

Ainhoa García Linares

Desarrollo y Evaluación de una Interfaz Cerebro-Ordenador (BCI) para el Control de Videojuegos: Un Enfoque Práctico Utilizando EEG

TRABAJO DE FIN DE GRADO

Dirigido por el Dr. Jordi Duch-Gavaldà

Doble grado Biotecnología e Ingeniería Informática



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

RESUM

Aquest Treball de Fi de Grau explora el desenvolupament i l'avaluació d'una Interfície Cerebell-Ordinador (BCI) per controlar un prototip de joc utilitzant ones cerebrals captades per EEG. La investigació se centra en crear una BCI de baix cost utilitzant tecnologies com el joguet MindFlex modificat i una placa Arduino UNO, demostrant que és possible crear sistemes BCI amb recursos limitats. El sensor inclou un algoritme que indicarà els nivells d'atenció i meditació de l'usuari, així com els valors de les bandes de freqüència.

El treball realitza una cerca àmplia de les tecnologies BCI existents actualment i fa una selecció entre elles. El disseny del sistema BCI comença amb el muntatge del maquinari i la configuració del programari per adquirir i processar les senyals EEG. A continuació, es van realitzar proves amb diversos subjectes i anàlisi estadística dels resultats. Amb això vam poder concloure en la precisió dels valors d'atenció i meditació recollits pel sensor, en relació amb les activitats que els usuaris duïen a terme.

Finalment, es defineix la creació d'una interfície gràfica o videojoc que respon a les ones cerebrals i proporciona al usuari retroalimentació en temps real.

RESUMEN

Este Trabajo de Fin de Grado explora el desarrollo y la evaluación de una Interfaz Cerebro-Ordenador (BCI) para controlar un prototipo de videojuego utilizando ondas cerebrales captadas por EEG. La investigación se centra en crear una BCI de bajo costo utilizando tecnologías el juguete MindFlex modificado y una placa Arduino UNO, demostrando que es posible crear sistemas BCI con recursos limitados. El sensor cuenta con un algoritmo que nos indicará los niveles de atención y meditación del usuario, así como los valores de las bandas de frecuencia.

El trabajo hace una amplia búsqueda de las tecnologías BCI existentes actualmente y realiza una selección entre ellas. El diseño del sistema BCI comienza con el montaje del hardware y la configuración del software para adquirir y procesar las señales EEG. A continuación, se realizaron pruebas con diversos sujetos y análisis estadísticos de los resultados. Con esto pudimos concluir en la precisión de los valores de atención y meditación recogidos por el sensor, en relación a las actividades que los usuarios llevaban a cabo.

Finalmente, se define la creación de una interfaz gráfica o videojuego que responde a las ondas cerebrales y proporciona al usuario feedback en tiempo real.

ABSTRACT

This Bachelor's Thesis explores the development and evaluation of a Brain-Computer Interface (BCI) for controlling a game using brain waves captured by EEG. The research focuses on creating a low-cost BCI using modified MindFlex toy technology and an Arduino UNO board, demonstrating that it is possible to create BCI systems with limited resources. The sensor includes an algorithm that will indicate the user's levels of attention and meditation, as well as the values of the frequency bands.

The work conducts a comprehensive search of the current BCI technologies and selects among them. The design of the BCI system begins with the assembly of the hardware and the configuration of the software to acquire and process the EEG signals. Subsequently, tests were conducted with various subjects and statistical analysis of the results. From this, we were able to conclude the accuracy of the attention and meditation values collected by the sensor, in relation to the activities that the users were undertaking.

Finally, the creation of a graphical interface or video game that responds to brain waves and provides real-time feedback to the user is defined.

ÍNDICE

1. Introducción	1
1.1 Estructura del Documento	1
2. El cerebro Humano	2
2.1 Electricidad: el Lenguaje del Cerebro	2
2.2 Bandas de Frecuencia	2
2.2.1 Algoritmos Biométricos	4
2.2.2 Técnicas de Captación de Ondas	6
3. Interfaz Cerebro-Ordenador	8
3.1 Análisis de la Tecnología	8
3.1.1 Dispositivos EEG que Contemplamos Utilizar	8
3.1.2 NeuroSky	11
3.2 Tecnología empleada.....	13
3.2.1 MindFlex	13
3.2.2 Arduino UNO	14
4. Estudio de la Planificación, Objetivos y Requisitos	16
4.1 Objetivos del Proyecto	16
4.2 Requisitos	16
4.3 Análisis de Riesgos.....	18
4.4 Planificación	18
4.4.1 Descripción de las Tareas	19
4.4.2 Diagrama de Gantt.....	21
4.5 Herramientas.....	21
4.6 Recursos y Presupuesto	22
4.6.1 Identificación y Estimación de los Costes.....	22
4.6.2 Análisis General del Presupuesto	25
5. Diseño del Sistema BCI	26
5.1 Esquema General del Diseño.....	26
5.2 Diseño de Hardware: Montaje del Detector	27
5.2.1 Montaje del Detector	27
5.3 Diseño de Software: Adquisición y Procesado de Señal.....	29
5.3.1 Arduino Sketch.....	29
5.3.2 Python.....	31

5.3.3	Resumen de Clases	41
6.	Recogida y Análisis de Datos	40
6.1	Recogida de Datos	40
6.1.1	Consulta al Departamento de Protección de Datos	40
6.1.2	Datos a Capturar	40
6.1.3	Diseño de las Pruebas	41
6.1.4	Diseño de las Entrevistas	42
6.2	Análisis Exploratorio de los Datos	45
6.2.1	Valores de Partida y Objetivo.....	45
6.2.2	Identificación de Correlaciones y Validez de las Señales de Entrada.....	45
6.2.3	Exploración de la Variación de los Valores o Distribuciones de Ondas para Diferentes Experimentos	50
6.2.4	Determinación de Diferencias o Similitudes en los Patrones de Ondas para un Mismo Tipo de Onda entre Diferentes Usuarios	58
7.	Diseño de un Prototipo de Videojuego para Validar el Sistema	62
7.1	Valores de Atención y Meditación	62
7.2	Determinación de la Funcionalidad de la Interfaz Gráfica.....	62
7.3	Diseño de la Estética y la Funcionalidad.....	63
7.3.1	Creación de los Sprites	64
7.3.2	Pantallas de Juego.....	65
7.4	Implementación del Videojuego.....	66
7.4.1	Resumen de Clases	69
7.5	Pruebas de Funcionamiento.....	69
7.5.1	TestOvniGame.py.....	69
7.5.2	ReceiveDataTest.py	71
8.	Evaluación , Discusión y Conclusiones del Proyecto	74
	Referencias	77
	Anexos.....	I

Índice de tablas

Tabla 1. Tabla resumen de las distintas ondas cerebrales con su amplitud en uV, frecuencia y casos donde predomina.....	4
Tabla 2. Tabla comparativa con las distintas tecnologías contempladas para el proyecto.....	11
Tabla 3. Coste temporal de los recursos humanos durante la gestión del proyecto.	22
Tabla 4. Coste de los recursos software para la elaboración del proyecto.	22
Tabla 5. Coste del hardware personal para la elaboración del proyecto.	23
Tabla 6. Coste del hardware personal para la elaboración del proyecto calculado según la amortización lineal de los dispositivos.....	23
Tabla 7. Coste de los recursos materiales para la elaboración del proyecto.	24
Tabla 8. Coste de los recursos software para la elaboración del proyecto.	24
Tabla 9. Tabla resumen con el costo monetario total para la realización del proyecto.....	25
Tabla 10. Tabla de ejemplo con los valores de correlación para la actividad de Segundito.	39
Tabla 11. Tabla resumen con la ruta, fichero y clases empleadas para el diseño del software.	41
Tabla 12. Tabla de correlación de los pares de variables para la actividad de meditación.	46
Tabla 13. Tabla de correlación de los pares de variables para la actividad de memory1.....	47
Tabla 14. Tabla de correlación de los pares de variables para la actividad de memory2.....	47
<i>Tabla 15. Tabla de correlación de los pares de variables para la actividad UP.....</i>	<i>48</i>
Tabla 16. Tabla de correlación de los pares de variables para la actividad DOWN.	48
Tabla 17. Tabla de correlación de los pares de variables para la actividad de segundito.	49
Tabla 18. Tabla resumen de los pares de correlaciones más relevantes para las distintas pruebas realizadas.....	49
Tabla 19. Tabla con los valores estadísticos promedios para el valor de atención durante todas las actividades.....	51
Tabla 20. Tabla con los valores estadísticos promedios para el valor de meditación durante todas las actividades.	51
Tabla 21. Tabla con los valores estadísticos promedios para el valor de onda delta durante todas las actividades.....	52
Tabla 22. Tabla con los valores estadísticos promedios para el valor de onda theta durante todas las actividades.....	52

Tabla 23. Tabla con los valores estadísticos promedios para el valor de onda high alfa durante todas las actividades	52
Tabla 24. Tabla con los valores estadísticos promedios para el valor de onda low alfa durante todas las actividades.	53
Tabla 25. Tabla con los valores estadísticos promedios para el valor de onda high beta durante todas las actividades.....	53
Tabla 26. Tabla con los valores estadísticos promedios para el valor de onda low beta durante todas las actividades.	53
<i>Tabla 27. Tabla de resultados para las relaciones entre distintas pruebas, su valor KS calculado y el p-value.....</i>	<i>57</i>
Tabla 28. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad Meditation, en relación a cada una de las ondas capturadas	58
Tabla 29. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad memory1, en relación a cada una de las ondas capturadas.....	59
Tabla 30. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad memory2, en relación a cada una de las ondas capturadas.....	59
Tabla 31. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad segundito, en relación a cada una de las ondas capturadas.	59
Tabla 32. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad UP, en relación a cada una de las ondas capturadas.....	60
Tabla 33. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad DOWN, en relación a cada una de las ondas capturadas.	60
Tabla 34. Tabla resumen con la ruta, ficheros y clases empleadas para la implementación de la interfaz gráfica.	69

Índice de figuras

Figura 1. Elementos de una onda (Fuente: http://www.estoy-aprendiendo.com/ondas.html)	2
Figura 2. Frecuencias de las distintas ondas cerebrales. Fuente: (Deposit Photos 464619208).....	3
Figura 3. Las 5 principales bandas de frecuencia y su relación con las otras. Fuente: Raja, L., Santhosh, R. (2021).....	6
Figura 4. Dispositivo MindWave. Fuente: (Neurosky, Inc).....	10
Figura 5. Juguete Mindflex (izquierda) y Star Wars Trainer (derecha). Fuente: (Ebay)	11
Figura 6. Especificaciones técnicas del chip TGAT de NeuroSky. Fuente: (NeuroSky, Inc).....	12
Figura 7. Diadema EEG del juguete MindFlex con partes seleccionadas. Fuente: (B.Serna, 2020)	13
Figura 8. Unidad de procesamiento (ThinkGear Chip) del dispositivo MindFlex. Fuente: (NeuroSky, Inc).....	13
Figura 9. Microprocesador Arduino Uno Rev3. Fuente: (Arduino Inc.).....	15
Figura 10. Captura de pantalla con la gestión de tareas en Notion.	21
Figura 11. Esquema general del funcionamiento de un sistema BCI adaptado a las características del proyecto. Fuente: Elaboración propia.....	26
Figura 12. Consola de salida serial de los valores de ondas cerebrales capturadas en el microcontrolador Arduino. El primer valor corresponde al string de los valores, el segundo (en blanco) errores en la transmisión encontrados durante la comunicación.	27
Figura 13. Fotografía de la diadema MindFlex desmontada.	27
Figura 14. Fotografías del proceso de hackeo de la diadema MindFlex.	28
Figura 15. Esquema general del funcionamiento del hardware BCI. Fuente: Elaboración propia ..	28
Figura 16. Esquema de la adquisición y procesado de señal de la librería Brain. Fuente: (Brain Hack, 2014).....	29
Figura 17. Captura de pantalla del funcionamiento de la librería BrainGrapher en Processing-4 ...	31
Figura 18. Interfaz de selección del tipo de dato registrado.....	35
Figura 19. Interfaz para guardar o descartar los datos capturados de un usuario concreto.	36
Figura 20. Interfaz para la selección del usuario y creación de nuevos perfiles.	36
Figura 21. Gráfica de ejemplo con los valores de atención y meditación registrados para la categoría de meditación y el usuario MOR.	38
Figura 22. Heatmap y gráfica para los valores de correlación de los pares de variables durante la actividad de segundito.....	39

Figura 23. Video utilizado durante una de las pruebas de las entrevistas. Fuente: https://www.youtube.com/watch?v=0zsE85khG6I	42
Figura 24. Obsequio para los participantes de las entrevistas.	44
Figura 25. Histogramas de distribución para la banda de frecuencia high alfa para los 6 eventos capturados.	55
Figura 26. Histogramas de distribución para el valor de <i>attention</i> para los 6 eventos capturados.	56
Figura 27. Histogramas de distribución para el valor de <i>meditation</i> para los 6 eventos capturados.	56
Figura 28. Pantalla del juego FlappyBird. Fuente: (FlappyBird, Inc).....	63
Figura 29. Captura dos condiciones del código play.py.....	63
Figura 30. Dibujo del Sprite de un ovni en PixelArt. Fuente: Elaboración propia	64
Figura 31. Sprites hechos con PixelArt. Fuente: Elaboración propia.....	64
Figura 32. Captura de la pantalla inicial del juego.....	65
Figura 33. Captura de la pantalla de calibrado del juego.	65
Figura 34. Captura de la pantalla principal del juego.....	66
Figura 35. Captura de la pantalla final del juego.	66
Figura 36. Batería de imágenes para las entrevistas. Juguete mingdflex.	XII
Figura 37. Batería de imágenes para las entrevistas. Juguete mingdflex. hackeado	XIII
Figura 38. Batería de imágenes para las entrevistas. Ondas cerebrales.	XIV
Figura 39. Mapa de calor para tablas de correlación en la actividad Meditación	XV
Figura 40. Mapa de calor para tablas de correlación en la actividad Segundoito	XV
Figura 41. Mapa de calor para tablas de correlación en la actividad UP	XVI
Figura 42. Mapa de calor para tablas de correlación en la actividad Down.....	XVI
Figura 43. Mapa de calor para tablas de correlación en la actividad Memory2.....	XVII
Figura 44. Mapa de calor para tablas de correlación en la actividad Memory1	XVII

Acrónimos

BCI - Interfaz cerebro-ordenador

EEG - Electroencefalograma

TGAM - ThinkGear AM

SNC – Sistema nervioso central

1. INTRODUCCIÓN

En la actualidad, la interacción entre los seres humanos y las máquinas se ha vuelto cada vez más común en diversos campos de la tecnología y la medicina. Una de las áreas más estudiadas de esta interacción es el desarrollo de Interfaces Cerebro-Ordenador que permiten a los usuarios controlar dispositivos externos únicamente con la actividad cerebral. Este trabajo de fin de grado se centra en el diseño, implementación y evaluación de una BCI de bajo costo que utiliza la tecnología de electroencefalograma (EEG) para manipular un juego, ofreciendo así una visión sobre la investigación en neurociencias y la ingeniería informática.

La relevancia de este proyecto se encuentra en su capacidad para proporcionar conocimientos sobre la dinámica del cerebro humano mientras interactúa con entornos virtuales, además de explorar aplicaciones prácticas en el ámbito del entretenimiento. La interfaz desarrollada se basa en la utilización tecnologías accesibles, como el juguete MindFlex modificado y la placa Arduino UNO formando un sistema semi-duplex, demostrando que es posible realizar contribuciones al campo de las BCI con recursos limitados.

1.1 ESTRUCTURA DEL DOCUMENTO

La estructura del documento está organizada para permitir al lector entender un contexto previo sobre el desarrollo de las BCI. Comienza con la introducción del marco teórico sobre el funcionamiento del cerebro humano, haciendo énfasis en cómo las señales eléctricas permiten la comunicación neuronal y la relevancia de las diferentes bandas de frecuencia de las ondas cerebrales. A continuación, se analizan diversas tecnologías BCI, justificando la selección de dispositivos EEG basados en su accesibilidad, costo y eficacia para el proyecto. En los apartados posteriores, se describe la planificación y los requisitos técnicos necesarios para llevar a cabo la investigación, incluyendo un análisis de riesgos y los objetivos correspondientes.

En cuanto al diseño del sistema BCI, se explica tanto el hardware como para el software desarrollado, destacando el proceso de montaje y configuración para la adquisición y procesamiento de señales EEG.

La sección de recogida y análisis de datos detalla las metodologías empleadas para la experimentación con usuarios y el análisis exploratorio de los datos obtenidos. El diseño y la implementación de la interfaz gráfica se discuten en términos de su funcionalidad y estética, culminando en el desarrollo de un videojuego controlado por la BCI.

Finalmente, se evaluaremos el rendimiento del sistema y se discuten los resultados en comparación con las expectativas iniciales mediante el diseño de un prototipo de videojuego para validar el sistema.

2. EL CEREBRO HUMANO

2.1 ELECTRICIDAD: EL LENGUAJE DEL CEREBRO

Todas las funciones llevadas a cabo por nuestro cerebro y cuerpo están impulsadas por señales eléctricas. Estas señales eléctricas están formadas por átomos, que a su vez están formados por protones, electrones y neutrones. Todos tienen diferentes cargas eléctricas:

- Los protones tienen una carga eléctrica positiva
- Los electrones tienen una carga eléctrica negativa
- Los neutrones tienen una carga eléctrica neutra

En las células, la comunicación se basa en el desequilibrio de la carga eléctrica en un átomo. El átomo se carga positiva o negativamente y permite que los electrones fluyan de un átomo a otro para mantener el equilibrio. Este flujo de electrones es lo que todos llamamos electricidad. Con alrededor de siete mil millones de billones de átomos en cada cuerpo humano, tienen la capacidad de generar cantidades significativas y medibles de electricidad.

Estas señales eléctricas (o "impulsos") de átomo a átomo son las que permiten que las células humanas se comuniquen entre sí. Nuestro sistema nervioso conduce estos impulsos, permitiéndoles saltar de célula en célula hasta llegar a su destino. Hay diferentes velocidades de transmisión de impulsos: los impulsos más rápidos pueden viajar a velocidades de aproximadamente 250 millas por hora. Esta es la razón por la que (por ejemplo) si te quemas la mano, tu cerebro puede eliminarla rápidamente de la fuente de calor antes de que el dolor se registre por completo [11].

2.2 BANDAS DE FRECUENCIA

Las neuronas se comunican entre ellas a través de estos **pequeños impulsos eléctricos** que se pueden medir, a esto le llamamos ondas cerebrales. Estas ondas tienen diferentes tipos de frecuencia, de modo que, si se separan a través de filtros, las podemos distinguir con más claridad. Para ellos deberemos realizar un EEG ¹ registrando la actividad eléctrica cerebral mediante distintas técnicas o tecnologías que se explicarán más adelante.

La frecuencia de las ondas cerebrales se mide en ciclos por segundo o Hertz (Hz). Un ejemplo práctico puede ser la corriente eléctrica en Europa, con una frecuencia de 50 Hz, lo que quiere decir que cada segundo tiene 50 ciclos.

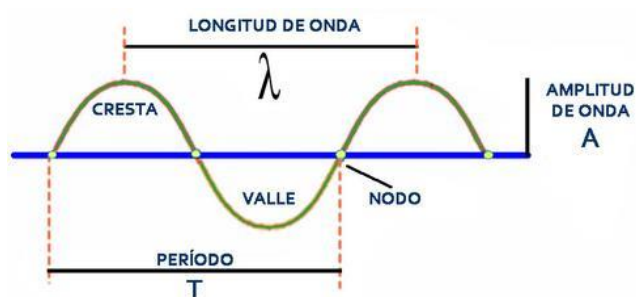


Figura 1. Elementos de una onda (Fuente: <http://www.estoy-aprendiendo.com/ondas.html>)

¹ Electroencefalograma. Estudio que mide la actividad eléctrica en el cerebro mediante pequeños electrodos colocados sobre el cuero cabelludo.

Las ondas cerebrales son un reflejo directo de la función del Sistema Nervioso Central (SNC). Nuestras ondas cerebrales son cambiantes y van a variar según lo que estamos haciendo y sintiendo. Cuando predominan las ondas cerebrales más lentas, nos sentiremos más cansados, lentos o perezosos. Por otro lado, cuando son las ondas con frecuencias más altas las que dominan, nos sentiremos más alertas o incluso ansiosos [1, 2].

Podemos asociar los diferentes tipos de ondas a diferentes estados de activación, de ánimo o de concentración. Cuando la actividad de cierto tipo de onda se encuentra alterada en alguna zona de nuestro cerebro es cuando producirá un síntoma específico [1, 2].

Dado que las ondas cerebrales son un reflejo de la actividad y función del sistema nervioso central, cuando éste sufre un desequilibrio se puede observar una alteración en la actividad de las ondas cerebrales.

Las ondas cerebrales son ondas de **baja amplitud**, en el caso del cerebro humano estamos hablando de microvoltios, y de diferentes tipos de frecuencia, siendo unas más rápidas y otras más lentas. Así pues, las ondas cerebrales se clasifican en diferentes tipos según su frecuencia. Las ondas registradas en el EEG, o mapa cerebral, se pueden separar en grupos para su estudio [1, 2]. Estas se dividirán según su frecuencia en:

- Ondas Delta: con una frecuencia de 0.2 – 3.5 Hz
- Ondas Theta: 4.0 – 8.0 Hz
- Ondas Alfa: 8.0 – 12.0 Hz
- Ondas Beta: 12.0 – 30.0 Hz
- Ondas Gamma: 30.0 – 90.0 Hz

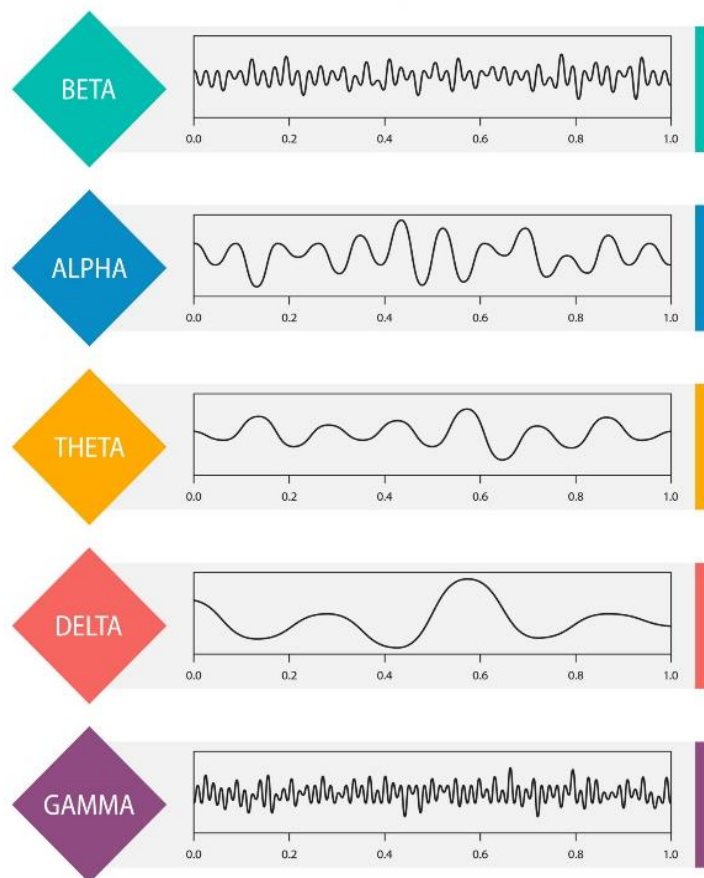


Figura 2. Frecuencias de las distintas ondas cerebrales. Fuente: (Deposit Photos 464619208)

2.2.1 Algoritmos Biométricos

Brain Wave	Amplitude (μV)	Frequency Spectrum (Hz)	Predominance
Delta	20 a 200 μV	0.2 – 3.5	<ul style="list-style-type: none"> · Actividades cognitivas · Estados inconscientes · Sueño profundo
Theta	< 30 μV	4.0 – 8.0	<ul style="list-style-type: none"> · Fase REM del sueño · Niños entre 1 y 6 años · Niveles altos de estrés
Alpha	20 a 60 μV	8.0 – 12.0	<ul style="list-style-type: none"> · Estados de relajación no de somnolencia · Estado de reposo
Beta	< 20 μV	12.0 – 30.0	<ul style="list-style-type: none"> · Actividad mental · Estado consciente
Gamma	< 10 μV	30.0 – 90.0	<ul style="list-style-type: none"> · Muy intensa actividad mental p.e durante los cálculos matemáticos

Tabla 1. Tabla resumen de las distintas ondas cerebrales con su amplitud en μV , frecuencia y casos donde predomina.

ATENCIÓN

El algoritmo de atención indica el rigor del "enfoco" o "atención" mental. El valor puede oscilar entre 0 y 100. El nivel de atención aumenta cuando un usuario **se enfoca en un solo pensamiento** o en un objeto externo, y **disminuye cuando se distrae** [8].

Los usuarios pueden observar su capacidad de concentración utilizando este tipo de algoritmos. En los juegos, la atención se utiliza para crear un control de "empuje" sobre los objetos virtuales.

MEDITACIÓN

El algoritmo del medidor de meditación puede indicarnos el nivel de "calma" o "relajación" mental. El valor oscila entre 0 y 100, y **aumenta cuando los usuarios relajan** la mente y **disminuye cuando están incómodos o estresados** [8].

Este valor cuantifica la capacidad de encontrar un estado interno de atención plena y, por lo tanto, puede ayudar a los usuarios a aprender a autocorregirse y encontrar un equilibrio interior para superar el estrés de la vida cotidiana. El algoritmo también se utiliza en una variedad de controles de diseño de juegos.

ONDAS DELTA

Son las ondas (o frecuencias) también conocidas como la onda del sueño. Son ondas muy lentas, pero también las que tienen mayor **amplitud**. Esta varía de los 20 a 200 μV , y su rango de frecuencia oscila entre 0,2 y 3,5 Hz [8].

También se observan en estados meditativos. En casos de lesiones cerebrales, problemas de aprendizaje o TDAH severo, las ondas delta pueden observarse de manera prominente.

ONDAS THETA

Las ondas theta predominan cuando los sentidos están procesando información interna y el individuo está desconectado del mundo exterior. Se presenta generalmente con la **activación del lóbulo temporal**. Su banda de frecuencia está entre los 4 y 8 Hz y su amplitud suele encontrarse en torno a los 30 μ V [8].

Este ritmo se suele presentar frecuentemente en niños o en adultos que se encuentran sometidos a altos niveles de estrés . Además, ocurren durante la transición entre la vigilia y el sueño, y durante la meditación profunda. Son las ondas que utilizamos durante la fase R.E.M. del sueño y están estrechamente relacionadas con el aprendizaje, la memoria y la intuición [6].

Los pacientes con problemas de atención a menudo tienen un exceso de ondas Theta. Este exceso puede controlarse y aprender a regularse con Neurofeedback [6].

ONDAS ALPHA

Alfa predomina cuando el SNC está en reposo, relajado pero despierto y atento. Las señales asociadas a este ritmo presentan una frecuencia comprendida entre los 8 y 12 Hz y una amplitud de entre 20 y 60 μ V. **La fuente principal de estas señales se encuentra en la región occipital del cerebro** [5, 6, 8].

Si hay deficiencia de alfa, el individuo tiene **dificultad para relajarse**. Diremos que el cerebro está inactivo, relajado, en reposo, pero al mismo tiempo listo para la acción si fuera necesario. Este ritmo se atenúa cuando la persona inicia alguna clase de actividad mental que requiera concentración [5, 6].

ONDAS BETA

Beta predomina durante el período de vigilia. Son las ondas con la velocidad de transmisión más alta. Su frecuencia oscila entre los 12 y los 35 Hz, y su amplitud en el rango entre los 2 y 20 μ V. Esta denota una intensa actividad mental, y está vinculada principalmente con los movimientos de las extremidades y actividades musculares. También puede asociarse a un estado mental de concentración por parte de la persona o durante la toma de decisiones. **Se pueden encontrar principalmente en la región frontal y central del cerebro** [8].

Aparece en estados en los que la atención se dirige a tareas cognitivas externas, **a diferencia de las ondas Theta**, que aparecen durante los estados cognitivos internos.

La versión beta se puede subdividir en:

- Beta 1 (12-15 Hz),
 - También se llama ritmo sensoriomotor. Estas ondas se producen cuando el cuerpo está quieto pero la mente está enfocada. Este estado **facilita la toma de medidas cuando sea necesario**. El aumento de este tipo de ondas puede producir una mejora en la atención y relajación.
- Beta 2 (15-22 Hz):
 - Cuando el SNC está ocupado en una tarea. Se asocia con pensar, ser consciente de uno mismo y del entorno. Estar alerta y activo, pero no agitado
- Beta 3 (22-30 Hz):

- Cuando el SNC se centra en cogniciones de alta complejidad, o en la integración de nuevas experiencias
- Puede significar un estado de ansiedad o excitación como sería el caso de las personas que sufren de Trastorno de Ansiedad Generalizada

Con el neurofeedback, se puede regular y disminuir la proporción de ondas Beta y así lograr la mejora de la ansiedad.

ONDAS GAMMA

Son las ondas más rápidas, ocurren en ráfagas cortas y se relacionan con el procesamiento de información simultánea en varias áreas del SNC. Son las frecuencias superiores a 30 Hz, y la amplitud más baja, con valores inferiores a los 10 μ V [8].

Las ráfagas de ondas gamma se observan cuando el cerebro está en un **estado de alta resolución**. (Ejemplo durante el cálculo de una fórmula matemática). Cuando nos encontramos en un momento de extrema atención y concentración o en momentos de alto estrés.

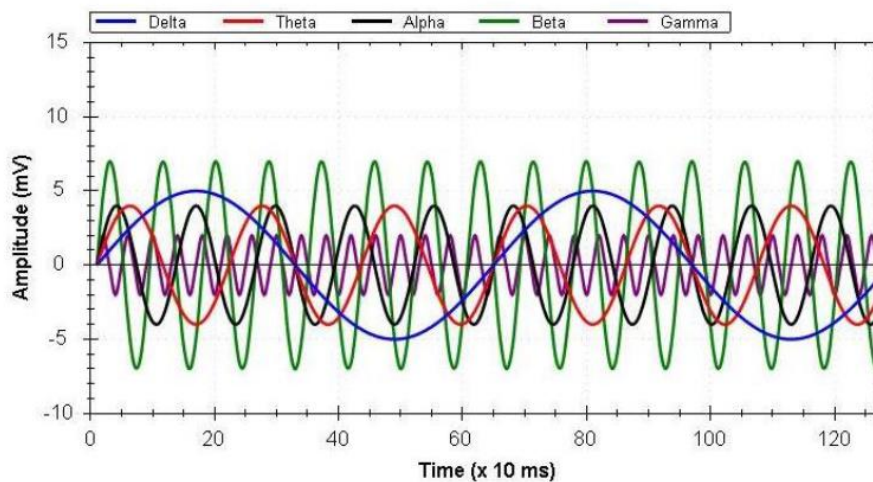


Figura 3. Las 5 principales bandas de frecuencia y su relación con las otras. Fuente: Raja, L., Santhosh, R. (2021)

2.2.2 Técnicas de Captación de Ondas

Las ondas cerebrales se pueden captar, distinguir y trabajar con diferentes métodos distinguidos entre invasivos o no invasivos. Dado que nuestro trabajo se basará en la utilización de estas ondas con distintos fines, contemplamos los distintos tipos [3, 8].

2.2.2.1 Invasivos

En los métodos invasivos, los sensores son implementados mediante neurocirugías en la materia gris del cerebro. Debido a que están directamente conectados al cerebro, producen señales más claras y de mejor calidad, con una menor sensibilidad al ruido.

Los métodos invasivos no son considerados apropiados para una BCI, ya que conlleva numerosos riesgos para la salud del paciente. Además, solo pueden ser realizados por un personal médico altamente cualificado.

2.2.2.2 No Invasivos

Los métodos no invasivos se caracterizan principalmente por **no suponer un riesgo para la salud del usuario**. Son métodos sencillos de colocar y retirar y no requieren cirugía.

No obstante, esto resulta en una resolución de más baja calidad, y un aumento de los factores de ruido. Además, estos sensores no son capaces de detectar señales de muy alta frecuencia. A continuación, se exponen algunas de las técnicas no invasivas más comunes.

- Magnetoencefalografía (MEG).
- Imágenes de resonancia magnética funcional (fMRI)
- Espectroscopia de infrarrojo cercano (NIRS).
- Electroencefalograma (EEG)

3. INTERFAZ CEREBRO-ORDENADOR

Una interfaz cerebro-computadora (BCI) es un sistema cuya tecnología facilita la comunicación directa entre el cerebro y un dispositivo externo. Esta interfaz permite el análisis de los impulsos cerebrales y su conversión en instrucciones que pueden manipular dispositivos externos, como computadoras, prótesis u otros sistemas eléctricos [2].

Actualmente los sistemas BCI cuentan varias aplicaciones, como la rehabilitación médica, tecnología de asistencia para personas con discapacidades, juegos y entretenimiento, e investigación sobre enfermedades y funciones cerebrales. El objetivo principal de la tecnología BCI es mejorar el nivel de vida facilitando nuevos modos de contacto y comunicación.

En este proyecto, nos proponemos definir distintos tipos de sistemas BCI de bajo coste y evaluar su precisión. Este proyecto estará diseñado para registrar las ondas cerebrales, realizar un análisis estadístico de las mismas, y utilizar los datos obtenidos para crear una interfaz gráfica controlada por valores o bandas de frecuencia específicos.

3.1 ANÁLISIS DE LA TECNOLOGÍA

Debido a su accesibilidad y capacidad de capturar la actividad cerebral de manera no invasiva y con alta resolución temporal, la tecnología EEG fue escogida para el desarrollo de este proyecto.

Es técnica consiste en colocar electrodos secos en el cuero cabelludo del sujeto para medir los cambios en el potencial eléctrico generado por la actividad neuronal del cerebro. Además, nos brinda la capacidad de registrar los patrones de actividad cerebral en tiempo real.

3.1.1 *Dispositivos EEG que Contemplamos Utilizar*

Una vez determinado el tipo de tecnología que utilizamos, hicimos una amplia búsqueda de los sistemas o tecnologías existentes que cumpliesen con unos requisitos básicos:

1. Utilice una tecnología EEG sencilla y con unas garantías de precisión.
2. Exista documentación o experiencias previas en su uso que poder utilizar como punto de referencia.
3. Tuviese un precio asequible para el desarrollo del TFG.

3.1.1.1 *Open EEG*

OpenEEG [12] nace como un proyecto abierto para aquellos interesados en la neuroretroalimentación o entrenamiento de biorretroalimentación de EEG. El propósito de OpenEEG es entrenar un aprendizaje consciente de la actividad generada por el cerebro. Este proyecto trata de producir planos y softwares para dispositivos EEG hechos artesanalmente de manera gratuita entre los aficionados. Ofrece una amplia gama de propuestas de software y hardware para la creación de sistemas EEG completamente nuevos.

Respecto al equipo principal, se compone de una placa base con un módulo de señales digitales, y un segundo módulo de señales analógicas. El módulo de señales analógicas configura los electrodos que se conectarían al usuario de ensayo, y que se habrían configurado previamente.

La lógica se enfoca en registrar las señales eléctricas emitidas por el usuario a través de electrodos (material médico), amplificarlas y convertirlas a señales digitales. La información se transmite al módulo digital después de haber sido digitalizada. Una vez filtrada la onda, el siguiente paso sería recopilar sus características y convertirlas en datos que puedan ser representados o utilizados.

Estos últimos datos, serían aquellos enviados al computador, a través de los cuales se basaría el resto del proyecto.

