



Uso de Machine Learning para detectar señales cerebrales de tipo P300 generando estímulos visuales y auditivos.

Alejandro Jesús Perdomo Cely.

Autor

Camilo Ernesto Pardo Beainy, MSc. Ph. D(c).

Director de tesis

Moshé Alonso Amarillo, Mg.

Codirector de tesis

Facultad de Ingeniería Electrónica, Universidad Santo Tomás seccional Tunja.

2 de abril de 2024.

CONTENIDO

1. Estudio Teórico del Proyecto:	14
1.1. Introducción:	14
1.2. Planteamiento del problema:	14
1.3. Justificación:	17
1.3.1. Impacto en Calidad de Vida:	17
1.3.2. Impacto Biomédico:	17
1.3.3. Impacto Neurocientífico:	17
1.4. Objetivos:	18
1.4.1. General:	18
1.4.2. Específicos:	18
1.5. Marco teórico:	18
1.6. Estado del arte:	27
2. Desarrollo del proyecto:	29
2.1. Fase 1, Recolección de datos de entrenamiento:	29
2.2. Fase 2, Análisis y procesamiento de datos:	49
2.3. Fase 3, Creación del sistema de Machine Learning:	62
2.4. Fase 4, Validación y métricas de evaluación:	65
3. Análisis de resultados y discusión:	68

3.1.	F1 Score Datos de Validación:	69
3.2.	F1 Score Datos de Prueba:	71
4.	Conclusiones y trabajos futuros:	72
5.	Referencias bibliográficas:	74
6.	Anexos:	77
6.1.	Programas Matlab:	77
6.1.1.	EEGLAB:	77
6.1.2.	Clasificación P300:	77
6.1.3.	Acoplador de Datos:	77
6.1.4.	Aumento de datos:	78
6.1.5.	Filtrado de Señal:	78
6.1.6.	Creación de Ventanas:	78
6.1.7.	Visualización de Potenciales Evocados:	78
6.2.	Programa Jupyterlab:	79
6.2.1.	Dataset a analizar:	79
6.2.2.	Detector de la señal P300:	79
6.2.3.	Datasets limpios:	79
6.2.4.	Respaldo Automático:	79
6.3.	Programa Psychopy:	79
6.4.	Datasets Generados:	80



6.5.	Consentimientos Informados:.....	80
6.6.	Videos:.....	80
6.7.	Manual de Usuario:	81

CONTENIDO DE FIGURAS

Figure 1 Arquitectura general del proyecto.....	29
Figure 2 Diagrama de flujo de la fase 1 del proyecto	30
Figure 3 Primera mitad del casco EEG importado en Ultimaker Cura	31
Figure 4 Primera mitad del casco EEG importado y laminado en Ultimaker Cura	31
Figure 5 Impresora 3D de la marca Creality CR 6 SE usada para el proyecto	32
Figure 6 Casco de electroencefalografía impreso en 3D, desglosamiento 1.....	33
Figure 7 Casco de electroencefalografía impreso en 3D, desglosamiento 2.....	33
Figure 8 Casco de electroencefalografía impreso en 3D, desglosamiento 3.....	34
Figure 9 Casco de electroencefalografía impreso en 3D, desglosamiento 4.....	34
Figure 10 Casco de electroencefalografía impreso en 3D, desglosamiento 5.....	35
Figure 11 Casco EEG armado en su totalidad.....	37
Figure 12 Interfaz Gráfica de Usuario del programa OpenBCI	38
Figure 13 Interfaz Gráfica de Usuario creado con Psychopy para los participantes.....	41
Figure 14 Pantalla de Psychopy en estado de estabilidad visual (superior) y cambio visual para la generación de P300 (inferior).....	42
Figure 15 Participantes durante pruebas de generación de P300: 1/A (izquierda), 2/C (Derecha superior), 3/Ma (Derecha central) y 4/Mo (Derecha inferior)	43
Figure 16 Interfaz gráfica de LabRecorder	44
Figure 17 Interfaz Gráfica de Usuario de EEGLab basado en Matlab.....	45
Figure 18 Señales EEG con marcadores de no aparición (línea vertical roja con un cero) y aparición de P300 (línea vertical verde con un uno).....	46
Figure 19 Comparación presencia y ausencia de potencial evocado en electrodo O1.....	47

Figure 20 Comparación presencia y ausencia de potencial evocado en electrodo O2.....	48
Figure 21 Diagrama de flujo fase 2	49
Figure 22 Señal P300 ideal artificial	50
Figure 23 Señal real ejemplificando la ausencia de P300.....	51
Figure 24 Señal real ejemplificando la presencia de P300.....	52
Figure 25 Características 1 y 2 de la extracción.....	53
Figure 26 Característica 3 de la extracción	54
Figure 27 Característica 4 de la extracción	55
Figure 28 Característica 5 de la extracción	56
Figure 29 Característica 6 de la extracción	58
Figure 30 Dataset con sus correspondientes características extraídas	59
Figure 31 Tamaño dataset sin data augmentation	59
Figure 32 Histogramas comparando entre columna de datos del dataset original (rojo) con el dataset aumentado (azul).	61
Figure 33 Diagrama de flujo fase 3	62
Figure 34 Importación del dataset a JupyterLab	63
Figure 35 Separación de dataset entre entrenamiento, validación y prueba	64
Figure 36 Modelos de Machine Learning con los datos de entrenamiento: LR (izquierda superior), Tree (derecha superior), SVM (izquierda inferior y KNN (derecha inferior).....	64
Figure 37 Diagrama de flujo fase 4.....	65
Figure 38 Matrices de confusión datos de validación (superiores) y datos de prueba (inferiores) con Regresión Logística (Verdes), Árbol de decisión (Morados), Máquina de Soporte Vectorial (Rojos) y K Vecinos más Cercanos (Azules) de participante 1/A.	66



Figure 39 Porcentaje Puntaje F1 de los cuatro participantes para datos de validación.....69

Figure 40 Porcentaje Puntaje F1 de los cuatro participantes para datos de prueba71

CONTENIDO DE ILUSTRACIONES

Ilustración 1 Señal con P300 (azul) y señal sin P300 (rojo) de una lectura EEG	15
Ilustración 2 Diagrama de tiempo de estímulos visuales	21
Ilustración 3 Regresión Logística.....	24
Ilustración 4 Árbol de decisión	24
Ilustración 5 Máquina de soporte vectorial	25
Ilustración 6 K Vecinos más Cercanos	26
Ilustración 7 Ejemplo de data augmentation.....	27
Ilustración 8 Distribución 10-20 real del proyecto (encerrados en rojo)	39



CONTENIDO DE TABLAS

Tabla 1 Elementos que conforman el hardware del proyecto de grado	37
Tabla 2 Distribución física de los electrodo correspondientes a su orden de OpenBCI	40



EXONERACIÓN DE RESPONSABILIDADES:

Se redacta que la Universidad Santo Tomás se ve exonerada de responsabilidades ante cualquier delito de plagio, debido a que toda la responsabilidad del documento redactado recae sobre el autor de este proyecto.



DEDICATORIA:

A mis padres, quienes me han apoyado en todo aspecto, de inicio a fin, en este proceso académico, que gracias al gran esfuerzo de ellos, he podido llegar muy lejos en mi formación.

A mis abuelos, quienes estando desde la lejanía me han apoyado de todo corazón en lograr terminar este proyecto de grado.

A mi hermana, quien ha estado para mí en las buenas y en las malas, sin importar la distancia o el tiempo.



AGRADECIMIENTOS:

A mi director de grado, quien me colaboró con todo lo relacionado con revisión, dirección y apoyo durante la creación de este documento.

A mi codirector de grado, quien me recibió en la Universidad de Los Andes, y me colaboró con el estudio de la temática principal de esta tesis, junto con material útil para el proyecto.

A mis docentes, quienes me guiaron durante toda la carrera para llegar a adquirir los conocimientos que hoy tengo.

A los participantes de este proyecto de grado, quienes me obsequiaron de su tiempo para poder realizar este estudio.

Y a todas las personas que me han apoyado y colaborado en este proceso, haya sido de manera directa o indirecta.

RESUMEN:

Existen distintas técnicas que permiten medir la actividad en tiempo real. Una de las más fácilmente implementables en humanos es la electroencefalografía. A partir de esta técnica, se han desarrollado distintas aplicaciones que permiten clasificar distintos estados cerebrales. En el presente proyecto implementamos un método de clasificación de potenciales evocados visuales P300. El proyecto inició con la impresión del casco electroencefalográfico, este sostiene 16 electrodos superficiales que son colocados en la superficie de la cabeza, con el fin de poder medir la actividad cerebral.

Se registraron cuatro participantes, quienes completaron una tarea que permitió desencadenar el potencial P300. Las señales fueron procesadas y se extrajeron 6 características por electrodo para diferenciar los ensayos que poseen un potencial P300 de los demás tipos de ensayos. Se aplicó el método de data augmentation a las características obtenidas para aumentar la precisión del algoritmo de clasificación. El conjunto de datos ampliados fue utilizado para entrenar 4 algoritmos de aprendizaje supervisado: Regresión Logística, Árbol de Decisión, Máquina de Soporte Vectorial y K Vecinos-más-Cercanos.

Por último, con el conjunto de prueba, se determinó que, en los cuatro participantes, los mejores clasificadores de la señal P300 corresponden a Máquina de Soporte Vectorial y K Vecinos más Cercanos, teniendo porcentajes de puntaje F1 de entre 70% y 80%, los cuales permiten predecir de manera confiable el componente P300.

1. ESTUDIO TEÓRICO DEL PROYECTO:

1.1. INTRODUCCIÓN:

En la intersección de la neurociencia y la inteligencia artificial, surge un campo de investigación orientado hacia la clasificación de señales cerebrales, en particular, la señal P300; esta señal representa una respuesta cerebral evocada que aparece cuando el participante detecta un estímulo visual o auditivo relevante o inesperado en medio de una serie de estímulos constantes. Esta señal, que se origina en la corteza cerebral, ha atraído la atención de investigadores, científicos de datos y clínicos por su potencial para revolucionar la forma en que interactuamos con el mundo que nos rodea y, más importante aún, para mejorar la vida de personas con problemas motores (Blanco Díaz & Ruiz Olaya, 2019).

