real, abriendo la puerta al desarrollo de aplicaciones basadas en la interfaz cerebro-máquina.

Apéndice A

Formato de los ficheros de entrada

El formato del **fichero de señales** contiene los siguientes elementos:

- **Cabecera.** La cabecera debe seguir la siguiente estructura:

```
Session: <nombre_de_la_sesión>
Data: dd-mm-aaaa
Time: hh:mm:ss
Duration: <duración_prueba> (en segundos)
Output rate: <frecuencia_de_muestreo> (en samples/sec)
```

- **Canales.** Los canales debe seguir la siguiente estructura:

- Una fila donde se indique la frecuencia de muestreo de cada uno de los sensores, separados por comas. Por ejemplo:

```
256 SPS,256 SPS,256 SPS,32 SPS,128 SPS,32 SPS,32 SPS
```

- Una fila donde se indique el nombre de los sensores, separados por comas. Por ejemplo (se muestran en dos filas, pero porque no se visualizan completamente en una sola fila):

```
Sensor-A:EEG,Sensor-B:EEG,Sensor-D:EOG,Sensor-E:SC/GSR,
Sensor-F:Temp.,Sensor-G:BVP,[G] Heart Rate
```

- Tantas filas como muestras haya de cada uno de los sensores, separados por comas. Por ejemplo:

```
1228.954,1341.187,-55.318,1.811,0.000,-69.115,70.000
7529.662,8279.951,690.997,1.811,0.000,-69.115,70.000
15528.828,17065.146,2181.196,1.811,0.000,-69.115,70.000
17231.315,18899.636,2791.035,1.811,0.000,-69.115,70.000
18367.166,20116.100,2867.227,1.811,0.000,-69.115,70.000
```

El formato del **fichero de marcas de tiempo** contiene los siguientes elementos:

- En la primera fila, el nombre de la sesión a la que hace referencia (debe ser el mismo que el que se puso en Session:, en el fichero de señales).
- Tantas filas como tareas se hayan realizado al sujeto. Estas filas contienen tres campos separados por comas:
 - Nombre de la tarea.
 - Hora de inicio de la tarea (en hh:mm:ss).
 - Hora de finalización de la tarea (en hh:mm:ss).

Un ejemplo de las tres primeras líneas de este fichero podrían ser:

```
SESSION-1
<closed_eyes>,08:57:23,08:58:23
<basal>,08:58:40,08:59:40
```

Apéndice B

Manual de Usuario

El manual de usuario para esta herramienta web, que recibe el nombre de EEG RTool, se encuentra alojado en la siguiente dirección web:

<https://jmssalas.github.io/EEG-RTool-UserManual/>

Glosario

ACI Análisis de Componentes Independientes. 64

AI Attention Index. 7, 29, 30, 35, 36, 52, 53, 61, 63

AIKE Artificial Intelligence Knowledge Engineering. 2, 64

AW Approach-Withdrawal Index. 7, 29, 30, 35–37, 52, 53, 61, 63

BS Basal o baseline. 27, 34, 36

BVP Blood Volume Pulse. 28

CANTAB Cambridge Neuropsychological Test Automated Battery. 27, 28

CSS Cascading Style Sheets. 12, 63

ECG Electrocardiograma. 28

EEG Electroencefalograma. 1–3, 5–8, 10, 11, 13, 23–28, 30–33, 41, 43, 63

EI Engagement Index. 7, 29, 30, 35, 36, 52, 53, 61, 63

EMG Electromiograma. 28

EOG Electrooculograma. 23–25, 28, 30–32, 41

ERT Emotion Recognition Task. 28

FIR Finite Impulse Response. 9–11, 24, 25, 55, 63

GFP Global Field Power. 35

GPS Global Power Spectrum. 22, 24, 33, 34

HR Heart Rate. 28

HTML HyperText Markup Language. 12, 63

IAF Individual Alpha Frequency. 16, 22, 24, 33, 34, 48–51, 55

IIR Infinite Impulse Response. 10, 11, 63

MI Memorization Index. 7, 29, 30, 35, 36, 52, 53, 61, 63

NiCE Neurociencia clínica y experimental. 2, 27, 64

OC Fase de ojos cerrados. 27

OTS One Touch Stockings. 28

PSD Power Spectral Density. 33, 48, 50, 55, 58–60

RMf Resonancia Magnética Funcional. 5

RMN Resonancia Magnética Nuclear. 5

RMSE Root Mean Square Error. 39–47, 49

RTI Reaction Time. 28

SC Skin Conductance. 28

SNC Sistema Nervioso Central. 5

SWM Spatial Working Memory. 28

TAC Tomografía Axial Computerizada. 5

TASK Tarea. 27, 29

TFG Trabajo Fin de Grado. 2, 15, 39, 63, 64

Bibliografía

- [1] BIOPAC Systems Inc. Iif vs. fir filters, 2014. URL <https://www.biopac.com/knowledge-base/iir-vs-fir-filters/>. (última visita 11/07/2017).
- [2] Electronics Tutorials. Band stop filter. URL <http://www.electronics-tutorials.ws/filter/band-stop-filter.html>. (última visita 11/07/2017).
- [3] Fundamentals of Statistics. Types of filters, 2012. URL http://www.statistics4u.com/fundstat_eng/cc_filtering.html. (última visita 11/07/2017).
- [4] Jean Gotman. Digital eeg from basics to advanced analysis, 2009. URL http://www.cnsfederation.org/uploads/content/cscn/documents/2009-digital_eeg-gotman.pdf. (última visita 11/07/2017).
- [5] Iowegian International Corporation. Digital signal processing faqs. URL <http://dspguru.com/dsp/faqs>. (última visita 11/07/2017).
- [6] Swartz Center for Computational Neuroscience. Matlab fir filtering eeglab/eegfilt, 1997. URL <https://github.com/jukka/eeglab/blob/master/functions/sigprocfunc/eegfilt.m>. (última visita 11/07/2017).
- [7] Marcelino Martínez Sober, Juan Gómez Sanchis, Luis Gómez Chova, Antonio Serrano López, and Joan Vila Francés. Diseño de filtros fir. Escuela Técnica Superior de Ingeniería, Universidad de Valencia., 2009-2010. URL http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema_3._diseno_de_filtros_fir.pdf. (última visita 11/07/2017).
- [8] Marcelino Martínez Sober, Juan Gómez Sanchis, Luis Gómez Chova, Antonio Serrano López, and Joan Vila Francés. Diseño de filtros iir. Escuela Técnica Superior de Ingeniería, Universidad de Valencia., 2009-2010. URL http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema_4_diseno_de_filtros_iir.pdf. (última visita 11/07/2017).
- [9] Research Institute Brainclinics. Real-time filtering in bioexplorer. URL <http://www.brainclinics.com/dynamic/media/1/documents/Onderzoeksinstituut/Filtering%20in%20BioExplorer.pdf>. (última visita 11/07/2017).
- [10] Andreas Widmann, Erich Schröger, and Burkhard Maess. Digital filter design for electrophysiological data - a practical approach. *Journal of Neuroscience Methods*, 2014.
- [11] Swartz Center for Computational Neuroscience. Eeglab, 2017. URL <https://sccn.ucsd.edu/eeglab/>. (última visita 11/07/2017).
- [12] Swartz Center for Computational Neuroscience. Matlab ocular correction eeglab/gratton, 2007. URL <https://github.com/ihrke/eeglab-gratton/blob/master/gratton.m>. (última visita 11/07/2017).