La construcción del dispositivo consideramos que sería uno de los aspectos más complejos del proceso, por lo que se descartó esta opción. Para crear los módulos, se requería un alto conocimiento de electrónica y una elevada cantidad de tiempo únicamente para la construcción. Es importante señalar que, además de la complejidad del proceso de montaje, los materiales médicos y de hardware son muy caros, lo que elevarían el costo del proyecto.

A pesar de las dificultades encontradas, la información proporcionada por la página web sobre la electrónica y los métodos por los cuales varios usuarios han creado software, así como la teoría del filtrado de las ondas y las bases fundamentales de cómo se lleva a cabo el proceso, ofrece un buen punto de partida y una base de información para apoyarse.

3.1.1.2 Mind set

NeuroSky es una empresa dedicada al desarrollo de tecnologías de sensores capaces de leer distintos tipos de bioseñales emitidas por el ser humano. En nuestro caso, el punto de interés recae sobre las señales EEG, para las cuales la empresa tiene un dispositivo propio [13].

NeuroSky MindSet es un dispositivo sencillo con el que llevar a cabo este proyecto: permite la posibilidad de conexiones inalámbricas y es compatible con distintos dispositivos. Durante la fabricación del producto, fue el propio desarrollador quien se encargó de controlar la compatibilidades entre dispositivos. Además, incorpora herramientas gratuitas para desarrolladores de la propia compañía.

El dispositivo es capaz de registrar las ondas cerebrales y determinar distintos tipos de estados de ánimo con un único sensor que se sitúa en la frente. Estos datos son directamente analizados por el dispositivo y son compartidos con el usuario, dando así información sobre la atención, estados de meditación, parpadeos, distintos tipos de frecuencias de onda (alfa, delta, theta, beta, etc.) y resto de señales sin procesar que también puede ser de interés para el usuario.

Además, la frecuencia y precisión de las ondas no es comparable con ningún otro dispositivo de los propuestos para el proyecto.

En este caso, el principal inconveniente es su elevado precio, ya que ronda alrededor de 300€. Esto nos llevó a descartarlo como opción.



Figura 4. Dispositivo MindWave. Fuente: (Neurosky, Inc)

3.1.1.3 *MindFlex* y *Star Wars Force Trainer*

MindFlex y *Star Wars Force Trainer* se tratan de dos juguetes desarrollados por la empresa NeuroSky, que incorporan el chip EEG diseñado para sus dispositivos de mayor coste [14, 15].

Las dos opciones ofrecieron una manera más económica de obtener los mismos datos utilizando una tecnología similar. Aunque actualmente están descatalogados, fueron muy comercializados alrededor del 2014, por lo tanto, se pueden encontrar fácilmente por alrededor de 15-20 € en tiendas de segunda mano o por internet en Wallapop, Vinted o Ebay.

Según la documentación del fabricante de cada uno de los dispositivos, el chip utilizado en las tres opciones (*MindSet*, *MindFlex* y *ForceTrainer*) es el mismo, pero muchas pruebas y proyectos encontrados en bibliografía en internet, muestran que cada uno ejecuta un firmware ligeramente diferente, lo que explica algunas variaciones en la salida de datos. Estos factores nos llevaron a decantarnos definitivamente por uno de ellos.

En el caso de *ForceTrainer*, no es capaz de registrar distintos valores en la banda de frecuencias del EEG (gamma, beta, alpha.), únicamente nos permite detectar una presencia o ausencia de señal en el caso de las ondas de 8-13Hz (alfa). Sin embargo, *MindFlex* si permite registrar estos valores de onda.

MindSet, a diferencia de los juguetes, además de las bandas EEG, también les brinda acceso a datos de ondas sin procesar, es decir, a los distintos tipos de variaciones que no siguen los esquemas o cánones establecidos por las bandas de frecuencia. En este caso, si tenemos en cuenta que el objetivo del TFG busca la recopilación y comprensión de ondas cerebrales, no consideramos esta una condición fundamental para la determinación del dispositivo.

Teniendo en cuenta todo esto, creemos que *MindFlex* representa la mejor opción en términos de precio y calidad. Este juguete se acerca en gran medida a los datos proporcionados por el dispositivo de alto coste de NeuroSky; *MindSet*, por un precio más razonable.



Figura 5. Juguete Mindflex (izquierda) y Star Wars Trainer (derecha). Fuente: (Ebay)

	Open EEG	Force trainer	MindFlex	Mind set
Descripción	Proyecto abierto para construir tu propio EEG	Juguete para levitar una bola con la mente	Juguete para levitar una bola con la mente	Casco oficial de NeuroSky
Valores de atención/meditación	No	Si	Si	Si
Bandas de frecuencias EEG	En función de la implementación	No	Si	Si
Ondas sin procesar / ruido	Si	No	No	Si
Precio	Depende de la implementación +100€	15-20€	15-20€	300€

Tabla 2. Tabla comparativa con las distintas tecnologías contempladas para el proyecto.

3.1.2 NeuroSky

La plataforma NeuroSky ofrece una base sólida para la creación de aplicaciones BCI, así como para estudios relacionados. Como se mencionaba previamente, esta empresa ofrece una variedad de productos con tecnología EEG que promueven la mejora de la agudeza mental, la memoria de trabajo, la concentración y el enfoque [14].

El módulo de PCB del sensor EEG, ThinkGear ASIC Module (TGAM), es uno de los dos tipos de productos de EEG que NeuroSky ofrece a los diseñadores de sistemas integrados [16].

3.1.2.1 ThinkGear AM

El módulo de PCB con sensor de EEG ThinkGear AM (TGAM) es el dispositivo por excelencia utilizado en más de 1 millón de módulos EEG incluyendo el dispositivo MindFlex de interés, según publica la propia empresa NeuroSky en su página web.

TGAM, el núcleo de la tecnología de detección de ondas cerebrales, permite a los usuarios de NeuroSky llevar al mercado tecnologías de consumo basadas en EEG de forma rápida y eficiente. Este chip junto con el electrodo seco, detecta las señales del cerebro humano, filtran el ruido e interferencias eléctricas y las convierten en señales digitales [13, 16].

DETALLES TÉCNICOS

Integrado en el TGAM, se encuentra el chip TGAT, un sensor de EEG de un solo chip totalmente integrado. El chip viene programado con NeuroSky eSense; software desarrollado por la propia empresa, A/D, detección y filtrado de ruido para EMG.

El dispositivo proporciona una interfaz de comunicación serie para transmitir datos de EEG a otros dispositivos o sistemas, en este caso UART (Universal Asynchronous Receiver-Transmitter).

Utilizaremos la transmisión UART para la transmisión de datos. El propio chip permite la transmisión a diferentes velocidades de baudios o bits por segundo (1200, 9600 y 57600), lo que nos permite configurar la velocidad de comunicación en función según las características de la transferencia de datos o estabilidad del sistema.

Además, la transmisión se hace en conjuntos de 8 bits de datos, sin bit de paridad para la detección de errores. Después de transmitir cada byte de datos, se agrega un bit de parada para indicar el final del byte. Destacar que estos parámetros son comunes en las implementaciones UART, y los escogidos por los desarrolladores para el chip TGAM de NeuroSky.

Destacar que la propia empresa resalta la extrema detección y caracterización de las señales EEG de bajo nivel. El chip cuenta además con algoritmos integrados para el procesamiento de las señales EEG que nos permite obtener información de salida de los valores de atención y meditación, así como las frecuencias de las ondas delta, theta, low alpha, high alpha, low beta, high beta, y ondas gamma [13, 16].

TGAT1/TGAM1

Features

- Direct connect to dry electrode
- One EEG channel + Reference + Ground
- Extremely low-level signal detection
- Advanced filter with high noise immunity
- RAW EEG at 512Hz

Data Outputs

- RAW EEG Signal
- Attention
- Meditation
- Delta, Theta, low alpha, high alpha, low beta, high beta and gamma waves
- Additional [algorithms available in SDK](#)

Dimensions

- TGAM1 Module: 27.9 x 15.2 x 2.5mm
- TGAT1 Chip: 9 x 9 x 1.6mm
- Weight (Max) 130mg

Specifications

- 512Hz sampling rate at 12 bits
- 3-100Hz frequency range
- ESD Protection: 4kV Contact Discharge; 8kV Air
- Max Power Consumption: 15mA @ 3.3V
- Operating voltage 2.97 ~3.63V
- UART (Serial):
 - 1200, 9600, 57600 baud
 - 8-bits
 - No parity
 - 1 stop bit

Figura 6. Especificaciones técnicas del chip TGAT de NeuroSky. Fuente: (NeuroSky, Inc)

3.2 TECNOLOGÍA EMPLEADA

3.2.1 MindFlex

Como concluimos en el apartado anterior, para el desarrollo de este TFG y la creación de nuestro sistema BCI, utilizamos la diadema EEG del dispositivo MindFlex; pieza clave que nos proporcionará los valores de las ondas cerebrales a tiempo real.

Este aparato es de un dispositivo de juguete que utiliza la tecnología BCI para permitir a los usuarios controlar objetos con su actividad cerebral. Fue desarrollado por la compañía Mattel utilizando la tecnología NeuroSky como una línea de juguetes [14].

Este dispositivo consta de tres componentes principales:

1. DIADEMA EEG

Esta diadema está formada por una banda de goma que se coloca en la cabeza del usuario. En los costados podemos encontrar la batería del dispositivo en el lado derecho, y el TGAM en el izquierdo.

Además, cuenta con 3 sensores que registran la actividad eléctrica del cerebro a través de la piel del cuero cabelludo: uno para el canal EEG (electrodo seco), que se coloca en la parte frontal de la cabeza, y dos en los apartados lobulares que se utilizan como toma de tierra.



Figura 7. Diadema EEG del juguete MindFlex con partes seleccionadas. Fuente: (B.Serna, 2020)

2. UNIDAD DE PROCESAMIENTO O THINKGEAR CHIP

La unidad TGAM recibe las señales cerebrales registradas por la diadema EEG y las procesa para interpretar los estados mentales del usuario. Este dispositivo nos proporciona la información necesaria para el desarrollo del proyecto.

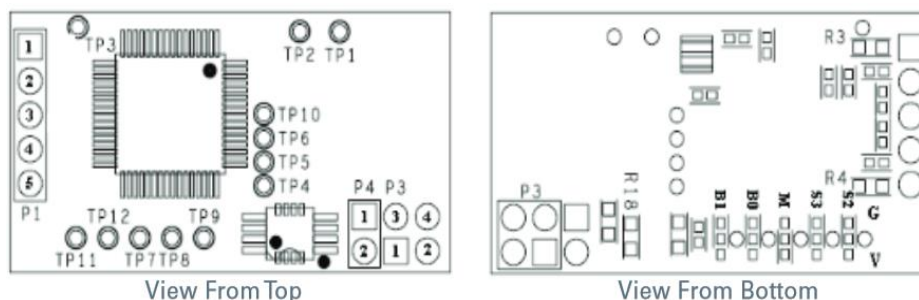


Figura 8. Unidad de procesamiento (ThinkGear Chip) del dispositivo MindFlex. Fuente: (NeuroSky, Inc)

3. DISPOSITIVO DE CONTROL

El dispositivo de control originalmente se trataba de una plataforma que permitía al usuario manipular un ventilador que utilizaba su actividad cerebral para hacer volar una pelota. El usuario coloca la diadema EEG y los sensores, y realiza una serie de acciones que se reflejarán en la unidad de procesamiento.

El chip TGAM será responsable de la toma de decisiones y comunica a la plataforma de control la potencia del ventilador. El funcionamiento es sencillo: cuanto más concentrado esté el usuario, más alta volará la bola, y cuanto más relajado esté, más baja estará la bola, lo que significa que tendremos más o menos potencia del ventilador.

En este caso, obviaremos el dispositivo de control inicial del juguete y emplearemos únicamente la diadema con el chip integrado.

3.2.2 Arduino UNO

La placa electrónica de hardware libre llamada Arduino incorpora un microcontrolador reprogramable con una serie de pines que permiten conectar el controlador a varios sensores, que son el "cerebro" de un circuito o maquinaria.

En este proyecto, utilizaremos el microcontrolador de Arduino como receptor y procesador de los datos del chip EEG en un **sistema semidúplex**. La placa será la encargada de procesar los paquetes enviados por la diadema MindFlex, procesar las distintas señales, y enviará de manera periódica los datos filtrados a nuestro ordenador mediante un puerto de comunicación serial, recibiendo de manera directa los datos limpios para trabajar.

Escogimos el modelo UNO de Arduino por ser una de las opciones más populares para proyectos participantes del estilo. Al igual que los requisitos que nos llevaron a decantarnos por la banda MindFlex, se trata de un dispositivo de uso sencillo, amplia documentación disponible en línea y destaca por ser uno de los productos más económicos de la marca [8].

DETALLES TÉCNICOS

Arduino UNO integra un microcontrolador ATmega328P de Atmel con una velocidad de reloj de 16 MHz y 32 KB de memoria flash. Cuenta además con 2KB de SRAM y 1 KB de EEPROM.

En nuestro caso, se alimentará a través de conexiones USB que nos proporciona una entrada de 5V. De igual manera, proporciona una salida de 5V para el caso de estas aplicaciones.

Cuenta con 14 pines digitales de E/S y 6 de entrada analógica. Para el proyecto en concreto, utilizaremos únicamente uno de los pines de entrada digital para la recepción de datos procedentes del MindFlex (Serial Port RXD / Digital Pin 0).

Además, cuenta con una serie de pines de alimentación y tierra (GRD), así como pines para la comunicación serial (UART), la comunicación I2C y la comunicación SPI. Ninguno de estos pines a excepción del pin de tierra serán necesarios para nuestra implementación en el proyecto [18].

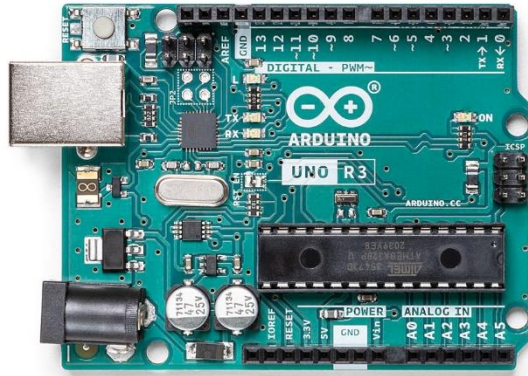


Figura 9. Microprocesador Arduino Uno Rev3. Fuente: (Arduino Inc.)

4. ESTUDIO DE LA PLANIFICACIÓN, OBJETIVOS Y REQUISITOS

El apartado actual se centra en la exposición de la planificación y los objetivos establecidos para el desarrollo del TFG. Consideramos que es fundamental explicar como se estructuró el plan de trabajo durante los meses que duró el proyecto, así como la coherencia y relevancia de los objetivos establecidos.

Por esta razón, se llevó a cabo una reunión con el director del TFG al comenzar el proyecto para debatir y determinar la planificación, así como los objetivos a corto y largo plazo. Para ello, se dividió el trabajo en distintas fases, estableciendo unos objetivos y tareas para cada una de ellas.

Desde el comienzo, planteamos cumplir unos objetivos realistas, que pudieran ser revisados a medida que íbamos avanzando.

4.1 OBJETIVOS DEL PROYECTO

El objetivo principal de este proyecto es desarrollar un sistema de interfaz cerebro-computadora (BCI) de bajo coste que permita registrar, analizar y utilizar las ondas cerebrales para controlar una interfaz gráfica. En este punto, se recogen los resultados que pretendimos alcanzar al comienzo del proyecto, así como lo que considerábamos las bases para realizar un buen trabajo. Para lograr esto, se han definido los siguientes objetivos específicos:

1. Estudio previo sobre la lógica del cerebro y las posibilidades del proyecto.
 - a. Conocer cómo las ondas cerebrales, las frecuencias y las conexiones neuronales se relacionan con el comportamiento humano.
 - b. Investigar también las técnicas de creación de sistemas BCI y las oportunidades de proyecto.
2. Utilizar algún dispositivo hardware, y desarrollar a un software propio para recoger ondas cerebrales.
3. Realizar ensayos con distintos usuarios para probar la captación de ondas cerebrales.
4. Estudiar la precisión y la correlación de los datos para determinar la funcionalidad que queremos darle a la recogida de ondas.
5. Determinar una aplicación o interfaz que utilice los valores de las ondas y nos permita visualizar feedback de las ondas a tiempo real.

4.2 REQUISITOS

A continuación, se detallan los requisitos necesarios para cumplir con los objetivos de trabajo, así como la explicación de que objetivos cumplen:

- **Estudio previo:**
 - Recopilar información detallada sobre las diferentes frecuencias de ondas cerebrales y su correlación con diferentes estados mentales y comportamientos humanos.
 - Definir las metodologías y técnicas existentes en el campo de las interfaces cerebro-computadora (BCI) y comprender cómo se han implementado en proyectos similares.

- **Requisitos de hardware y software:**

- Seleccionar un dispositivo hardware compatible con la captación de ondas cerebrales, que cumpla con los estándares de calidad y precisión requeridos.
 1. El dispositivo tiene que ser compatible con el software a desarrollar y permitir una fácil comunicación. Debe contar con una interfaz sencilla de comunicación.
 2. Debe permitir incluir mecanismos de calibración que permitan ajustar los parámetros de captación de acuerdo con las características individuales de cada usuario.
 3. El dispositivo ha de ser económico y fácilmente conseguible.
- Desarrollar un software personalizado capaz de interactuar con el dispositivo hardware seleccionado, recoger y procesar los datos de ondas cerebrales de manera eficiente y precisa.
- Desarrollar una interfaz sencilla y con retroalimentación clara sobre las acciones realizadas.
- Garantizar una recogida de datos sin pérdidas y determinar un formato accesible de datos.
 1. Incluir algoritmos de procesamiento de señales para filtrar el ruido.
 2. Incluir algoritmos de extracción de las características relevantes de los datos.

- **Ensayos con usuarios:**

- Diseñar protocolos de ensayo claros y estructurados que permitan probar la efectividad del dispositivo y el software en la captación de ondas cerebrales.
- Seleccionar una muestra representativa de usuarios para participar en los ensayos.
 1. Obtener el consentimiento informado de todos los participantes antes de realizar los ensayos.
- Obtener la aprobación del departamento de Protección de Datos para la realización de las entrevistas con los usuarios.

- **Análisis de datos:**

- Establecer criterios claros para evaluar la precisión y correlación de los datos recopilados durante los ensayos con usuarios.
- Utilizar técnicas de análisis estadístico para determinar la relación entre los datos de ondas cerebrales y el comportamiento observado, y extraer conclusiones significativas sobre la funcionalidad del sistema.
 1. Identificar patrones, tendencias o relaciones significativas entre las variables.

2. Analizar la correlación entre los datos de las señales de ondas cerebrales y el comportamiento observado de los participantes durante los ensayos.
- Interpretar los resultados del análisis de datos y extraer conclusiones significativas sobre la efectividad del sistema de captación de ondas cerebrales.
 - **Desarrollo de la interfaz**
 - Definir el propósito y la funcionalidad de la aplicación o interfaz, considerando los datos y conclusiones obtenidas del análisis de ondas cerebrales.
 1. Determinar cómo los usuarios interactuarán con la aplicación utilizando las ondas cerebrales.
 2. Implementar mecanismos para proporcionar **feedback en tiempo real** al usuario sobre la interacción basada en las ondas cerebrales.

4.3 ANÁLISIS DE RIESGOS

El análisis de riesgos es parte fundamental de la planificación, pues nos permite anticipar posibles obstáculos y dificultades. Cuando comenzamos con la idea de crear un sistema BCI, asumimos los riesgos que esto podría conllevar.

Uno de los principales problemas era el no encontrar un dispositivo o técnicas eficientes para recopilar ondas cerebrales y que se ajustara a las necesidades del proyecto. Los requisitos era que el dispositivo BCI tuviera una tecnología sencilla, con alguna documentación o experiencias previas en su uso y con un precio razonable.

Otro riesgo podía ser la falta de valores precisos durante la captura o que no tuviesen una correlación objetiva con las actividades que se llevaban a cabo.

Podríamos haber obtenido valores irreales o que no siguiesen una correlación una vez acabada la BCI.

Con respecto a los posibles obstáculos, consideramos que era necesario consultar al Departamento de Protección de Datos debido a que nuestra investigación contaba con la participación de usuarios externos y utilizaba datos relacionados con la actividad cerebral recopilados durante las entrevistas.

4.4 PLANIFICACIÓN

En esta sección se abordará la planificación empleada lograr los objetivos previamente establecidos. Al comienzo se decide organizar el trabajo en fases se definen los subapartados dentro de cada fase. Establecimos 5 fases del proyecto basándonos en cada uno de los objetivos previamente descritos. En cada etapa, desarrollamos los subapartados y objetivos más específicos para cada situación.

Para la revisión del trabajo, se propusieron reuniones regulares con el responsable del proyecto, el Dr. Jordi Duch, y comunicación semanal a través del correo electrónico. La idea era poder resolver las preguntas o especificaciones necesarias en cada situación.

4.4.1 Descripción de las Tareas

A continuación, se proporcionará un análisis detallado de las tareas específicas que se llevaron a cabo para alcanzar cada uno de los objetivos del proyecto. Destacar una especificación muy concreta de cada uno de los pasos y apartados, que nos permitió llevar un control mucho más preciso de cada uno de los aspectos que consideramos más relevantes.

FASE 1. DOCUMENTACIÓN (11 SEMANAS)

Estudio previo sobre la lógica del cerebro y las posibilidades del proyecto.

F1.1 Análisis bibliográfico

1. Conocer la electrónica básica de las conexiones neuronales y del cerebro.
2. Estudios y bibliografía básica sobre las ondas cerebrales
3. Conocer la implicación y relación entre las actividades humanas y su equivalencia en ondas cerebrales

F1.2 Análisis tecnológico

4. Metodologías de captación de ondas cerebrales: dispositivos, costes, y precisión.
5. Selección de un dispositivo y determinación de las posibilidades

FASE 2. PREPARACIÓN SOFTWARE Y HARDWARE (11 SEMANAS)

Preparación del materia hardware, desarrollo del software y exploración de datos

F2.1 Hardware

6. Conseguir el dispositivo de captación de ondas, materiales eléctricos necesarios y otras cosas.
7. Hacking de la placa del dispositivo MindFlex.
8. Pruebas básicas y determinación de la validez del dispositivo MindFlex.
9. Verificación de la calidad y correlación de los datos.
10. Familiarización con el resto de las tecnologías (Arduino).

F2.2 Software

11. Preparación del Software de comunicación entre MindFlex-Arduino.
12. Limpiador de los datos procedentes de MindFlex con Arduino.
13. Creación de las bases de datos para los valores registrados por MindFlex.
14. Preparación de interfaces específicas para la recogida de datos con el dispositivo MindFlex con Python.

F2.3 Datos

15. Selección de los tipos de datos de interés para el proyecto.
16. Exploración de las posibilidades del proyecto con los datos y preparación para la tercera fase.

FASE 3. RECOGIDA DE DATOS (3 SEMANAS)

Recogida de datos con el dispositivo MindFlex. Primeras pruebas reales.

F3.1 Diseño del procedimiento

17. Determinación de los datos de las ondas de interés y estudio sobre como obtener cada una de ella de manera aislada.
18. Diseño de las pruebas a realizar a distintos usuarios.
19. Diseño de las entrevistas con los usuarios: breve explicación del proyecto, familiarización con la tecnología, desarrollo de las pruebas, detalle de agradecimiento con los usuarios.

F3.2 Entrevistas

20. Captación de los usuarios: familiares, amigos, compañeros de clase, profesores o gente de un entorno cercano.
21. Realización de las entrevistas de manera individual.

F3.3 Consultas al departamento de protección de datos

FASE 4. EVALUACIÓN DE DATOS (4 SEMANAS)

Evaluación de los datos.

22. Preparación y limpieza de los datos: fases de calibrado, valores nulos o posibles incidencias durante las entrevistas.
23. Construcción de los gráficos resultantes de los procedimientos de las ondas cerebrales.
24. Realización de análisis estadísticos, mapas de calor, valores de correlatividad.
 - a. Análisis estadísticos generales: medias, desviación estándar de los datos.
 - b. Relación entre parámetros del mismo experimento: tablas de correlación entre valores registrados.
 - c. Diferencia entre parámetros para dos experimentos diferentes.
 - d. Diferencias entre las distribuciones de dos usuarios para un mismo experimento: aplicación de test de Kolmogorov–Smirnov.
25. Evaluar el resultado de los primeros datos y toma de decisiones para las siguientes fases.

FASE 5. CREACIÓN DE LA INTERFAZ GRÁFICA (3 SEMANAS)

26. Determinación del propósito y funcionalidad de la aplicación.
27. Creación de los sprites y la estética del videojuego para la interfaz gráfica de usuario.
28. Implementación del videojuego y pruebas de funcionamiento.
29. Refinado del videojuego y conectividad.

4.4.2 Diagrama de Gantt

Durante todo el procedimiento, se hicieron estimaciones semanales en base al tiempo que nos supondría cada una de las tareas, y esto se fue registrando periódicamente en un diagrama de Gantt. Esta herramienta nos permite una visualización clara de la planificación del proyecto.

De esta manera, podemos concluir y determinar a simple vista la importancia que tuvo un buen trabajo previo en el desarrollo de este proyecto, tanto de investigación como de preparación, frente a la propia ejecución del proyecto. Ver Anexo I y Anexo II.

4.5 HERRAMIENTAS

Durante el desarrollo del proyecto se utilizaron distintas herramientas que nos facilitaron una gestión eficiente, coordinación con el responsable y una organización sencilla de las diferentes tareas.

COMUNICACIÓN

Teams. Debido a que me encontraba realizando una estancia erasmus durante el desarrollo del trabajo, las reuniones regulares de revisión se hicieron remotamente a través de esta plataforma.

Correo electrónico. Para la comunicación semanal con el Dr. Jordi Duch-Gavaldà, se utilizó mensajería.

GESTIÓN DE LAS TAREAS

Notion. Esta herramienta nos permitió crear y organizar todas las partes del proyecto. Permite crear listas de tareas, documentos, tableros, bases de datos etc., que nos facilitaba el seguimiento y el control diario de las tareas y el estado del proyecto.

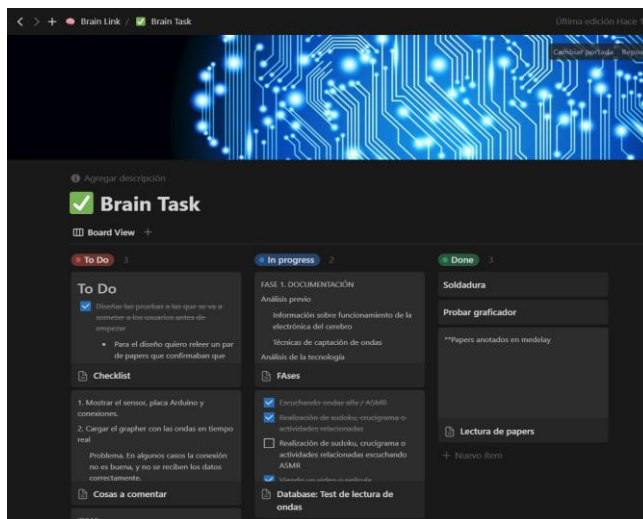


Figura 10. Captura de pantalla con la gestión de tareas en Notion.

GESTIÓN DEL CÓDIGO

GitHub. Esta herramienta fue parte fundamental de la gestión de versiones del código. Creamos una rama principal (master) y ramas subyacentes para las nuevas funciones que se iban implementando. Se haría un registro periódico con *commits* en cada rama, y una vez

terminada y testeada cada una de las funciones, se fusionaría con la rama master mediante un *merge*.

4.6 RECURSOS Y PRESUPUESTO

A continuación, se propone un presupuesto estimado asociado al coste del desarrollo del TFG. El presupuesto se ha dividido en costos directos e indirectos. Para los primeros tendremos en cuenta únicamente aquellos relacionados a los recursos humanos y costes de hardware especializado. Por otro lado, respecto a los costes indirectos, se detallarán a continuación los costos de software y costos materiales y de suministros.

4.6.1 Identificación y Estimación de los Costes

4.6.1.1 Recursos Directos

4.6.1.1.1 Recursos humanos

En este apartado se deberían incluir los salarios y honorarios del personal responsable del desarrollo. En vista de las características del proyecto, decidimos detallar las funciones elementales realizadas, así como un tiempo de trabajo estimado basadas en el diagrama de Gantt y sus respectivas tasas:

Producto	Coste temporal (semanas)
Planificación y estudio del proyecto	9
Análisis y elección de la tecnología	5
Creación de base de datos	2
Software específico para el tratamiento de datos	6
Diseño y realización de las entrevistas y captación de usuarios	3
Análisis de datos	4
Creación de la interfaz gráfica	3
TOTAL	31 semanas

Tabla 3. Coste temporal de los recursos humanos durante la gestión del proyecto.

4.6.1.1.2 Recursos hardware

Estos recursos incluyen el costo de los materiales utilizados en la fase de montaje del material hardware.

- Juguete MindFlex
- Microcontrolador Arduino UNO

Producto	Coste para el proyecto (€)
MindFlex	15
Arduino UNO	28
TOTAL	43€

Tabla 4. Coste de los recursos software para la elaboración del proyecto.

4.6.1.2 Recursos Indirectos

4.6.1.2.1 Hardware Propio

A continuación se expone el hardware propio utilizado durante el desarrollo del proyecto.

- Huawei MateBook D 15 AMD Ryzen 7 4800H/16GB/1TB SSD – Ordenador personal .
- iPhone 11 128 GB ROM / 4 GB RAM – Móvil personal utilizado para grabar las entrevistas.
- iPad 2019 128 GB RAM / 3 GB RAM – Equipo utilizado para el desarrollo de las entrevistas.

Producto	Coste (€)
Huawei D15	929,00 €
iPhone 11	599,00 €
iPad 2019	405,00€
TOTAL	1.933,00€

Tabla 5. Coste del hardware personal para la elaboración del proyecto.

Dado que suponemos que el material ha sido utilizado en más ocasiones además de este proyecto, se decide no tener en cuenta el coste total del material para este único proyecto.

Se expone a continuación una amortización lineal teniendo en cuenta la vida útil promedio de cada dispositivo. Consideraremos en este caso la vida útil de cada dispositivo en semanas y calcularemos el coste equivalente a la duración del proyecto (28 semanas), o el número de semanas de utilización de cada dispositivo. Con este supuesto, el coste se reduce a **67,82 €**.

Producto	Coste unitario (€)	Amortización (años)	Amortización (semanas)	Uso durante el proyecto (semanas)	Coste para el proyecto (€)
Huawei D15	929,00 €	8	417,1432	32	62,36 €
iPhone 11	599,00 €	4	208,5716	1	2,87 €
iPad 2019	405,00€	8	156,4287	1	2,59 €
TOTAL					67,82 €

Tabla 6. Coste del hardware personal para la elaboración del proyecto calculado según la amortización lineal de los dispositivos

4.6.1.2.2 Recursos Materiales

En este caso, se detallan los gastos relacionados con la compra de materiales o suministros necesarios para el desarrollo del software.

- **Pilas**
- **Impresión de documentos**
 - Guiones para las entrevistas.
 - Papers para el estudio.
 - Tarjetas de agradecimiento para los usuarios participantes en las entrevistas.
- **Regalo** de agradecimiento para los usuarios de las entrevistas.
- **Equipo de soldadura.** En este caso, un amigo familiarizado en micro-soldaduras nos hizo el favor, por lo que no incluiremos el coste.

- **Bridas.**
- **Destornillador** de estrella de punta fina.
- **Cable usb A- usb B** para la conexión Arduino-ordenador.
- **Cables** con recubierta de silicona de 0.32mm².

Producto		Unidades	Coste para el proyecto (€)
Fase 1	Pilas AA	4	1,99
	Bridas	2	0,90
	Destornillador estrella punta fina	1	2
	Coste de soldadura	1	-
Fase 2	Cable 0.32mm ²	2	7,98
	Cable usb A - usb B	1	3,20
Fase 3	Impresión de documentos	~ 50	~ 10
	Regalos entrevistas	2	5,50
TOTAL			31,57€

Tabla 7. Coste de los recursos materiales para la elaboración del proyecto.

4.6.1.2.3 Recursos Software

Debido a que todos los programas y herramientas utilizados en el desarrollo estaban disponibles bajo licencia gratuita, no se incluyen los costos de adquisición de software.

Se optó por utilizar herramientas de código abierto y programas disponibles gratuitamente, lo que permitió minimizar los costos.

De igual manera, se cita a continuación el software utilizado:

Elementos	Coste (€)
GitHub	0
Visual Studio Code	0
Arduino IDE	0
Processing-4	0
PixelArt	0
TOTAL	0 €

Tabla 8. Coste de los recursos software para la elaboración del proyecto.

4.6.2 *Análisis General del Presupuesto*

El análisis general de los costos incluye el equipo personal (costo equivalente a la duración del proyecto), los recursos hardware, recursos materias y los recursos software. Se excluyen los costos de recursos humanos en basándonos en que se trata de la elaboración de un proyecto TFG. A pesar de esto, se expone previamente (apartado 4.6.1.1.1) el desglose de los recursos y su costo horario.

Elementos	Coste (€)
Recursos humanos	-
Equipo personal	67,82
Hardware	43
Material	31,57
Software	0
TOTAL	142,39€

Tabla 9. Tabla resumen con el costo monetario total para la realización del proyecto

5. DISEÑO DEL SISTEMA BCI

5.1 ESQUEMA GENERAL DEL DISEÑO

En esta sección describiremos los aspectos claves del diseño detrás de nuestro sistema. Para todo BCI, podemos diferenciar tres partes esenciales como son la **adquisición de la señal**, el **procesado de los datos** y la **aplicación** de estos. En este caso, ajustaremos estas tres partes específicamente para nuestro proyecto en concreto:

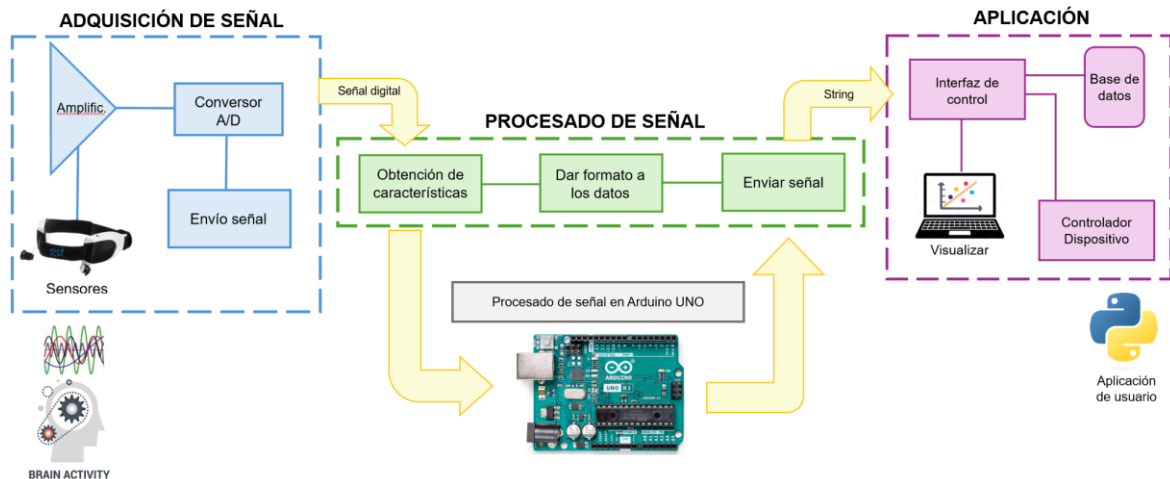


Figura 11. Esquema general del funcionamiento de un sistema BCI adaptado a las características del proyecto. Fuente: Elaboración propia.