En este contexto, el presente trabajo de grado tiene como objetivo comparar el desempeño de 4 clasificadores que permitan identificar la presencia del componente P300.

El proyecto de grado se estructura en torno a una serie de objetivos: Comprender en la señal P300, explorar y evaluar los algoritmos de clasificación más adecuadas para la detección del componente P300; diseñar e implementar una tarea que utilice estímulos visuales y auditivos para desencadenar una señal con el componente P300; proponer las posibles aplicaciones prácticas de la detección de la señal P300 en el campo de las interfaces cerebro-computadora y la rehabilitación.

1.2. PLANTEAMIENTO DEL PROBLEMA:

La señal P300 es una respuesta cerebral que se produce cuando un individuo detecta un estímulo relevante o inesperado dentro de un flujo de información. Su detección y análisis son cruciales en aplicaciones en interfaces cerebro-computadora (BCI) donde destaca la rehabilitación de personas

con discapacidades neuromotoras (Macías Macías, Ramírez Quintana, Méndez Aguirre, Chacón Murgia, & Corral Sáenz, 2020).

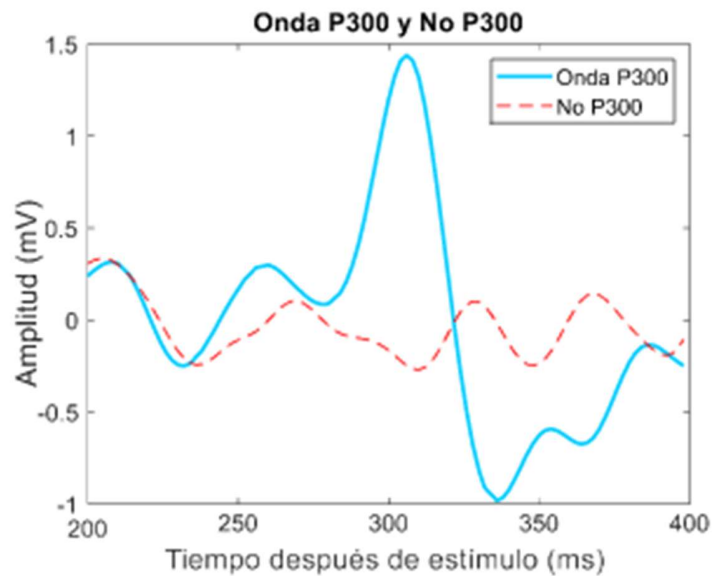


Ilustración 1 Señal con P300 (azul) y señal sin P300 (rojo) de una lectura EEG

Fuente: (Macías Macías, Ramírez Quintana, Méndez Aguirre, Chacón Murgia, & Corral Sáenz, 2020)

La señal P300 se puede ver como un pico de voltaje positivo aproximadamente trescientos milisegundos después de presentarse el estímulo visual inesperado, aunque puede cambiar entre diferentes personas, teniendo rangos de aparición de entre 250 a 400 ms.

A pesar de los avances en el estudio de la señal P300 y los sistemas BCI basados en ella, se presentan varios obstáculos significativos en la detección eficiente de esta señal. Entre los problemas principales se encuentran:

- Ruido en las señales: Las señales P300 se ven afectadas por ruido electromagnético, lo que dificulta su detección confiable en pruebas reales. Esta variabilidad puede deberse a factores como el poco contacto entre el cuero cabelludo y el electrodo, movimientos involuntarios que muevan los electrodos, o en los casos de electrodos de cableados largos

el balancearse ligeramente hace que entren frecuencias no deseadas a la señal (Hinrichs, 2004).

- **Diseño de estímulos:** La generación de estímulos visuales y auditivos que desencadenen respuestas P300 confiables es clave en estos estudios. Estos estímulos óptimos pueden variar entre participantes, ya que a unos puede intensificarse más un cierto estímulo, mientras que a otros participantes no les afecte en gran medida.
- **Eficiencia computacional:** El procesamiento en tiempo real de las señales cerebrales y la toma de decisiones basada en Machine Learning pueden verse ralentizadas por las capacidades que presenten los equipos computacionales, especialmente en aplicaciones de BCI donde la latencia sea alta.
- **Optimización de parámetros y generalización:** La optimización de parámetros de algoritmos de Machine Learning para adaptarse a diferentes usuarios y situaciones es un problema crítico, ya que se requieren de bases de datos bastante grandes de bastantes participantes que permitan hallar hiperparámetros cercanos a los comportamientos cerebrales comunes.

Ante estos desafíos, se desarrolló un algoritmo de clasificación capaz de detectar señales cerebrales de tipo P300. Además, se desarrolló una tarea donde se presentaban estímulos visuales y auditivos que se alinearon temporalmente con la señal de electroencefalografía (Macías Macías, Ramirez Quintana, Méndez Aguirre, Chacón Murgia, & Corral Sáenz, 2020).

A partir del presente trabajo se busca contribuir al avance en la detección de la señal P300 utilizando técnicas de clasificación automática a través de la utilización de estímulos visuales y auditivos (Peirce, 2019).

1.3. JUSTIFICACIÓN:

Los estudios acerca de la detección de señales cerebrales de tipo P300 usando Machine Learning tienen valor en la actualidad, tanto por el ámbito científico y tecnológico, como en una perspectiva humanitaria y social, tal como se describe a continuación.

1.3.1. IMPACTO EN CALIDAD DE VIDA:

Esta investigación tiene un impacto en la vida de personas con discapacidades neuromotoras, ya sea que sufran de paraplejía o tetraplejía, los cuales son condiciones en las que la persona no tiene movilidad desde la cintura hacia abajo (primer caso) o no tengan movilidad en todo el cuerpo (segundo caso). Estos individuos constantemente se ven limitados en términos de comunicación y movilidad, por ende, reduciendo su independencia y su interacción con cosas cotidianas. La detección de la señal P300 y su aplicación en sistemas de interfaz cerebro-computadora pueden ser el inicio de la creación de más sistemas que permitan realizar tareas sencillas a través del control cerebral, siendo herramientas poderosas para recuperar su autonomía y mejorar su calidad de vida.

1.3.2. IMPACTO BIOMÉDICO:

La detección de la señal P300 no se limita únicamente a aplicaciones de comunicación. También tiene un potencial significativo en la rehabilitación de pacientes con lesiones de médula espinal y otras condiciones neurológicas. El uso de BCI's basados en la señal P300 puede facilitar la recuperación de habilidades motoras y cognitivas, lo que es esencial para la rehabilitación exitosa.

1.3.3. IMPACTO NEUROCIENTÍFICO:

Desde una perspectiva científica, esta investigación contribuye al avance de la neurociencia al profundizar en la comprensión de cómo funcionan las señales cerebrales y cómo se pueden utilizar para mejorar la interacción entre humanos y tecnología.

1.4. OBJETIVOS:

A continuación, se presentan los objetivos que se plantean para el trabajo de grado.

1.4.1. GENERAL:

Comparar cuatro algoritmos de clasificación de señales P300 desencadenadas por estímulos visuales y auditivos.

1.4.2. ESPECIFICOS:

- Desarrollar una tarea en la que se desencadene el potencial P300 a partir de una serie de estímulos visuales y auditivos.
- Extraer las características de las señales EEG relacionadas con la señal P300.
- Evaluar los algoritmos de clasificación para la detección de la señal P300.
- Determinar el mejor algoritmo a partir de las métricas de desempeño para los datos de prueba.

1.5. MARCO TEÓRICO:

La electroencefalografía (EEG) es una técnica de medición de voltaje que permite estimar la actividad eléctrica del cerebro. Las señales producidas tienen su origen en conjuntos de varias neuronas que reaccionan ante estímulos, comunicándose unas con otras a través de mensajes químicos y eléctricos. El movimiento de iones por parte de las neuronas produce un campo electromagnético que se propaga hasta el cuero cabelludo, viéndose reducido a medida que se aleja de su lugar de origen. Debido a que los voltajes son demasiado pequeños para ser medidos con instrumentos comunes, se aplican técnicas de amplificación y filtrado de las señales. Normalmente se usan amplificadores en configuración diferencial, para poder medir las fluctuaciones de la corteza cerebral (Nunez & Srinivasan, 2006). Existen dos tipos de Electroencefalografía, el primero

que es de tipo invasivo, en el que electrodos en forma de aguja son colocados directamente en el cerebro. En este método se realiza una cirugía cerebral, y es frecuentemente realizado para obtener información acerca de focos epilépticos y con el objetivo de tratar enfermedades como el Párkinson (Ríos P. & Álvarez D., 2013). Existe un método menos invasivo y más fácilmente implementable que consiste en colocar electrodos superficiales sobre el cuero cabelludo, los cuales hacen contacto directo con la piel de la cabeza. Este método tiene una menor precisión, debido al ruido electromagnético producido por dispositivos cercanos, la actividad muscular de la cabeza y el ruido propio del sistema de amplificación que se superpone a la señal captada (Electroencefalografía (EEG), 2018). Además, el hueso del cráneo, las meninges, las capas de piel del cuero cabelludo y la resistencia de la interfaz electrodo-piel, generan impedancias relativamente altas, que están por el orden de los $k\Omega$. En consecuencia, se han desarrollado distintas estrategias para obtener medicas precisas. En primer lugar, los electrodos tienen localizaciones específicas reconocidas a nivel internacional, esta forma de localización es conocida como sistema 10-20, el cual hace referencia a repartir a nivel longitudinal de la cabeza en 4 secciones de 20% la longitud total, y 2 secciones de 10% la misma longitud total. Esto se hace tanto para la distancia desde el nasión (punto donde el cráneo tiene una cavidad en medio de las cejas) y el inión (protuberancia ósea craneal localizada detrás de la cabeza), al igual para la longitud entre los dos puntos preauriculares (delante del pabellón auditivo) (Broggi Angulo, Koc Gonzáles, & Martinez Esteban, 2022).