- [13] MathWorks (2017). Matlab, 2017. URL <https://es.mathworks.com/products/matlab.html>. (última visita 11/07/2017).
- [14] R Core Team (2017). R: A language and environment for statistical computing, 2017. URL <https://www.R-project.org/>. (última visita 11/07/2017).
- [15] John W. Eaton. Gnu octave, 2017. URL <https://www.gnu.org/software/octave/>. (última visita 11/07/2017).
- [16] Nathaniel E. Helwig. Toolkit for electroencephalography data. Technical report, R Project, 2015.
- [17] World Wide Web Consortium. W3c html, 2017. URL <https://www.w3.org/html/>. (última visita 11/07/2017).
- [18] World Wide Web Consortium. Cascading style sheets (css), 2017. URL <https://www.w3.org/Style/CSS/>. (última visita 11/07/2017).
- [19] World Wide Web Consortium. Javascript web apis, 2017. URL <https://www.w3.org/standards/webdesign/script>. (última visita 11/07/2017).
- [20] RStudio (2017). Shiny - the basic parts of a shiny app, 2017. URL <https://shiny.rstudio.com/articles/basics.html>. (última visita 08/08/2017).
- [21] R Core Team (2017). Rstudio: Open source and enterprise-ready professional software for r., 2017. URL <https://www.rstudio.com>. (última visita 11/07/2017).
- [22] The PHP Group. Php, 2017. URL <http://php.net>. (última visita 11/07/2017).
- [23] R Core Team (2017). plot function, 2017. URL <https://www.rdocumentation.org/packages/graphics/versions/3.4.0/topics/plot>. (última visita 11/07/2017).
- [24] Hadley Wickham. ggplot2, 2017. URL <http://ggplot2.tidyverse.org/index.html>. (última visita 11/07/2017).
- [25] RStudio (2017). dygraphs por r, 2017. URL <https://rstudio.github.io/dygraphs/>. (última visita 11/07/2017).
- [26] Joshua Kunst. Highcharter, 2017. URL <http://jkunst.com/highcharter/>. (última visita 11/07/2017).
- [27] Leland Wilkinson. *The Grammar of Graphics*. Springer-Verlag New York, 2 edition, 2005.
- [28] Grabiele Gratton, Michael G.H. Coles, and Emanuel Donchin. A new method for off-line removal of ocular artifact. *Electroencephalography and clinical Neurophysiology*, page 55(4):468–484, 1983.
- [29] Hugo Álvarez Chaves. Análisis de eeg para la evaluación de las emociones. Universidad de Murcia, September 2016.
- [30] Wolfman Klimesch. Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain research reviews*, page 29(2):169–195., 1999.
- [31] F Ramos-Argüelles, G Morales, S Egozcue, R.M Pabón, and M.T Alonso. Técnicas básicas de electroencefalografía: principios y aplicaciones clínicas. *Anales del Sistema Sanitario de Navarra*, pages 32 (Supl. 3): 69–82, 2009.

- [32] Yissel Rodríguez Aldana, Tahimy González Rubio, Enrique Marañón Reyes, Arquímedes Montoya Pedrón, and Frank Sanabria Marcías. Aplicación de la corrección de artefactos en el electroencefalograma para el monitoreo del estado anestésico. *Revista Cubana de Neurología y Neurocirugía*, pages 5(Supl.1): S9–S14, 2015.
- [33] Teresa Talamillo García. Manual básico para enfermeros en electroencefalografía. *Enfermería Docente*, pages 94: 29–33, 2011.
- [34] Giovanni Vecchiato, Patrizia Cherubino, Anton Giulio Maglione, Maria Trinidad Herrera Ezquierro, and Franco Marinoszzi. How to measure cerebral correlates of emotions in marketin relevant tasks. *Cognitive Computation*, pages 6(4):856–871, 2014.
- [35] Wolfman Klimesch, M Doppelmayr, H Russegger, T Pachinger, and J Schwaiger. Induced alpha band power changes in the human eeg and attention. *Neuroscience Letters*, pages 244(2):73–76, 1998.
- [36] Giovanni Vecchiato, J Toppy, L Astolfi, F Cincotti, F De Vico Fallani, A.G. Maglione, G Borghini, P Cherubino, D Mattia, and F Babiloni. The added value of the electrical neuroimaging for the evaluation of marketing stimuli. *Bulletin of the Polish Academy of Sciences: Technical Sciences.*, pages 60(3):419–426, 2012.
- [37] Fabio Babiloni, P Cherubino, I Graziani, A Trettel, G Bagordo, C Cundari, G Borghini, P Aricò, A Maglione, and G Vecchiato. The great beauty: a by neuroelectric imaging during the observation of the real michelangelo's moses sculpture. *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE., 2014.
- [38] F G Freeman, P J Mikulka, L J Prinzel, and M W Scerbo. Evaluation of an adaptive automation system usign three eeg indices with a visual tracking task. *Biological Psychology*, pages 50(1):61–76, 1999.
- [39] Alan T Pope, Edward H Bogart, and Debbie S Bartolome. Biocybernetic system evaluates indices of operator engagement in automated task. *Biological Psychology*, pages 40(1):187–195, 1995.
- [40] Michael Teplan. Fundamentals of eeg measurement. *Measurement Science Review*, pages 2(2):1–11, 2002.
- [41] Emilia Gómez Gutiérrez. Introducción al filtrado digital. Technical report, Escola Superior de Musica de Catalunya, November 2009.
- [42] RStudio (2017). Shiny: A web application framework for r, 2017. URL <https://shiny.rstudio.com>. (última visita 11/07/2017).
- [43] RStudio (2017). Shiny - reactivity: An overview, 2017. URL <https://shiny.rstudio.com/articles/reactivity-overview.html>. (última visita 08/08/2017).
- [44] RStudio (2017). Shiny - modularizing shiny app code, 2017. URL <https://shiny.rstudio.com/articles/modules.html>. (última visita 08/08/2017).