- Adquisición de la señal:** En esta etapa se captura la señal procedente del cerebro. Como explicamos anteriormente, elegimos la electroencefalografía (EEG) como la técnica más adecuada para desarrollar nuestro BCI, en este caso con los sensores de la banda MindFlex. En este paso, el propio MindFlex se encargará de filtrar la señal, transmitiendo únicamente las frecuencias deseadas de manera digital.
- Procesado de la señal:** El procesado de la señal implica todo el acondicionamiento de la señal para las características que se requieren para la BCI en cuestión. En este proyecto, este paso consistirá en la preparación y transmisión mediante comunicación serial desde el dispositivo MindFlex al microcontrolador Arduino. La **implementación es semidúplex**: Utiliza el pin RX de Arduino para transmitir los datos y este será el que los envíe al ordenador. Trataremos la señal procedente de la banda EEG y transmitiremos un string en formato CSV con todos los valores y la calidad de la señal. Podemos ver un ejemplo en la figura 12.
- Aplicación:** En el último paso se determinarán las acciones a realizar en función de la señal recibida. Trabajaremos con Python creando una base de datos con los valores recibidos desde el microcontrolador Arduino. A partir de esta, desarrollaremos la propia interfaz de control y la aplicación de usuario, en este caso, un videojuego.

```

Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5') Both NL & CR 9600 baud
20:51:12.439 -> 26,0,0,514935,478895,87998,237163,47420,131939,98729,88341
20:51:13.432 ->
20:51:13.432 -> 26,0,0,620256,261363,172245,389348,91981,98141,111232,100789
20:51:14.448 ->
20:51:14.448 -> 26,0,0,353380,489692,499329,130719,264716,231276,75623,63420
20:51:15.414 ->
20:51:15.447 -> 26,0,0,276352,499211,158344,199955,77810,107088,62733,45888
20:51:16.427 ->
20:51:16.427 -> 26,0,0,987524,766643,238385,433217,78758,90778,77366,56070
20:51:17.406 ->
20:51:17.438 -> 26,0,0,237847,1017861,305209,84602,103021,48097,94952,121829
20:51:18.425 ->
20:51:18.425 -> 26,0,0,161404,957169,235845,461874,43037,125671,114524,77245

```

Figura 12. Consola de salida serial de los valores de ondas cerebrales capturadas en el microcontrolador Arduino. El primer valor corresponde al string de los valores, el segundo (en blanco) errores en la transmisión encontrados durante la comunicación.

5.2 DISEÑO DE HARDWARE: MONTAJE DEL DETECTOR

El desarrollo de un BCI funcional y eficiente requiere del diseño de hardware y el montaje del sistema. En este apartado, nos centraremos en el proceso de diseño y ensamblaje de los componentes físicos que constituyen nuestro sistema BCI: desde la preparación de los sensores hasta la configuración del sistema en su conjunto.

Este apartado se centra además en los preparativos para el apartado de adquisición de la señal que se explicaba anteriormente.

Cabe destacar que no somos los primeros en utilizar el dispositivo MindFlex para proyectos recreativos o de investigación, pues en internet se puede encontrar información al respecto. Es por eso, que para el montaje del detector nos basamos en información relevante y videotutoriales que encontramos en internet.

5.2.1 Montaje del Detector

Para el montaje del detector, seguiremos un tutorial que podemos encontrar en [19].

Primero de todo, desmontaremos el protector izquierdo de la diadema MindFlex dejando ver la placa TGAM de NeuroSky que contiene el juguete, adyacente a la placa del propio dispositivo.



Figura 13. Fotografía de la diadema MindFlex desmontada.

Buscamos en la placa TGAM los pines T y R. Estos pines son los responsables de la comunicación en serie con el microcontrolador Arduino. Soldaremos un cable al punto TX (transmisión de datos) ejemplificado en la figura 15 con una línea azul, que se comunicará con el punto Serial Port RXD (recepción de datos) de la placa Arduino UNO.

Por otra parte, buscamos la toma de tierra del dispositivo MindFlex (en color naranja) y soldaremos un cable que irá directamente conectado con la misma toma en la placa Arduino.

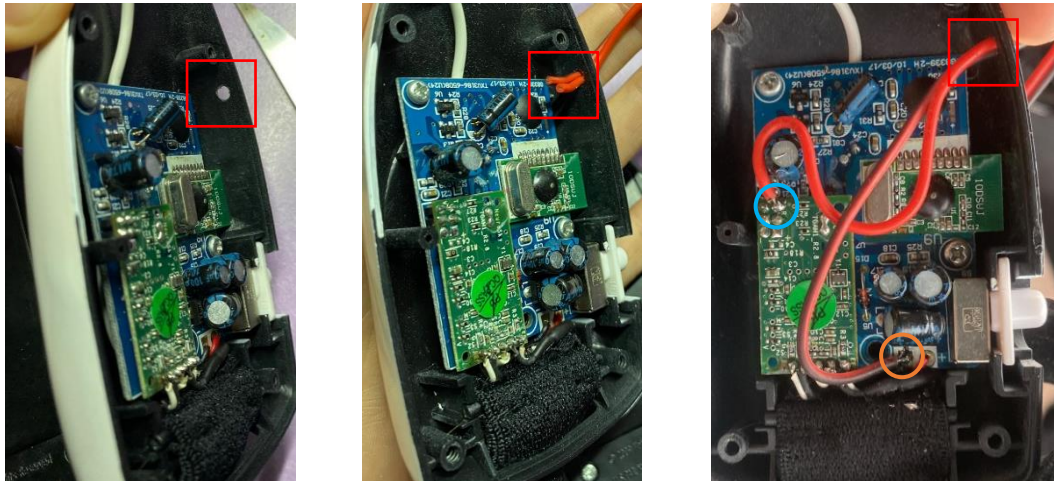


Figura 14. Fotografías del proceso de hackeo de la diadema MindFlex.

Una vez terminado, y por recomendación de algunas de las páginas revisadas, agujeramos la carcasa del MindFlex para pasar los cables del interior al exterior y poder cerrar caja protectora, aliviando así la tensión de los cables que irán conectados a la placa Arduino.

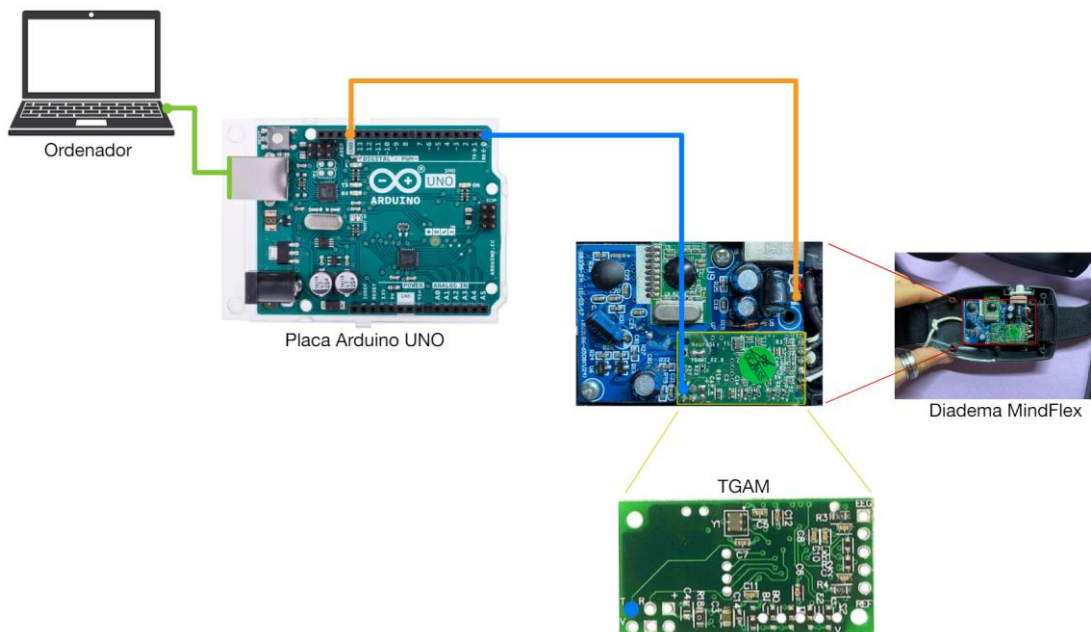


Figura 15. Esquema general del funcionamiento del hardware BCI. Fuente: Elaboración propia

5.3 DISEÑO DE SOFTWARE: ADQUISICIÓN Y PROCESADO DE SEÑAL

5.3.1 Arduino Sketch

Para la adquisición y primer procesado de la señal, utilizaremos el microcontrolador Arduino. En este caso, utilizamos una librería de código abierto publicada en <https://github.com/kitschpatrol/Brain>. Este sketch será responsable de la captura de señales de EEG, el procesamiento de datos utilizando la librería mencionada y la transmisión de datos en formato CSV a través del puerto serie.

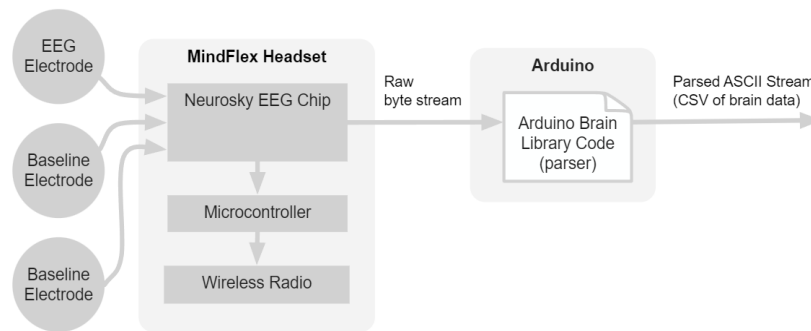


Figura 16. Esquema de la adquisición y procesado de señal de la librería Brain. Fuente: (Brain Hack, 2014)

Esta librería cuenta con funciones específicas para el dispositivo MindFlex y Star Wars Force Trainer que simplifican el envío de cadenas ASCII en serie a partir de la información procesada y recibida de los dispositivos.

Utilizamos además el código base para hacer comprobaciones sobre la velocidad, de transmisión de los datos. Se realizaron pruebas exhaustivas del código para evaluar un correcto funcionamiento con el dispositivo EEG, pues este no había sido revisado desde 2014.

A este código se añadieron además algunas funciones de optimización para garantizar un rendimiento eficiente y asegurar una correcta transmisión de datos con el menor desfase posible. Hay que tener en cuenta que la implementación se trata de un **sistema semidúplex** por lo que siempre contaremos con un pequeño intervalo de tiempo entre la captura y la transmisión de los datos.

Algunas de las funciones más relevantes empleadas por el sistema Arduino Brain son las siguientes:

Inicialización y control:

```
boolean update();
```

Lee los valores de la banda EEG en el bucle principal

```
char* getCSV();
```

Devuelve una cadena de caracteres con todos los valores cerebrales en formato CSV. Los datos se devuelven en este orden:

“signal strength, attention, meditation, delta, theta, low alpha, high alpha, low beta, high beta, low gamma, high gamma”

```
byte readSignalQuality();
```

Devuelve la última lectura de calidad de señal. 200 indica que no hay señal, 0 indica que hay una buena señal.

Devuelve los valores de aplicación de los algoritmos de atención y meditación:

```
byte readAttention();
```

Devuelve el valor de atención "eSense" de NeuroSky.

```
byte readMeditation();
```

Devuelve el valor de meditación "eSense" de NeuroSky.

Devuelve los valores de todas las ondas:

```
unsigned long readDelta();
```

Devuelve el valor de la onda delta (1–3 Hz).

```
unsigned long readTheta();
```

Devuelve el valor de la onda theta (4–7Hz).

```
unsigned long readLowAlpha();
```

Devuelve el valor de la onda alfa bajas (8–9 Hz).

```
unsigned long readHighAlpha();
```

Devuelve el valor de onda alfa alta(10–12 Hz).

```
unsigned long readLowBeta();
```

Devuelve el valor de la onda beta baja (13–17 Hz).

```
unsigned long readHighBeta();
```

Devuelve el valor de la onda beta alta (18-30 Hz).

```
unsigned long readLowGamma();
```

Devuelve el valor de la onda gamma bajo (31-40 Hz)

```
unsigned long readMidGamma();
```

Devuelve el valor de la onda gamma alta (41-50 Hz).

Como conclusión, el sistema Arduino transmitirá por el puerto serie 9600 con una frecuencia de transmisión de 2 strings/segundo.

5.3.1.1 BrainGrapher

BrainGrapher es otro software de código abierto desarrollado por el mismo usuario, que permite visualizar de una manera más intuitiva los datos recibidos en tiempo real en Processing-4².

El código BrainGrapher funciona con los valores transmitidos por Arduino BrainSerialOut, y será utilizado únicamente a nivel informativo y para ayudarnos con el desarrollo del proyecto. En este caso, no se hizo ninguna modificación ni optimización del código.

Adicionalmente, se utilizó esta herramienta para el desarrollo de las entrevistas para el registro de datos, donde utilizábamos el software para mostrar al participante sus ondas cerebrales a tiempo real.

² Processing-4 es un lenguaje de programación que permite una visualización de datos y creación de gráficos dinámicos en tiempo real.

Podemos encontrar el código en <https://github.com/kitschpatrol/BrainGrapher>.

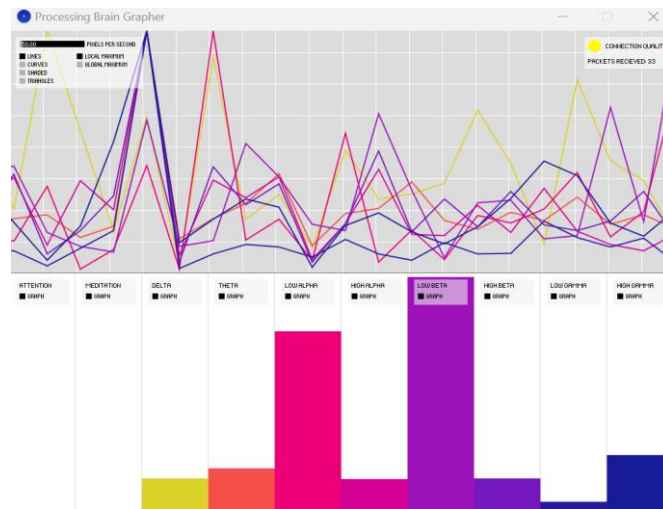


Figura 17. Captura de pantalla del funcionamiento de la librería BrainGrapher en Processing-4

5.3.2 Python

Para el procesado de la señal utilizaremos Python. El programa principal está estructurado como una librería de scripts independientes que nos permite cumplir con los todos los requisitos establecidos de manera individual. Para entender el funcionamiento general del proyecto, explicaremos cada uno de los archivos de manera individual y su funcionalidad en cada caso.

Se explican además a continuación las distintas bibliotecas utilizadas para la implementación del proyecto.

BIBLIOTECAS EMPLEADAS

- **os.** Utilizamos os para manejar rutas, crear y eliminar archivos y directorios, y ejecutar comandos del sistema.
- **random:** Utilizamos esta biblioteca para generar números aleatorios y realizar operaciones relacionadas con la aleatoriedad.
- **threading:** Permite la ejecución de código paralelo utilizando hilos (threads). La utilizaremos para realizar la lectura de ondas de forma concurrente.
- **logging:** Permite la captura de eventos y la generación de registros (logs). Utilizamos esta biblioteca para la depuración de errores y comprensión del estado del programa en cada momento.
- **serial:** La biblioteca serial (o pySerial) facilita la comunicación con dispositivos a través de puertos seriales. Utilizamos esa biblioteca para recibir los datos desde el microcontrolador Arduino.
- **datetime:** Permite manipular y operar con fechas y horas.
- **pandas:** Crea estructuras de datos como DataFrames y nos facilita la manipulación, agregación, limpieza y análisis de los datos.
- **numpy:** Utilizaremos la biblioteca numpy para el cálculo matemático y numérico de algunas funciones.
- **scipy:** La biblioteca scipy amplía las capacidades de numpy para los análisis estadísticos y procesamiento de datos.
- **matplotlib:** Esta biblioteca se utiliza para crear los gráficos de representación de los datos registrados en 2D.

ESTRUCTURA DE FICHEROS

BrainLink/

```
|— game/
    |— __pycache__/
    |— Assets/
    |— play.py
    |— receiveDataGame.py
    └— test_ovni_game.py

|— data/
    |— __pycache__/
    |— receiveData.py
    |— trainingData.py
    |— csv_functions.py
    |— receiveData/
    └— trainingData/

|— interfaces/
    |— __pycache__/
    |— dataType_interface.py
    |— saveData_interface.py
    └— user_interface.py

|— graphs/
    |— graphicator_main.py
    |— plot_waves.py
    └— graphics_waves/

|— estadistics/
    |— media.py
    |— correlation.py
    |— histogramas.py
    |— testKS.py
    |— estadisics_main.py
    └— estadisticResults/
```

5.3.2.1 Data

Esta carpeta se compone de varios scripts para la captura de señales de EEG procedente de Arduino en formato CSV, originada en el dispositivo MindFlex. Las señales capturadas se almacenan en archivos de registro para su posterior análisis y procesamiento.

En función de la fase en la que nos encontremos, se desarrolló un script diferente para cumplir con las necesidades en cada momento.

RECEIVEDATA.PY

Este script nos permite una toma de contacto con el recibimiento y procesado de los datos. Primero conecta el dispositivo Arduino a un puerto serie y recibe datos en tiempo real.

Filtrar los valores vacíos, los muestra por pantalla y los almacenan en una lista. La lista de datos se guarda en un archivo CSV en la subcarpeta Data/ con el nombre del dateTime (basado en la fecha actual), cuando se finaliza la ejecución del script. A continuación, una muestra la general:

```
ser = serial.Serial('COM5', 9600)
while True:
    try:
        data = ser.readline().decode().strip()
        if data:
            print("Datos recibidos desde Arduino:", data)
            data_list.append(data)
    except KeyboardInterrupt:
        ser.close()      # Cerrar el puerto serie al finalizar
        print("Puerto serie cerrado")
        file_name = "data\\" + datetime.now().strftime('%Y_%m_%d') +
        ".csv"
        save_data(folder_path, file_name, data_list)
```

TRAININGDATA.PY

Este script está dedicado a la captura de los datos, selección del usuario y determinación de la prueba que se está llevando a cabo durante la entrevista.

El programa se inicia con una selección del usuario mediante interfaces específicas. En este apartado se permite la creación de un nuevo perfil y salir del proceso. Las interfaces se encuentran definidas en la carpeta Interfaces que definiremos más adelante.

Para permitir la ejecución simultánea de otras operaciones en el programa, el proceso de recepción de datos se lleva a cabo en un hilo de ejecución diferente.

```
if selected_user:
    # Crear un hilo para recibir los datos en segundo plano
    receptionThread = threading.Thread(target=recibir_datos)
    receptionThread.daemon = True
    receptionThread.start()
```

```
print("-----Selección tipo de datos-----")
data_window = TI.DataTypeWindow(selected_user)
save = data_window.create_window()
```

A continuación, conecta el dispositivo Arduino a través del puerto serie 9600, recibe datos del dispositivo y los procesa en tiempo real. Los datos son filtrados, procesados y almacenados en listas separadas mediante funciones específicas. Una vez comprobada que la conexión establecida es correcta, comienza la ejecución.

```
def recibir_datos():
    while True:
        if data:
            process_data(raw_data)
```

El procesado de datos filtra los 4 primeros valores recibidos como calibrado y comprueba que la señal recibida desde el sistema semi-duplex formado con Arduino, el valor *signal strength* sea superior a 200.

Selecciona los valores de interés, añade el tiempo de captura mediante un formato timestamp, y almacena los valores en una lista en el siguiente formato:

[timestamp, attention, meditation, deltatheta, low alpha, high alpha, low beta, high beta, low gamma, high gamma]

```
def process_data(raw_data):
    timestamp = int(time.time())
    if (calibrationCounter > 3) and (array[0] > 200):
        calibrated_values = array[1:9]
        calibrated_values = np.append([timestamp,
data_window.data_type], calibrated_values)
        procesed_data.append(calibrated_values)
```

CSV FUNCTIONS.PY

Este script define la clase `csvFunctions` y proporciona funciones que permiten trabajar con archivos CSV: realizar operaciones como mostrar datos, verificar y crear CSV, guardar datos o añadir datos a un CSV existente. Las funciones de esta clase fueron utilizadas por el resto de los scripts para almacenar datos. Algunas de las funciones declaradas son las siguientes:

- `show_data(data)` imprime los datos almacenados en una lista en formato de matriz.
- `verify_and_create(path)` verifica si un archivo CSV existe en una carpeta y, si no existe, crea un nuevo archivo CSV vacío en la carpeta especificada.
- `save_data(path, raw_data)` guarda los datos almacenados en una lista en un archivo CSV.
- `save_procesed_data(path, data)` guarda los datos procesados almacenados en una lista en un archivo CSV.
- `append_data(data,)` añade datos a un archivo CSV existente.

Se decidió almacenar los datos relativos al proyecto en ficheros en formato CSV debido a que no contábamos con datos masivos de información. Este formato permite una lectura y escritura sencilla mediante scripts de Python y con ampliamente compatibles con todos los programas.

5.3.2.2 Interfaces

Esta carpeta define diferentes clases dedicadas a crear interfaces específicas utilizadas durante la recogida de datos y las entrevistas (explicado más adelante).

DATA TYPE INTERFACE.PY

Proporciona una interfaz de usuario para seleccionar el tipo de datos que se están registrando. La clase tiene una lista predefinida de tipos de datos disponibles, y crea *checkboxes* para cada tipo de dato en la lista. Permite al usuario seleccionar uno a la vez.

`create_window()` inicializa y muestra la ventana de selección de datos. Al cerrar la ventana, devuelve un valor booleano (`save`) que indica si se debe guardar la información o no.

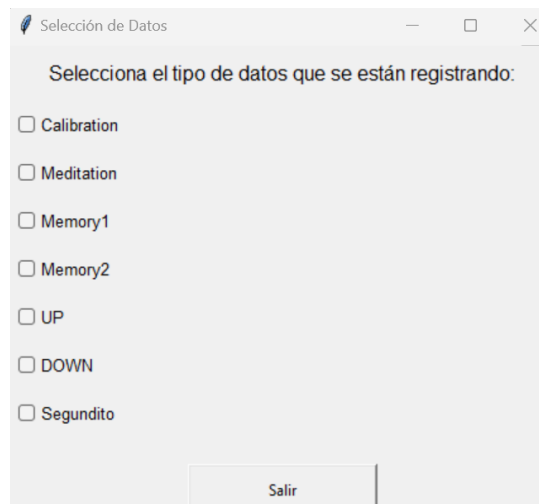


Figura 18. Interfaz de selección del tipo de dato registrado.

SAVE DATA INTERFACE.PY

Esta clase proporciona una manera sencilla de guardar datos en un archivo CSV y luego visualizarlos en un DataFrame de Pandas para su posterior análisis.

La clase toma el nombre del archivo CSV y los datos a guardar como argumentos en su inicialización.

Crea una ventana que muestra el nombre del usuario cuyos datos se van a guardar. Esta ventana ofrece dos botones: "Guardar" y "No guardar".

`DI_cargar_archivo()` se encarga de guardar los datos en un archivo CSV y luego cargar este archivo en un DataFrame de Pandas para visualizar su correcto guardado.

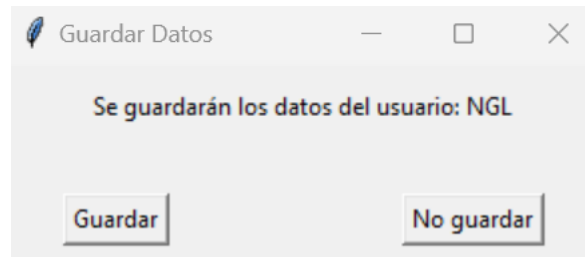


Figura 19. Interfaz para guardar o descartar los datos capturados de un usuario concreto.

USER_INTERFACE.PY

Esta clase permite que seleccionar un perfil de usuario existente o crear uno nuevo antes de continuar con el programa.

La clase toma una ruta de carpeta como argumento en su inicialización.

Crea una ventana Tkinter que muestra una lista de botones, cada uno correspondiente a un perfil de usuario existente.

Ofrece dos botones adicionales: "Nuevo perfil" para crear un nuevo perfil de usuario y "Salir" para cerrar la ventana. Cuando se hace clic en el botón "Nuevo perfil", solicita al usuario que introduzca el nombre del nuevo perfil y luego crea un archivo CSV vacío con ese nombre en la carpeta especificada.

`user_window()` inicializa y muestra la ventana de selección de usuario. Devuelve el nombre del usuario seleccionado.

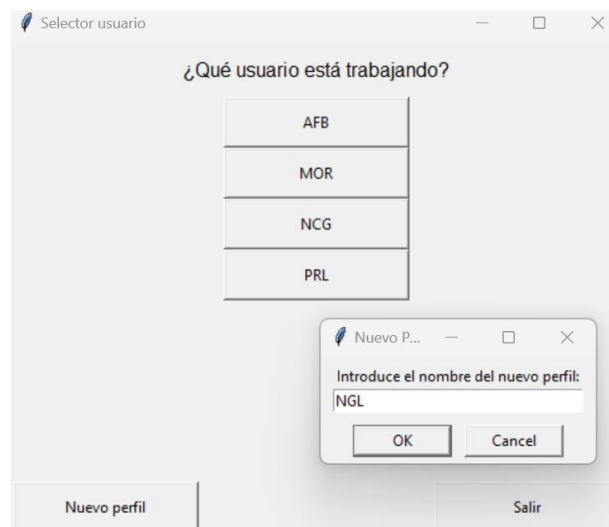


Figura 20. Interfaz para la selección del usuario y creación de nuevos perfiles.

5.3.2.3 Graphs

Esta carpeta permite graficar los valores obtenidos y guardados para cada uno de los usuarios. Facilita una manera sencilla de visualizar todos los datos una vez terminada la captura de estos.

PLOT WAVES.PY

Este script define la clase `AllDataGraphic` que permite crear gráficos de las ondas almacenados en archivos CSV de la captura de cada uno de los usuarios. Se utiliza principalmente para tener una visión general de como han ido las entrevistas y si fuese necesario, mostrárselas a los usuarios al acabar.

La clase toma una ruta de carpeta como argumento en su inicialización, donde se encuentran los archivos CSV con los datos de interés.

Se definen además tres métodos para generar diferentes tipos de gráficos. Cada método recibe el nombre del archivo CSV que contiene los datos y una carpeta de salida opcional donde se guardarán los gráficos generados. Los gráficos se guardarán en archivos PNG en la carpeta de salida especificada o se mostrarán en una ventana emergente.

- `_read_csv ()` lee un archivo CSV, asignar nombres de columnas y convertir la columna de tiempo a formato de fecha y hora.
- `plot_Alldata ()`: Genera gráficos para todas las categorías de datos especificadas.
- `meditation_and_attention ()`: Genera gráficos de atención y meditación para cada categoría.
- `plot_waves ()`: Genera gráficos de ondas cerebrales para las categorías especificadas.

Para cada categoría de datos, los métodos generan un gráfico que muestra la evolución de las diferentes variables u ondas cerebrales a lo largo del tiempo de registro.

GRAPHICATOR MAIN.PY

Este script procesa todos los archivos CSV de una carpeta de entrada y genera gráficos utilizando las funciones de la clase `AllDataGraphic` en función de las necesidades en cada momento.

Crea una instancia de la clase `AllDataGraphic` y se llama a la función `process_csv_files ()` con las rutas de entrada y salida como argumentos, lo que inicia el proceso de generación de gráficos para los archivos CSV y guarda los gráficos resultantes en la carpeta de salida.

```
def process_csv_files(input_folder, output_folder):
    graphic_processor = AllDataGraphic(input_folder)

    # Procesa todos los archivos CSV en la carpeta de entrada
    for file_name in os.listdir(input_folder):
        if file_name.endswith(".csv"):
            file_path = os.path.join(input_folder, file_name)
            graphic_processor.meditation_and_attention(file_path,
output_folder)

            graphic_processor.plot_Alldata(file_path, output_folder)
            graphic_processor.plot_waves(file_path, output_folder)
```

```
# Rutas de entrada y salida
input_folder = "..\\TFG BrainLink \\trainingData"
output_folder = "..\\TFG BrainLink \\trainingData\\Graphics"

process_csv_files(input_folder, output_folder)
```

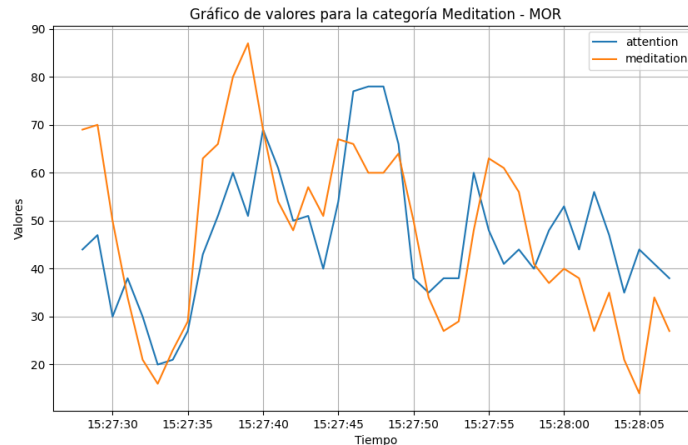


Figura 21. Gráfica de ejemplo con los valores de atención y meditación registrados para la categoría de meditación y el usuario MOR.

5.3.2.4 *Estadistics*

MEDIA.PY

El script es utilizado para unificar todos los datos de todos los documentos CSV existentes. Nos permite juntar todos los valores existentes de los distintos usuarios y así organizarlos por categorías. Es un paso previo a los análisis estadísticos.

CORRELATION.PY

Este archivo define una clase `Correlation` y se encarga de leer un archivo CSV, calcular una matriz de correlación entre las diferentes variables y mostrar distintos tipos de visualizaciones según interés. Para esto utiliza la función específicas de la biblioteca Pandas, `corr()`

- `correlacion_todas_categorias()`: Calcula y muestra la tabla de correlación general sin hacer distinción en el tipo de dato archivo CSV. Guarda esta tabla de correlación en dos formatos de archivo diferentes (CSV y TXT).

Utiliza las funciones `correlation_heatmap()`, `correlation_dispersion()` y `correlation_bars()` para generar tres tipos de visualizaciones para cada tipo de dato: un mapa de calor de correlación, gráficos de dispersión y gráficos de barras de correlación,

- `correlacion_por_categorias()`: Calcula y muestra las tablas de correlación para cada tipo de dato específico de entre todas las pruebas realizadas en la entrevista. Itera sobre cada tipo de dato especificado y filtra el DataFrame original para obtener solo los datos de ese tipo.

```
tipos_especificos = ["Meditation", "Memory1", "Memory2", "UP", "DOWN", "Segundito"]
```

Utiliza las funciones para generar tres tipos de visualizaciones para cada tipo de dato: un mapa de calor de correlación, gráficos de dispersión y gráficos de barras de correlación.

SEGUNDITO

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
attention	1.0000							
meditation	0.1590	1.0000						
delta	0.0573	-0.0310	1.0000					
theta	-0.3379	0.1577	0.3517	1.0000				
low alpha	-0.2206	0.2181	0.3958	0.5463	1.0000			
high alpha	-0.1959	0.2296	0.3815	0.7151	0.4841	1.0000		
low beta	-0.1151	0.1778	0.4394	0.6306	0.5005	0.6664	1.0000	
high beta	-0.0888	0.2211	0.4636	0.7028	0.5633	0.7151	0.7324	1.0000

Tabla 10. Tabla de ejemplo con los valores de correlación para la actividad de Segundito.

- **correlation_heatmap(matrix, tipo)**: Crea un mapa de calor de correlación utilizando Matplotlib a partir de la matriz de correlación calculada y el tipo de dato como entrada. Los valores más altos de correlación se representan con colores más cálidos y los valores más bajos con colores más fríos. Finalmente, guarda esta visualización como una imagen PNG.
- **correlation_dispersion(matrix, df_tipo, tipo)**: Crea gráficos de dispersión para visualizar las relaciones entre las variables del tipo de dato especificado. Utiliza Matplotlib para generar una matriz de subgráficos, donde cada subgráfico muestra la relación entre dos variables. Guarda esta visualización como una imagen PNG.
- **correlation_bars(matrix, tipo)**: Crea gráficos de barras para visualizar las correlaciones más altas y bajas entre las variables del tipo de dato especificado. Utiliza Matplotlib para generar un gráfico de barras horizontal, donde las correlaciones positivas se muestran en un color y las correlaciones negativas en otro. Esto permite una forma rápida de identificar las relaciones más fuertes y débiles entre las variables. Guarda estos gráficos como una imagen PNG.

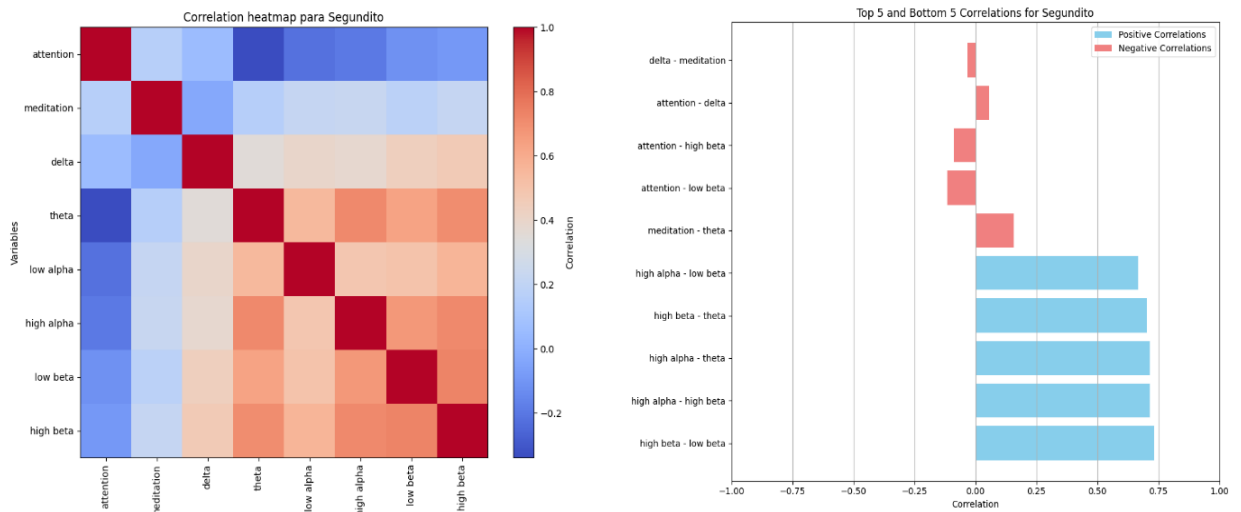


Figura 22. Heatmap y gráfica para los valores de correlación de los pares de variables durante la actividad de segundito.

HISTOGRAMS.PY

Define la clase **Histograms** que proporciona métodos para generar histogramas y distribuciones normales. Permite distinguir los datos conjuntos, así como entre categoría de datos. Utiliza la función **norm()** de la librería Scipy para calcular las distribuciones.

- **histogram_distribution_todas_categorias()**: Este método crea histogramas y distribuciones normales para todas las categorías de datos sin hacer distinción en el tipo de dato archivo CSV. Los resultados se guardan como imágenes PNG en una carpeta de salida.
- **histogram_distrib_por_categorias()**: Este método crea histogramas y distribuciones normales para cada tipo de dato específico de entre todas las pruebas realizadas en la entrevista.

Utiliza **seleccionar_categorias()** para agrupar los datos por categoría y luego itera sobre cada categoría para calcular las estadísticas y generar los histogramas correspondientes. Los resultados se guardan como imágenes PNG en una carpeta de salida.