El sistema 10-20 permite registrar distintos puntos a lo largo de la cabeza. Cada punto se denomina con una letra y un número que corresponde con la localización anatómica del cerebro adyacente. La región frontal, que es aquella que, como su nombre lo indica, se encuentra localizada en la parte frontal de la cabeza, desde la frente hasta la altura de los puntos preauriculares; luego la región occipital, se encuentra en la parte posterior de la cabeza, justo encima del punto inión; la región

parietal está entre la región occipital y frontal; la región temporal es aquella que se encuentra encima de los puntos preauriculares ; y por último la región Central, el cual como su nombre lo indica, se encuentra en toda la línea central de la cabeza dividiendo el lóbulo izquierdo del derecho (Sherman & Walterspacher, 2006).

Los registros electroencefalográficos son series temporales que pueden descomponerse en frecuencias, su origen y función son aún desconocidas, sin embargo, correlacionan con distintos estados de conciencia. Las ondas Delta son aquellas que oscilan entre los 0,5 Hz y los 4 Hz, y se caracterizan por ser halladas durante el sueño profundo de la persona; las ondas Theta se encuentran entre los 4 Hz y los 8 Hz, y son presentes durante el sueño ligero; las ondas Alfa son las que oscilan entre los 8 Hz y los 13 Hz, y se manifiestan cuando la persona está despierta con los ojos cerrados o a inicios del sueño; y por último las ondas Beta son aquellas que se encuentran entre los 13 Hz y los 30 Hz, cuando la persona se encuentra despierta y alerta (García, 2011).

La señal P300 se produce en el cerebro cuando un patrón de estímulos frecuentes es modificado repentinamente. Uno de los paradigmas más utilizados para desencadenar esta señal, implica observar una serie de imágenes con características constantes, y en un tiempo aleatorio mostrar imágenes con características distintas. Típicamente las imágenes son figuras geométricas cuadrados cuyo color varía (Macías Macías, Ramirez Quintana, Méndez Aguirre, Chacón Murgia, & Corral Sáenz, 2020).

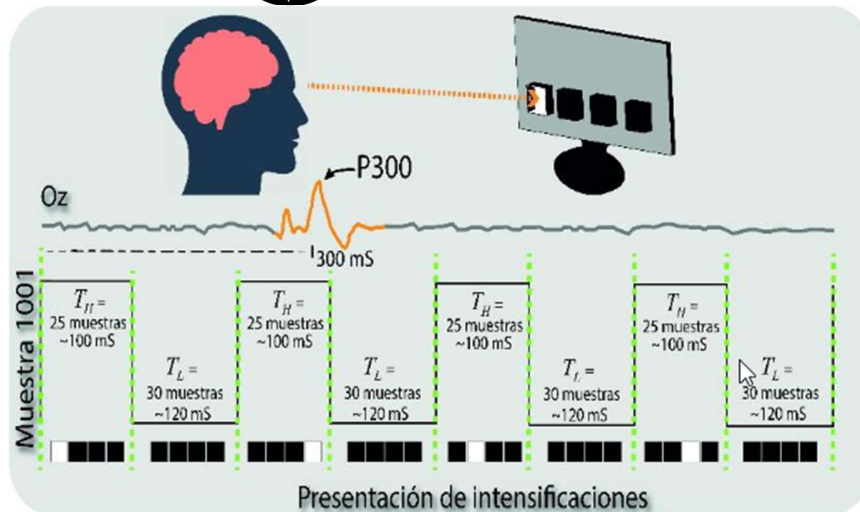


Ilustración 2 Diagrama de tiempo de estímulos visuales

Fuente: (Macías Macías, Ramirez Quintana, Méndez Aguirre, Chacón Murgia, & Corral Sáenz, 2020)

En el momento en el que se produce ese cambio visual, aproximadamente 300 ms después, se genera un voltaje positivo fuerte en la corteza cerebral cercanos a las regiones occipitales, donde se procesa la información visual en el cerebro (Wood, Truett, Goff, Williamson, & Spencer, 2008).

Este trabajo de grado usará el kit de electroencefalografía el cual contiene la placa electrónica conocida como la tarjeta de biosensado Cyton, cuya función es captar las señales EEG y las amplifica y digitaliza, dejándolas listas para su envío de manera inalámbrica hacia el computador. Esto último se hace posible a través del “Cyton Dongle” el cual capta las señales inalámbricas de la placa y las importa hacia el programa OpenBCI que está instalado en el computador. Este programa, de las cuales las siglas BCI corresponden a Interfaz Cerebro-Computadora en inglés, es la interfaz gráfica de usuario que permite visualizar en tiempo real la información que se obtiene de los electrodos, tanto en el dominio del tiempo como en el de la frecuencia, además de permitir realizar filtrados en frecuencia, siendo el más importante la remoción de la frecuencia de la red eléctrica, que en América es de 60 Hz. OpenBCI no solo tiene esas funciones mencionadas, sino



también tiene la posibilidad de enviar la información a través de otros medios, como UDP, Serial, OSC, y LSL, a este último, LSL (Lab Streaming Layer), será utilizado en el presente proyecto pues permite enviar múltiples cadenas de datos de manera simultánea, para dar uso al procesamiento de la información.

Psychopy es un programa basado en Python, el cual permite crear interfaces gráficas que pueden consistir en imágenes, figuras, sonidos, e implementarse el uso del teclado, el micrófono, o el ratón, para crear actividades interactivas para el usuario. Este programa permite crear marcadores de distintos eventos en la presentación de estímulos, ya sea cuando aparece una imagen establecida, o cuando cambie de color una figura, los marcadores se sincronizan con la señal de EEG; es decir, se puede ejecutar OpenBCI, comunicar la información de los electrodos a través de LSL hacia Psychopy, y este último envía marcadores a las señales de EEG. Una vez se termine el proceso, se almacena la información y en el registro quedará guardado las señales EEG con los marcadores implementados, los cuales serán leídos a través de Matlab, el cual permitirá realizar la extracción de las características de las señales obtenidos con sus marcadores (Wallisch, et al., 2014).

Con las señales EEG y los respectivos marcadores basados en los estímulos visuales que se crearán con Psychopy, se procedería a usar Machine Learning para el análisis de las señales. Machine Learning se traduce como “Aprendizaje de máquina”, herramienta con la cual se ejecutan procedimientos de inteligencia artificial. El Machine Learning tiene la capacidad de usar la computación para crear sistemas de regresión y de clasificación, donde por medio del uso de datos de entrada y de salida, logra formular el proceso que más se asemeja al intermedio de las entradas y las salidas. Esto es una herramienta muy potente cuando se tienen datos que no parecen relacionarse entre sí, al menos de una forma coherente, porque el Machine Learning tiene la capacidad de procesar los datos y hallar lo que más se asemeje a la etapa intermedia entre los datos

de entrada y los datos de salida. El aprendizaje de máquina requiere especificar las características más importantes de los datos, para así calcular los procesos matemáticos más cercanos a la relación de entrada contra salida. Además, dentro de Machine Learning, se encuentra un subconjunto conocido como Deep Learning, el cual se basa en múltiples datos de entrada en el que automáticamente se extraen las características más importantes, donde se crean capas de procesamiento, se entrelazan para crear sistemas matemáticos más complejos, hasta obtener una salida que se compara con la verdadera salida. Todo el sistema se realimenta y se repite, dando más precisión entre más cantidad de información sea ingresada, (Zhuang, et al., 1996).

Otra forma de trabajar el Machine Learning es por medio del tipo de aprendizaje. El primero es conocido como “Aprendizaje No Supervisado” donde los datos no se encuentran etiquetados. Esto implica que la información debe ser clasificada en grupos pero que no tienen una definición clara, sino que un nuevo dato a estimar ingresará a uno u otro grupo. Contrario a este tipo de aprendizaje, es conocido como “Aprendizaje Supervisado”, donde los datos sí se encuentran etiquetados por clases (Gannouni, Aledaily, Belwafi, & Aboalsamh, 2021).

Por último, entre el grupo de Machine Learning llamado Aprendizaje Supervisado, se definirá cada uno de los modelos a trabajar en el proyecto de grado.

La “Regresión Logística” es un sistema que clasifica normalmente en dos grupos, en el cual el sistema automáticamente empieza a crear una línea divisora en medio de la nube de datos, y empieza a darle inclinaciones de diferentes ángulos. Este proceso se va realizando y evaluando, hasta hallar el mejor ángulo y posicionamiento de la línea divisora que clasifique de manera más óptima los datos entre uno u otro grupo, basado en distancias entre la línea y los datos. (Hojas)

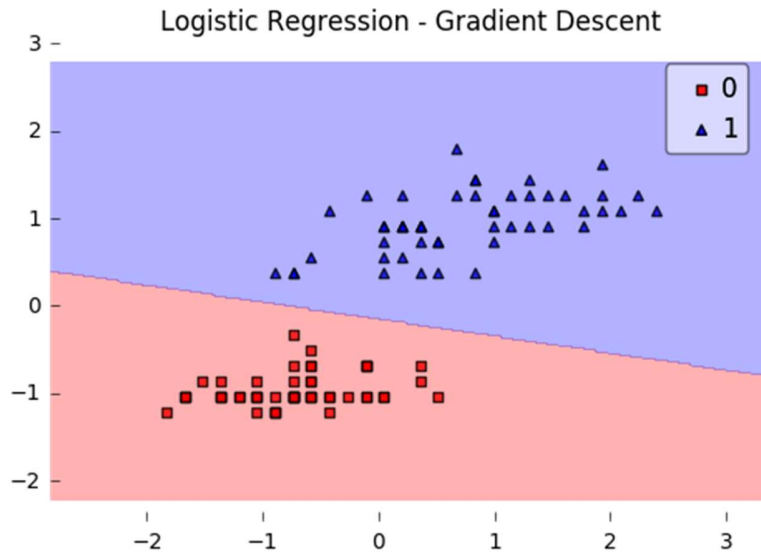


Ilustración 3 Regresión Logística

Fuente: (Hojas)