TESTKS.PY

Este script inicializa la clase **Test** y realiza pruebas estadísticas y análisis comparativos entre diferentes grupos de datos contenidos en un archivo CSV. El script itera sobre cada par de tipos de datos y realiza las pruebas y análisis para cada variable en esos grupos de datos. Utiliza las funciones **ks_2samp()** y **ttest_ind()** de la librería Scipy para calcular los valores.

Los resultados se guardan en un archivo de texto y los gráficos se guardan como imágenes PNG. A continuación, una descripción de las funciones principales:

- **KS_test()**: Realiza la prueba de Kolmogorov-Smirnov para comparar las distribuciones de dos conjuntos de datos. Calcula el estadístico KS y el valor p, e imprime los resultados.
- **tStudent_test()**: Realiza la prueba t-Student para comparar las medias de dos conjuntos de datos. Calcula el estadístico t y el valor p, e imprime los resultados.
- **estadistica_descriptiva()**: calcula estadísticas descriptivas (media, desviación estándar, etc.) para cada conjunto de datos y los imprime.
- **crear_plots()**: Esta función crea un histograma comparativo para cada variable en los dos grupos de datos y lo guarda como una imagen PNG.
- **save_plot()**: Esta función guarda los gráficos creados por **crear_plots** en una carpeta específica.

ESTADISTICS_MAIN.PY

Este script procesa todos los archivos CSV de una carpeta de entrada. Realiza los análisis estadísticos y crea gráficos utilizando todas las funciones y clases previamente descritas

Crea distintas instancias las clases **Test**, **Histograms** y **Correlation** y llama a las distintas funciones de interés en cada circunstancia.

5.3.3 Resumen de Clases

Ruta	Fichero	Clase	Funcionalidad
Data/	trainingData.py	trainigData	Desarrollo de las entrevistas. Captura ondas, selecciona usuario y determina que actividad se esta llevando a cabo.
	csv_functions.py	csvFunctions	Proporciona métodos para trabajar con CSV.
Interfaces/	datatype_interface.py	dataType	Crea interfaz de usuario para seleccionar que datos se están registrando
	saveData_interface.py	saveData	Crea una interfaz que permite seleccionar si guardar o no los datos capturados durante la ejecución de un programa.
	User_interface.py	User	Crea una interfaz que permite seleccionar un perfil de usuario existente o crear uno nuevo para la recogida de datos.
Graphs/	Plot_waves.py	AllDataGraphic	Define funciones para crear gráficos de las ondas cerebrales almacenadas en CSV.
	Graphicator_main.py	Main	Procesa todos los CSV que se le pasen por una ruta de entrada y utiliza la clase AllDataGraphic para crear diversos plots.
Estadistics/	Media.py	-	Comprueba el formato de los datos de distintos archivos CSV y los unifica en un nuevo fichero previo a procesar los datos.
	Correlation.py	Correlation	Lee un fichero CSV y calcula una matriz de correlación. Incorpora además diversas funciones para representar las matrices de correlación.
	Hispograms.py	Histograms	Proporciona métodos para generar histogramas y distribuciones normales
	testKS.py	Test	Incorpora funciones que permiten realizar pruebas estadísticas y análisis comparativos entre diferentes grupos de datos contenidos en un archivo CSV
	Estadistics_main.py	Main	Procesa todos los archivos CSV de una carpeta de entrada y realiza análisis estadísticos y graficaciones utilizando las clases de la carpeta estadistics.

Tabla 11. Tabla resumen con la ruta, fichero y clases empleadas para el diseño del software.

6. RECOGIDA Y ANÁLISIS DE DATOS

6.1 RECOGIDA DE DATOS

Una vez los apartados de adquisición y procesado de la señal están preparados, procedimos con la parte de aplicación. Contamos con el software previamente descrito que permitió un correcto desarrollo de las entrevistas, los análisis y la recogida de datos.

Comenzamos un recogida de los datos exploratoria que utilizaremos como base del estudio. Esta primera toma de contacto nos permite conocer la fiabilidad del aparato, los valores promedios que registra, así como otra información relevante.

Primero de todo, hemos de tener claro que valores almacenaremos, y bajo que circunstancias. Hemos de recordar que las ondas cerebrales son emitidas de manera pasiva por nuestro cerebro en función de las condiciones en las que se encuentre o a las que se someta al usuario.

La idea fue diseñar 5 pruebas cortas, de aproximadamente 1 minuto, que no se hiciesen pesadas para el usuario y nos permitiesen estimular distintas situaciones para recoger todos los valores de ondas posibles.

6.1.1 *Consulta al Departamento de Protección de Datos*

Dadas las características del proyecto, antes de comenzar con la recogida de datos contactamos por correo con el equipo de soporte a la Protección de Datos de la universidad para que evaluaran y analizaran las características y riesgos del proyecto, y nos diesen un criterio y unas pautas sobre como actuar en este caso.

Primero de todo nos comunicaron la necesidad de informar a las personas participantes y que estas firmasen un consentimiento de participación en el TFG. Además, debido a las características, se procedió a una evaluación de impacto para evaluar los posibles riesgos a los que se encontrarían expuestos los datos personales en consecuencia al desarrollo del trabajo. Esta parte fue evaluada y determinada por el DPD, por lo que únicamente les facilitamos la documentación del TFG y esperamos su aprobación. Los informes de consentimiento se añaden en el Anexo III y IV.

6.1.2 *Datos a Capturar*

En esta primera parte, nos centramos en guardar todos los valores que registraba la banda MindFlex: atención, meditación, ondas delta, ondas theta, low alpha, high alpha, low beta, high beta, low gamma, high gamma.

Recordar que las ondas EEG se clasifican de acuerdo a su frecuencia (numero de veces que una onda se repite en un segundo) en cinco diferentes bandas: delta, theta, gamma, alfa y beta. Además, el dispositivo envía valores de atención y meditación calculados en base a algoritmos biométricos. Revisar el apartado de ondas cerebrales 2.2.1 para conocer las especificaciones de cada una.

- Valor calculado de **atención**
- Valor calculado de **meditación**

- **Onda Delta:** u onda del sueño. Su amplitud varía de los 20 a 200 μV , y su rango de frecuencia oscila entre 0,2 y 3,5 Hz.
- **Onda Theta (θ):** su banda de frecuencia está entre los 4 y 8 Hz y su amplitud ronda valores inferiores a 30 μV . Se suele presentar en niños o en adultos que se encuentran sometidos a altos niveles de estrés.
- **Onda Alfa (α):** corresponde al estado de relajación, en donde la persona tiene poca actividad mental o se encuentra con los ojos cerrados. Suelen presentar unas frecuencias de 8 y 12 Hz, una amplitud de entre 20 y 60 μV .
- **Onda Beta (β):** su amplitud se encuentra en el rango entre los 2 y 20 μV y su frecuencia entre los 12 y 30 Hz, está vinculado principalmente a un estado mental de concentración por parte de la persona.
- **Onda Gamma:** Son las frecuencias superiores a 30 Hz, suelen tener la frecuencia más alta y se relacionan con actividades cerebrales intensas.

6.1.3 *Diseño de las Pruebas*

Inicialmente se escogieron 5 pruebas que estimularan todo el rango de las diferentes bandas de frecuencia emitidas por el cerebro humano. Estas pruebas fueron orientadas o directamente reproducidas de estudios específicos para cada una de las frecuencias de interés.

Comenzamos con una prueba de meditación favoreciendo las frecuencias de tipo alfa. El objetivo de comenzar con una prueba de meditación consiste en establecer un punto de partida común para todos los usuarios.

Hemos de tener en cuenta que tanto las situaciones personales, como la hora del día o actividades previas al comienzo de las pruebas son factores claves que pueden alterar los resultados y la producción de las diferentes ondas. Es por eso, que según [5] establecer una meditación inicial nos ayuda a estandarizar los posibles resultados.

PRUEBA 1. MEDITACIÓN

La prueba inicial consiste en situar a la persona en un lugar de su confianza, e invitarlo a que se siente o se tumben y cierre los ojos. La persona participante se centrará en realizar respiraciones profundas y comenzará con una cuenta atrás de 50 a 0.

Paralelamente, el entrevistador reproducirá música calmada que induzca aún más al usuario a un estado de relajación.

PRUEBA 2. MEMORY TASK

Una vez establecido el punto de partida, comenzaremos con una tarea de memoria de trabajo. En esta prueba, el entrevistador pronunciará un conjunto de dígitos de tamaño creciente (rango: 2 -9) a un ritmo de un dígito por segundo, y los participantes tendrán que repetirlos verbalmente en el orden correcto. Podemos encontrar el conjunto de dígitos exactos en las pautas para el desarrollo de la entrevista en el anexo V.

En una segunda fase de esta misma prueba, los participantes tendrán que repetir otros conjuntos de números completamente diferentes en orden inverso al dictado por el entrevistador.

PRUEBA 3. UP-DOWN

El objetivo de esta prueba es recrear el diseño inicial del juguete MindFlex, simulando un movimiento ascendente y descendente de una pelota de unos 5cm de diámetro con la “*mente del usuario*”. En esta prueba buscamos observar las oscilaciones entre los valores de atención y de meditación de la propia banda.

El entrevistador situará una pequeña pelota entre sus dedos índice y pulgar frente al participante, e invitará al usuario a hacerla levitar con su mente. Se especificará a los participantes que no pierdan contacto visual con la pelota durante el momento del ascenso. Una vez la pelota ha llegado al punto más alto, el usuario tendrá que perder contacto visual con la propia pelota e iniciar un periodo de relajación de unos pocos segundos.

El entrevistador realizará el movimiento ascendente muy lentamente, hasta un tope previamente marcado, y una vez arriba del todo hará de nuevo descender la pelota hasta el límite inferior mientras el usuario pierde contacto con esta.

Se realizará todo el procedimiento up-down dos veces continuadas, donde los participantes repetirán el mismo ejercicio de la misma manera.

PRUEBA 4. CORTO SEGUNDITO

En esta última prueba se busca estimular las ondas beta o de concentración, manteniendo un estado de atención. Escogimos un video 1’54’’ minutos [23], con el objetivo de mantener una atención constante por parte del usuario, sin que este perdiese la concentración.

El desarrollo de la prueba es sencillo, pues únicamente consiste en la reproducción del video por parte del entrevistador y la visualización de este mismo por parte del participante .



Figura 23. Video utilizado durante una de las pruebas de las entrevistas. Fuente: <https://www.youtube.com/watch?v=0zsE85khG6I>

6.1.4 Diseño de las Entrevistas

Escogimos las entrevistas como método de recopilación de datos ya que consideramos primordial que el entrevistador guiase al entrevistado en todo momento. Uno de los factores clave fue buscar un entorno familiar y de confianza para los usuarios, puesto que debido al carácter del trabajo y los datos de interés, las condiciones externas podían afectar al resultado (variaciones en las ondas cerebrales).

Destacar que se hizo entrega de un consentimiento informado al participante al comienzo de la entrevista. Anexos III y IV.

Para la planificación de las entrevistas, tendremos en cuenta los siguientes puntos clave:

- **Objetivo de la recopilación de datos.**

El objetivo principal es obtener valores de ondas cerebrales relacionados con distintas actividades.

- **Participantes.**

Los participantes fueron personas de un entorno cercano con las que pudimos contactar de manera directa. No consideramos relevante establecer unas condiciones específicas como la edad o el sexo, pues debido al carácter de los datos de interés, estos factores se consideraron irrelevantes.

- **Conjunto de pruebas.**

El conjunto de pruebas que se llevaron a cabo se especifican en el apartado 6.1.3

- **Preparar documentación a entregar.**

Al comienzo de la entrevista todos los participantes fueron provistos de un consentimiento informado el cual firmaron al comienzo. Se firman dos copias en cada caso, una para el participante y una segunda para el entrevistador.

Además, se les facilitó un PDF con imágenes y fotografías que permitiría al usuario entender algunos de los conceptos clave que se explicarían durante el trascurso de la entrevista.

- **Verificar un equipo de grabación**

Todas las entrevistas fueron grabadas utilizando equipo propio, iPhone 11 y iPad 2019, bajo el consentimiento verbal de los participantes.

- **Organizar horario y lugar**

Como se comentaba en el punto “*participantes*”, todos los usuarios fueron personas de un entorno conocido, por lo que las entrevistas pudieron ser organizadas específicamente para cada una de las personas con menos de 24h de antelación. Además, todas fueron llevadas a cabo en sus casas u habitaciones individuales, donde se habilitó una zona específica para el equipo.

Establecimos aproximadamente las 12 del medio día como horario de referencia para eliminar posibles variaciones en el resultado debido al paso del día, cansancio u estrés acumulado de los participantes. Esto es debido a que, como se comentaba anteriormente, la hora de captura, el lugar o las condiciones externas podían afectar al resultado de las ondas emitidas.

6.2 ANÁLISIS EXPLORATORIO DE LOS DATOS

Un análisis exploratorio de datos es una técnica en ciencia de datos que permite comprender conjuntos de datos antes de realizar cualquier análisis estadístico más avanzado. El objetivo principal es resumir las características y encontrar las relaciones principales utilizando técnicas visuales y estadística descriptiva.

En este caso, trabajaremos con datos previamente limpiados tras las entrevistas. Se eliminaron valores faltantes o erróneos en el conjunto de datos, así como resultados de las entrevistas no deseados.

Además, se descartaron datos capturados durante interrupciones en las entrevistas o valores cuyas características durante la captura de datos no eran de interés. Por ejemplo, una llamada de teléfono durante la recogida de datos, o un ruido muy fuerte en un momento durante una de las entrevistas podrían alterar la emisión de ondas del propio usuario.

De esta manera, garantizamos que los datos sean lo más representativos y precisos posibles, lo que nos permite comprender mejor las relaciones y patrones presentes en los datos.

6.2.1 Valores de Partida y Objetivo

Se dispone de los datos de seis usuarios diferentes de edades comprendidas entre 20 y 23 años, tres hombres y tres mujeres. Se realizaron las pruebas descritas en el apartado 6.1.3 y almacenaron en una base de datos en archivos individuales identificados con una secuencia alfanumérica única para cada usuario.

El objetivo de esta parte del proyecto fue estudiar las diferencias entre patrones de ondas y su relación con la actividad que se estaba llevando a cabo, de modo que pudiésemos predecir o identificar características únicas para el estado mental de las personas en cada situación. Estructuramos el análisis en tres partes:

1. Identificación de las correlaciones de los valores de entrada y la validez de la lectura de las señales.
2. Para cada uno de los diferentes experimentos, explorar la variación de los valores o distribuciones de ondas.
3. Para cada usuario, determinar las posibles diferencias o similitudes en los patrones para un mismo tipo de onda.

6.2.2 Identificación de Correlaciones y Validez de las Señales de Entrada

En este apartado el objetivo es investigar las correlaciones entre los valores de entrada, como la atención y la meditación, y las bandas de frecuencia.

La correlación entre dos variables estadísticas se define como – *“la medida de la fuerza y dirección de una relación lineal y proporcionalidad entre ellas”*. Se establece que existe correlación cuando los valores de una variable varían de manera sistemática en relación a los valores correspondientes de la otra variable. Esto implica verificar si dos mediciones capturadas están realmente reflejando los estados mentales de los usuarios y varían con una misma tendencia.

Esperamos encontrar correlaciones significativas entre las medidas de atención/meditación y las actividades específicas realizadas por los usuarios. Por ejemplo, una alta correlación

entre las ondas beta/gamma y tareas de alta concentración, y entre las ondas alfa/theta y estados de meditación.

EJEMPLO: Cálculo de la correlación

Cogeremos de ejemplo los valores de meditación para atención (at) y meditación (me),

1. Primero calcularemos el sumatorio apra cada uno de los grupo de valores.

$$\begin{matrix} \Sigma at & \Sigma at \cdot me & \Sigma me^2 \\ \Sigma me & \Sigma at^2 & \end{matrix} \quad (1)$$

2. Aplicamos la formula de la correlación de Pearson

$$r = \frac{n(\Sigma at \times me) - (\Sigma at)(\Sigma me)}{\sqrt{[n\Sigma at^2 - (\Sigma at)^2] \times [n\Sigma me^2 - (\Sigma me)^2]}} \quad (2)$$

Para esta premisa, separaremos las diferentes actividades evaluadas durante las entrevistas y determinaremos las correlaciones entre cada par de variables. Se escogió representar los valores en formato de tabla y heatmap que permitiese una correcta visualización de los resultados. Los gráficos de calor pueden encontrarse en el anexo VII.

6.2.2.1 Resultados

MEDITACIÓN

Durante esta actividad el usuario mantiene una alta concentración en la realización de una cuenta atrás mientras descansa. Se reprodujo además música que permitiese al usuario entrar en un estado de relajación.

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
attention	1.0000							
meditation	0.8639*	1.0000						
delta	0.3008*	0.2070*	1.0000					
theta	-0.3507*	-0.2051*	0.0100	1.0000				
low alpha	-0.2324*	-0.1310*	0.0410	0.4722*	1.0000			
high alpha	-0.3342*	-0.2436*	0.0351	0.7306*	0.4923*	1.0000		
low beta	-0.3514*	-0.3108*	0.1677*	0.4698*	0.4498*	0.5200*	1.0000	
high beta	-0.3758*	-0.2888*	0.0361	0.6736*	0.5083*	0.6827*	0.6914*	1.0000

Tabla 12. Tabla de correlación de los pares de variables para la actividad de meditación.

En este caso, destacaremos una correlación positiva fuerte entre los estados de atención y meditación, lo que indica que estos dos estados mentales están altamente relacionados. A medida que aumenta la atención, también lo hace la meditación. Este supuesto coincide con los valores que podríamos esperar de los resultados de esta prueba.

Además, podemos destacar relaciones entre distintos pares de frecuencias como pueden ser la Theta y High Alpha, o High Beta y Low Beta que nos arroja información sobre como estas ondas se relacionan entre si. En este caso, no se aprecian correlaciones negativas relevantes.

MEMORY 1 Y MEMORY 2

Para la actividad de memoria, se hizo por duplicado, de modo que se espera que los valores de ambas tablas sean similares.

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
<i>attention</i>	1.0000							
<i>meditation</i>	0.7503*	1.0000						
<i>delta</i>	0.2096*	0.1701*	1.0000					
<i>theta</i>	-0.1522*	0.0354	0.2750*	1.0000				
<i>low alpha</i>	-0.1291*	0.0754	0.1815*	0.5839*	1.0000			
<i>high alpha</i>	-0.1525*	0.0421	0.2371*	0.6852*	0.5195*	1.0000		
<i>low beta</i>	-0.0745	0.1128*	0.2443*	0.5407*	0.5233*	0.5731*	1.0000	
<i>high beta</i>	-0.0785	0.1423*	0.2752*	0.6982*	0.6147*	0.7644*	0.6886*	1.0000

Tabla 13. Tabla de correlación de los pares de variables para la actividad de memory1.

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
<i>attention</i>	1.0000							
<i>meditation</i>	0.6534	1.0000						
<i>delta</i>	-0.1195*	-0.1082	1.0000					
<i>theta</i>	0.0176*	0.0215	0.3177*	1.0000				
<i>low alpha</i>	0.0689*	0.1356*	0.2527*	0.5678*	1.0000			
<i>high alpha</i>	0.0730*	0.1618*	0.1687*	0.6520*	0.6124*	1.0000		
<i>low beta</i>	0.1464	0.1645*	0.2351*	0.5925*	0.6398*	0.6392*	1.0000	
<i>high beta</i>	0.1508	0.1758*	0.2304*	0.6841*	0.6652*	0.6795*	0.7094*	1.0000

Tabla 14. Tabla de correlación de los pares de variables para la actividad de memory2.

Podemos ver de nuevo que, para los valores de atención y meditación, ambas condiciones muestran una fuerte correlación positiva, aunque ligeramente menor en Memory 2 en comparación con Memory 1. Este suceso puede deberse a modificaciones en las condiciones externas al usuario durante el proceso de la entrevista.

Podemos apreciar también en las ondas high alpha y high beta una fuerte correlación, mostrando de nuevo una correlación ligeramente superior en Memory 1 frente a Memory 2.

UP-DOWN

Las pruebas Up-Down intentan simular el propósito inicial de las bandas MindFlex, intentando simular los cambios de atención y relajación de los usuarios. Cabe destacar que, durante la realización de esta prueba, muchos de los usuarios no terminaron de comprender el propósito y realización de la misma, debido a que no disponían de un feedback que correlacionase su estado mental con la acción que estaban realizando.

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
<u>attention</u>	1.0000							
<u>meditation</u>	0.5781*	1.0000						
<u>delta</u>	-0.3741*	-0.3414	1.0000					
<u>theta</u>	-0.1992*	0.0835	0.4714*	1.0000				
<u>low alpha</u>	-0.0648	0.1254*	0.4497*	0.6739*	1.0000			
<u>high alpha</u>	-0.1742*	0.1727*	0.3288*	0.7201*	0.5680*	1.0000		
<u>low beta</u>	-0.1378*	0.1721*	0.3763*	0.7982*	0.5583*	0.7120*	1.0000	
<u>high beta</u>	-0.0665	0.1652*	0.2954*	0.7625*	0.6528*	0.6647*	0.8318*	1.0000

Tabla 15. Tabla de correlación de los pares de variables para la actividad UP.

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
<u>attention</u>	1.0000							
<u>meditation</u>	0.7948*	1.0000						
<u>delta</u>	-0.3005*	-0.0571	1.0000					
<u>theta</u>	0.1167	0.1476	0.5123*	1.0000				
<u>low alpha</u>	0.1275	0.2723	0.3873*	0.6950*	1.0000			
<u>high alpha</u>	0.1579	0.1707	0.4394*	0.8900*	0.7866*	1.0000		
<u>low beta</u>	0.2095	0.2372	0.4403*	0.7699*	0.7980*	0.8378*	1.0000	
<u>high beta</u>	0.1856	0.2059	0.4451*	0.7844*	0.5052*	0.7492*	0.7348*	1.0000

Tabla 16. Tabla de correlación de los pares de variables para la actividad DOWN.

Las correlaciones entre las condiciones UP y DOWN indican patrones consistentes en varias áreas y diferencias clave en otras. Podemos destacar principalmente en este caso las relaciones fuertes entre atención y meditación en DOWN, frente a una disminución de esta para el caso de UP. Como era de esperar, los valores de UP estaban pensados para estimular la atención y disminuir los valores de meditación.

Además, en este caso apreciamos también unas correlaciones muy altas para el caso de las ondas theta y high alpha, así como para high y low beta.

VIDEO SEGUNDITO

	attention	meditation	delta	theta	low alpha	high alpha	low beta	high beta
attention	1.0000							
meditation	0.1590*	1.0000						
delta	0.0573	-0.0310	1.0000					
theta	-0.3379*	0.1577	0.3517*	1.0000				
low alpha	-0.2206*	0.2181*	0.3958*	0.5463*	1.0000			
high alpha	-0.1959*	0.2296*	0.3815*	0.7151*	0.4841*	1.0000		
low beta	-0.1151*	0.1778*	0.4394*	0.6306*	0.5005*	0.6664*	1.0000	
high beta	-0.0888	0.2211*	0.4636*	0.7028*	0.5633*	0.7151*	0.7324*	1.0000

Tabla 17. Tabla de correlación de los pares de variables para la actividad de segundito.

La correlación positiva muy débil entre los valores de atención y meditación en este caso sugiere que pueden aumentar ligeramente juntas, pero la relación para esta actividad no es significativa. Las ondas theta y alpha altas están fuertemente correlacionadas, lo que puede indicar un estado de relajación combinado con un procesamiento cognitivo moderado.

De nuevo, podemos confirmar que estos valores son consecuentes con el tipo de actividad llevadas a cabo.

6.2.2.2 Conclusiones

A continuación, se muestran las correlaciones más destacadas y de las que podemos extraer algunas conclusiones. Podemos ver gráficas con los valores más relevantes en el anexo IV.

Condición	Attention-Meditation	Attention-Theta	Theta-High Alpha	High Alpha-High Beta	Low Alpha-Theta	Low Beta-Theta	Delta-Meditation
Meditación	0.5781	-0.3379	0.7151	0.7151	0.5463	0.6306	-0.0310
Memory 1	0.7503	-0.1522	0.6852	0.7644	0.5839	0.5407	0.1701
Memory 2	0.6534	0.0176	0.6520	0.6795	0.5678	0.5925	-0.1082
UP	0.1590	-0.3379	0.7151	0.7151	0.5463	0.6306	-0.0310
DOWN	0.7948	0.1167	0.8900	0.7492	0.6950	0.7699	-0.0571
Segundito	0.1590	-0.3379	0.7151	0.7151	0.5463	0.6306	-0.0310

Tabla 18. Tabla resumen de los pares de correlaciones más relevantes para las distintas pruebas realizadas

Para los valores de atención y meditación, podemos observar una fuerte relación entre las variables en muchas de las pruebas llevadas a cabo, y apenas correlación en otras. Esto se puede concluir en que realmente si existe una relación entre el contexto y la naturaleza de la tarea desempeñada, sus valores de atención y meditación. Por ejemplo, para los resultados

de UP (0.5781) frente a DOWN (0.7948) podemos ver como el cambio de atención supone un factor clave durante estas actividades. Esto queda reflejado apreciar en su diferencia de valores de ondas para estos casos.

Las correlaciones entre las diferentes bandas de frecuencia proporcionan una visión sobre cómo la actividad cerebral se ajusta durante esta tarea específica. Estas observaciones pueden ser útiles para entender cómo diferentes tipos de estados afectan la actividad cerebral. La fuerte correlación entre theta y high alpha, así como entre high alpha y high beta, es notablemente consistente y podría ser un indicador clave para el desarrollo de aplicaciones BCI.

6.2.3 Exploración de la Variación de los Valores o Distribuciones de Ondas para Diferentes Experimentos

En esta etapa, nos enfocamos en explorar cómo varían los valores de atención, meditación, y varias bandas de frecuencia de las ondas cerebrales (delta, theta, low alpha, high alpha, low beta, high beta) en los diferentes experimentos.

La distribución de datos nos indica la manera en la que se distribuyen los valores en un rango determinado. Esto nos proporciona información sobre la forma, dispersión y tendencia de los valores. Esto nos permite comprender cómo cambian las características de las ondas en respuesta a diferentes actividades o estímulos.

En este caso, calculamos los valores de estadísticas descriptivas y representamos los datos utilizando gráficos de dispersión e histogramas que nos ayudasen a reconocer las variaciones existentes de datos. Podemos encontrar la representación de todos los histogramas en el anexo VIII. Finalmente, realizaremos la prueba de Kolmogorov-Smirnov (KS) para determinar la comparativa entre pares de muestras [22].

EJEMPLO: Cálculo de las estadísticas descriptivas.

Cogeremos de ejemplo los valores de meditación para atención (a).

- **Media**

$$\mu = \frac{1}{N} \sum_{i=1}^N a_i \quad (3)$$

Donde N es el número de valores registrados y a_i es cada valor individual de atención

- **Desviación Estándar**

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (a_i - \mu)^2} \quad (4)$$

EJEMPLO: Cálculo Estadístico KS

El valor p de la prueba KS se calcula utilizando la distribución de Kolmogorov para determinar la probabilidad de observar una diferencia tan extrema como $D_{n,m}$ bajo la hipótesis nula de que ambas muestras siguen la misma distribución.

- **Ordenación de las muestras.** Antes de comenzar ordenaremos los valores de las dos muestras.
- **Cálculo de las ECDFs³:**
Para dos muestras independientes $X = \{ X_1, X_2, \dots, X_n \}$ y $Y = \{ Y_1, Y_2, \dots, Y_m \}$ las ECDF $F_n(x)$ y $G_m(y)$ se definen como:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{X_i \leq x\}} \quad (5)$$

Donde $\mathbf{1}_{\{X_i \leq x\}}$ es una función que nos indica que vale 1 si $\{X_i \leq x\}$ y 0 en el caso contrario.

$$G_m(y) = \frac{1}{m} \sum_{j=1}^m \mathbf{1}_{\{Y_j \leq y\}} \quad (6)$$

Donde $\mathbf{1}_{\{Y_j \leq y\}}$ es una función que nos indica que vale 1 si $\{Y_j \leq y\}$ y 0 en el caso contrario.

- **Encontrar la máxima diferencia absoluta:**

$$D_{n,m} = \max |F_n(x) - G_m(x)| \quad (7)$$

6.2.3.1 Estadística Descriptiva

ATTENTION

	Mean	Std Dev	Min	Max
Meditation	41.15	37.44	0.0	100.0
Memory1	41.12	27.81	0.0	97.0
Memory2	47.76	22.55	1.0	100.0
UP	49.12	17.42	1.0	88.0
DOWN	42.53	16.05	1.0	67.0
Segundito	59.46	20.80	27.0	100.0

Tabla 19. Tabla con los valores estadísticos promedios para el valor de atención durante todas las actividades.

MEDITATION

	Mean	Std Dev	Min	Max
Meditation	37.57	31.54	0.0	87.0
Memory1	44.39	28.79	0.0	93.0
Memory2	51.05	17.93	1.0	94.0
UP	58.21	21.86	1.0	96.0
DOWN	59.53	20.41	11.0	96.0
Segundito	60.58	13.02	1.0	93.0

Tabla 20. Tabla con los valores estadísticos promedios para el valor de meditación durante todas las actividades.

³ Empirical Cumulative Distribution Function

ONDAS DELTA

	Mean	Std Dev	Min	Max
Meditation	37.57	31.54	0.0	87.0
Memory1	44.39	28.79	0.0	93.0
Memory2	51.05	17.93	1.0	94.0
UP	58.21	21.86	1.0	96.0
DOWN	59.53	20.41	11.0	96.0
Segundito	60.58	13.02	1.0	93.0

Tabla 21. Tabla con los valores estadísticos promedios para el valor de onda delta durante todas las actividades.

ONDAS THETA

	Mean	Std Dev	Min	Max
Meditation	387404.45	438942.31	3,241	1,919,861
Memory1	443756.05	470957.91	8,628	2,579,112
Memory2	354193.96	478485.54	2,766	2,716,078
UP	468407.78	590944.47	4,652	2,559,328
DOWN	424924.38	495806.20	7,713	1,947,973
Segundito	380099.83	492132.26	1,907	2,437,801

Tabla 22. Tabla con los valores estadísticos promedios para el valor de onda theta durante todas las actividades.

HIGH ALFA

	Mean	Std Dev	Min	Max
Meditation	108069.41	147899.77	508	747235
Memory1	115157.63	158587.94	397	765291
Memory2	75425.21	132677.44	346	934969
UP	133064.61	205387.82	648	1094339
DOWN	85742.85	157376.91	1199	697554
Segundito	89049.74	121091.36	148	666296

Tabla 23. Tabla con los valores estadísticos promedios para el valor de onda high alfa durante todas las actividades.

LOW ALFA

	Mean	Std Dev	Min	Max
Meditation	99120.27	151978.78	402	1164700
Memory1	112982.12	164097.04	152	1082709
Memory2	84263.89	140605.01	223	912858
UP	140674.90	247922.02	500	1678274
DOWN	92079.32	112017.07	2099	553926
Segundito	110032.78	171003.22	231	1346552

Tabla 24. Tabla con los valores estadísticos promedios para el valor de onda low alfa durante todas las actividades.

HIGH BETA

	Mean	Std Dev	Min	Max
Meditation	37293.00	42425.87	183.0	264172.0
Memory1	34823.49	41033.15	557.0	232059.0
Memory2	26284.29	44962.04	317.0	328382.0
UP	60263.24	100573.42	257.0	545691.0
DOWN	42565.00	74647.33	333.0	373928.0
Segundito	30196.24	35886.42	656.0	238257.0

Tabla 25. Tabla con los valores estadísticos promedios para el valor de onda high beta durante todas las actividades.

LOW BETA

	Mean	Std Dev	Min	Max
Meditation	49121.25	65679.66	856.0	536180.0
Memory1	45183.82	63757.24	482.0	390425.0
Memory2	32263.14	51587.58	253.0	382014.0
UP	65111.45	98775.23	410.0	612457.0
DOWN	50983.26	79576.54	151.0	405761.0
Segundito	50091.87	64419.69	594.0	441362.0

Tabla 26. Tabla con los valores estadísticos promedios para el valor de onda low beta durante todas las actividades.

Analizando el conjunto de los valores para las distintas actividades llevadas a cabo podemos concluir en que ciertas actividades muestran unos valores específicos que se correlacionan con la lógica esperada. Por ejemplo, ver el video "Segundito" es la actividad que más captura la atención del usuario, con una media de 59.46 y una variabilidad casi nula. Además, induce el estado de meditación más profundo, con una media de 60.58. Estos resultados son

coherentes con lo que se podría esperar de una actividad centrada en el enfoque y visualmente estimulante.

Durante las tareas altamente cognitivas, las ondas theta y alfa presentan una variabilidad significativa, lo que sugiere que dichas actividades tienen un impacto importante en estos estados cerebrales. Además, las ondas beta, tanto en su forma alta como baja, son más elevadas durante las actividades que implican un aumento de la concentración.

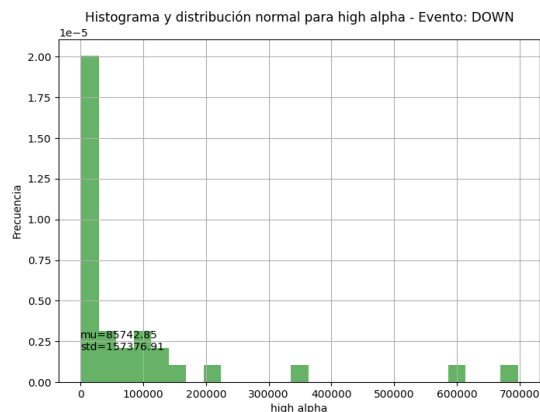
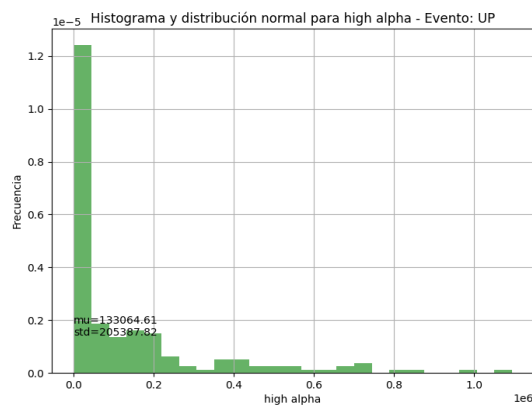
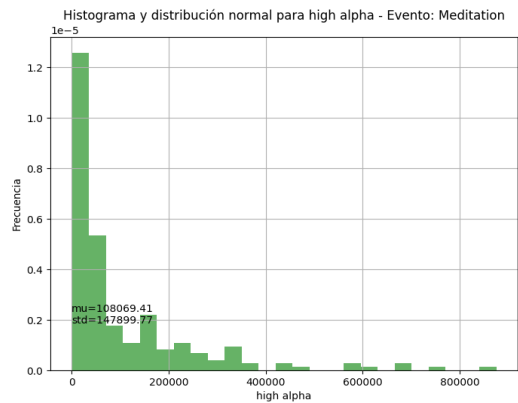
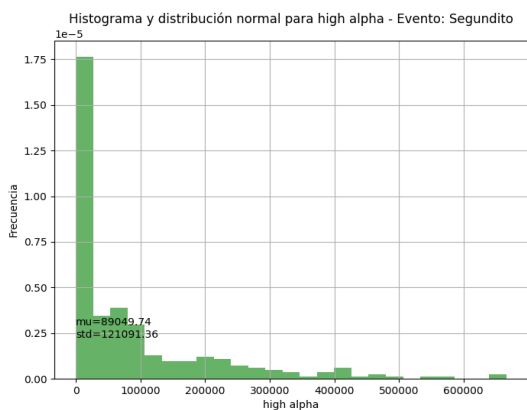
Para los valores de atención, las actividades de "Meditation" y "Memory1" tienen medias similares (aproximadamente 41), pero la actividad "Meditation" muestra una mayor dispersión (desviación estándar de 37.44) que podría reflejar diferentes niveles de habilidad o de esfuerzo entre los participantes en mantener la atención, coherente con las pruebas.

En contraste, los ejercicios de memoria presentan medias más bajas y una mayor variabilidad en ambos estados de atención y meditación, lo que podría indicar una respuesta menos uniforme a estas tareas cognitivas en comparación con otras actividades más relajantes o visualmente dirigidas.

6.2.3.2 Histogramas

Decidimos representar con histogramas la distribución de los valores para las distintas actividades. En este caso, obtuvimos que todas las ondas seguían patrones prácticamente idénticos, de manera que a simple vista no se podían apreciar variaciones significativas consecuentes al tipo de actividad llevada a cabo.

A continuación, se muestran seis histogramas con la representación de los valores capturados para la banda de frecuencia High Alpha y los distintos eventos llevados a cabo



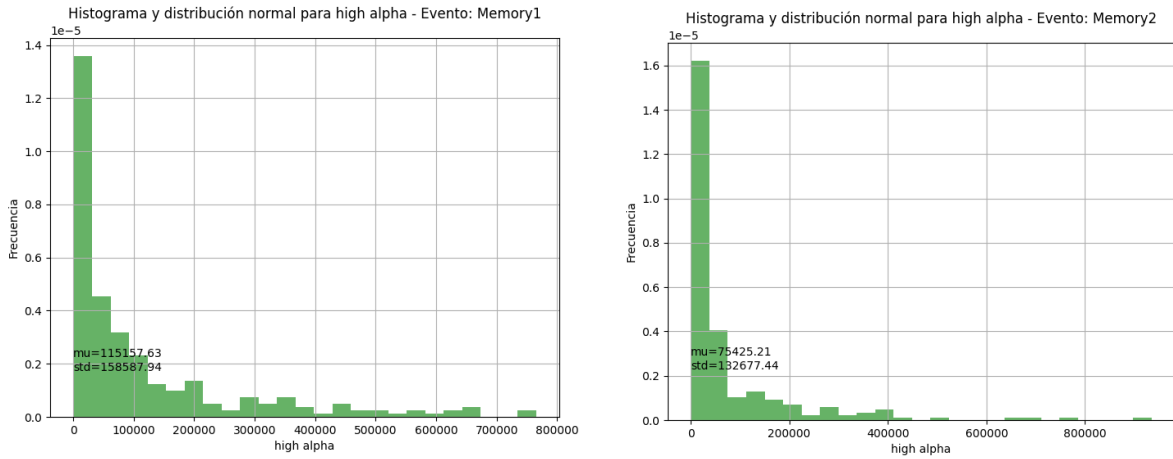
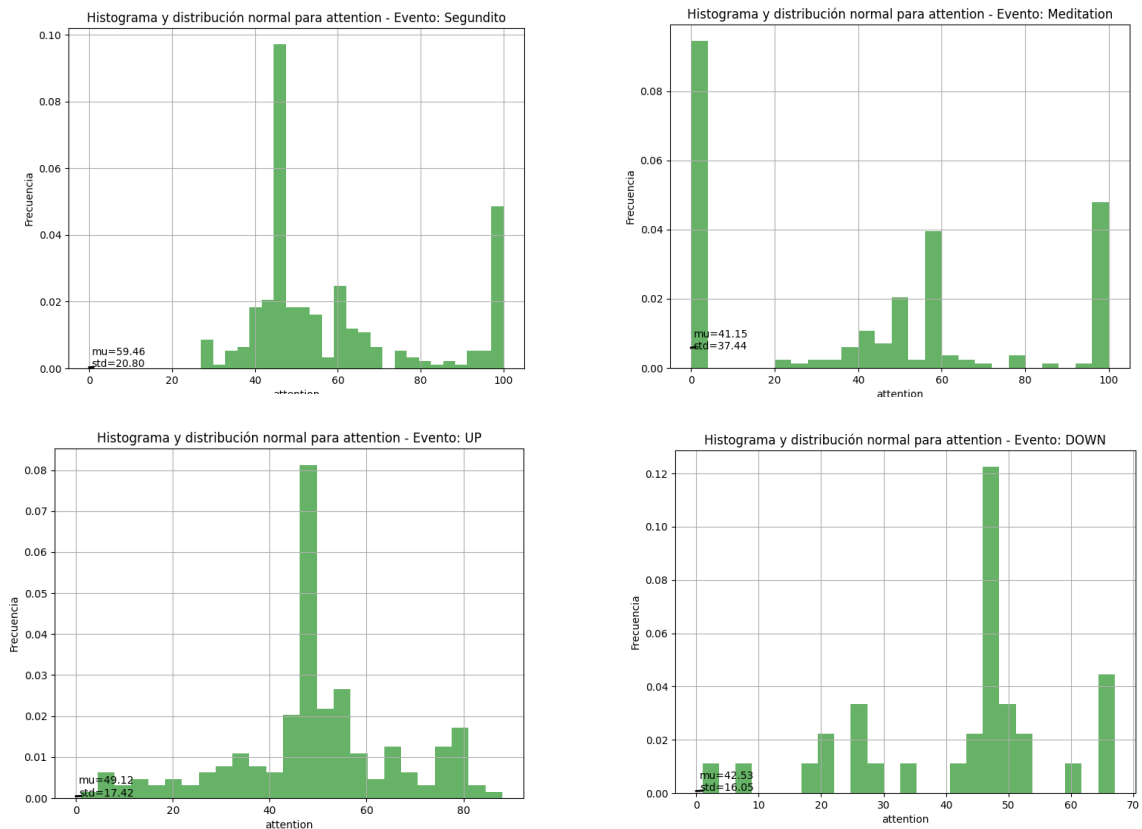


Figura 25. Histogramas de distribución para la banda de frecuencia high alfa para los 6 eventos capturados.

Como se puede apreciar, aunque existe cierta variación en los valores, la misma onda sigue patrones muy similares para todos los eventos. Esto mismo ocurre para todo el resto de las bandas de frecuencia.

Por otro lado, para los valores de atención y meditación, si que existen diferencias significativas y observables para los distintos eventos. Esto se correlaciona con las diferencias observadas en los análisis previos.

ATTENTION



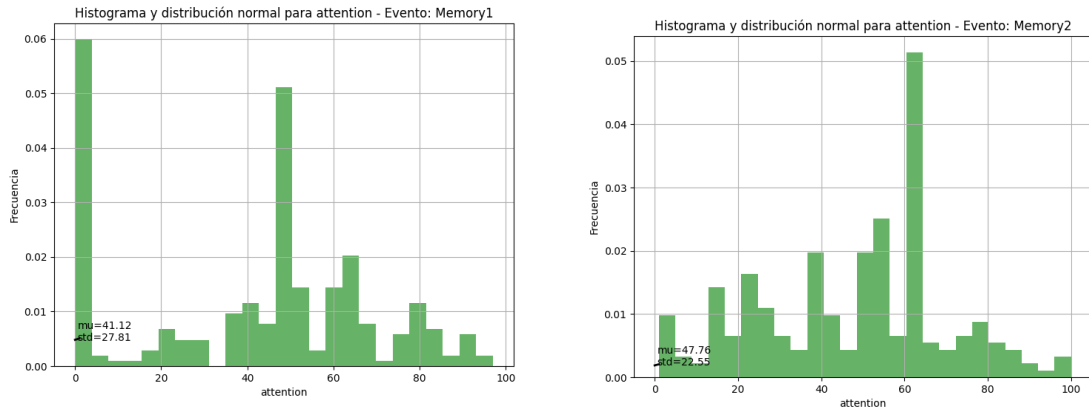


Figura 26. Histogramas de distribución para el valor de *attention* para los 6 eventos capturados.

MEDITATION

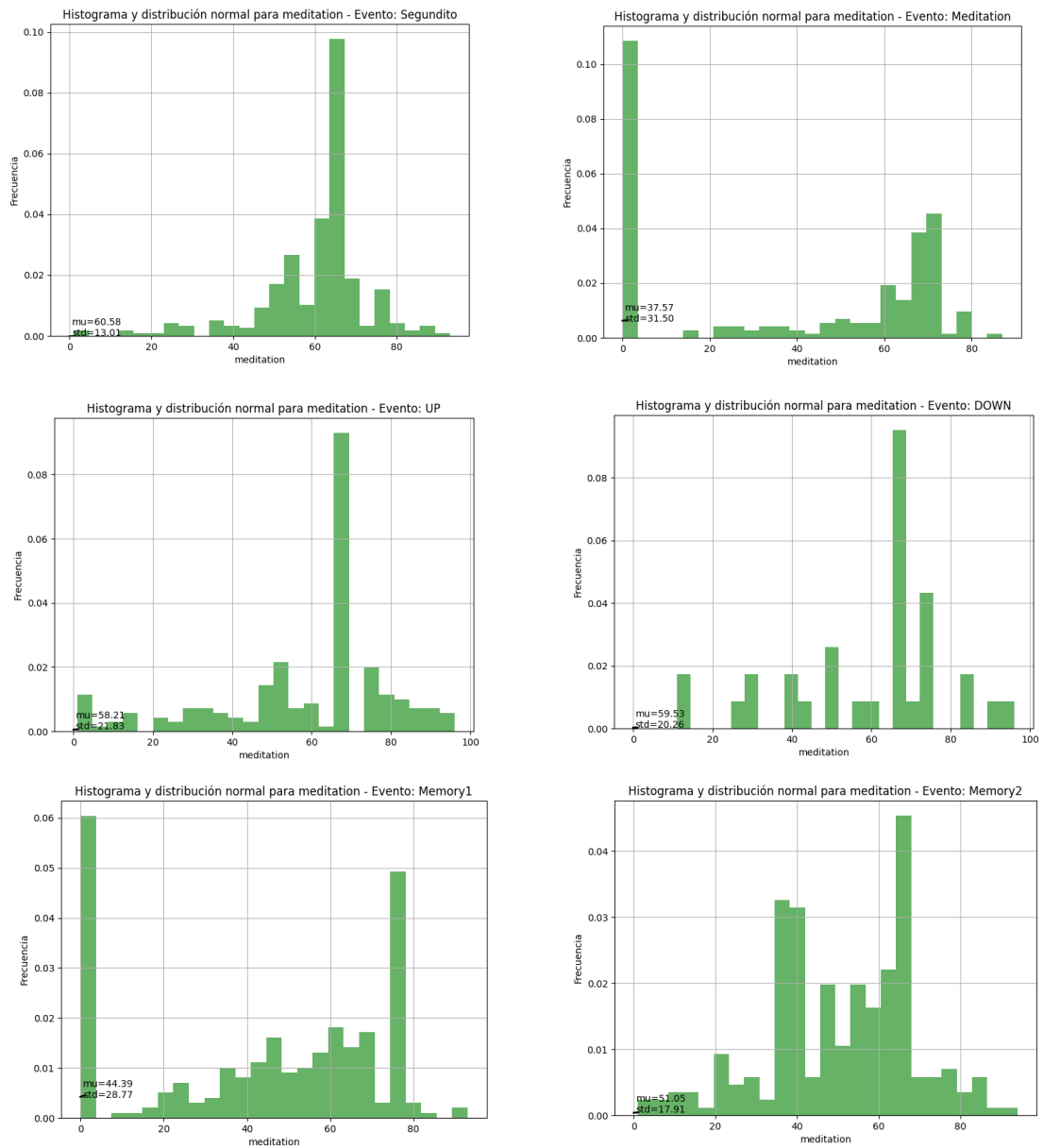


Figura 27. Histogramas de distribución para el valor de *meditation* para los 6 eventos capturados.

6.2.3.3 Prueba de Kolmogorov-Smirnov (KS)

La prueba de Kolmogorov-Smirnov (KS) es una herramienta que nos permite comparar dos muestras y determinar si provienen de la misma distribución. Esta prueba nos permitirá determinar si las bandas de frecuencia siguen o no una distribución similar en función de la actividad realizada, y si pueden ser utilizadas para determinar los estados mentales del usuario.

Por otra parte, analizaremos si existen diferencias significativas para los valores de atención y meditación en las distintas actividades. Esto podría significar que, si existe una correlación entre la actividad llevada a cabo y los valores de atención y meditación, de modo que estos podrían ser utilizados para las posteriores aplicaciones.

Relación	Variable	Estadístico KS	Valor p
Meditation - Memory1	Attention	0.1887*	8.8051e-08
	Meditation	0.1742*	1.0740e-06
Meditation - Memory2	Attention	0.3780*	1.8494e-28
	Meditation	0.3780*	1.8494e-28
Meditation - UP	Attention	0.3780*	1.9211e-25
	Meditation	0.3780*	1.9211e-25
Meditation - DOWN	Attention	0.3780*	5.3490e-08
	Meditation	0.3780*	5.3490e-08
Meditation - Segundito	Attention	0.3876*	2.6902e-34
	Meditation	0.3929*	2.9570e-35
Memory1 - Memory2	Attention	0.2247*	1.8524e-11
	Meditation	0.2247*	1.8524e-11
Memory1 - UP	Attention	0.2343*	5.5194e-11
	Meditation	0.2911*	8.7181e-17
Memory1 - DOWN	Attention	0.2639*	3.3776e-04
	Meditation	0.3218*	4.7255e-06
Memory1 - Segundito	Attention	0.3034*	4.6243e-24
	Meditation	0.3564*	3.1434e-33
Memory2 - UP	Attention	0.1628*	3.2214e-05
	Meditation	0.3056*	2.1000e-17
Memory2 - DOWN	Attention	0.2903	6.4677e-05
	Meditation	0.3363**	1.7902e-06
Memory2 - Segundito	Attention	0.2919*	1.3771e-20
	Meditation	0.3402*	6.0853e-28
UP - DOWN	Attention	0.1953	0.0213
	Meditation	0.0745*	0.8808
UP - Segundito	Attention	0.1775*	6.8128e-07
	Meditation	0.1482*	6.2149e-05
DOWN - Segundito	Attention	0.2851*	6.5659e-05
	Meditation	0.1701	0.0502

Tabla 27. Tabla de resultados para las relaciones entre distintas pruebas, su valor KS calculado y el p-value.

Los resultados refuerzan la idea de que las medidas de atención y meditación son sensibles a diferentes tipos de estímulos y actividades. Se observan valores de p muy bajos ($p < 0.001$) en la mayoría de los casos, lo cual señala diferencias estadísticamente significativas. Las mayores diferencias se muestran al contrastar las actividades de meditación con otras tareas, así como entre los ejercicios de memoria y distintas actividades. Estos hallazgos sugieren

que diversas actividades tienen un impacto notable en la atención y el estado de meditación del individuo.

6.2.3.4 Conclusiones

A partir del análisis de las estadísticas descriptivas, la representación de los histogramas y el cálculo de la prueba de Kolmogorov-Smirnov (KS), se ha observado que las bandas de frecuencia de las ondas cerebrales presentan variaciones significativas y poco concluyentes.

Estas variaciones parecen estar influenciadas por factores externos a las pruebas realizadas, lo que sugiere la presencia de ruido, interferencias externas o factores personales que afectan la consistencia de las mediciones. Aunque los valores p obtenidos en la prueba KS indican diferencias significativas entre las distintas condiciones, tanto las estadísticas descriptivas y los histogramas muestran una dispersión no controlada en los datos de las distintas pruebas. Por lo tanto, debido a la influencia de estas variaciones no controladas, no es posible determinar un estado concreto y confiable de las ondas cerebrales basado en las condiciones de las pruebas actuales.

No obstante, en el caso de las variables de atención y meditación, los análisis realizados sí muestran una relación clara entre la actividad llevada a cabo y los resultados obtenidos. Estas variables reflejan cambios significativos que corresponden con las diferentes actividades, lo que sugiere que, a pesar de las variaciones externas, la atención y la meditación son indicadores más robustos y sensibles a las condiciones experimentales.

6.2.4 Determinación de Diferencias o Similitudes en los Patrones de Ondas para un Mismo Tipo de Onda entre Diferentes Usuarios

El objetivo en este apartado es analizar las diferencias o similitudes en los patrones de ondas entre diferentes usuarios para un mismo tipo de onda. Esto nos ayudará a identificar si existe características únicas para el estado mental de cada usuario y entender la diversidad en los patrones de ondas entre individuos.

Para esto, realizamos test KS entre la relación de cada par de ondas cerebrales y cada una de las actividades llevadas a cabo, entre los distintos usuarios. A continuación, se muestra un ejemplo de la realización de estos análisis para dos usuarios concretos: MOR y AFB.

6.2.4.1 Resultados para MOR y AFB

MEDITATION

Measurement	KS Statistic	P-value
Attention	0.4899	0.0000
Meditation	0.7000	0.0000
Delta	0.1585	0.5784
Theta	0.5612	0.0000
Low Alpha	0.6298	0.0000
High Alpha	0.7686	0.0000
Low Beta	0.6899	0.0000
High Beta	0.7686	0.0000

Tabla 28. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad Meditation, en relación a cada una de las ondas capturadas.

En la actividad de meditation, aunque muchas de las medidas mostraron diferencias significativas, Delta mostró una diferencia mucho menos pronunciada (estadística KS de 0.1585 y valor P de 0.5784), lo que podría sugerir que durante la meditación, la actividad en la banda de ondas Delta tiende a ser más similar entre los usuarios comparados con otras bandas de frecuencia. Esto tiene implicación en los valores de atención ya que, como explicábamos anteriormente, este valor se calcula a partir de la magnitud de las ondas delta en relación con las alfa.

MEMORY 1

Measurement	KS Statistic	P-value
Attention	0.4656	0.0000
Meditation	0.7521	0.0000
Delta	0.1097	0.7216
Theta	0.5035	0.0000
Low Alpha	0.6243	0.0000
High Alpha	0.7402	0.0000
Low Beta	0.5807	0.0000
High Beta	0.7378	0.0000

Tabla 29. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad memory1, en relación a cada una de las ondas capturadas.

MEMORY 2

Measurement	KS Statistic	P-value
Attention	0.5185	0.0000
Meditation	0.6296	0.0000
Delta	0.4683	0.0001
Theta	0.6858	0.0000
Low Alpha	0.6782	0.0000
High Alpha	0.9184	0.0000
Low Beta	0.7133	0.0000
High Beta	0.8196	0.0000

Tabla 30. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad memory2, en relación a cada una de las ondas capturadas.

SEGUNDITO

Measurement	KS Statistic	P-value
Attention	0.7024	0.0000
Meditation	0.6429	0.0000
Delta	0.7214	0.5784
Theta	0.9036	0.0000
Low Alpha	0.8417	0.0000
High Alpha	0.9256	0.0000
Low Beta	0.8381	0.0000
High Beta	0.9018	0.0000

Tabla 31. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad segundito, en relación a cada una de las ondas capturadas.

Las actividades **Memory1**, **Memory2**, y **Segundito** muestran consistencia en la diferenciación de las ondas cerebrales en la mayoría de las mediciones, con la mayoría de las estadísticas KS altas y valores P muy bajos, reflejando diferencias claras y consistentes entre los dos usuarios.

UP

Measurement	KS Statistic	P-value
Attention	0.9767	0.0000
Meditation	0.7442	0.0000
Delta	0.8861	0.0000
Theta	1.0000	0.0000
Low Alpha	0.9351	0.0000
High Alpha	1.0000	0.0000
Low Beta	0.9792	0.0000
High Beta	1.0000	0.0000

Tabla 32. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad UP, en relación a cada una de las ondas capturadas.

DOWN

Measurement	KS Statistic	P-value
Attention	1.0000	0.0015
Meditation	1.0000	0.0015
Delta	0.6591	0.0982
Theta	1.0000	0.0015
Low Alpha	1.0000	0.0015
High Alpha	0.9091	0.0073
Low Beta	0.9091	0.0073
High Beta	1.0000	0.0015

Tabla 33. Tabla con los resultados del test KS para los usuarios MOR y AFB en la actividad DOWN, en relación a cada una de las ondas capturadas.

Entre las actividades UP y DOWN, podemos destacar su alta variabilidad entre los usuarios.

Para la actividad UP, todas las mediciones tienen estadísticas KS muy altas y valores P de 0.0000, lo que indica diferencias significativas entre los usuarios en todas las categorías de ondas cerebrales durante esta actividad. En el caso de DOWN, observamos una situación similar a UP excepto en el caso de delta.

Para los resultados, no se tomaron muy en serio los resultados de las actividades DOWN y UP, ya que, como se explicó previamente, durante el desarrollo de las entrevistas nos dimos cuenta que esta actividad no se llevaba a cabo de manera correcta por el usuario.

6.2.4.2 Conclusiones

Los datos muestran resultados de pruebas de KS que comparan la distribución de medidas de ondas cerebrales entre dos usuarios (MOR y AFB) durante diferentes actividades. Las pruebas KS miden tanto la estadística KS, que indica qué tan diferentes son las distribuciones entre dos muestras, como el valor P, que nos dice si estas diferencias son estadísticamente significativas (generalmente, un valor P menor a 0.05 indica significancia estadística).

Los resultados indican que hay algunas diferencias significativas en las respuestas cerebrales entre dos usuarios en varias actividades, lo que puede ser de interés para estudios personalizados o aplicaciones en neurofeedback donde se busca ajustar actividades o tratamientos basados en perfiles de onda cerebral individuales.

Dado que las actividades mostraron variabilidad en varias bandas de frecuencia, los programas basados en neurofeedback deberían ser adaptados a las características individuales del cerebro de cada usuario.

Dada la variabilidad observada, y de cara a futuros proyectos, sería conveniente realizar más estudios para explorar las causas de estas diferencias y controlar variables como la edad, el estado de salud mental, y el nivel de experiencia en las actividades. Además, deberíamos considerar variables de confusión que puedan afectar las mediciones de ondas cerebrales, como el ambiente donde se realizan las mediciones o el estado emocional y físico del usuario durante la actividad.

A pesar de que cada usuario genere una distribución diferente, las ondas de atención y meditación de cada usuario sí que presentan una correlación con los estados cognitivos de los usuarios. Los rangos de valores correspondientes a la atención del usuario pueden variar entre ellos, pero todos demuestran una correlación significativa. Esta justificación junto a los resultados de los puntos 6.2.2 y 6.6.3, nos hicieron decantarnos por la utilización de estos valores para desarrollar una aplicación de neurofeedback.

7. DISEÑO DE UN PROTOTIPO DE VIDEOJUEGO PARA VALIDAR EL SISTEMA

Tras la realización de los análisis estadísticos y evaluación de los resultados, decidimos implementar un prototipo de videojuego que nos permitiese validar los resultados y conclusiones extraídas de los análisis.

Concluimos en utilizar los valores de atención y meditación en tiempo real que nos envía el dispositivo MindFlex para la toma de decisiones en un videojuego. Esta nos permite ver feedback en tiempo real sobre las ondas cerebrales del usuario y demostrar una prueba de concepto en relación al trabajo realizado.

7.1 VALORES DE ATENCIÓN Y MEDITACIÓN

Los valores de atención y meditación son valores eSense que se pueden calcular a partir de las ondas cerebrales y oscilan en un rango de 0 a 100. Un valor eSense es una medida utilizada por dispositivos de neurofeedback, como el MindFlex, para cuantificar los niveles de atención y meditación de una persona.

- El valor de **atención**, se calcula a partir de la frecuencia beta y alfa. Un mayor nivel de actividad beta en el cerebro suele indicar un estado de mayor atención, mientras que un mayor nivel de actividad alfa puede indicar un estado de relajación o menor atención. Este valor es el resultado de realizar un filtrado de las ondas cerebrales, cálculo de la relación alfa-beta y una normalización de los valores, obteniendo así una escala de 0 a 100.
- El valor de **meditación**, se calcula a partir de las ondas theta y delta, asociando un mayor nivel de ondas theta a la meditación y relajación. El proceso para obtener el valor de meditación es similar al anterior: filtrado de las ondas, medición de la amplitud de las ondas theta en comparación con la delta y, en algunos casos, con otras ondas de referencia, y la normalización de los datos en una escala de 0 a 100.

En este caso, y como explicábamos anteriormente, el propio dispositivo MindFlex nos proporciona estos valores calculados.

Decidimos utilizar los valores de atención y meditación ya que pudimos concluir en los análisis exploratorios anteriores que la respuesta que producían era muy precisa. Los valores dan una respuesta óptima en función de la actividad a la que se encontraba sometido el usuario, haciendo de estos un punto clave para el proyecto. Además, pudimos encontrar que anteriormente se habían utilizado estos mismos valores como puntos de referencia para el desarrollo de proyectos similares de BCI [9, 10].

7.2 DETERMINACIÓN DE LA FUNCIONALIDAD DE LA INTERFAZ GRÁFICA

Para la creación del videojuego prototipo, nos inspiramos en una funcionalidad similar a la que podemos encontrar en juegos como FlappyBird o el Dinosaurio de Google Chrome, con órdenes sencillas para el movimiento el elemento principal: subir y bajar.

A lo largo del desarrollo del juego, aparecerán elementos móviles por el lateral derecho de la pantalla, que irán avanzando a medida que se desarrolle la partida. El objetivo de este juego será **controlar con la mente un elemento principal** que tendrá que recoger estos elementos móviles. Cuantos más elementos recojamos, más puntos sumaremos al marcador.



Figura 28. Pantalla del juego FlappyBird. Fuente: (FlappyBird, Inc)

Determinamos un rango para el valor de atención, basándonos en los análisis exploratorios y la bibliografía citada. Esto serán las variables que recibiremos del dispositivo MindFlex en tiempo real, y las que determinarán el movimiento del Sprite principal.

- Para valores de atención del MindFlex (en un rango de 0 a 100) inferiores a 46, se determinará la orden de ‘bajar’.
- Para los casos donde el valor de atención supera 51, se determinará la orden ‘subir’.
- Los valores intermedios [47 a 50] se consideran valores fluctuantes de cambio de estado, por lo que se considerará que el usuario mantiene el estado anterior hasta recibir nuevos valores.

```
if attention < 46:
    self.interpretation = False
elif attention > 51:
    self.interpretation = True
# else: se mantiene el estado anterior
```

Figura 29. Captura dos condiciones del código play.py

Planteamos dos objetos principales, la interfaz y un marcador:

- **Elemento principal.** El elemento principal realizará movimiento vertical en el eje Y con las ordenes subir y bajar determinadas por el valor de atención del dispositivo MindFlex. Este elemento simula el pájaro o el dinosaurio en los juegos de inspiración.
- **Obstáculos.** Se crearán obstáculos aleatorios en la pantalla, los cuales el elemento principal tendrá que recoger. Estos obstáculos simulan las tuberías o los cactus en los juegos de ejemplo.
- **Interfaz.** La interfaz presentará un paisaje estable de fondo. Los obstáculos aparecerán por el lateral derecho de la pantalla y desaparecerán por el izquierdo. El elemento principal se situará en una posición estable en el eje x de la pantalla y realizará movimiento vertical en el eje y.
- **Marcador.** En la esquina superior derecha aparecerá un marcador con el número de puntos conseguidos en función de la cantidad de obstáculos recogidos.

Hay que recalcar que el objetivo del proyecto no fue el desarrollo de un videojuego, por lo que no buscamos una gran jugabilidad. Esto es una forma de evaluación de las conclusiones en relación al trabajo realizado.

7.3 DISEÑO DE LA ESTÉTICA Y LA FUNCIONALIDAD

Para la implementación del videojuego, buscamos una estética sencilla pero atractiva para el usuario. La idea principal es controlar un objeto, en este caso un OVNI, que debe recoger las vacas que aparecen en la pantalla para ir sumando puntos a su marcador. El

usuario controla el movimiento vertical del OVNI utilizando sus niveles de atención y meditación.

Decidimos llamar a nuestro prototipo “*Abduce las vacas*”.

7.3.1 Creación de los Sprites

Con idea de darle un carácter único al prototipo, decidimos crear nuestros propios Sprites con la página web pixelArt [24]. Crearemos cada uno de los elementos móviles pixel a pixel, siguiendo la estética deseada.

7.3.1.1 Elemento Principal

El elemento principal del videojuego *Abduce las vacas* será un OVNI. Realizamos dos versiones del mismo: uno con el **fondo verde** y un segundo con el **fondo amarillo**. Durante la implementación del videojuego realizaremos cambios entre los Sprites, dando una sensación de iluminación en el elemento principal.

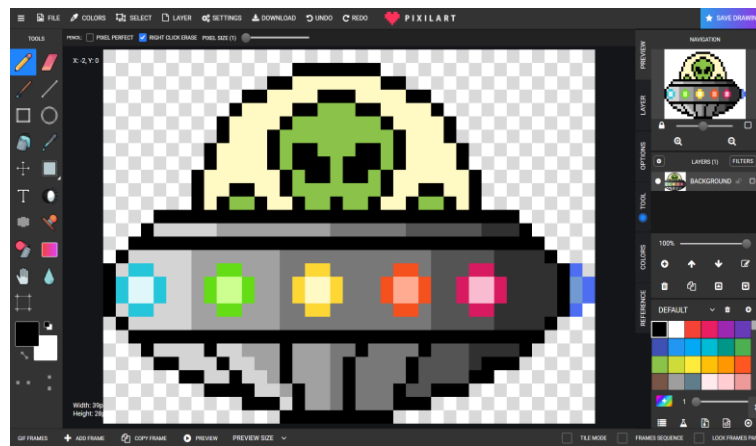


Figura 30. Dibujo del Sprite de un ovni en PixelArt. Fuente: Elaboración propia

7.3.1.2 Obstáculos

Para los obstáculos, en este caso las vacas, definimos dos diferentes para darle variabilidad al juego. Estas se generarán y aparecerán de manera aleatoria.

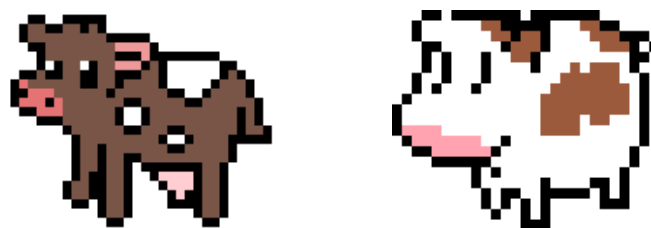


Figura 31. Sprites hechos con PixelArt. Fuente: Elaboración propia.

7.3.1.3 Interfaz

Para la interfaz escogimos un fondo del espacio con opacidad reducida, de modo que destacasen los Sprites sobre el fondo. Incluiremos también un sistema de puntuación que se mostrará en la esquina superior derecha e irá aumentando a medida que toquemos los obstáculos con el elemento principal. A continuación se muestra una imagen con la pantalla principal del juego y los elementos incluidos. Ver figura 34.

7.3.2 Pantallas de Juego

Definimos 4 pantallas de juego: una pantalla principal, una pantalla de calibrado, la pantalla de juego, y una pantalla final.

7.3.2.1 Pantalla Inicial

La pantalla inicial se muestra al comienzo del juego. Espera la introducción de cualquier tecla por teclado para comenzar con el juego.

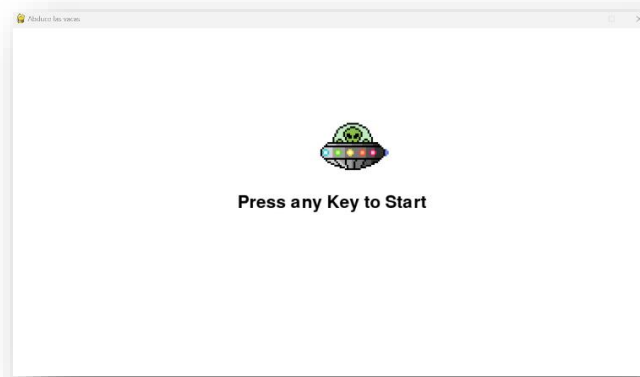


Figura 32. Captura de la pantalla inicial del juego.

7.3.2.2 Pantalla de Calibrado

La pantalla de calibrado muestra una cuenta atrás de 3 a 0. Esta pantalla es esencial para asegurarnos de que los valores del MindFlex están llegando correctamente. En caso contrario, el juego se reiniciará.

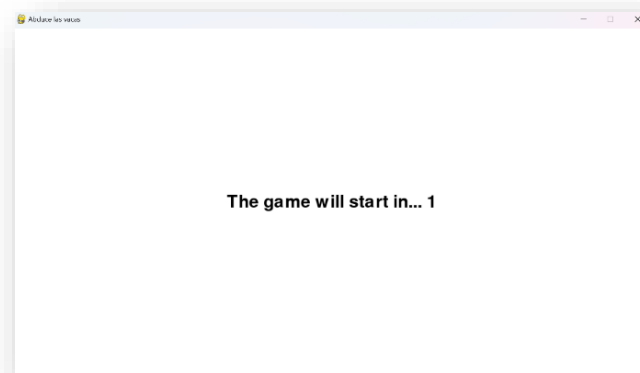


Figura 33. Captura de la pantalla de calibrado del juego.

7.3.2.3 Pantalla de Juego

La pantalla de juego muestra la interfaz previamente descrita, con el elemento principal, el fondo, los obstáculos y un sistema de puntuación. Esta se ejecuta automáticamente después de la pantalla de calibrado.

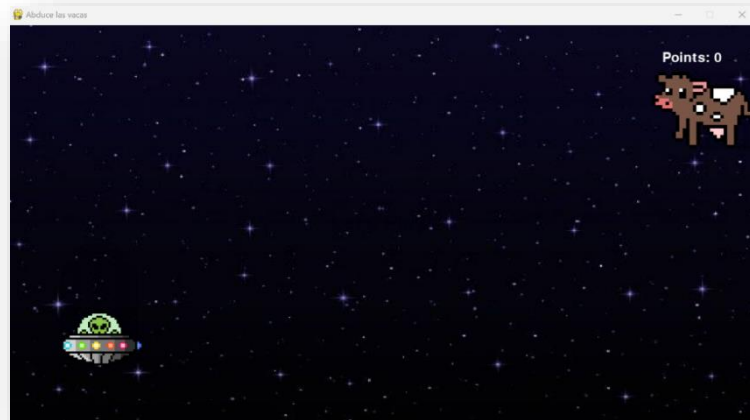


Figura 34. Captura de la pantalla principal del juego.

7.3.2.4 Pantalla Final

Una vez el usuario sale del juego, se le muestra la pantalla final con la puntuación obtenida. Permite al usuario volver a comenzar una partida presionando nuevamente cualquier tecla del teclado.

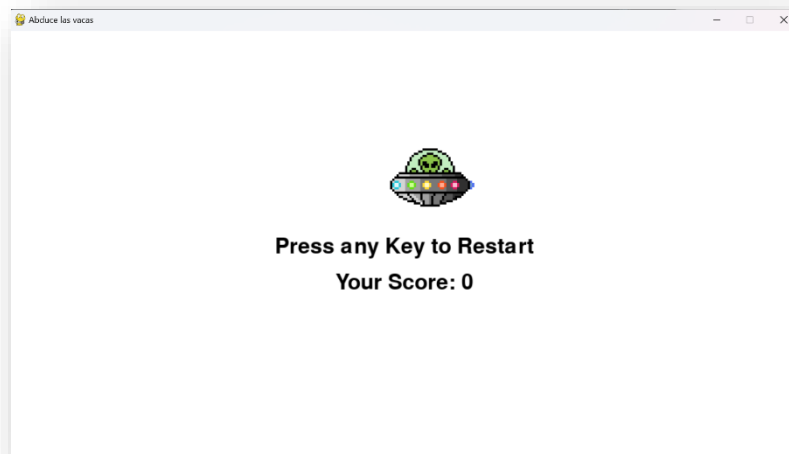


Figura 35. Captura de la pantalla final del juego.

7.4 IMPLEMENTACIÓN DEL VIDEOJUEGO

Para esta última parte, partimos de una señal previamente adquirida con el dispositivo MindFlex, limpiada con Arduino y procesada en Python. Podemos encontrar el código completo en el anexo X.