El siguiente modelo se llama “Árbol de Decisión” donde a cada columna de datos se le deriva una característica específica, donde se evalúa si se cumple o no, dependiendo de eso, se le extrae otra característica a cada caso, resultando en más derivaciones, y así sucesivamente, hasta llegar a concluir si hace parte de uno u otro grupo. La imagen representa un ejemplo más claro (Romo):

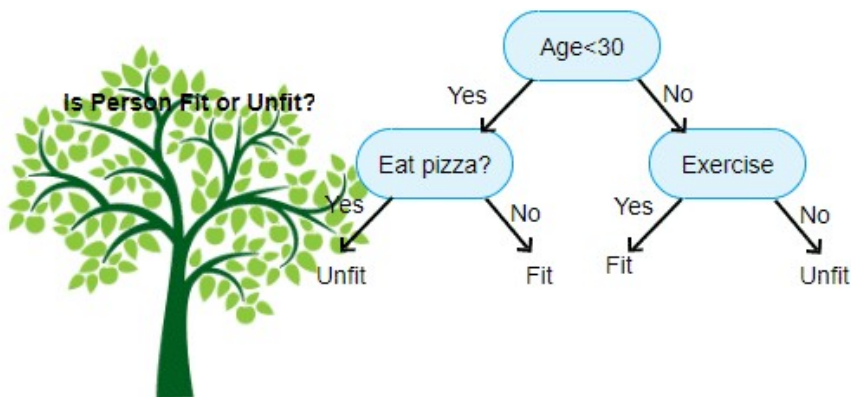


Ilustración 4 Árbol de decisión

Fuente: (Romo)

La “Máquina de Soporte Vectorial” es un sistema similar a la regresión logística en el hecho de la clasificación por medio del uso de una línea divisora, solo que tiene en cuenta unos vectores de soporte, los cuales son datos que tienen una distancia definida entre ellos con la línea, conocida como frontera, y generan así un área de separación entre los dos grupos de datos. Hay veces en las que el sistema requiere aplicar un “kernel” al conjunto de datos, el cual es el proceso de aumentar las dimensiones de los datos, de n dimensiones a $n+m$ dimensiones, las que el sistema considere necesario, para poder clasificar mejor los datos (Chandra Poonia, Singh, & Ranjan Nayak, 2022).

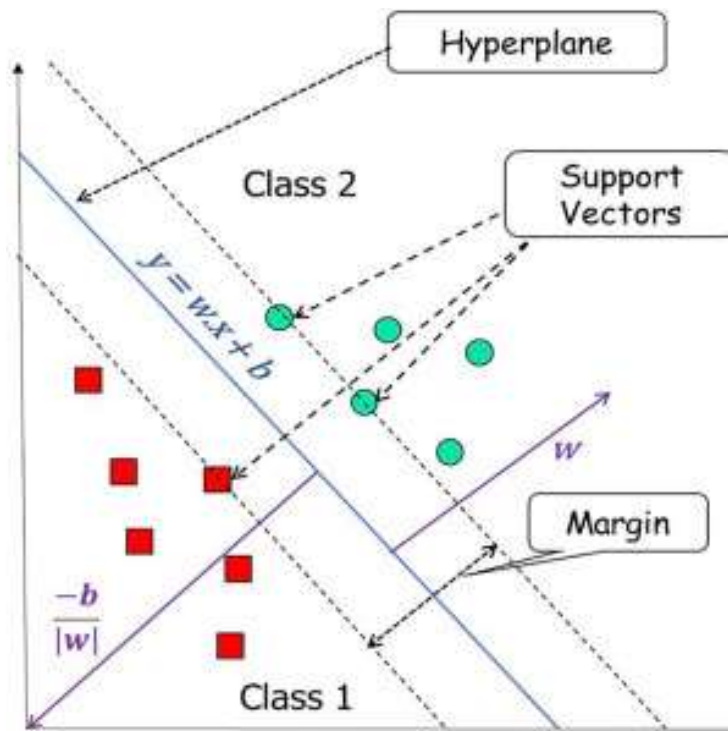


Ilustración 5 Máquina de soporte vectorial

Fuente: (Chandra Poonia, Singh, & Ranjan Nayak, 2022)

Por último, se tiene el sistema clasificador por “K Vecinos más Cercanos” en el cual se tienen los dos conjuntos de datos, en el cual, en el momento de agregar un nuevo dato, se buscan los k vecinos más cercanos a ese dato, y el que mayor cantidad de vecinos tenga de un conjunto, mandará el dato

a ese conjunto. Cuando k es 3 por ejemplo, el nuevo dato empezará a ampliar una circunferencia alrededor de él, hasta encerrar a 3 datos dentro de la circunferencia, y si hay por lo menos 2 datos del conjunto A y 1 dato del conjunto B, el nuevo dato pertenecerá al grupo A (Gonzales, 2019).

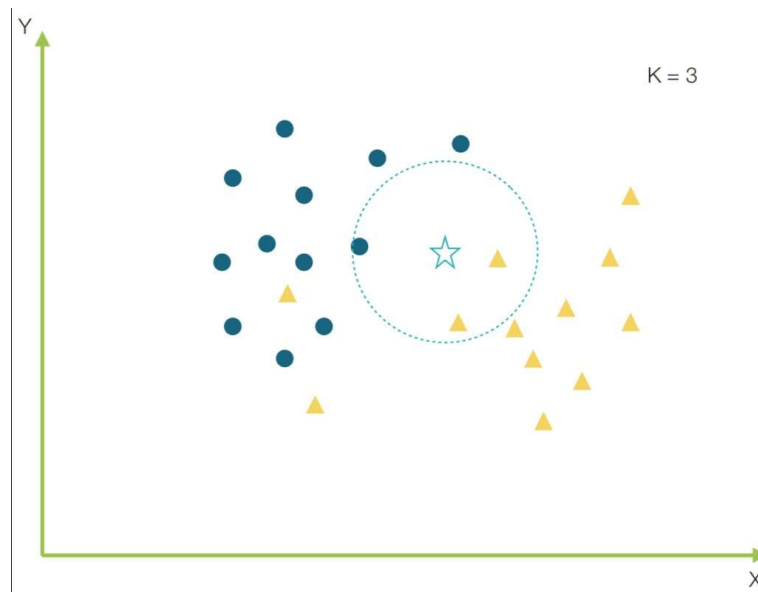


Ilustración 6 K Vecinos más Cercanos

Fuente: (Gonzales, 2019)

Estos son las definiciones de los modelos de Machine Learning a utilizar en el trabajo de grado.

Entre lo que se ve involucrado en el trabajo es el aumento de datos, para mejorar los entrenamientos de los modelos de Machine Learning. El aumento de datos, conocido en inglés como “data augmentation”, es el proceso en el cual se replican los datos que se tienen originalmente y se realizan pequeñas modificaciones a cada conjunto, creando así copias del conjunto de datos original, pero agregándose ruido, o ligeras variaciones, los cuales hacen que sean diferentes al dataset original, pero sin alterar en gran medida el conjunto original, con el fin de no perder sus características primordiales (Kumar, 2019). La siguiente imagen es un ejemplo ilustrativo de lo que consiste el data augmentation:

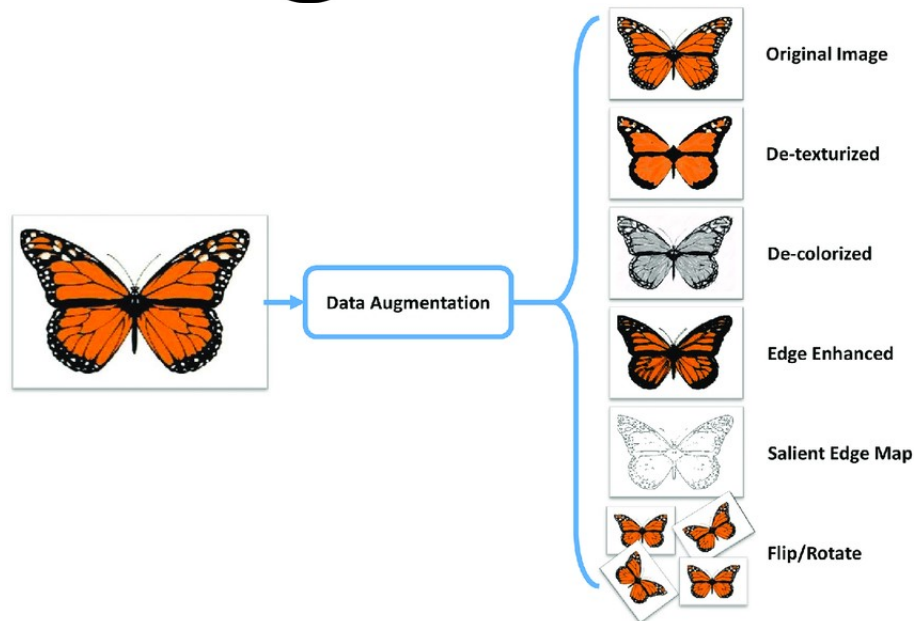


Ilustración 7 Ejemplo de data augmentation

Fuente: (Kumar, 2019)

Esto optimiza el aprendizaje supervisado de los modelos de Machine Learning.

1.6. ESTADO DEL ARTE:

El uso de Machine Learning para la detección de señales cerebrales de tipo P300 ha sido un campo de investigación activo y prometedor en la intersección de la neurociencia y la inteligencia artificial. Algunos aspectos clave incluyen:

- Interfaces Cerebro-Computadora (BCI): Se han venido desarrollando BCI's basadas en la señal P300 para ayudar a personas con discapacidades físicas a comunicarse y controlar dispositivos. Estas aplicaciones van desde el control de sillas de ruedas eléctricas hasta la escritura de texto. Se incluyen estudios como potenciales evocados, con el cual se logran extraer características de las señales EEG luego de haber aplicado etapas de pre-procesamiento y filtrado (Blanco Díaz & Ruiz Olaya, 2019).