A continuación utilizaremos de nuevo scripts de Python que nos permitirá darle funcionalidad a las ondas registradas. El programa define una interfaz para el usuario con feedback respecto a la producción de las ondas específicas.

ReceiveDataGame.py

Este script define la clase `receiveData` que se encargará de recibir los datos mediante comunicación serial desde la placa Arduino, y filtrar los valores de atención y meditación.

Este programa determinará el valor de la variable `interpretation` en función del valor de meditación y atención con las condiciones previamente descritas:

```
if attention < 46:
    self.interpretation = False
elif attention > 51:
    self.interpretation = True
```

Esta variable será accesible desde el juego principal, proporcionando así los valores para el movimiento de los sprites. Para los casos donde la variable booleana `interpretation` es cierta, los sprites se moverán hacia arriba, mientras que cuando la variable cuente con un valor falso, los sprites descenderán respecto a su posición actual.

Se definen a continuación las funciones principales del programa:

- **execute ()** : Este método es el núcleo del programa y se ejecuta en un bucle infinito para recibir y procesar continuamente los datos del dispositivo MindFlex. Lee los datos del puerto serie y los procesa para extraer los valores de atención y meditación.

Filtra los datos, descartando las primeras lecturas para la calibración. Guarda los valores de atención y meditación y actualiza la interpretación de los niveles de concentración. Establece un flag para indicar la disponibilidad de nuevos datos.

- **updateNewData ()** : Permite establecer el flag `new_data` en **False** después de que se han procesado los datos desde el programa principal (el videojuego)
- **getInterpretation ()** : Retorna el valor de la variable `interpretation`, que indica si la concentración es alta o baja.
- **getNewData ()** : Retorna el valor de la variable `new_data`, que indica si se han recibido nuevos datos desde la última vez que se llamó al método `updateNewData`.
- **stopReading ()** : Este método detiene la lectura de datos del dispositivo MindFlex. Establece el flag de **fin** en **True**, lo que hace que el bucle principal deje de ejecutarse.

OVNI.py

Este script implementa el juego 'Abduce las vacas'. Para la implementación del juego, se utilizó la librería **pygame**. Esta permite crear videojuegos y aplicaciones multimedia interactiva, proporcionando gran cantidad de funciones específicas para el desarrollo. Podemos ver el código completo en el anexo X.

Este programa implementa distintas clases y funciones:

1. Ovni:

- **Descripción:** Representa al OVNI controlado por el jugador. Controla el movimiento del en respuesta a los datos recibidos y dibuja el OVNI en la pantalla.
- **Métodos:**

- **update ()** : Actualiza la posición del OVNI en función de los datos recibidos de la variable **interpretation**.
- **draw ()** : Dibuja el OVNI en la pantalla. Alternará entre la imagen del ovni verde y el ovni amarillo gracias a un contador.

2. **Obstacle:**

- **Descripción:** Clase base para los obstáculos en el juego. Controla el movimiento y la representación aleatoria de los obstáculos en la pantalla.
- **Métodos:**
 - **update ()** : Actualiza la posición del obstáculo en la pantalla.
 - **draw ()** : Dibuja el obstáculo en la pantalla.

3. **Cow** (hereda de **Obstacle**):

- **Descripción:** Representa una vaca, un tipo específico de obstáculo en el juego. Controla el comportamiento y la representación de las vacas en la pantalla.
- **Métodos:** No introduce métodos adicionales, únicamente hereda de **Obstacle**.

A continuación se destacan las funciones principales:

- **main ()** : Se trata de la función principal que controla el bucle principal del juego. Inicializa el juego, maneja eventos, actualiza la pantalla y controla la lógica del juego.
- **calibrationScreen (player)** : Función que muestra una pantalla de calibración con una cuenta atrás antes de iniciar el juego. Espera a que se complete la calibración de los valores recibidos por la banda mindflex antes de iniciar el juego.
- **menuScreen ()** : muestra el menú principal del juego: inicio y finalización del juego. Permite al jugador iniciar al comienzo o reiniciar el juego. Cuando el juego es reiniciado, se muestran los puntos obtenidos en la partida anterior.

A continuación se muestra la lógica simplificada que sigue el cuerpo principal del juego en la función **main**:

```
def main():
    inicializaciones ()
    # Inicializa recibimiento de datos del Mindflex en un thread
    receiver_thread = threading.Thread(target=player.receiver.execute)
    calibrationScreen (player)

    while run:
        if fin_juego:
            run = False
        background()      #Inicializa el fondo
        if new_value_mindflex:
            player.update()
            if no_obstáculos:
```

```

generarObstáculosRandom()
if obstacles:
    pintarObstáculo()
# Resetea después de la actualización
new_value_mindflex = False
pygame.display.update()

finLectura()
receiver_thread.join()

```

7.4.1 Resumen de Clases

Ruta	Fichero	Clase	Funcionalidad
Game/	receiveDataGame.py	receiveData	Recibe datos mediante comunicación serial de la placa Arduino. Actualiza variables de control para la implementación del juego
	OVNI.py	Ovni	Se encarga del movimiento e inicialización principal del videojuego. Inicializa imágenes, fondo y los elementos móviles. Es la arquitectura principal del videojuego.

Tabla 34. Tabla resumen con la ruta, ficheros y clases empleadas para la implementación de la interfaz gráfica.

7.5 PRUEBAS DE FUNCIONAMIENTO

La realización de las pruebas de funcionamiento es fundamental para asegurar la calidad y el correcto funcionamiento del juego desarrollado. A continuación se detallan las principales pruebas realizadas para los ficheros principales del videojuego.

7.5.1 TestOvniGame.py

7.5.1.1 Pruebas unitarias

Inicialización del Ovni

- Verificar que el **Ovni** se inicializa correctamente.
- Comprobar que los atributos como **ovni_run**, **step_index**, **image**, **ovni_rect.x**, y **ovni_rect.y** están configurados correctamente.
- Confirmar que **self.receiver** se inicializa llamando a **receiveData()**.

Código:

```

def test_ovni_initialization():
    ovni = Ovni()
    assert ovni.ovni_run == True
    assert ovni.step_index == 0
    assert ovni.image is not None
    assert ovni.ovni_rect.x == Ovni.X_POS
    assert ovni.ovni_rect.y == Ovni.Y_POS
    assert ovni.receiver is not None

```

Movimiento del Ovni

- Simular diferentes valores de interpretación recibidos (**True** o **False**).
- Verificar que el OVNI se mueve hacia arriba o hacia abajo según la interpretación recibida.

Código:

```
def test_ovni_movement():
    ovni = Ovni()
    initial_y = ovni.ovni_rect.y

    # Simular atención alta
    ovni.receiver.getInterpretation = lambda: True
    ovni.update()
    assert ovni.ovni_rect.y == initial_y - ovni.MOVE_SPEED

    # Simular atención baja
    ovni.receiver.getInterpretation = lambda: False
    ovni.update()
    assert ovni.ovni_rect.y == initial_y
```

Inicialización de Obstáculos

- Verificar que los obstáculos se inicializan correctamente y que sus atributos como `rect.x` y `rect.y` están configurados adecuadamente.

Código:

```
def test_obstacle_initialization():
    cow = Cow(COW1)
    assert cow.rect.x == SCREEN_WIDTH
    assert 50 <= cow.rect.y <= 550
```

Actualización de Obstáculos

- Verificar que los obstáculos se mueven hacia la izquierda con el paso del tiempo.
- Comprobar que los obstáculos se eliminan al final de la pantalla.

Código:

```
def test_obstacle_update():
    obstacles = [Cow(COW1)]
    initial_x = obstacles[0].rect.x
    obstacles[0].update()
    assert obstacles[0].rect.x == initial_x - game_speed

    # Simular obstáculo saliendo de la pantalla
    obstacles[0].rect.x = -obstacles[0].rect.width
    obstacles[0].update()
    assert len(obstacles) == 0
```

7.5.1.2 Pruebas de Integración

Inicio de Juego

- Verificar que la pantalla de inicio se muestra correctamente.
- Comprobar que al presionar una tecla, se inicia el juego.

Código:

```
def test_game_start():
    # Tecla presionada
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            main()
    # Juego ha comenzado
    assert DEATH_COUNT == 1
    assert POINTS == 0
```

Pantalla de Calibración

- Verificar que la pantalla de calibración se muestra correctamente y que el juego comienza después de la calibración.

Código:

```
def test_calibration_screen():
    player = Ovni()
    player.receiver.getCalibration = lambda: 0
    calibrationScreen(player)
    # La pantalla de calibración si se muestra correctamente
    assert pygame.display.get_surface() is not None
```

7.5.1.3 Pruebas de Rendimiento

Rendimiento del Juego

- Medir el tiempo de fotograma para asegurar que el juego se ejecuta a 30 FPS.
- Verificar que el juego no experimenta caídas de rendimiento significativas.

Código:

```
def test_game_performance():
    clock = pygame.time.Clock()
    start_time = pygame.time.get_ticks()
    main()
    end_time = pygame.time.get_ticks()
    assert (end_time - start_time) / 1000 <= 1 / 30
```

7.5.2 ReceiveDataTest.py

7.5.2.1 Pruebas de funcionamiento

Inicialización de objeto

- Verificar que la clase **receiveData** se inicializa correctamente y que las variables de instancia tienen los valores esperados.

Código:

```
def test_initialization():
    receiver = receiveData()
    assert receiver.folder_path == "D:\\TFG BrainLink\\BrainLink"
    assert receiver.data_list == []
    assert receiver.attention_values == []
    assert receiver.meditation_values == []
    assert receiver.calibration_count == 4
    assert receiver.fin == False
    assert receiver.new_data == False
    assert receiver.interpretation == None
```

Recepción y Procesamiento de Datos

- Verifica que los datos recibidos desde el puerto serie se procesan y almacenan correctamente. Comprueba además que la interpretación se realice correctamente según los valores de atención.

Código:

```
def test_data_processing():
    receiver = receiveData()
    mock_data = "0,50,60,100,200,300,400,500,600,700,800"
    receiver.calibration_count = 1 # Calibración ya ha ocurrido

    def mock_readline():
        return mock_data.encode()

    ser.readline = mock_readline

    receiver.execute()

    assert receiver.attention_values == [50]
    assert receiver.meditation_values == [60]
    assert receiver.getInterpretation() == None # Mantiene estado
    previo
    assert receiver.getNewData() == True
```

Interrupción de la Lectura de Datos

- Verificar que el método **stopReading** interrumpe correctamente la lectura de datos.

Código:

```
def test_stop_reading():
    receiver = receiveData()
    receiver.stopReading()
    assert receiver.fin == True
```

Calibración

- Verificar que el contador de calibración decrece correctamente con cada lectura de datos hasta que llega a 0.

Código:

```
def test_calibration_count():
    receiver = receiveData()
    receiver.calibration_count = 3
    mock_data = "0,50,60,100,200,300,400,500,600,700,800"

    def mock_readline():
        return mock_data.encode()

    ser.readline = mock_readline

    receiver.execute()

    assert receiver.calibration_count == 2
```

8. EVALUACIÓN, DISCUSIÓN Y CONCLUSIONES DEL PROYECTO

El desarrollo de la interfaz cerebro-ordenador y su prototipo de validación del sistema para el control del juego “Abduce las vacas” cumplen con los objetivos establecidos en la fase inicial del proyecto, pero sobre todo con la posibilidad de visualización del feedback en tiempo real.

Los resultados y el análisis del proyecto confirmaron la posibilidad de capturar y procesar señales EEG con precisión utilizando un sistema semidúplex, lo que nos permitió utilizar estos valores para desarrollar un prototipo, y controlar las funciones de un videojuego. Se observó principalmente una correlación significativa entre los niveles de atención y meditación medidos por el sensor EEG y las acciones realizadas por los usuarios en el juego durante las entrevistas, lo que nos permitió desarrollar una interfaz que responde de manera efectiva a estos estados cognitivos del usuario. Esto se correlaciona además con otros estudios relacionados [7, 9, 10] que responden con resultados similares.

Sin embargo, a pesar de cumplir con los propósitos generales, identificamos algunas limitaciones. La primera tiene que ver con la velocidad de respuesta del sistema BCI, ya que durante la implementación del videojuego, nos dimos cuenta de que el sistema recibe un valor de actividad cerebral cada segundo. Esto resulta en una interacción relativamente lenta para la implementación de un videojuego, pudiendo afectar así a la experiencia de usuario.

Cabe destacar que la causa principal de esta limitación no se debe a la programación o al diseño del software, sino a la frecuencia de envío de señales del dispositivo MindFlex. Este dispositivo permite únicamente enviar señales con una frecuencia de 9600 baudios. Este retraso en la transmisión hace que la posible interacción se vea comprometida en entornos de juego donde es esencial la respuesta inmediata.

Hay que aclarar que el objetivo principal no era determinar una gran experiencia de usuario, sino crear un prototipo que nos permitiese validar el sistema. Desde esta perspectiva, podemos dar por válidos los resultados obtenidos.

Otra limitación identificada durante el desarrollo del trabajo es el actual tamaño de la muestra de usuarios para la realización de las pruebas y de los análisis estadísticos. Una muestra única de seis usuarios restringe la generalización y puede limitarnos la precisión de los resultados.

Además, durante el proyecto se establecieron umbrales específicos para el valor de atención para realizar los movimientos ascendentes o descendentes en el juego: valores menores de 46 indican movimientos descendentes y mayores de 51 indicarían movimientos ascendentes. Estos umbrales fueron definidos en base a la muestra tomada de seis usuarios y podrían no ser óptimos para todas las personas debido a diferencias individuales en las señales de EEG de cada cerebro. Cabe recordar que la emisión de las ondas EEG depende no solo de las condiciones externas o de las actividades llevadas a cabo, sino también de la persona así como de posibles condiciones neurodivergentes (p.e TDAH) [4].

Como posible sugerencia para mitigar esta variabilidad y mejorar la personalización del programa para el usuario, se propone el desarrollo de un sistema de inteligencia artificial. Esta red podría realizar un ejercicio de calibrado al inicio del juego o de las pruebas realizadas para cada usuario, ajustando automáticamente los umbrales de atención y meditación según las características individuales del EEG.

Para concluir la discusión, podemos afirmar que se han cumplido con éxito los objetivos propuestos inicialmente. Se han discutido las diversas limitaciones encontradas a lo largo

del proyecto, como la velocidad de respuesta del dispositivo MindFlex y el tamaño reducido de la muestra, que podrían impactar en la generalización de los resultados obtenidos.

Finalmente, han propuesto algunas soluciones para abordar estas limitaciones en futuros trabajos como podría ser la implementación de un sistema de inteligencia artificial para personalizar los umbrales de interacción del usuario con la BCI. Estas mejoras no solo permitirían optimizar la funcionalidad de nuestra BCI, sino que también enriquecerían la experiencia del usuario, haciendo que estos sistemas sean más accesibles y se adapten mejor a las necesidades individuales.

En conclusión, este trabajo demuestra que sí posible crear sistemas BCI funcionales y económicos con tecnologías accesibles y herramientas limitadas.

Referencias

- [1] Jeevareha, R., & Tharini, M. (2020). EEG Based Brain Controlled Keypad and Devices. *International Research Journal of Multidisciplinary Technovation*, 2(3), 27–33. <https://doi.org/10.34256/IRJMT2035>
- [2] Wang, A. I., Hokland, J., & Larsen, E. A. (2011). Classification of EEG Signals in a Brain-Computer Interface System. 70. <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/252472>
- [3] Carrasco, F., & Alvaro, J. (2017). *Estudio y diseño de un sistema interfaz cerebro-computador de bajo coste con módulo Arduino*.
- [4] Fink, A., Schwab, D., & Papousek, I. (2011). Sensitivity of EEG upper alpha activity to cognitive and affective creativity interventions. *International Journal of Psychophysiology*, 82(3), 233–239. <https://doi.org/10.1016/J.IJPSYCHO.2011.09.003>
- [5] Angelakis, E., Lubar, J. F., Stathopoulou, S., & Kounios, J. (2004). Peak alpha frequency: An electroencephalographic measure of cognitive preparedness. *Clinical Neurophysiology*, 115(4), 887–897. <https://doi.org/10.1016/j.clinph.2003.11.034>
- [6] Klimesch, W. (1999). EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research. Brain Research Reviews*, 29(2–3), 169–195. [https://doi.org/10.1016/S0165-0173\(98\)00056-3](https://doi.org/10.1016/S0165-0173(98)00056-3)
- [7] (PDF) *Sistema Embebido para Detección de Somnolencia en Conductores mediante señal EEG*. (n.d.). Retrieved May 25, 2024, from https://www.researchgate.net/publication/301223033_Sistema_Embebido_para_Deteccion_de_Somnolencia_en_Conductores_mediante_senal_EEG
- [8] Politécnica De Madrid, U., José, L., & Figueroa, G. (n.d.). *DEPARTAMENTO DE AUTOMÁTICA, INGENIERÍA ELÉCTRICA Y ELECTRÓNICA E INFORMÁTICA Análisis de señales EEG para detección de eventos oculares, musculares y cognitivos*.
- [9] Katona, J., Farkas, I., Ujbanyi, T., Dukan, P., & Kovari, A. (2014). Evaluation of the NeuroSky MindFlex EEG headset brain waves data. *SAMI 2014 - IEEE 12th International Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 91–94. <https://doi.org/10.1109/SAMI.2014.6822382>
- [10] Serna, B., Baltazar, R., Casillas, M., Saavedra, Y., Alanis, A., & Meza, A. (2020). Medical Diagnostic Through a Mobile Application Controlled by Brain Waves: ConsultApp. *Smart Innovation, Systems and Technologies*, 148, 185–193. https://doi.org/10.1007/978-981-13-8679-4_16

WEBGRAFÍA

- [11] *Despolarización y repolarización de la membrana celular - Ciencia de Hoy*. (n.d.). Retrieved November 8, 2023, from <https://cienciadehoy.com/despolarizacion-y-repolarizacion-de-la-membrana-celular/>
- [12] *The OpenEEG project*. (n.d.). Retrieved November 28, 2023, from <https://openeeg.sourceforge.net/>
- [13] *EEG - ECG - Biosensors*. (n.d.). Retrieved November 28, 2023, from <https://neurosky.com/>
- [14] *Mindflex*. (n.d.). Retrieved November 28, 2023, from <https://store.neurosky.com/products/mindflex>
- [15] *Force Trainer - Wikipedia*. (n.d.). Retrieved November 28, 2023, from https://en.wikipedia.org/wiki/Force_Trainer
- [16] *ThinkGear AM - Brainwave Sensor*. (n.d.). Retrieved November 30, 2023, from <https://www.seeedstudio.com/ThinkGear-AM-Brainwave-Sensor-p-1441.html>
- [17] (PDF) *EEG Based Brain Controlled Keypad and Devices*. (n.d.). Retrieved October 16, 2023, from https://www.researchgate.net/publication/347714624_EEG_Based_Brain_Controlled_Keypad_and_Devices
- [18] *Arduino Uno Rev3 — Arduino Official Store*. (n.d.). Retrieved January 25, 2023, from <https://store.arduino.cc/products/arduino-uno-rev3>

- [19] How to Hack Toy EEGs | Frontier Nerds. (n.d.). Retrieved December 20, 2023, from <https://frontiernerds.com/brain-hack>
- [20] *kitschpatrol/BrainGrapher: Processing-based visualizer for NeuroSky EEG brainwave data output from the Arduino Brain library.* (n.d.). Retrieved December 20, 2023, from <https://github.com/kitschpatrol/BrainGrapher>
- [21] *kitschpatrol/Brain: Arduino library for reading Neurosky EEG brainwave data. (Tested with the MindFlex and Force Trainer toys.)*. (n.d.). Retrieved December 20, 2023, from <https://github.com/kitschpatrol/Brain>
- [22] *Kolmogorov–Smirnov test - Wikipedia.* (n.d.). Retrieved May 4, 2024, from https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test
- [23] “SEGUNDITO” - Corto de ANIMACIÓN sobre el TIEMPO [MUY DIVERTIDO] - YouTube. (n.d.). Retrieved April 17, 2024, from <https://www.youtube.com/watch?v=0zsE85khG6I>
- [24] *Pixilart - Free Online Art Community and Pixel Art Tool.* (n.d.). Retrieved May 25, 2024, from <https://www.pixilart.com/>

ANEXOS

Anexo I.	Diagrama de Gantt fases 1 y 2.....	II
Anexo II.	Diagrama de Gantt Fases 3, 4 y 5.....	III
Anexo III.	Consentimiento informado	IV
Anexo IV.	Consentimiento informado DPD	V
Anexo V.	Diseño de las entrevistas.....	VIII
Anexo VI.	Imágenes para las entrevistas	XII
Anexo VII.	Análisis exploratorio de datos	XV
Anexo VIII.	Histogramas para las distintas actividades evaluadas.....	XVIII
Anexo IX.	Tablas de correlación para las variables más significativas	XXIV
Anexo X.	Código Python ‘Abduce las vacas’	XXV

Anexo III. Consentimiento informado

CONSENTIMIENTO INFORMADO DEL PARTICIPANTE

Análisis de ondas cerebrales mediante tecnología EEG para TFG

Garantía de confidencialidad y de participación voluntaria: El fin de la encuesta será meramente para la investigación llevada a cabo en un trabajo de fin de grado para el grado de ingeniería informática en la URV. Todos los datos obtenidos se codificarán de forma que no aparezca ningún nombre, ni ninguna otra información que permita la identificación de los participantes. Sólo las personas relacionadas con el proyecto tendrán acceso a los archivos. Estos procedimientos están sujetos a la Ley Orgánica 15/1999 del 13 de diciembre sobre protección de datos de carácter personal. La participación en este estudio es totalmente voluntaria, por lo que el sujeto puede retirar su permiso en cualquier momento. En este caso, tus datos serían retirados de la base de datos.

Firma

Fecha.....

.....

Nombre y apellidos del voluntario:

.....

En caso que más adelante usted quiera hacer alguna pregunta o comentario sobre este proyecto, o bien si quiere revocar su participación en el mismo, porfavor contacte con:

Ainhoa García Linares
ainhoa.garcial@estudiants.urv.cat

Lugar, fecha y firma del investigador:.....

Ejemplar para el participante / Ejemplar para el investigador

Anexo IV. Consentimiento informado DPD



Full de consentiment informat

Títol de l'estudi: Desarrollo y Evaluación de una Interfaz Cerebro-Ordenador (BCI) para el Control de Videojuegos: Un Enfoque Práctico Utilizando EEG

Dades de contacte de l'investigador principal: *Ainhoa.garcial@estudiants.urv.cat*

Jo¹ amb DNI.....:

- He llegit el full d'informació al participant sobre l'estudi del qual se m'ha entregat una còpia.
- He pogut fer preguntes i resoldre els meus dubtes sobre l'estudi i la meva participació.
- Comprenc la meva participació a l'estudi d'acord amb allò expressat al full d'informació al participant sobre l'estudi i de les respostes a les meves preguntes, així com els riscos i beneficis que comporta.
- Accepto que la meva participació és voluntària i dono lliurement la meva conformitat per participar a l'estudi.
- Conec que em puc retirar en qualsevol moment de la participació a l'estudi sense que això em pugui causar cap perjudici.
- Estic informat sobre el tractament que es realitzarà de les meves dades personals.
- Dono el meu consentiment per a l'accés i utilització de les meves dades en les condicions detallades al full d'informació al participant sobre l'estudi.

Sí No

- ²Dono el meu consentiment per a la difusió de les meves dades personals junt amb la publicació dels resultats de l'estudi.

Sí No

- ³Un cop finalitzada la investigació, és possible que hi hagi mostra sobrant. En relació a aquestes, s'ofereixen les següents opcions:

La **destrucció** de la mostra sobrant.

La seva **utilització en futurs projectes** d'investigació biomèdica relacionats amb el mateix tema

⁴ I per expressar aquest consentiment, el participant signa en data i lloc aquest full de consentiment:

¹ Indicar el nom i cognoms del participant.

² Només si en els resultats de l'estudi que es publicaran apareix el nom de les persones o dades que permetin identificar la persona, així com la seva imatge o la seva veu sense tècniques d'anonimització.

³ Si no hi ha tractament de mostres biològiques es pot eliminar.

⁴ Si el participant pot prestar lliurement el consentiment utilitzar aquest redactat fins a la següent nota el redactat del qual es pot eliminar.

Signatura del participant

⁵ I per expressar aquest consentiment, el representant legal del participant signa en data i lloc aquest full de consentiment:

Nom del representant legal.....

Relació del representant legal amb el participant

Signatura del representant legal

⁵ Si el participant no pot llegir o escriure, és menor de 14 anys o per qualsevol altre motiu no pot prestar lliurement el consentiment és necessari que el seu consentiment el presti el tutor o representant legal. En aquest cas utilitzarem aquest redactat i eliminarem el corresponent a la nota anterior.

INFORMACIÓ DE PROTECCIÓ DE DADES PERSONALS	
Responsable	El responsable del tractament de les seves dades personals és la Universitat Rovira i Virgili amb CIF Q9350003A i amb domicili fiscal al carrer de l'Escorxador, s/n, 43003 de Tarragona.
Finalitat	Participar en el treballs de fi de Grau en els termes que es descriuen al full d'informació al participant. En el cas que l'estudi prevegi la publicació, difusió i reutilització dels resultats obtinguts incloent dades personals, les dades personals seran utilitzades per a aquesta finalitat sempre que l'interessat hagi atorgat el seu consentiment.
Drets	Pot exercir els drets d'accés, rectificació, supressió, portabilitat, limitació o oposició al tractament, mitjançant un escrit adreçat al Registre General de la URV a la mateixa adreça del domicili fiscal o mitjançant la seva presentació al Registre General de la Universitat, presencialment o telemàtica, segons s'indica a https://seuelectronica.urv.cat/registre.html .
Informació addicional	Pot consultar informació addicional sobre aquest tractament de dades personals denominat ⁶ i els seus drets al Registre d'Activitats del Tractament de la URV publicat a https://seuelectronica.urv.cat/rgpd on també s'hi pot consultar la Política de Privacitat de la URV. Així mateix, pot adreçar als nostres delegats de protecció de dades qualsevol consulta sobre protecció de dades personals a la direcció de correu electrònic del dgd@urv.cat .

⁶ Indicar en cursiva "*Projectes de recerca de la URV*", "*Treballs de Fi de Grau o de Màster de la URV*" o "*Tesis Doctorals de la URV*" segons correspongui

Fase de testeo BrainLink

ANTES DE EMPEZAR

Tener preparado:

- Placa Arduino conectada a los cables correctamente. Sensor con las pilas y limpio.
 - Portatil enchufado a la corriente
 - Pelotas azules a mano
 - BrainGrapher abierto
 - Movil e ipad preparados para grabar.
 - Video "segundito" en el ipad cargado.
 - Youtube con video relajante.
 - Botella de agua y folio a modo de techo.
 - Python abierto y programa a punto para cargar. *Hacer previa prueba de conexión*
 - Recompensa agradecimiento.
-

INTRODUCCIÓN

Con el usuario

A continuación se le explica al usuario brevemente todos y cada uno de los ejercicios que se van a llevar a cabo durante las pruebas y la finalidad de esto.

Primero de todo, muchísimas gracias por querer formar parte de esto, de verdad que me ayuda muchísimo con el proyecto que estoy llevando a cabo. Este proyecto que estoy haciendo para mi trabajo TFG es una idea que se me ocurrió el año pasado durante una de las clases, y tengo cierto cariño.

Es muy importante saber que esto dura en total unos 20 minutos y necesito que durante ese tiempo estés concentrado/concentrada conmigo. Si crees que no tienes tiempo, o prefieres hacerlo en otro momento coméntamelo antes de empezar para que los resultados sean válidos.

Mi trabajo se basa en utilizar las denominadas ondas cerebrales para crear un juego que pueda moverse con la mente (comentar que es el objetivo final aunque luego pueda variar).

ONDAS CEREBRALES

Nuestro cerebro cuando pensamos, cuando dormimos, cuando hacemos todas las actividades produce una cosa que se llaman ondas cerebrales. Estas ondas cerebrales no es mas que esa "energía" que producimos los humanos simplemente porque nuestro cuerpo funciona.

Estas ondas cerebrales pueden diferenciarse en 5 tipos en función de lo que estemos haciendo

**Mostrar imágenes de los anexos

SENSOR ¿CÓMO SE REGISTRAN ESTÁS ONDAS?

Quiero enseñarte primero el sensor que utilizo para esto. Esto era parte de un juguete, no se si lo conoces.

Este juguete se comercializaba allá por 2014 y consistía en modular la velocidad de un ventilador pensando arriba o abajo. ****Mostrar imágenes y explicar brevemente.**

Lo que yo hice, es coger el aparato que utilizaba este juguete, reprogramar la placa base del juguete y conectarlo a mi placa Arduino que es lo que recibe los datos y se los pasa a mi ordenador.

RECOGIDA DE DATOS ¿PORQUÉ ES IMPORTANTE TU PAPEL EN ESTO?

Mi objetivo final con este juguete, es hacer un juego tipo flappybird o el dinosaurio de Google, que podamos controlar con la mente. Es decir, si tu piensas que el personaje se mueva hacia arriba, este salte, y si piensas que baje hacia abajo, este descienda.

Para esto, es muy importante una previa recogida de datos que me permita partir desde una base, probar la funcionalidad del aparato. Tener una referencia

Por ejemplo: tengo que enseñar a mi programa, que imagina que te pones el sensor, tu piensas arriba y el sensor recibe un valor de 1000. Pues entonces yo ya se, que cuando mi sensor me este dando un valor de 1000, el personaje tiene que subir. Y así con todo. Imagina que tu piensas ahora en bajar, y el aparato registra un 10, pues entonces mi programa aprende que cuando este reciba un 10, el pajarito tiene que bajar.

(breve explicación; con los primeros usuarios que pregunte, determinar si la explicación es sencilla, útil para ellos o si omitirla más adelante)

COLOCAR EL SENSOR AL USUARIO

A continuación, quiero que te coloques el sensor. Este aparato cuenta con 3 pines, uno en la parte frontal y dos en las lobulares.

Colócatelo con cuidado y dime si te molesta o no. Es importante que todo el sensor esté perfectamente colocado y tocando con la superficie de la piel. Comprueba también que no te apriete ni te moleste.

Muy importante, desde el momento en el que te lo pones, siempre que quieras manipularlo avísame. Si te molesta, notas algo, quieres quitártelo en algún momento.

BRAINGRAPHER

Puedo hacerte una breve demostración si quieres de tu cerebro ahora mismo, y ves como estas pensando y lo que esto registra.

Ofrecer al usuario la posibilidad de visionar las ondas que se registran en cada momento con el brainGrapher.

LOS EJERCICIOS

Para comenzar con las pruebas lo que vamos a hacer son 5 ejercicios muy sencillitos, de un par de minutos cada uno mientras llevas el sensor.

Primero de todo, te explicaré cada uno de los ejercicios que vamos a hacer. Una vez explicado, con todas las preguntas que tengas por hacerme y te haya quedado todo claro, yo voy a conectar el sensor, comenzar con la ejecución y empezar con el ejercicio.

Una vez acabado cada uno, cerraré el programa y empezaremos de nuevo con el siguiente.

EJERCICIO DE MEDITACIÓN

Primero de todo, vamos a empezar relajándonos y entrando un poco en situación.

Quiero que te sientes cómodo o cómoda, y voy a ponerte una música que te resulte relajante (dar la posibilidad de elegir al usuario una música o ponerla yo). Quiero que, cuando estés listo/lista, cierres los ojos y comiences una cuenta atrás desde 30.

EJERCICIO DE MEMORIA

Ahora que estamos relajados y nuestra cabeza tranquila, vamos a jugar a un juego de memoria muy simple.

Yo voy a empezar a decirte números del 1 al 40 y tu tienes que memorizarlos mientras yo te los digo, y luego repetírmelos.

- Por ejemplo, yo comienzo diciéndote una cifra: 10
- Y cuando yo acabe tu me dices 10.
- A continuación te digo dos valores, 10- 7
- Y tu me dices 10-7

Y así hasta 9 números en total. No te preocupes si te equivocas o cualquier cosa, que lo importante aquí no es eso.

Una vez acabado el ejercicio de memoria, vamos a hacer lo mismo pero a la inversa. Yo te iré diciendo los números y tu tienes que decírmelos en el orden inverso al que yo te los dije:

- Por ejemplo, yo comienzo diciéndote 10- 7
- Tu me tienes que responder 7- 10

Y así de nuevo con 9 números.

DERECHO:

8	19							
3	20	16						
7	17	4	18					
13	10	8	19	3				
1	11	9	17	18	18			
8	6	18	2	3	20	5		
9	12	19	7	17	20	12	9	

REVÉS:

17	20							
12	1	1						
17	2	15	18					
19	4	6	14	16				
15	2	15	3	3	6			
5	20	15	19	17	18	18		
1	2	4	16	14	13	3	9	

UP-DOWN

A continuación, como ya somos expertos en esto y sabemos como funciona voy a ponerte un ejercicio de los más importantes para estas pruebas.

Vamos a imaginar que estamos haciendo levitar una pelota. Voy a ponerte un techo y quiero que durante la prueba vamos a llevar esta pelota desde abajo hacia en techo, una vez toque arriba la pelota va a bajar, y de nuevo pensamos en hacerla subir.

Es importante que no pierdas contacto visual con el centro de la pelota en ningún momento excepto cuando la pelota toque el techo. Una vez tocas el techo, deja de mirar el centro la pelota hasta que esta vuelve a estar abajo del todo.

Hay a gente que le sirve pensar en su cabeza un “sube sube sube” repetidas veces, o pensar que la mueve muy fuerte, pensar que mi mano es la suya y que ha está haciendo levitar etc. Prueba con lo que a ti te sea más cómodo.

¿quieres hacer una de práctica?

CORTO SEGUNDITO

Por último, ya terminamos. Te voy a poner un video de dos minutos que a mi me parece bastante gracioso y no tienes que hacer nada más. Simplemente verlo y disfrutar, que te lo has ganado.

MOSTRAR AL USUARIO LOS RESULTADOS

Cargar los valores con el programa de pyhton y graficarlos en un momento.

Explicar al usuario las diferencias que hay a simple vista cuando se llevaban a cabo cada una de las pruebas en función de las cosas que observemos para cada uno.

Ser muy conciso y comentar las cosas por encima.

AGRADECIMIENTO

Entregar la tarjeta de agradecimiento y el detalle.

Anexo VI. Imágenes para las entrevistas



Figura 36. Batería de imágenes para las entrevistas. Juguete MindFlex.