- Algoritmos de Machine Learning: Los investigadores han venido aplicando una variedad de algoritmos de Machine Learning, como sistemas SVM, redes neuronales como Deep Learning, y técnicas de procesamiento de señales para detectar y decodificar las señales P300 de manera más precisa y eficiente. Estos estudios inician con la comunicación entre el hardware de los electrodos, con el software que crea los estímulos, al igual que el software que entrelaza las lecturas EEG (Razavi, et al., 2021).
- Aplicaciones Clínicas: Se han explorado aplicaciones clínicas, como la rehabilitación de pacientes con lesiones de médula espinal y la mejora de la calidad de vida de personas con paraplejia y tetraplejia. Además, se mencionan estudios relacionados con la creación de electrodos no invasivos y con reducción de ruido efectivo (Biarnés Rabella, 2018).
- Optimización de Estímulos Visuales: Científicos han estado investigando la creación de estímulos visuales efectivos que pudieran desencadenar respuestas P300 confiables, lo que es crucial para la precisión de los sistemas BCI, realizando análisis en datos de series de tiempo, y explicando las dificultades y correcciones a los problemas principales que se producen en la captación de señales EEG (Cohen, 2014).
- Colaboraciones Interdisciplinarias: La investigación en este campo requiere colaboraciones interdisciplinarias entre neurocientíficos, expertos en procesamiento de señales y científicos de datos, de tal forma que se logra aliar la psicología con la biomedicina y los diferentes comportamientos cerebrales (Urgilés Cárdenas & Vásquez Rodríguez, 2017).

Este trabajo investigativo está soportado en cada uno de los conceptos previos, para el uso de métodos de análisis y procesamiento de datos basados en las señales electroencefalográficas.

2. DESARROLLO DEL PROYECTO:

El proyecto se dividió en 4 fases, los cuales consisten en fabricación del casco EEG y recolección de datos, extracción de características y procesamiento de datos, entrenamiento de modelos de aprendizaje supervisado, y métricas finales, los cuales se ilustran como una arquitectura general de la siguiente forma:

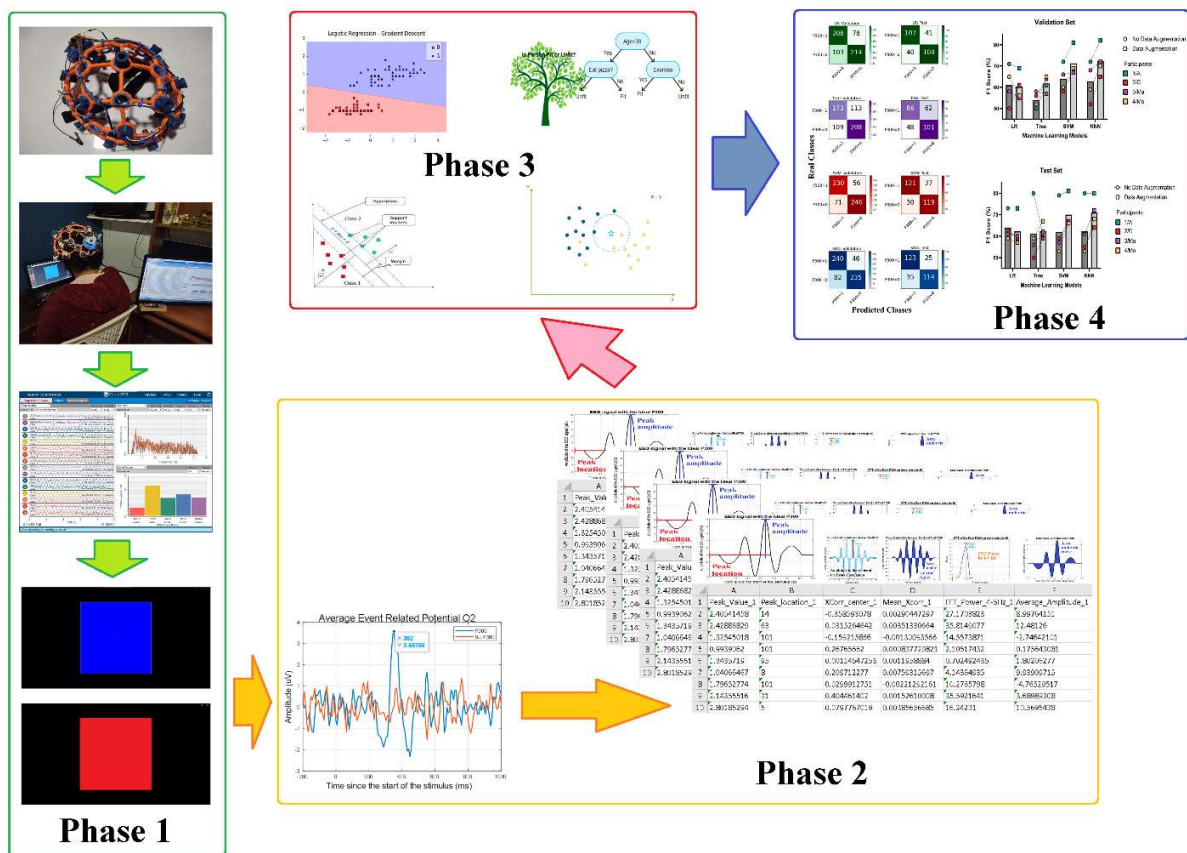


Figure 1 Arquitectura general del proyecto

Cada una de estas fases será descrita de forma detallada, incluyendo sus respectivos resultados.

2.1. FASE 1, RECOLECCIÓN DE DATOS DE ENTRENAMIENTO:

El siguiente es el diagrama de flujo de la fase 1 que ilustra cada uno de los pasos desarrollados para la respectiva ejecución:

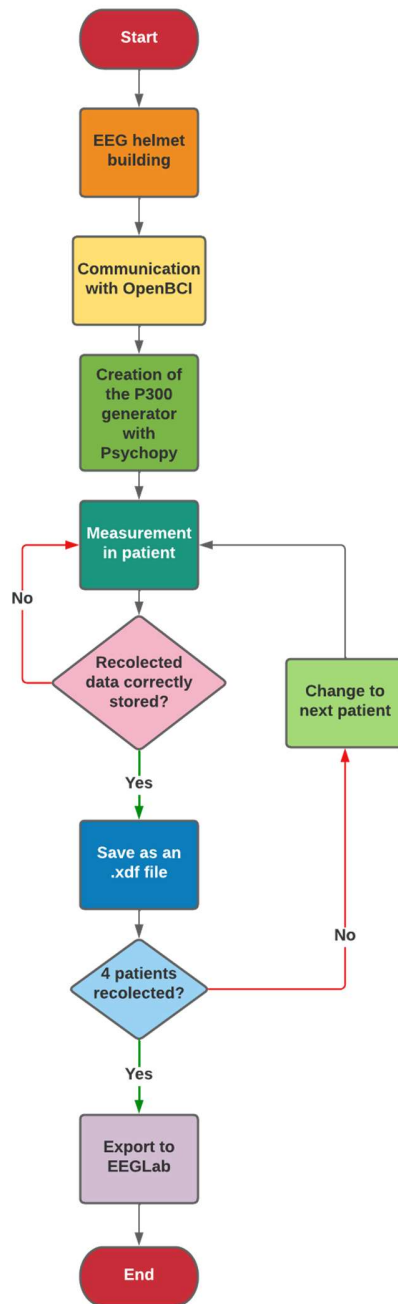


Figure 2 Diagrama de flujo de la fase 1 del proyecto

Esta fase inició en la fabricación del casco electroencefalográfico, el cual permitió ensamblar los electrodos en las posiciones requeridas para el estudio, los cuales son en las regiones parietal, temporal y occipital. OpenBCI no solo brinda el paso a paso de ensamblar todo el casco completo,

sino que también suministra en su página web el modelo en 3D del casco, para ser impreso con plástico. Se descargaron los modelos en formato .stl y se importaron en un programa llamado Ultimaker Cura, el cual es una aplicación que permite importar modelos en 3D, ya sea en formatos .stl o .3mf, como se observará en la siguiente figura:

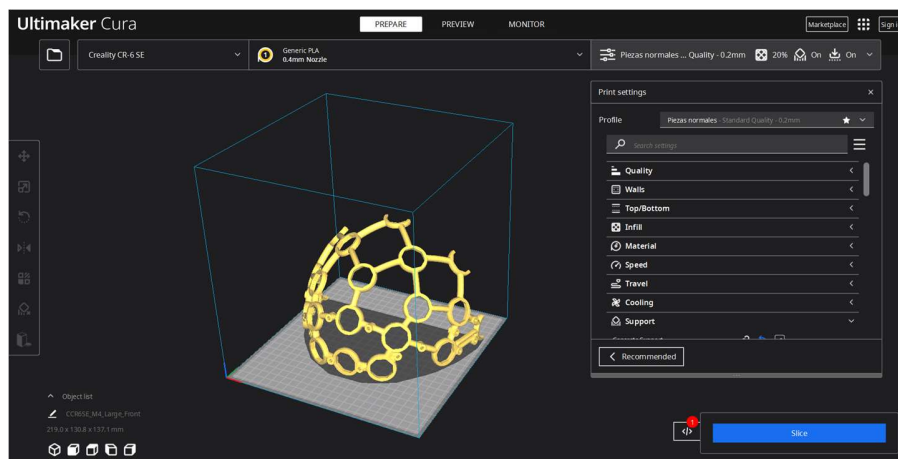


Figure 3 Primera mitad del casco EEG importado en Ultimaker Cura

Y después los procesa para laminar la pieza en capas, generando un archivo de código G creado específicamente para lenguaje de G-code para la impresora 3D. Este archivo logra estimar el tiempo total de impresión, al igual que la cantidad total de material que se invierte en la impresión.