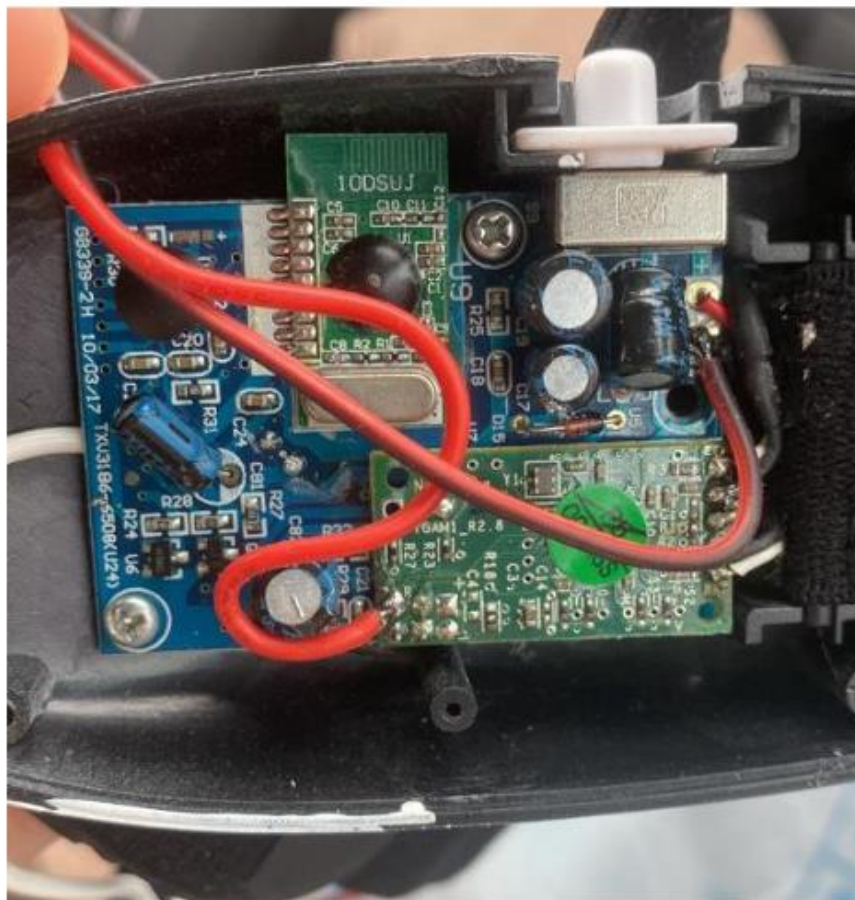


Figura 37. Batería de imágenes para las entrevistas. Juguete MindFlex. hackeado

- Ondas Delta: con una frecuencia de 0.2-4 Hz
- Ondas Theta: 4-8 Hz
- Ondas Alfa: 8-12 Hz
- Ondas Beta: 12-30 Hz
- Ondas Gamma: 30-90 Hz

HUMAN BRAIN WAVES

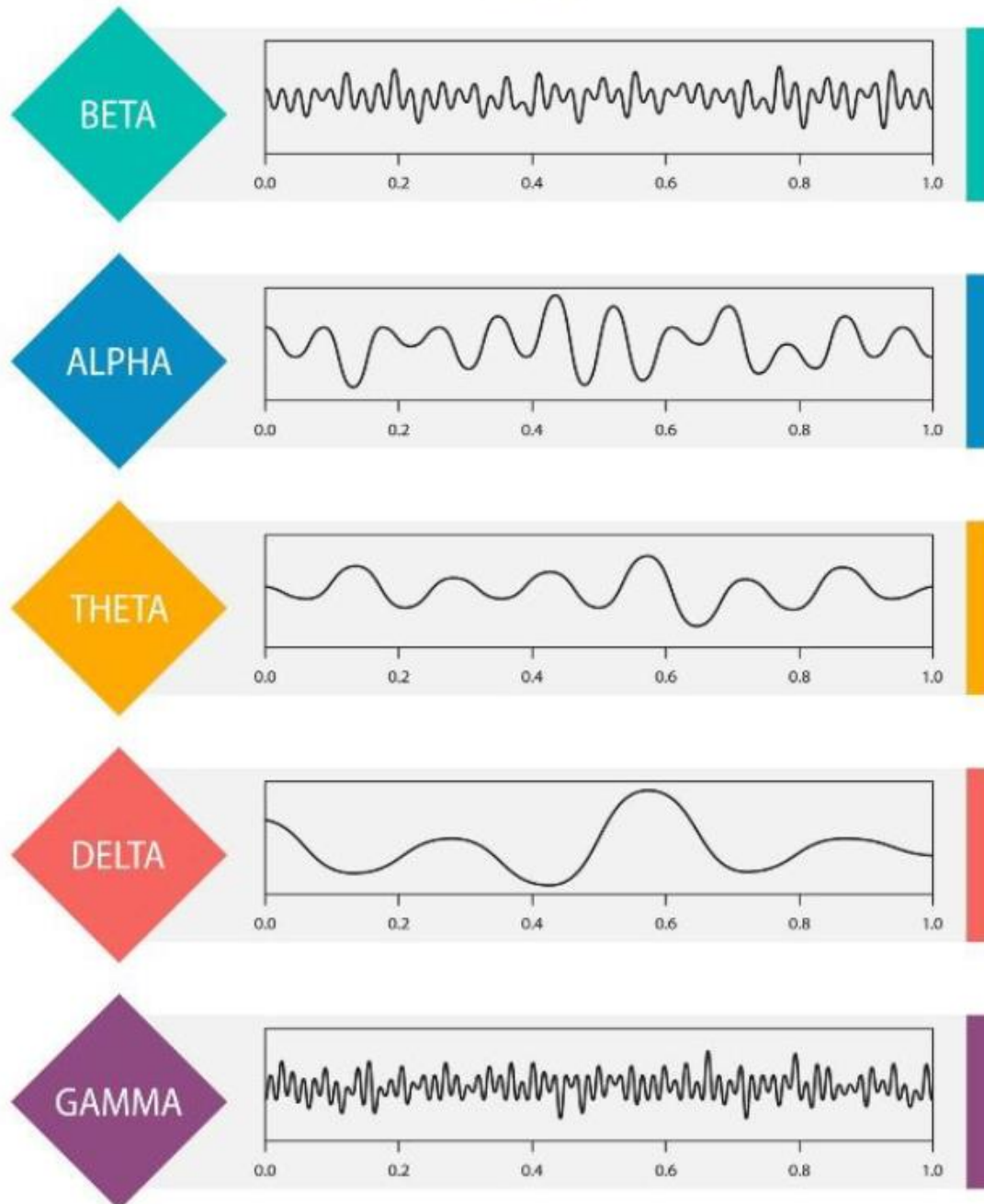


Figura 38. Batería de imágenes para las entrevistas. Ondas cerebrales.

Anexo VII. Análisis exploratorio de datos

Identificación de las correlaciones de los valores de entrada y la validez de la lectura de las señales.

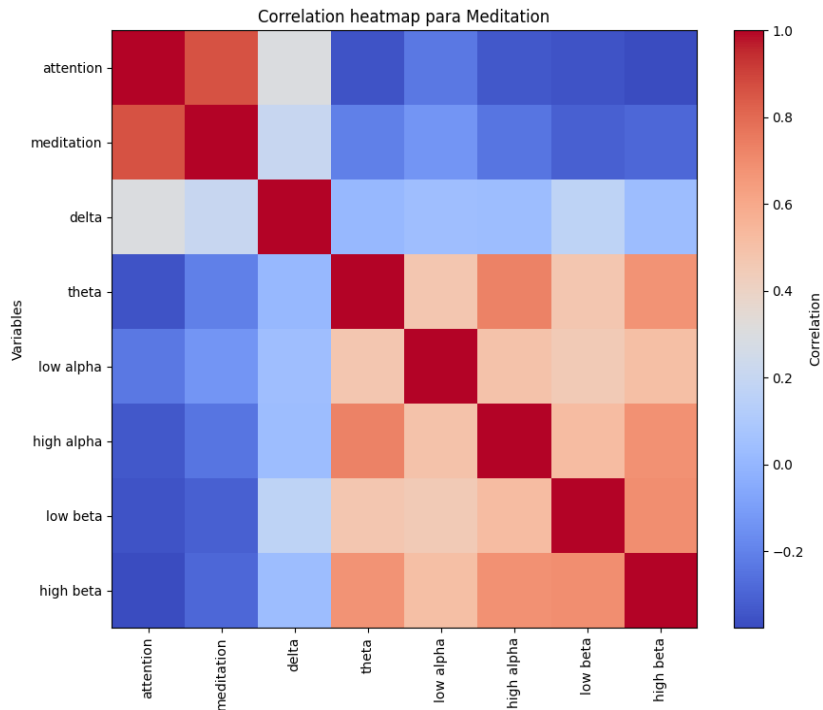


Figura 39. Mapa de calor para tablas de correlación en la actividad Meditación

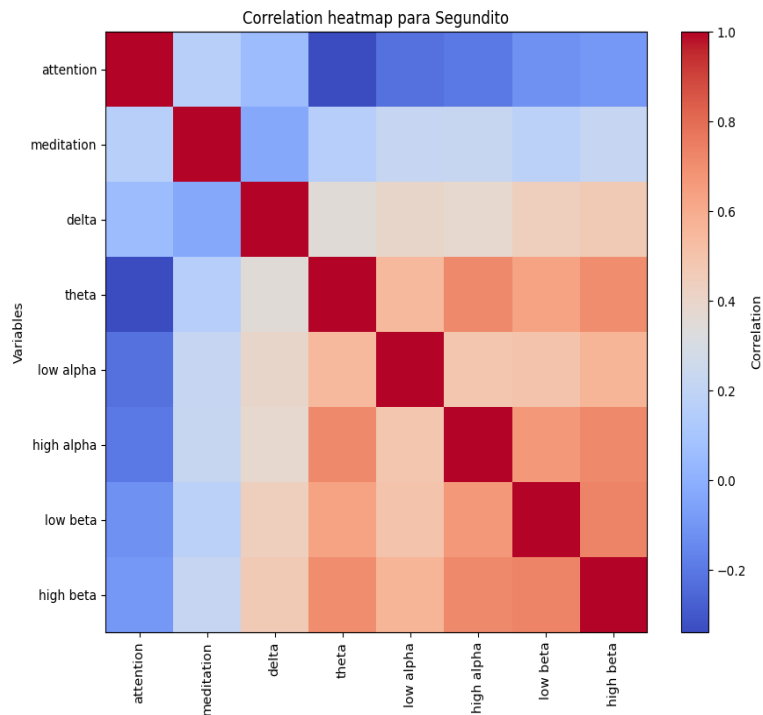


Figura 40. Mapa de calor para tablas de correlación en la actividad Segundito

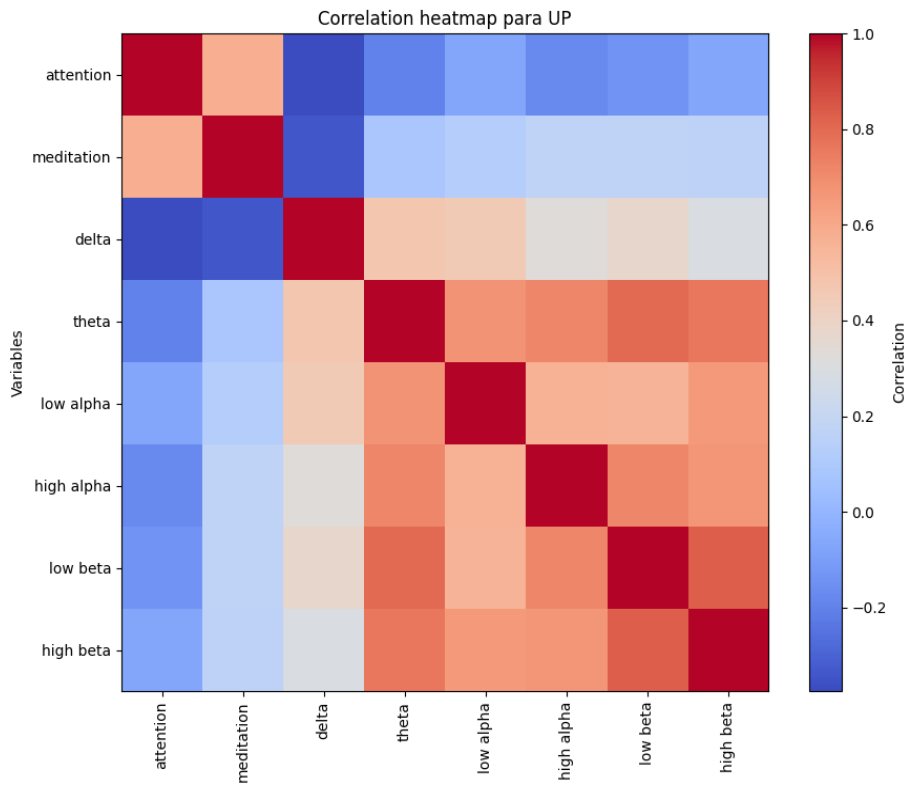


Figura 41. Mapa de calor para tablas de correlación en la actividad UP

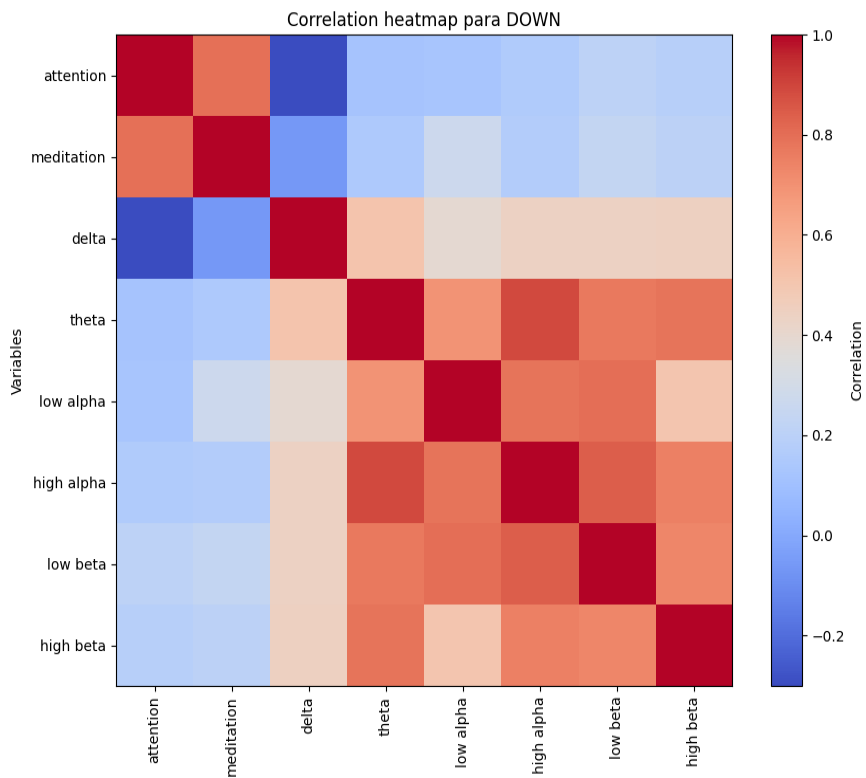


Figura 42. Mapa de calor para tablas de correlación en la actividad Down

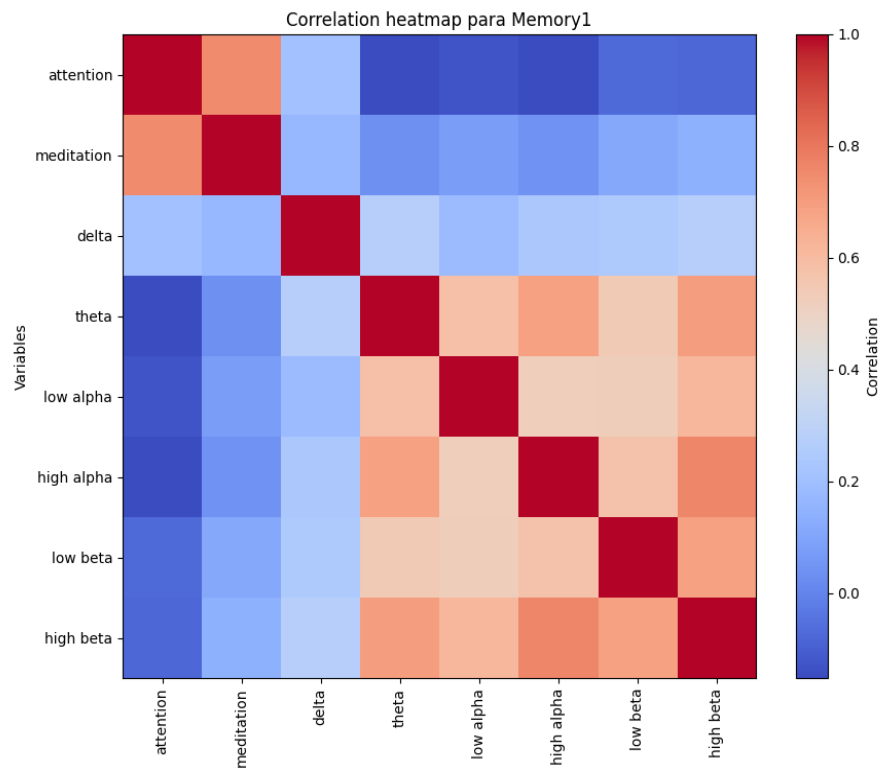


Figura 44. Mapa de calor para tablas de correlación en la actividad Memory1

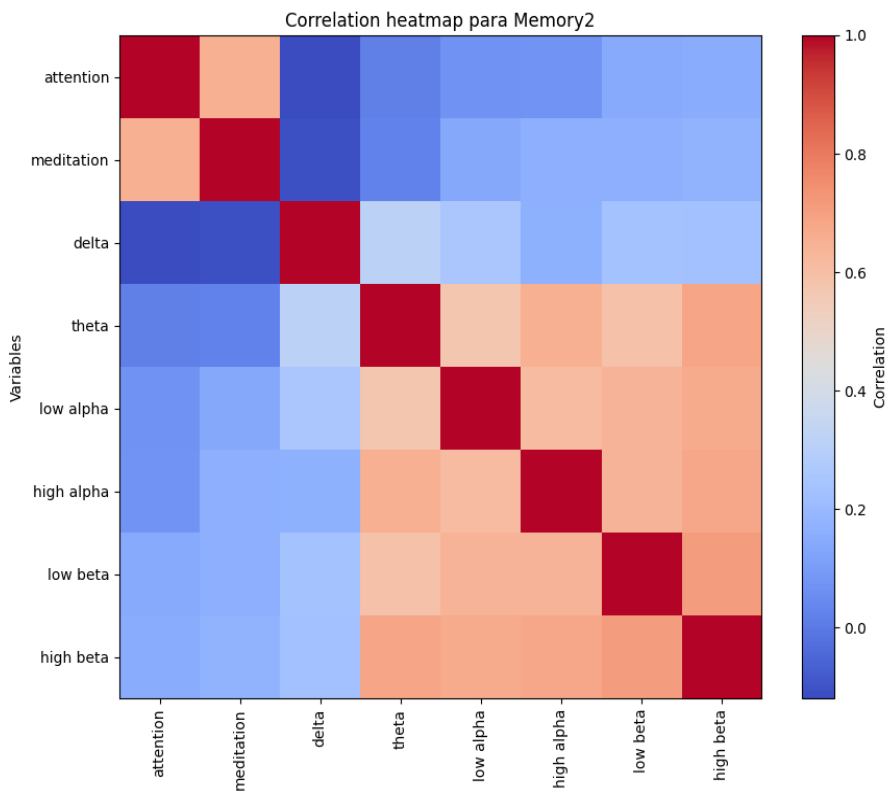
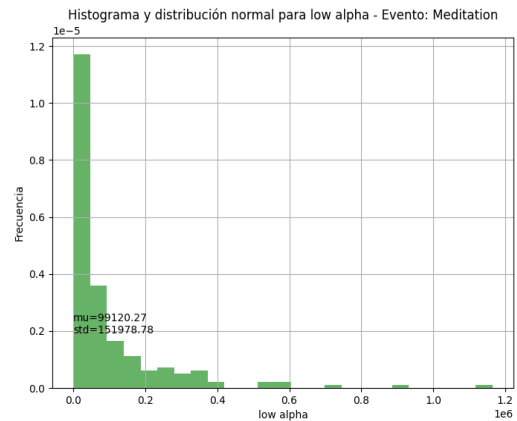
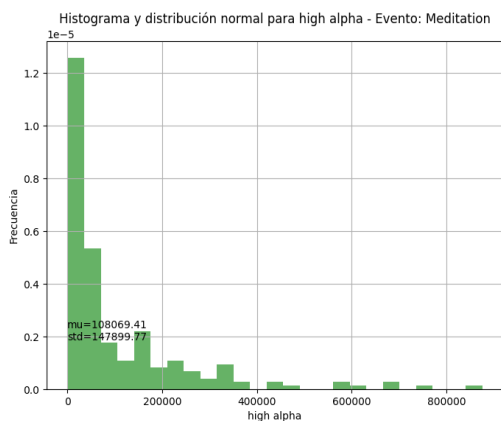
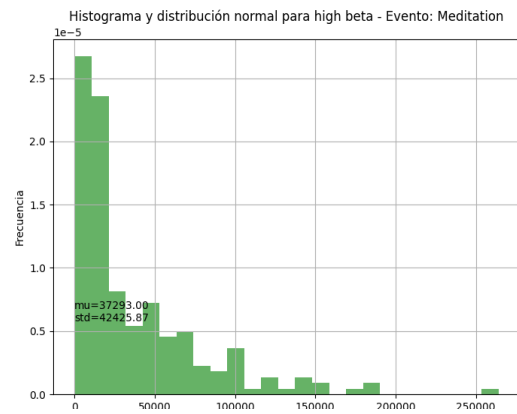
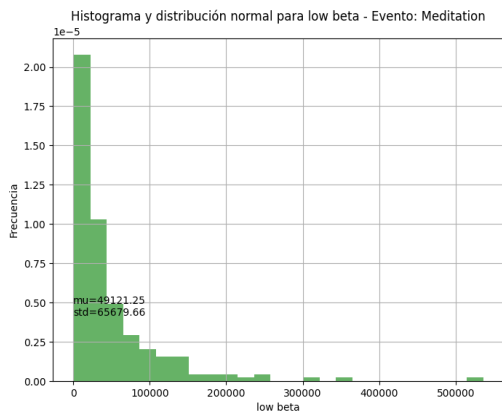
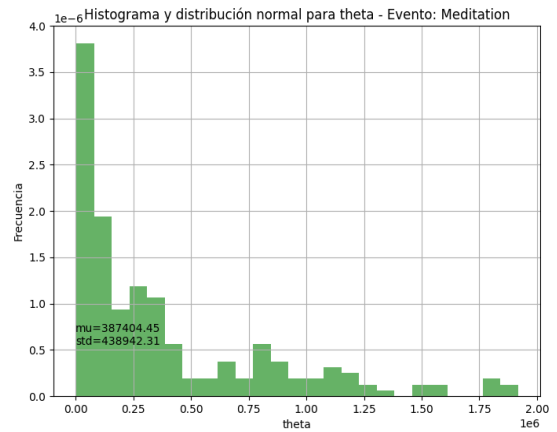
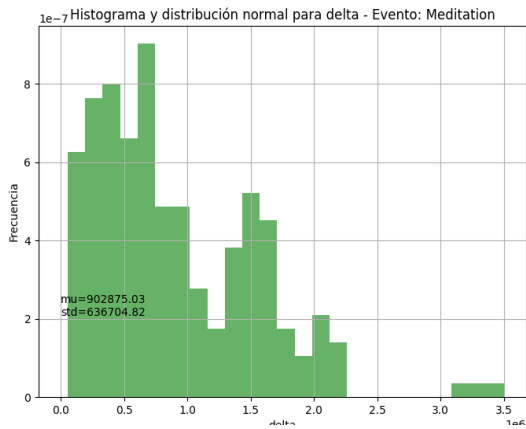
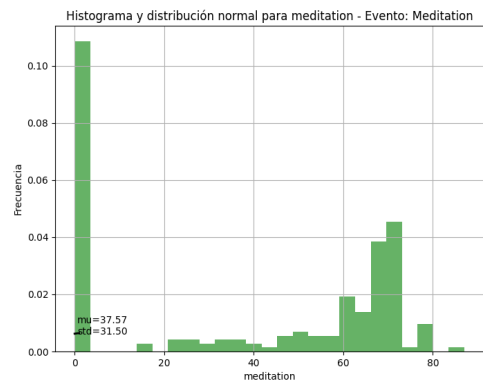
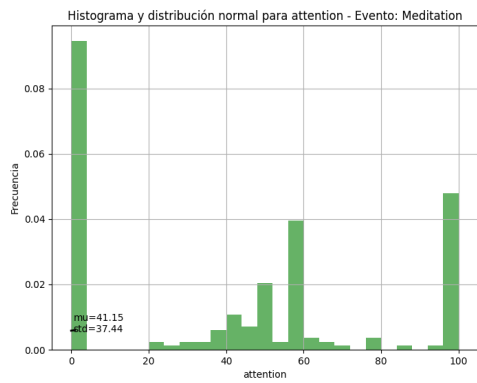


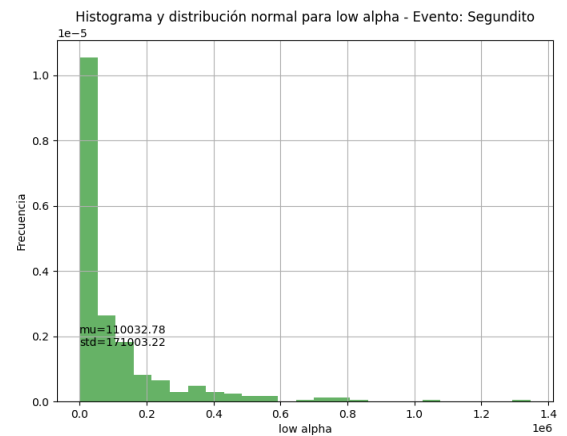
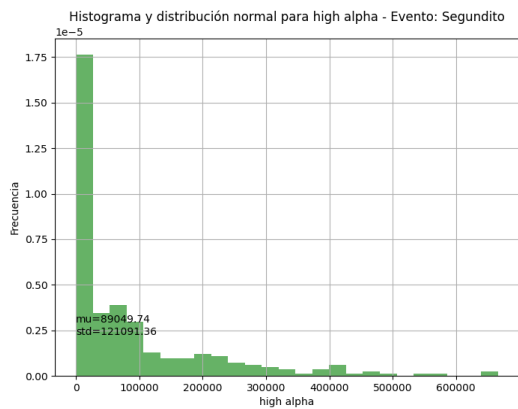
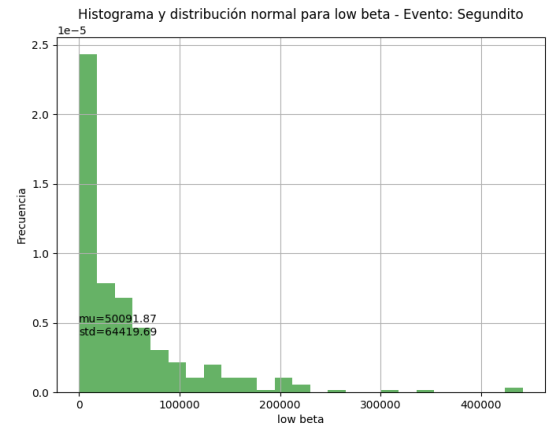
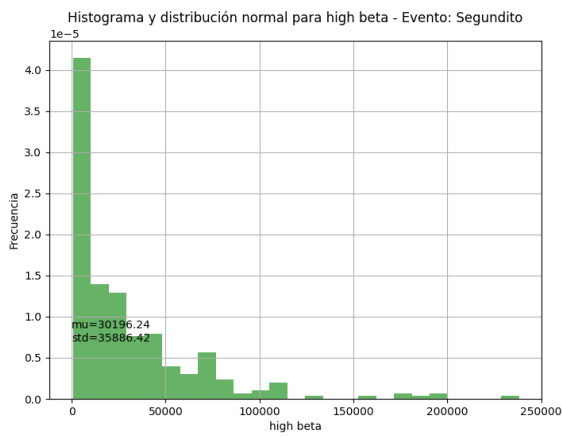
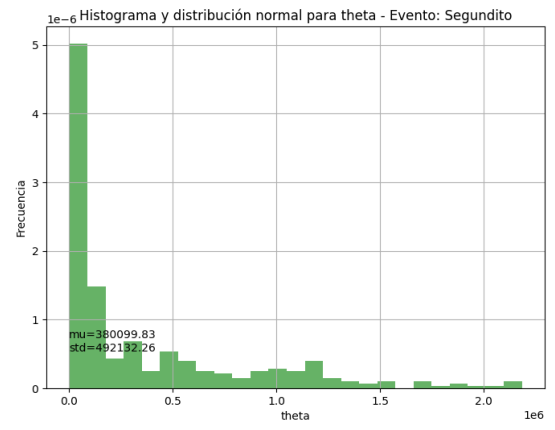
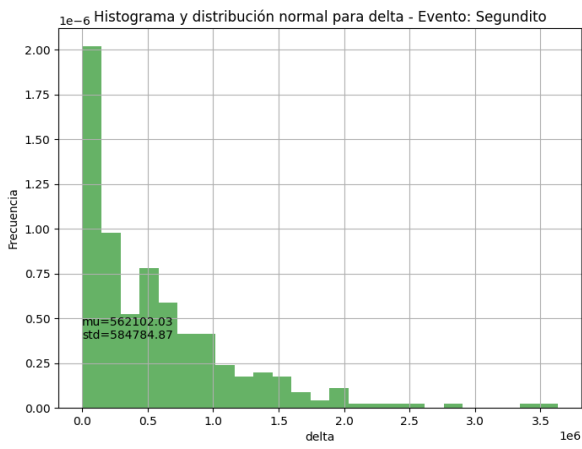
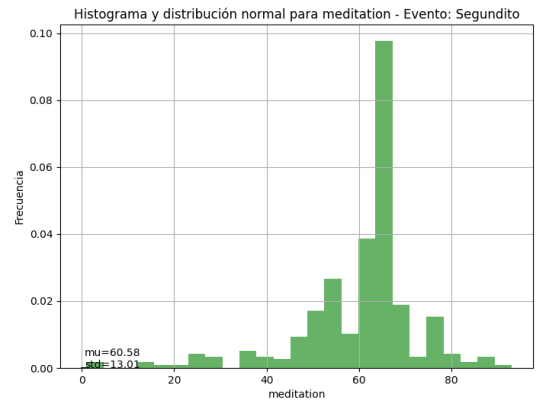
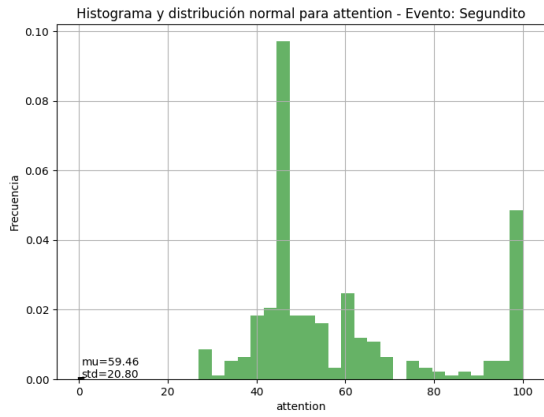
Figura 43. Mapa de calor para tablas de correlación en la actividad Memory2

Anexo VIII. Histogramas para las distintas actividades evaluadas

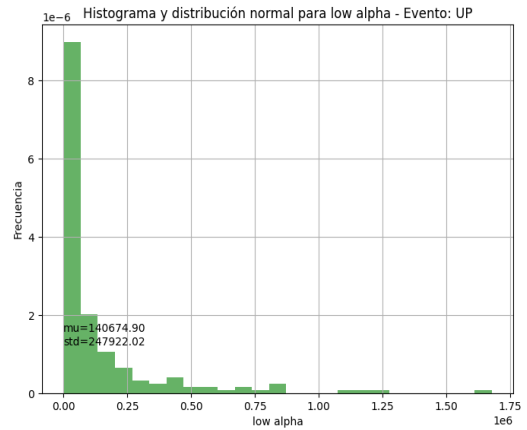
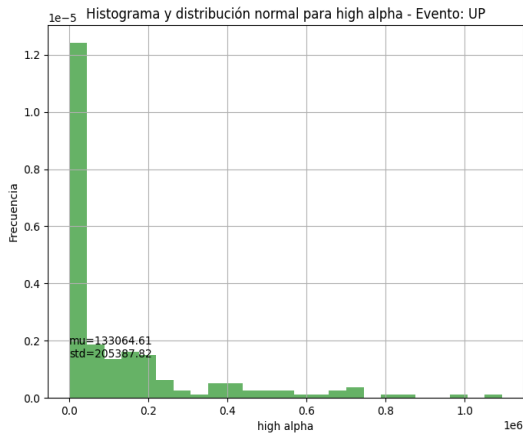
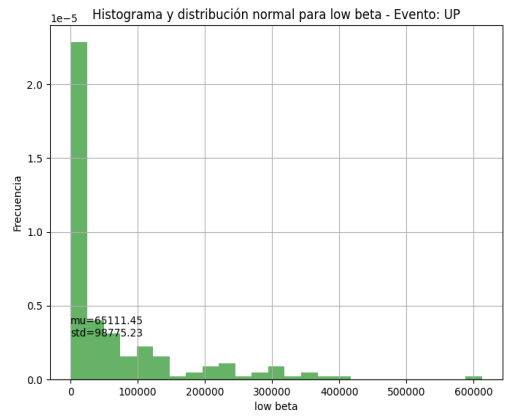
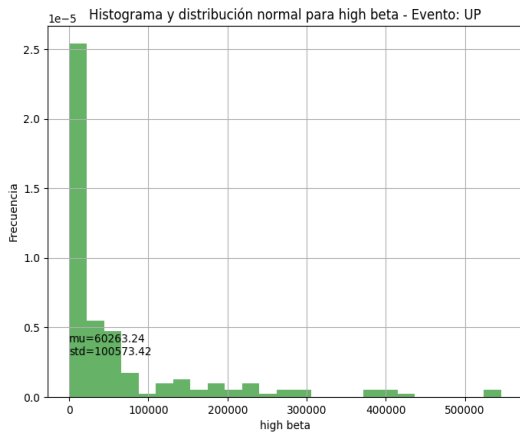
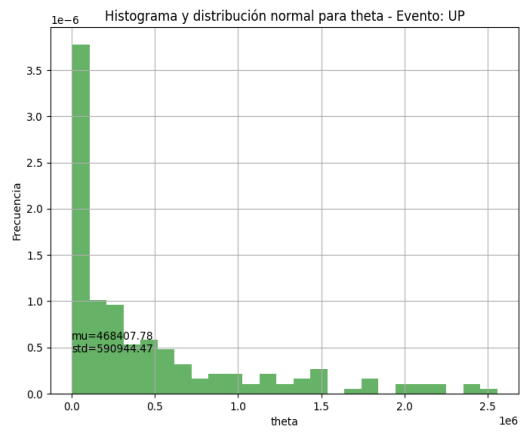
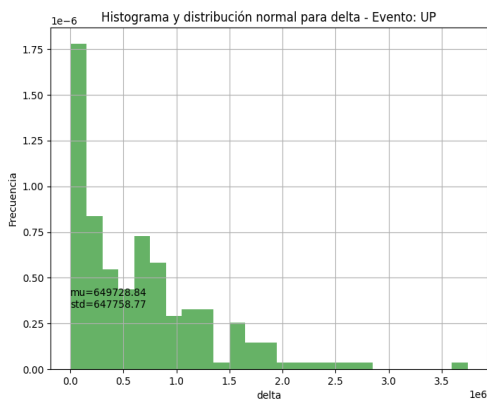
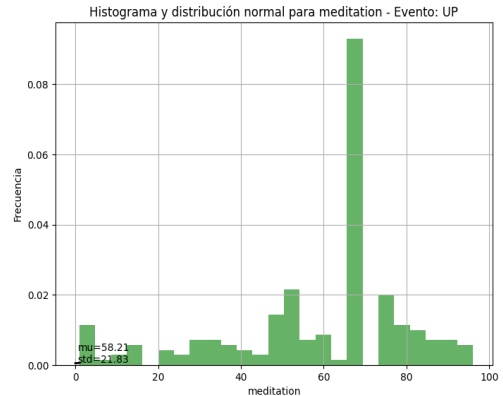
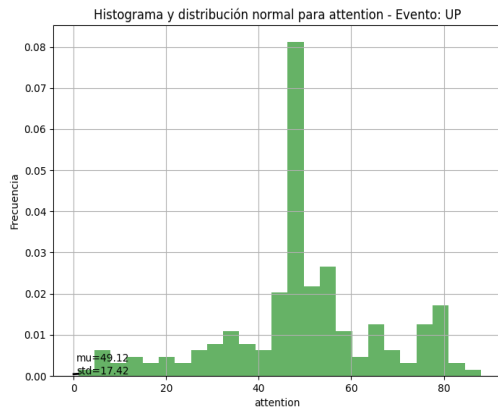
Meditation