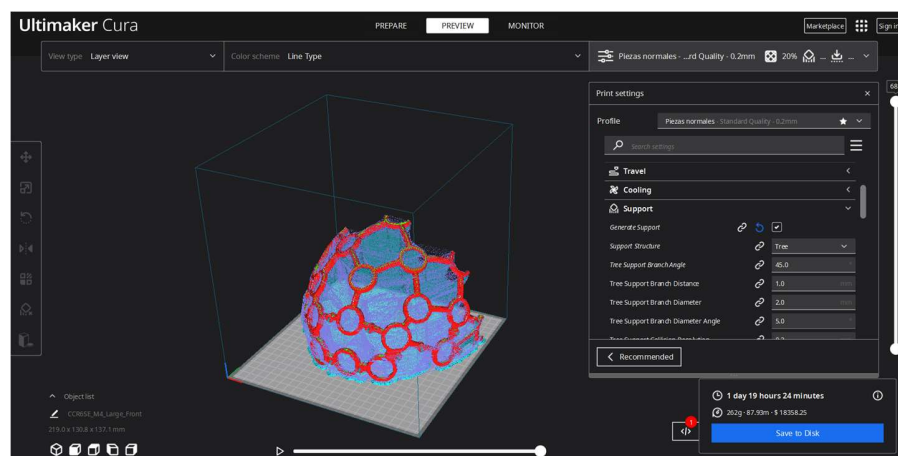


Figure 4 Primera mitad del casco EEG importado y laminado en Ultimaker Cura

El autor de este documento cuenta con una impresora Creality CR 6 SE, con dimensiones máximas de ancho, largo y alto de 235*235*250mm respectivamente, el cual tiene una configuración de tipo i3, el cual corresponde a que realiza movimientos del eje Y para la cama, y los movimientos del eje X y Z en la boquilla, que es por donde se extruye el material:



Figure 5 Impresora 3D de la marca Creality CR 6 SE usada para el proyecto

Sin embargo, por el hecho de que el casco era muy grande para ser impreso en una sola pieza completa, se imprimieron 2 piezas y se unieron al final de la impresión, una vez fueran removidos los soportes. Una vez fueron impresas las dos partes del casco, junto con elementos complementarios como la caja que contendrá la placa de circuitos y la batería, las tuercas que permiten atornillar los electrodos, y clips que organizan mejor los cables, se obtuvo el siguiente resultado final, del cual se ilustrará una lista de las partes esenciales del casco:

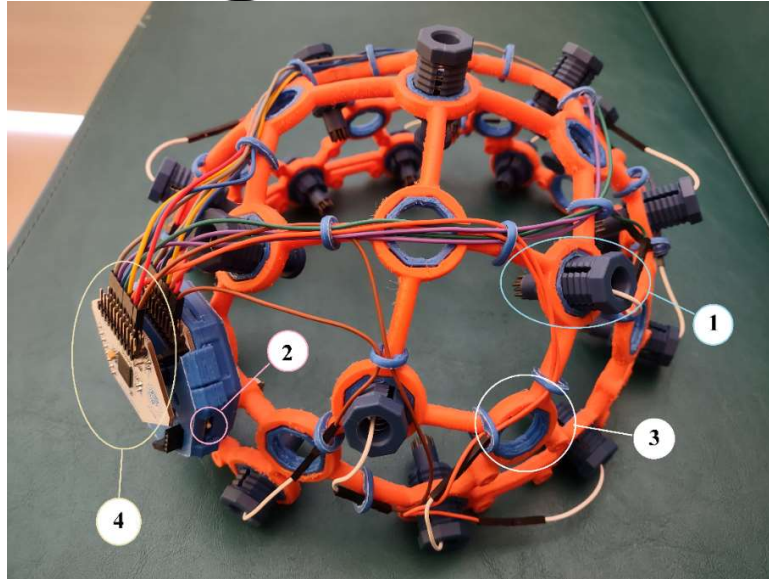


Figure 6 Casco de electroencefalografía impreso en 3D, desglosamiento 1



Figure 7 Casco de electroencefalografía impreso en 3D, desglosamiento 2

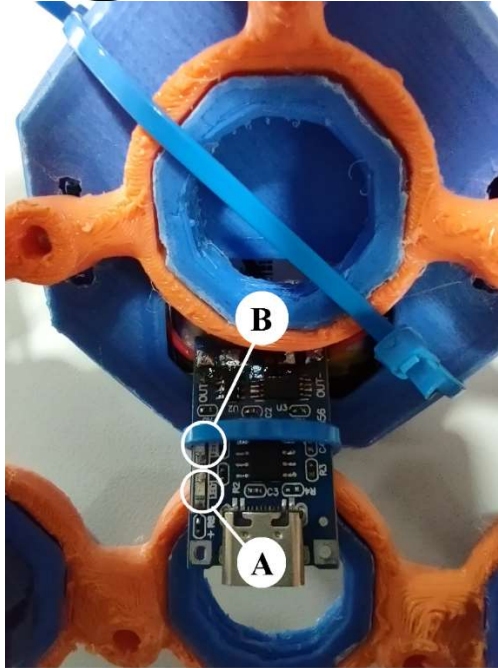


Figure 8 Casco de electroencefalografía impreso en 3D, desglosamiento 3

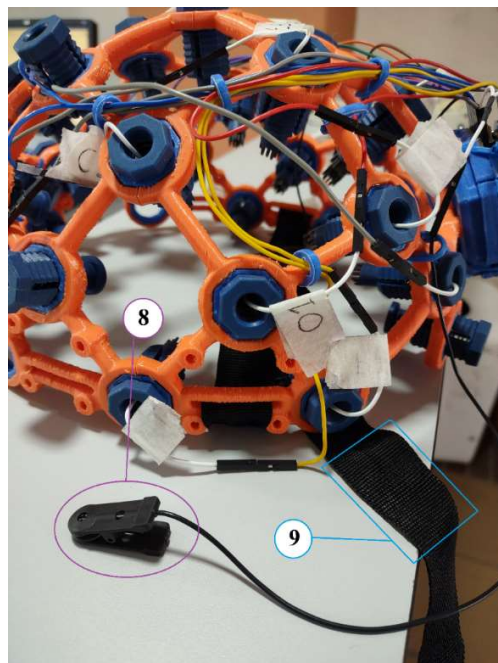


Figure 9 Casco de electroencefalografía impreso en 3D, desglosamiento 4

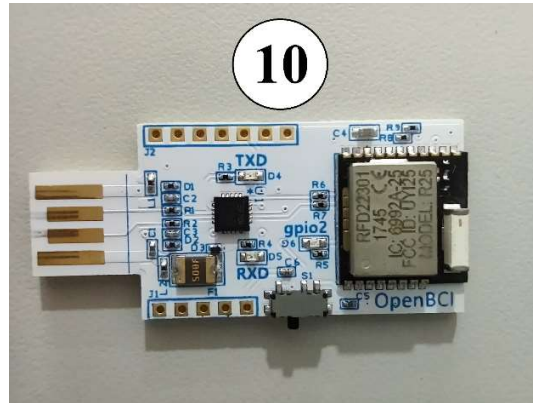


Figure 10 Casco de electroencefalografía impreso en 3D, desglosamiento 5

La tabla que sigue describe cada elemento y su función principal dentro del kit:

# de elemento	Nombre	Función
1	Electrodo	Realizar la lectura de voltajes medidos en uV de la zona craneal que esté haciendo contacto directo.
2	Interruptor	Encender o apagar la placa de circuitos del casco, con posiciones de encendido normal, apagado y encendido con almacenamiento en tarjeta Micro SD.
3	Tuerca para electrodos	Soportar el enroscamiento de los electrodos o soportes de goma, que sostienen firmemente entre el elemento y el casco.

4	Placa de circuitos	Captar y enviar todos los datos de los 16 electrodos de forma inalámbrica desde el casco hacia el computador.
5	Batería	Alimentar la placa de circuitos del casco, siendo una batería de LiPo.
6	Soportes de goma	Generar soporte suave sobre la cabeza del participante para reducir la presión de netamente los electrodos sobre el cuero cabelludo.
7	Cargador tipo C	Cargar de forma segura la batería, al igual que proteger la batería durante descarga, con conector de USB tipo C.
7.A	LED indicador de proceso de carga	Indicar con luz de color rojo el proceso de carga de la batería.
7.B	LED indicador de carga completa	Indicar con luz de color azul la carga completa de la batería.
8	Pinzas para las orejas	Generar la tierra y referencia en ubicaciones bajas de la cabeza para mejorar la calidad de señales captadas en el cráneo.

9	Correas	Dar mayor firmeza del casco sobre la cabeza, reduciendo el movimiento y por ende el ruido producido por fricción.
10	Receptor de señal USB	Comunicar las señales provenientes del casco de forma inalámbrica, hacia el computador, por medio de USB.

Tabla 1 Elementos que conforman el hardware del proyecto de grado

Una vez se tienen las piezas ensambladas, el casco se vea de la siguiente manera:

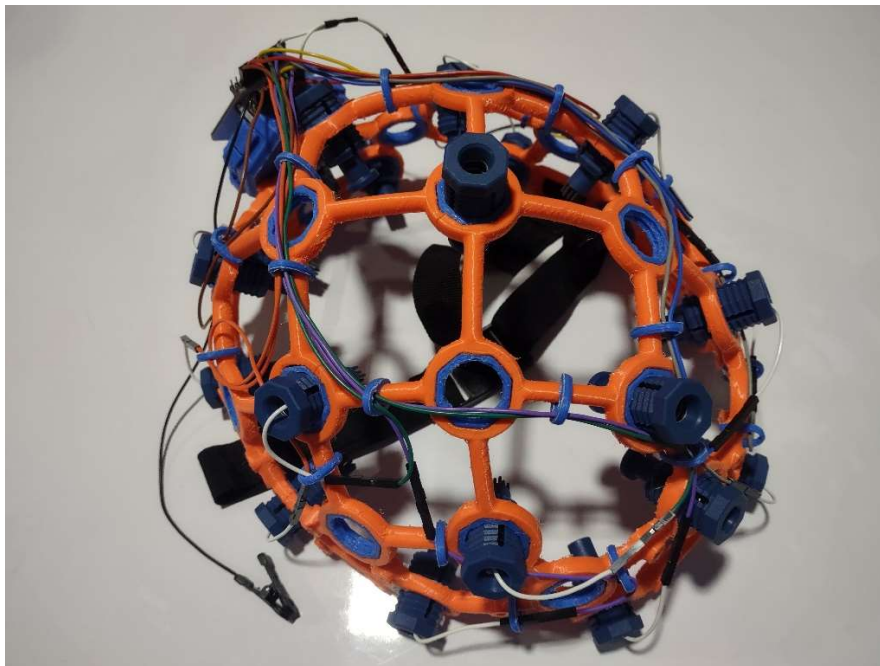


Figure 11 Casco EEG armado en su totalidad

Con esto se le dio fin a la fabricación del hardware del proyecto. El siguiente paso que se realiza, es asegurar la correcta comunicación con el programa OpenBCI, el cual es la interfaz cerebro-computadora que permite visualizar la actividad cerebral por medio de los voltajes generados en el cerebro. El programa de OpenBCI es gratuito, y se puede descargar desde su página web oficial,

aunque no operará correctamente hasta instalarse un driver llamado “FTDI” el cual asegura la comunicación correcta entre el dispositivo USB y el computador. Finalizado la instalación, se llega a esta interfaz, el cual nos ilustra los voltajes captados de los 16 electrodos, en unidades de micro voltios, junto con sus frecuencias y potencias por bandas de frecuencia:

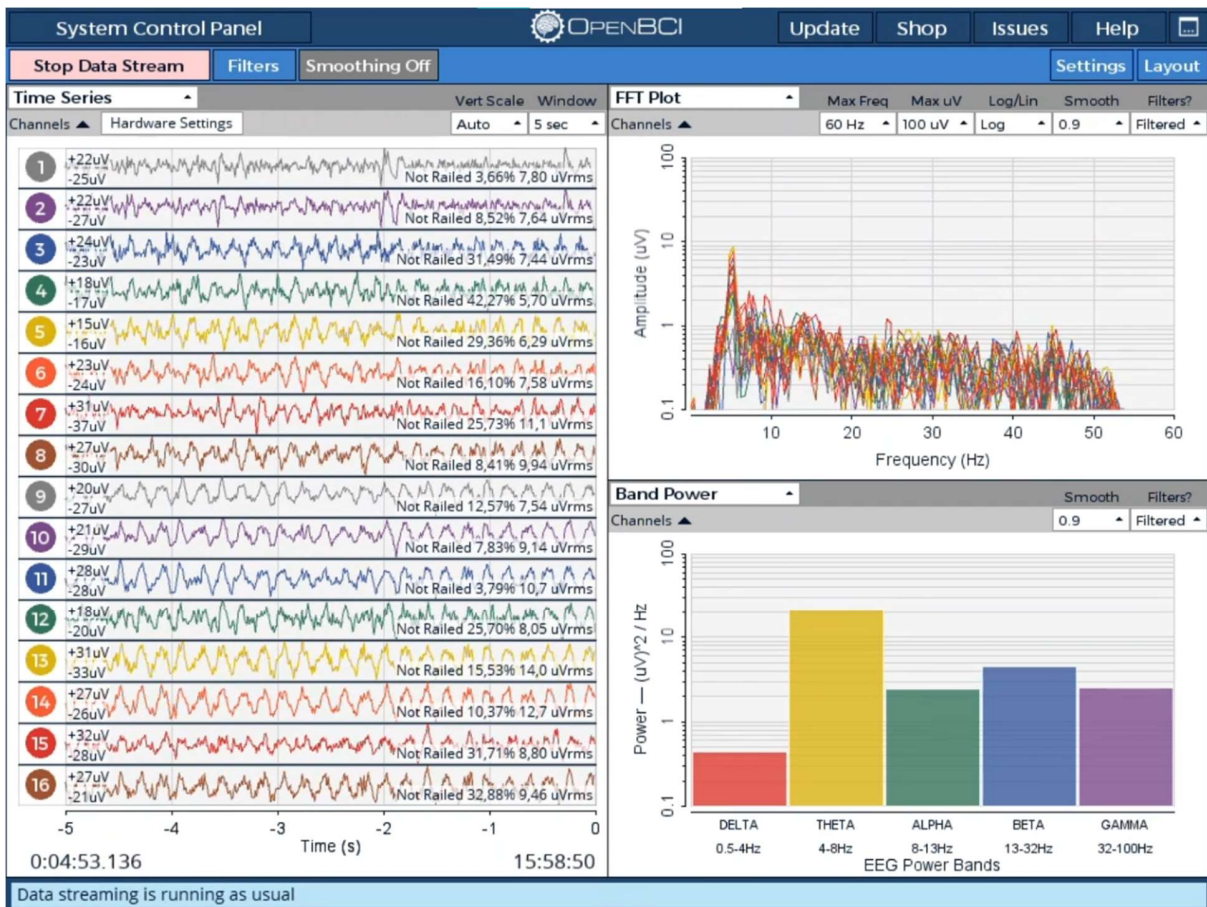


Figure 12 Interfaz Gráfica de Usuario del programa OpenBCI

Y ahora se muestra las ubicaciones de los electrodos físicamente, con las respectivas enumeraciones dadas por OpenBCI:

7	O1
8	O2
9	CP1
10	CP2
11	PO7
12	PO8
13	T7
14	T8
15	P3
16	P4

Tabla 2 Distribución física de los electrodo correspondientes a su orden de OpenBCI

Con esto, una vez se logró la correcta recepción de datos desde el casco EEG hacia el computador, se inició con el segmento de generación de estímulos visuales y auditivos utilizando Psychopy. La aplicación permite crear marcadores en ubicaciones de tiempo específicas, donde es el usuario quien define en qué momentos generar los marcadores y de qué tipo, con apoyo de figuras en la pantalla o sonidos emitidos.

A partir de este concepto, se diseñó un programa que comienza mostrando una pantalla de inicio para el participante, en el que se le pide que sujete un mando de consola, y que presione el botón B para iniciar, y cada vez que aparezca un cuadrado de color diferente al azul, mientras el computador emite un sonido, debe presionar el botón B y contar mentalmente el total de cuadrados de diferente color. Esta primera interfaz se muestra a continuación:

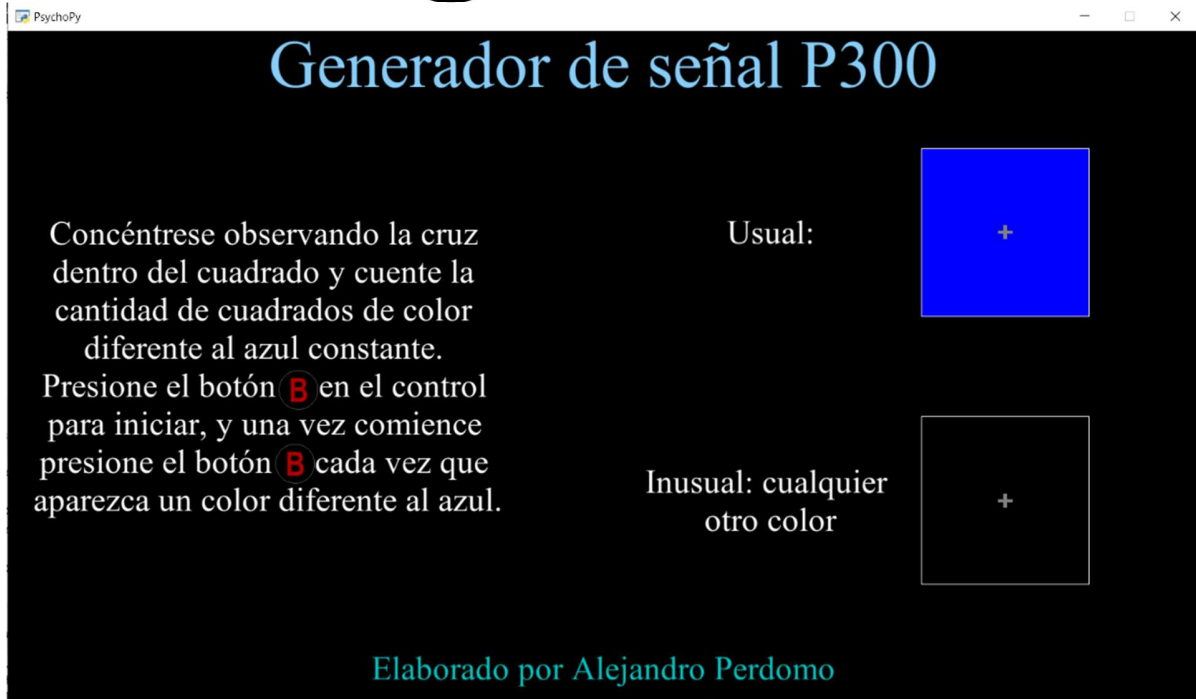


Figure 13 Interfaz Gráfica de Usuario creado con Psychopy para los participantes

Una vez el participante presiona el botón B la primera vez, se le es presentado una interfaz como en la parte superior de la Figure 14, y en un tiempo aleatorio cambia a otro color mientras se emite un tono de sonido, como en la parte inferior de la figura 14. Cada vez que era presentado un cuadrado azul, el cual no se notaba cambio visual ni auditivo para el participante, se generaba un marcador de referencia 0, indicativo de ausencia de señal P300. En cambio, en el momento en el que al participante se le presentaba un cuadrado de un color diferente mientras sonaba un tono auditivo, por el generarse un cambio repentino e inesperado para la persona, se produciría una señal P300 en el cerebro, lo cual se agregaba un marcador con referencia 1 indicando la presencia de esa señal P300 en ese tiempo exacto.