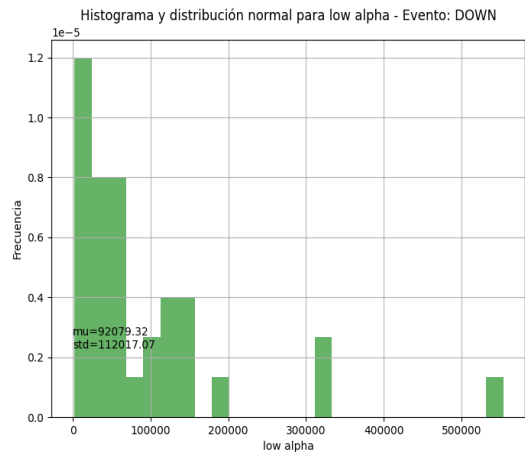
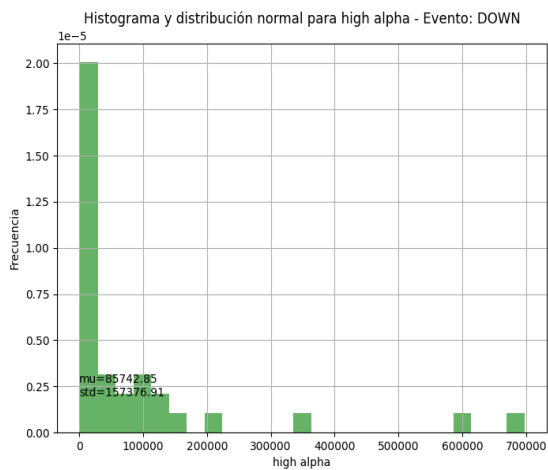
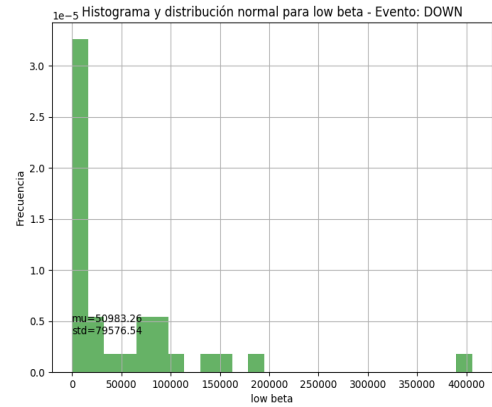
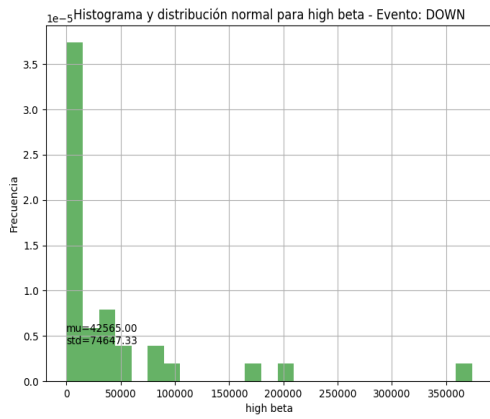
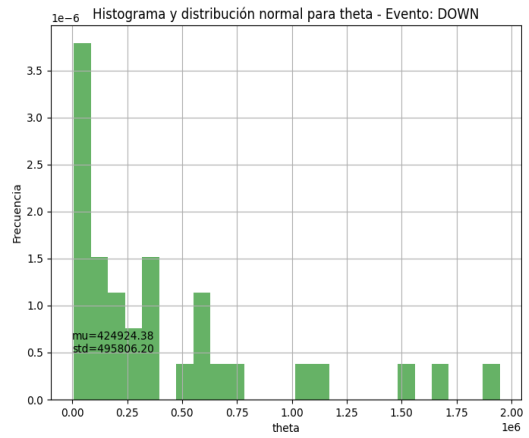
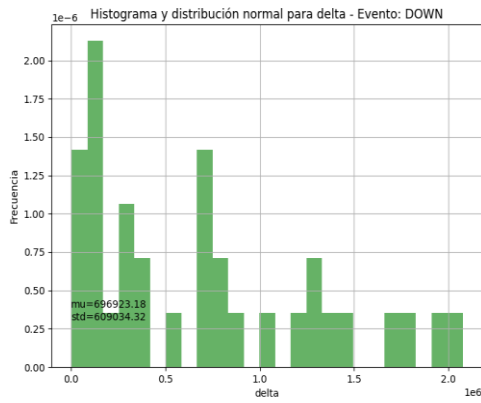
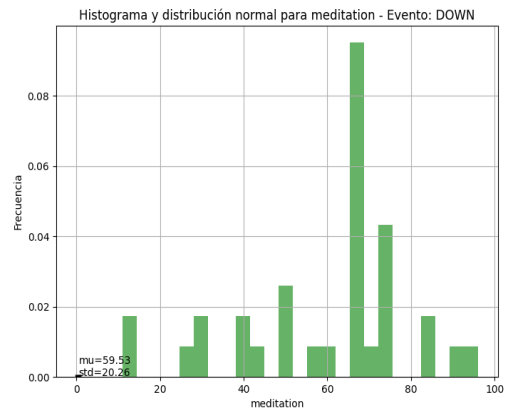
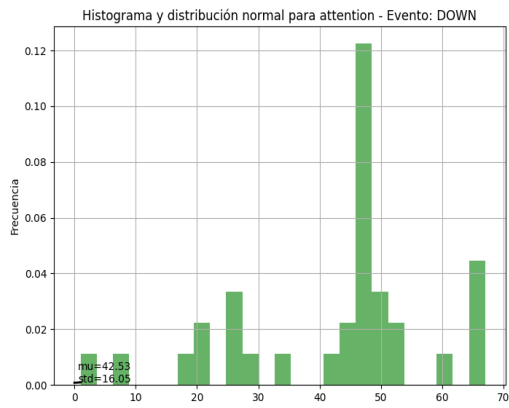
Segundito



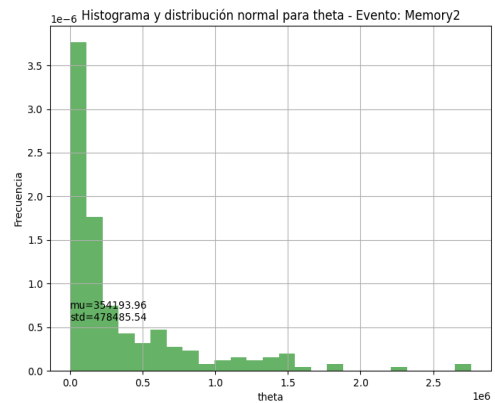
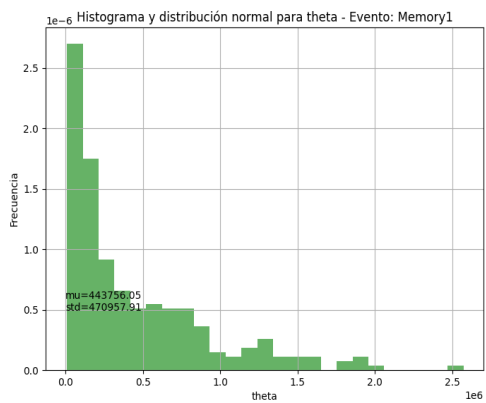
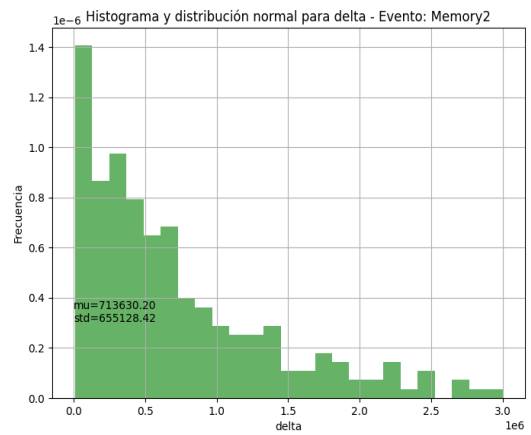
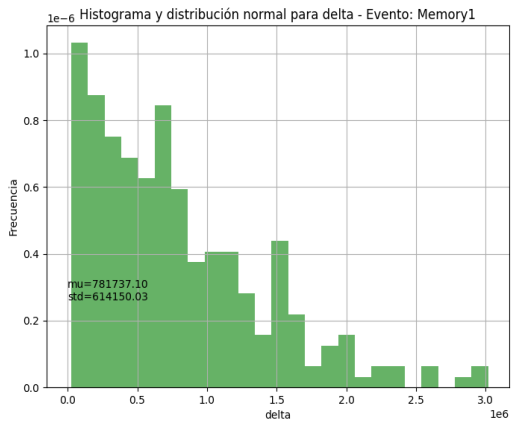
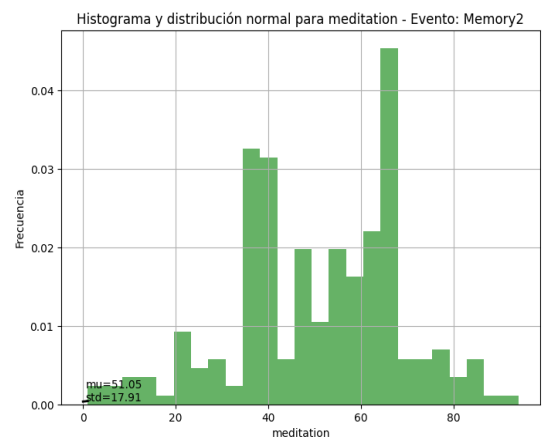
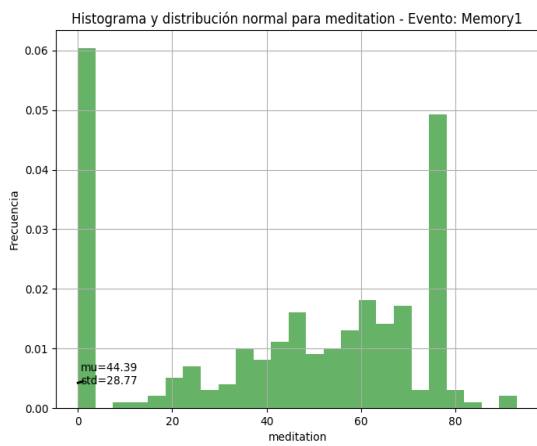
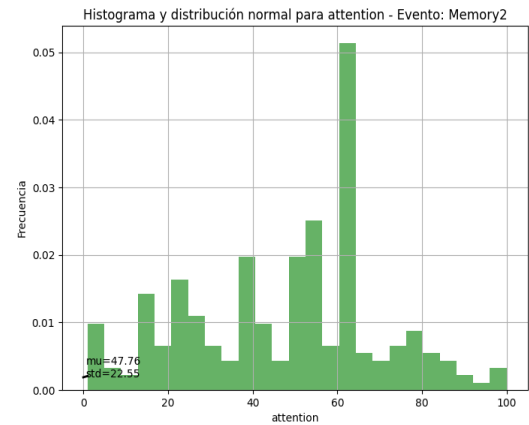
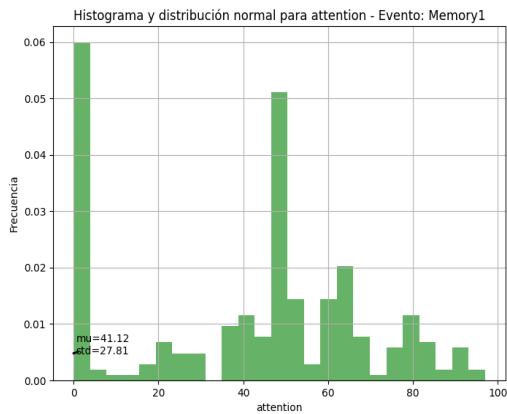
UP

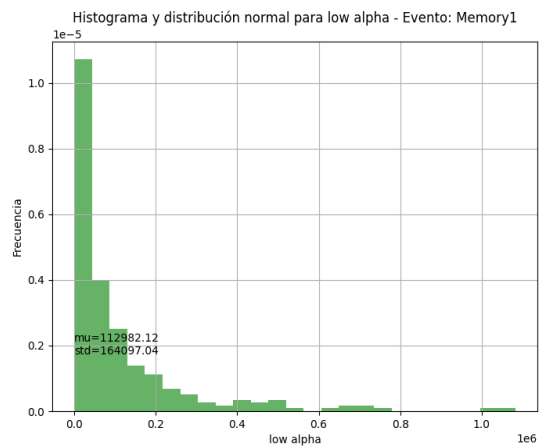
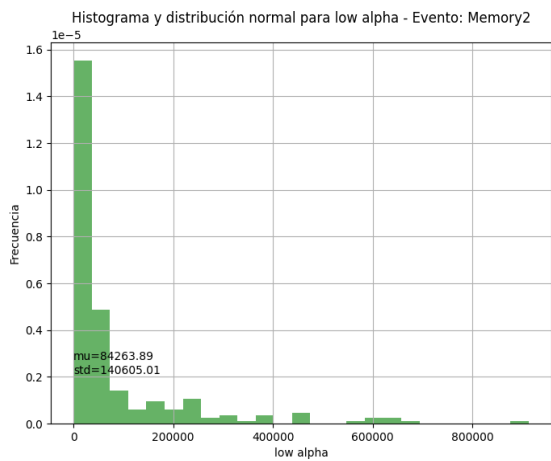
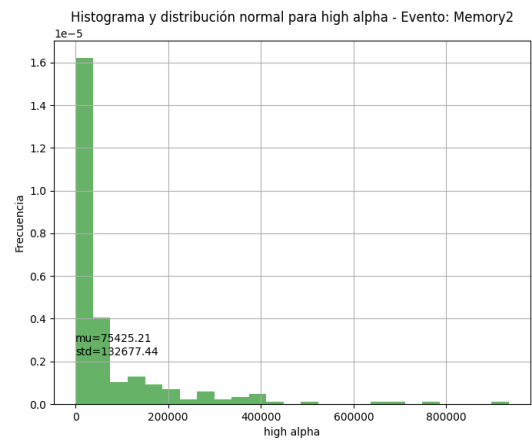
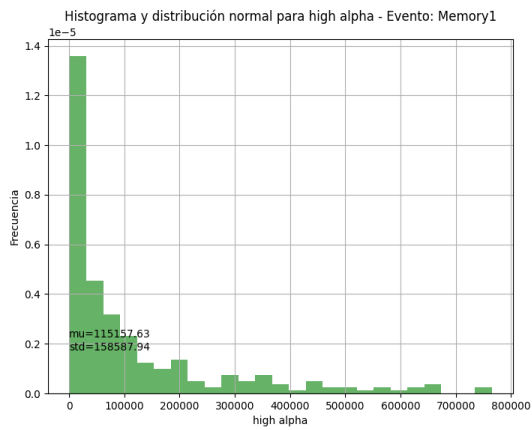
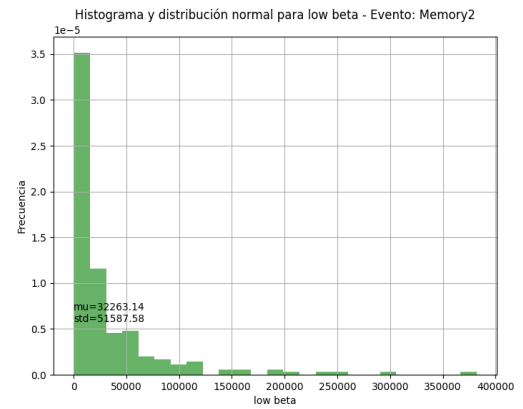
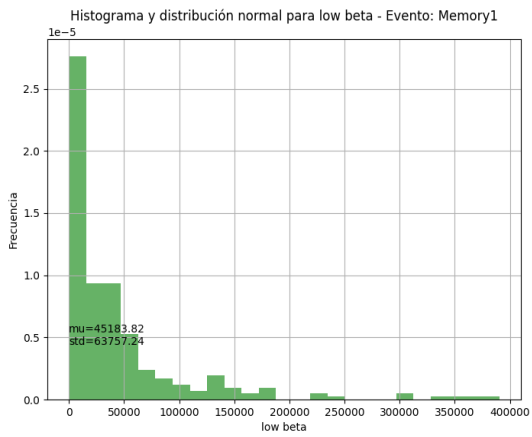
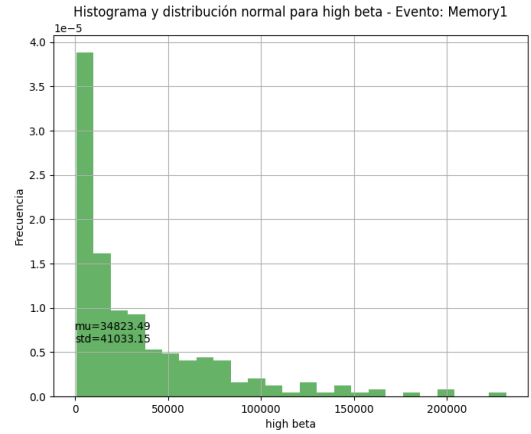
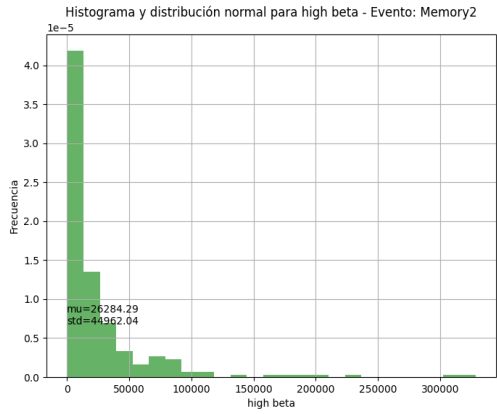


DOWN

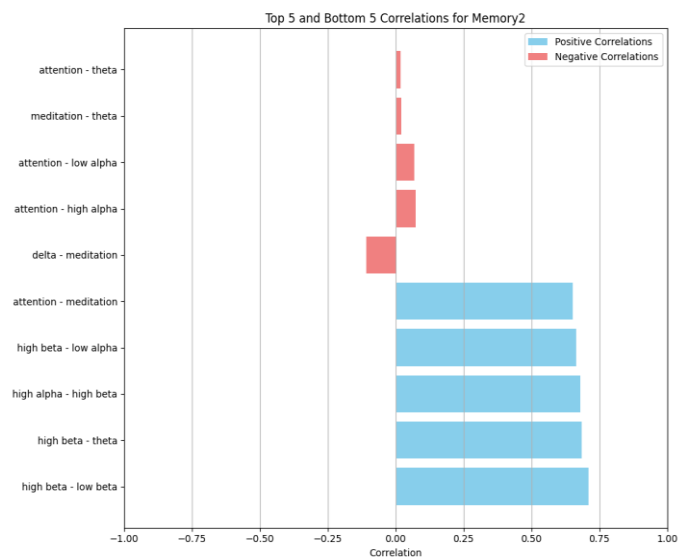
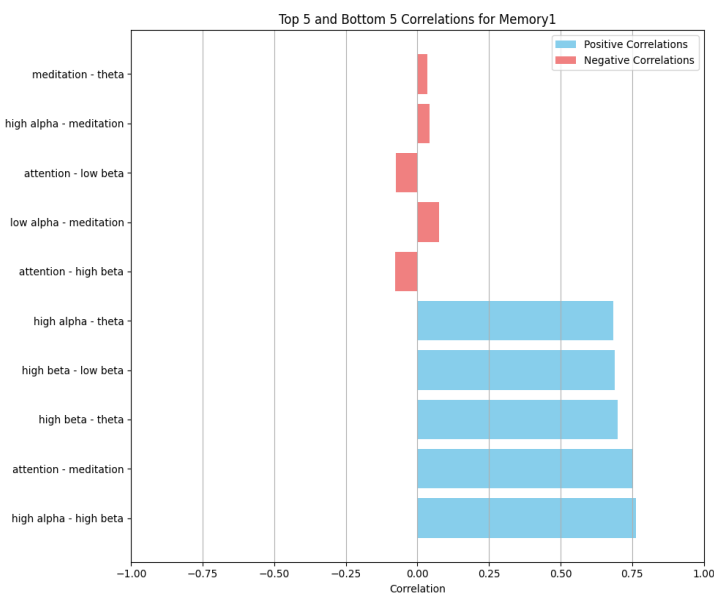
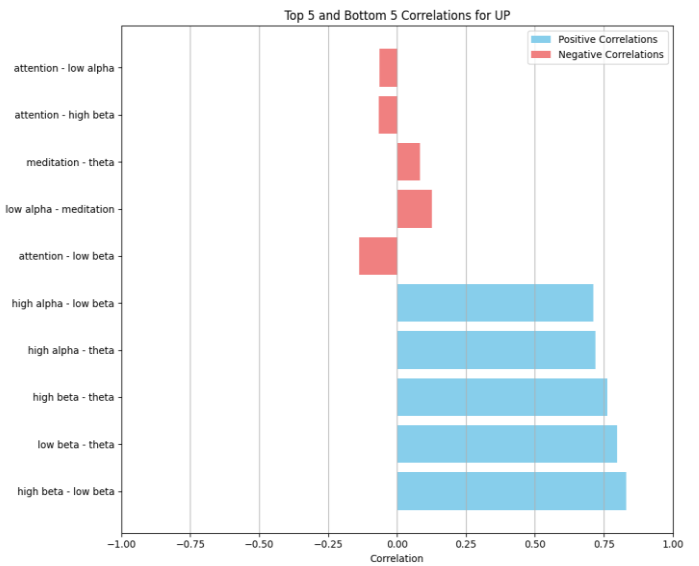
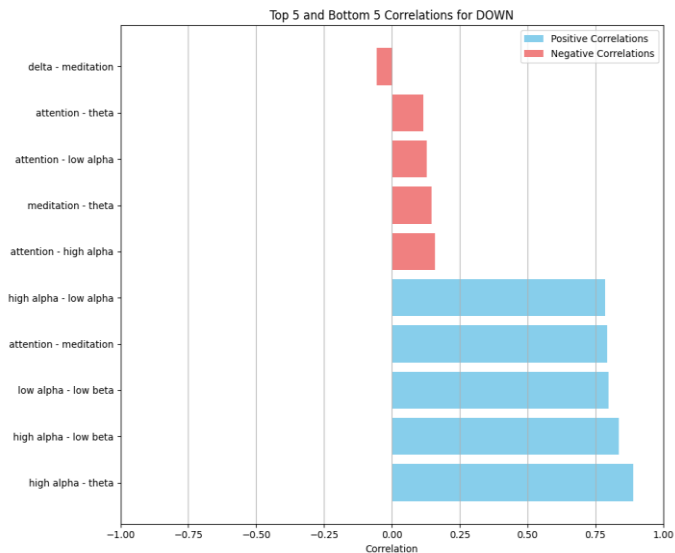
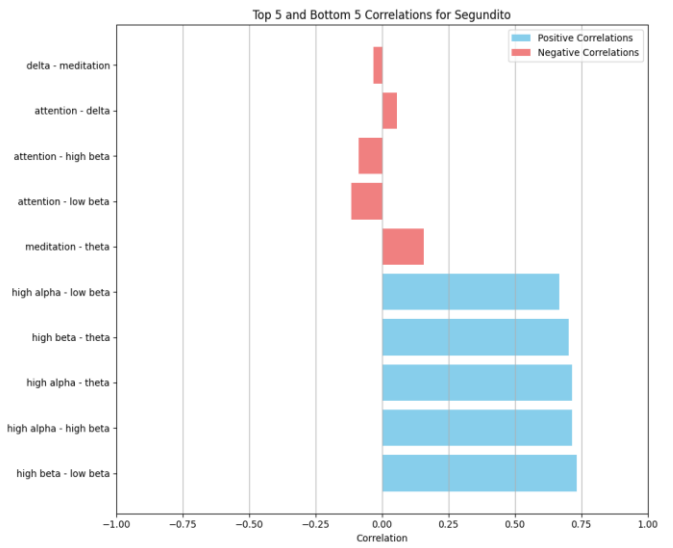
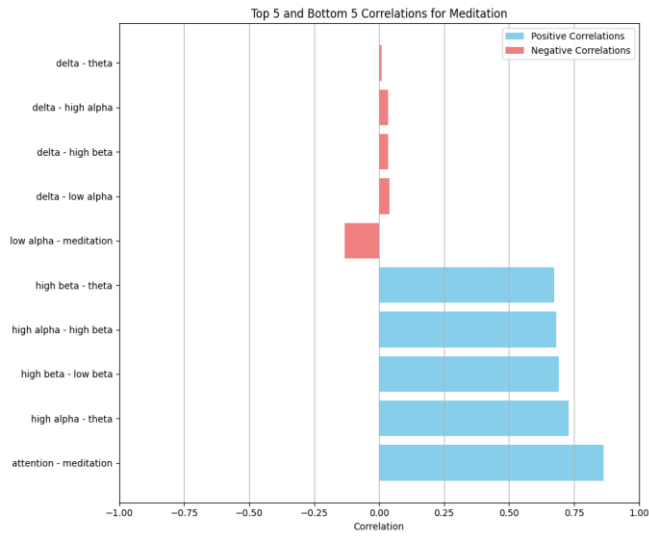


Memory1/Memory2





Anexo IX. Tablas de correlación para las variables más significativas



Anexo X. Código Python 'Abduce las vacas'

Play.py

```
import pygame
import os
import random
import sys
import threading
import logging

# Configurar el módulo logging
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s - %(levelname)s -
%(message)s')

# Agregar la ruta del directorio 'python' al sys.path
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), '..',
'python'))))

from ReceivingData import receiveData

pygame.init()
pygame.font.init()

# Global Constants
SCREEN_HEIGHT = 600
SCREEN_WIDTH = 1100
DEATH_COUNT = 0
POINTS = 0
SCREEN = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
pygame.display.set_caption("Abduce las vacas")

# Imágenes del OVNI
RUNNING = [pygame.image.load(os.path.join("Game/Assets/Ovni", "OvniRun1.png")),
           pygame.image.load(os.path.join("Game/Assets/Ovni", "OvniRun2.png"))]

COW1 = pygame.image.load(os.path.join("Game/Assets/Cow", "Cow1.png"))
COW2 = pygame.image.load(os.path.join("Game/Assets/Cow", "Cow2.png"))

BACKGROUND = pygame.image.load(os.path.join("Game/Assets/Other",
"background2.jpg"))
BACKGROUND = pygame.transform.scale(BACKGROUND, (SCREEN_WIDTH, SCREEN_HEIGHT))
BACKGROUND.set_alpha(220) # Establecer opacidad

class Ovni:
    X_POS = 80
    Y_POS = 310
    MOVE_SPEED = 10

    def __init__(self):
        logging.info("Inicializando el OVNI")
        self.run_img = RUNNING
        self.ovni_run = True
        self.step_index = 0
        self.image = self.run_img[0]
        self.ovni_rect = self.image.get_rect()
        self.ovni_rect.x = self.X_POS
        self.ovni_rect.y = self.Y_POS

        logging.info("Iniciando receiveData")
```

```

        self.receiver = receiveData()

def update(self):
    if self.ovni_run:
        self.run()

    if self.step_index >= 10:
        self.step_index = 0

    if self.receiver.getInterpretation(): # Si atención es True
        if self.ovni_rect.top > 0:
            self.ovni_rect.y -= self.MOVE_SPEED
            logging.debug(f"OVNI moviéndose hacia arriba:
{self.ovni_rect.y}")
        elif not self.receiver.getInterpretation(): # Si atención es False
            if self.ovni_rect.bottom < SCREEN_HEIGHT:
                self.ovni_rect.y += self.MOVE_SPEED
                logging.debug(f"OVNI moviéndose hacia abajo: {self.ovni_rect.y}")

def run(self):
    self.image = self.run_img[self.step_index // 5]
    self.step_index += 1

def draw(self, SCREEN):
    SCREEN.blit(self.image, self.ovni_rect.topleft)

class Obstacle:
    def __init__(self, image):
        logging.info("Inicializando un obstáculo")
        self.image = image
        self.rect = self.image.get_rect()
        self.rect.x = SCREEN_WIDTH

    def update(self):
        self.rect.x -= game_speed
        if self.rect.x < -self.rect.width:
            logging.debug("El obstáculo ha salido de la pantalla")
            obstacles.pop()

    def draw(self, SCREEN):
        SCREEN.blit(self.image, self.rect)

class Cow(Obstacle):
    def __init__(self, image):
        super().__init__(image)
        self.rect.y = random.randint(50, 550)

def main():
    global game_speed, obstacles, POINTS, DEATH_COUNT
    logging.info("Iniciando el juego")
    POINTS = 0
    DEATH_COUNT = 1

    run = True
    clock = pygame.time.Clock()
    player = Ovni()
    obstacles = []
    game_speed = 15
    font = pygame.font.Font("freesansbold.ttf", 20)

```

```

# Ejecutar receiver.execute() en un hilo separado
logging.info("Iniciando hilo")
receiver_thread = threading.Thread(target=player.receiver.execute)
receiver_thread.start()

calibrationScreen(player)

def score(addPoints):
    global POINTS, game_speed
    if addPoints:
        POINTS += 10

    text = font.render("Points: " + str(POINTS), True, (255, 255, 255))
    SCREEN.blit(text, (960, 40))

def background():
    temp_background = BACKGROUND.copy()
    temp_background.set_alpha(220)
    SCREEN.blit(temp_background, (0, 0))

logging.info("While Run")
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            logging.info("Saliendo del juego")
            run = False
            player.receiver.stopReading()

    background()

    if player.receiver.getNewData():
        player.update()
        player.draw(SCREEN)

        if len(obstacles) == 0:
            if random.randint(0, 1) == 0:
                obstacles.append(Cow(COW1))
            elif random.randint(0, 1) == 1:
                obstacles.append(Cow(COW2))

        for obstacle in obstacles:
            obstacle.draw(SCREEN)
            obstacle.update()
            if player.ovni_rect.collidect(rect(obstacle.rect)):
                pygame.draw.rect(SCREEN, (250, 0, 0), player.ovni_rect, 2)
                score(True)

        score(False)
        player.receiver.updateNewData() # Resetea new_data después de la
actualización

        pygame.display.update()
        clock.tick(30)

player.receiver.stopReading()
receiver_thread.join()

```

```

def calibrationScreen(player):
    countdown = 3 # Duración de la cuenta regresiva en segundos
    while player.receiver.getCalibration() > 1:
        for i in range(countdown, 0, -1):
            SCREEN.fill((255, 255, 255))
            font = pygame.font.Font('freesansbold.ttf', 30)
            text = font.render(f"The game will start in... {i}", True, (0, 0, 0))
            textRect = text.get_rect()
            textRect.center = (SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2)
            SCREEN.blit(text, textRect)
            pygame.display.update()
            pygame.time.wait(500) # Esperar 1 segundo

        SCREEN.fill((255, 255, 255))
        font = pygame.font.Font('freesansbold.ttf', 30)
        text = font.render("Let's Play!", True, (0, 0, 0))
        textRect = text.get_rect()
        textRect.center = (SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2)
        SCREEN.blit(text, textRect)
        pygame.display.update()

def menuScreen():
    global POINTS, DEATH_COUNT
    run = True
    while run:
        SCREEN.fill((255, 255, 255))
        font = pygame.font.Font('freesansbold.ttf', 30)

        if DEATH_COUNT == 0:
            text = font.render("Press any Key to Start", True, (0, 0, 0))
        elif DEATH_COUNT > 0:
            text = font.render("Press any Key to Restart", True, (0, 0, 0))
            score = font.render("Your Score: " + str(POINTS), True, (0, 0, 0))
            scoreRect = score.get_rect()
            scoreRect.center = (SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2 + 50)
            SCREEN.blit(score, scoreRect)
        textRect = text.get_rect()
        textRect.center = (SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2)
        SCREEN.blit(text, textRect)
        SCREEN.blit(RUNNING[0], (SCREEN_WIDTH // 2 - 20, SCREEN_HEIGHT // 2 -
140))
        pygame.display.update()
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                logging.info("Saliendo del menú")
                pygame.quit()
                run = False
                return
            if event.type == pygame.KEYDOWN:
                logging.info("Iniciando el juego desde el menú")
                main()

    pygame.quit()

menuScreen()

```

receivingData.py

```
import serial
from datetime import datetime
import os
import logging

# Configurar el módulo logging
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s - %(levelname)s -
%(message)s')

# Configurar el puerto serie
ser = serial.Serial('COM5', 9600)

class receiveData:

    def __init__(self):
        self.folder_path = "D:\\OneDrive\\URV\\5º Curso\\TFG
BrainLink\\BrainLink"

        # Lista para almacenar los datos recibidos
        self.data_list = []
        self.attention_values = []
        self.meditation_values = []

        self.calibration_count = 4

        self.fin = False

        # Variable para indicar nuevos datos
        self.new_data = False

        # Variable para interpretación
        self.interpretation = None

        logging.info("Inicializando receiveData")

    def execute(self):
        while not self.fin:
            """
            "Signal strength, attention, meditation, delta, theta, low alpha,
            high alpha, low beta, high beta, low gamma, high gamma"

            Signal strength ranges de 0 a 200. Contrariamente a la intuición, 0
            significa que la unidad se ha conectado correctamente,
            y 200 significa que no hay señal.

            IMPORTANTE: descartar siempre los primeros 3 valores de la lectura y
            tomarlos como calibrado.
            Además, filtrar el rango de la señal.
            """
            logging.info("Try")
            try:
                # Leer una línea de datos desde el puerto serie
                data = ser.readline().decode().strip()

                # Filtrar datos vacíos y procesar los datos
                logging.info("Waiting data")
                if data:
```

```

logging.info(f"Datos recibidos desde Arduino: {data}")

# Convertir los valores a números enteros o flotantes
values = data.split(',')
attention = int(values[1])
meditation = int(values[2])

if self.calibration_count > 1:
    self.calibration_count = self.calibration_count - 1
    logging.info(f"Calibración en progreso...
{self.calibration_count}")
    continue

# Filtrar por rango de la señal (por ejemplo, 0-200)
if values[0] != 200:
    # Almacenar los valores de atención y meditación
    self.saveValues(attention, meditation, data)
    self.new_data = True
    logging.info("new_data se ha actualizado a True")

    # Interpretar los niveles de concentración
    # concentration levels lower than 40 are equivalent to
the word "NO"
    # If the concentration is higher than 51, it is
equivalent to the word "YES"
    if attention < 46:
        self.interpretation = False
    elif attention > 51:
        self.interpretation = True
    # else: se mantiene el estado anterior

except UnicodeDecodeError as e:
    logging.error(f"Error de decodificación: {e}")
    continue # Continúa con la siguiente iteración del bucle

# Asegurarse de que el puerto serie se cierre correctamente
if ser.is_open:
    ser.close()
    logging.info("Puerto serie cerrado")

# Guardar los datos recibidos en un archivo CSV
self.guardarCSV()

def guardarCSV(self):
    if self.data_list:
        if not os.path.exists(self.folder_path):
            os.makedirs(self.folder_path)

        file_name = os.path.join(self.folder_path, "data_" +
datetime.now().strftime('%Y_%m_%d_%H_%M_%S') + ".csv")
        with open(file_name, 'w') as file:
            for item in self.data_list:
                file.write(f"{item}\n")
            logging.info(f"Datos guardados en {file_name}")

def saveValues(self, attention, meditation, data):
    self.attention_values.append(attention)
    self.meditation_values.append(meditation)

```

```
        logging.info(f"Atención: {attention}, Meditación: {meditation}")

        self.data_list.append(data)

    def updateNewData(self):
        self.new_data = False
        logging.info("new_data se ha actualizado a False")

    def getInterpretation(self):
        return self.interpretation

    def getNewData(self):
        return self.new_data

    def stopReading(self):
        self.fin = True

    def getCalibration(self):
        return self.calibration_count

# Uso individual:
if __name__ == "__main__":
    receiver = receiveData()
    receiver.execute()
```