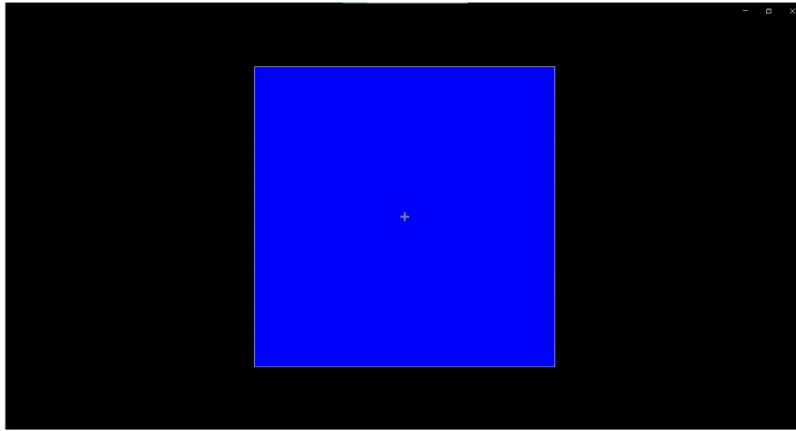


Figure 14 Pantalla de Psychopy en estado de estabilidad visual (superior) y cambio visual para la generación de P300 (inferior)

La prueba fue realizada en los 4 participantes, donde fueron hechas 3 sesiones de 5 minutos cada una, por participante, de las cuales cada sesión generaba 50 marcadores de ausencia de P300, y 50 marcadores de presencia de P300, es decir 50 1's y 50 0's, dando un total de 300 marcadores por participante.



Figure 15 Participantes durante pruebas de generación de P300: 1/A (izquierda), 2/C (Derecha superior), 3/Ma (Derecha central) y 4/Mo (Derecha inferior)

Mientras Psychopy producía los marcadores durante cada estímulo, OpenBCI producía las señales cerebrales captadas en tiempo real. Para poder unir ambos en un solo archivo, era necesario empatarlos a través de un enlace directo, por lo que se acudió al uso de un programa llamado LabRecorder, el cual unía los datos transmitidos de dos tramos de datos en un solo archivo de formato .xdf, usando comunicación por LSL (Lab Streaming Layer) de los dos tramos de datos. El programa se visualiza a continuación:

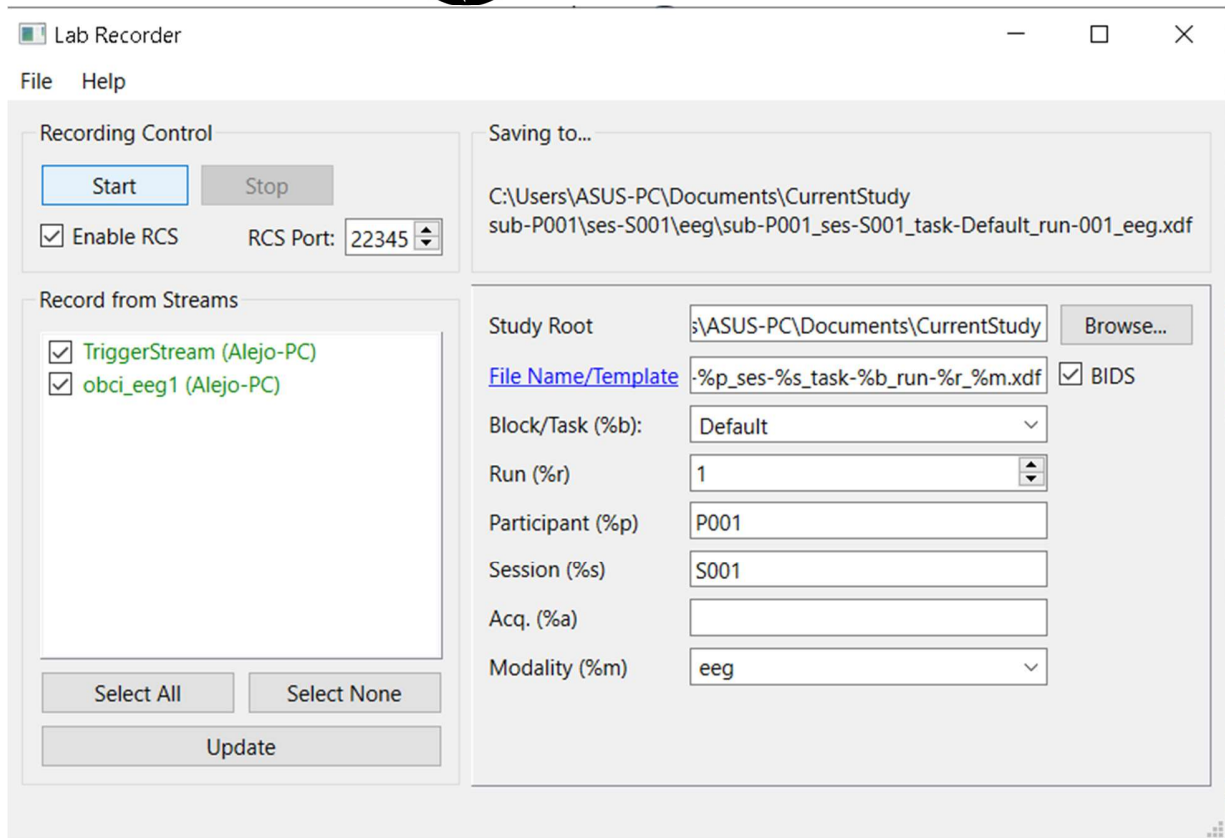


Figure 16 Interfaz gráfica de LabRecorder

Donde con el inicio de transmisión de datos con LSL de ambos OpenBCI y Psychopy, se unen en un solo archivo, de formato procesable por Matlab. Estos archivos que unían EEG con marcadores fueron los que se recolectaron de los 4 participantes, y dieron paso al siguiente segmento de la fase 1.

Una vez obtenidos los archivos .xdf, se abría el programa de Matlab conocido como EEGLab, el cual es un programa que permite realizar diferentes tipos de análisis de señales electroencefalográficas, de los cuales procesa también marcadores en zonas específicas en tiempo.

La siguiente es la interfaz de EEGLab de Matlab:

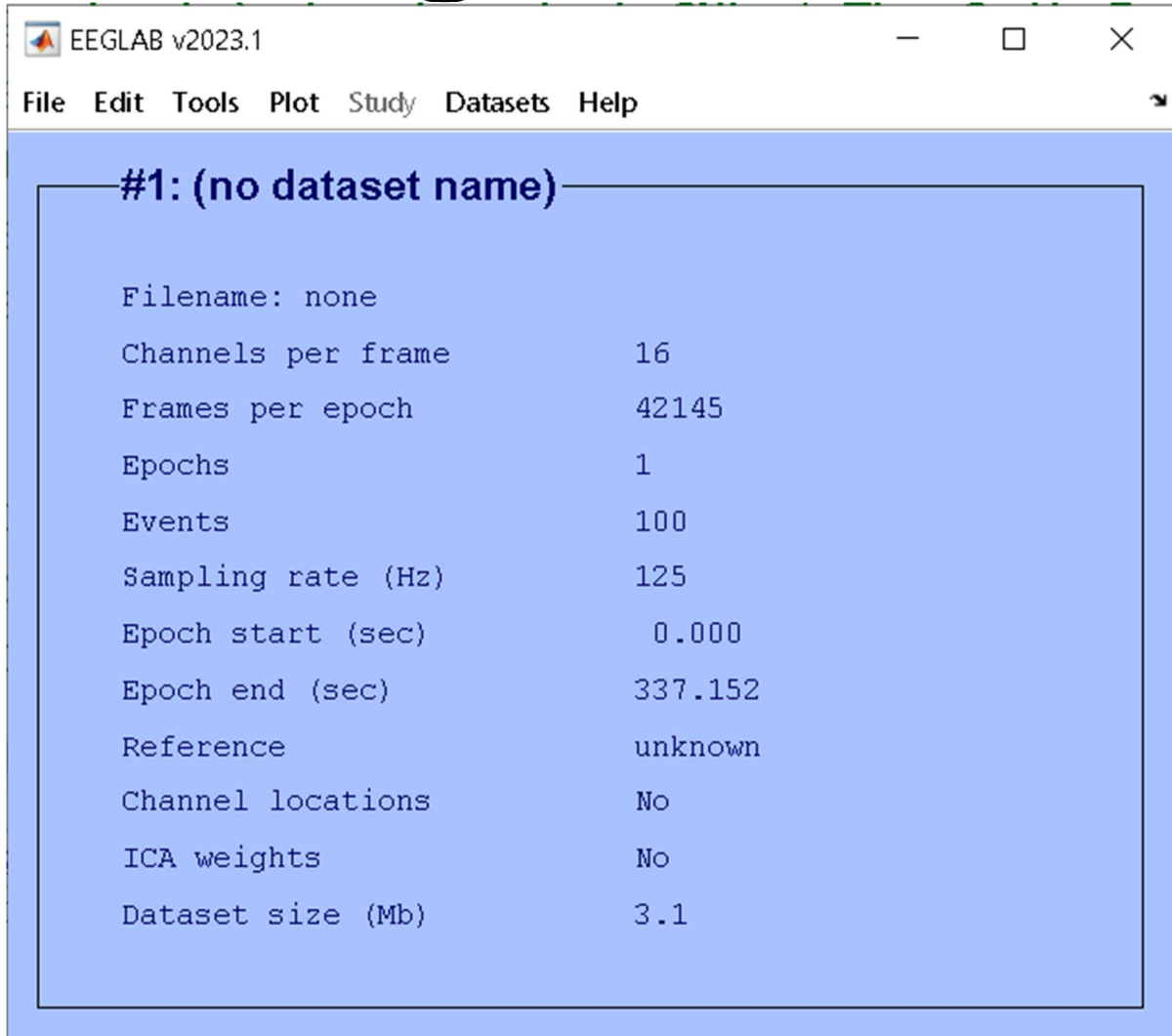


Figure 17 Interfaz Gráfica de Usuario de EEGLab basado en Matlab

Al importar el archivo .xdf era posible visualizar los resultados obtenidos, tanto de los 16 electrodos, como de los marcadores indicando presencia o ausencia de señal P300. Con esto, era posible notar ciertas diferencias con respecto al comportamiento de las señales cerebrales en la diferencia de ausencia y presencia de estímulos aleatorios, como se visualiza en la siguiente imagen:

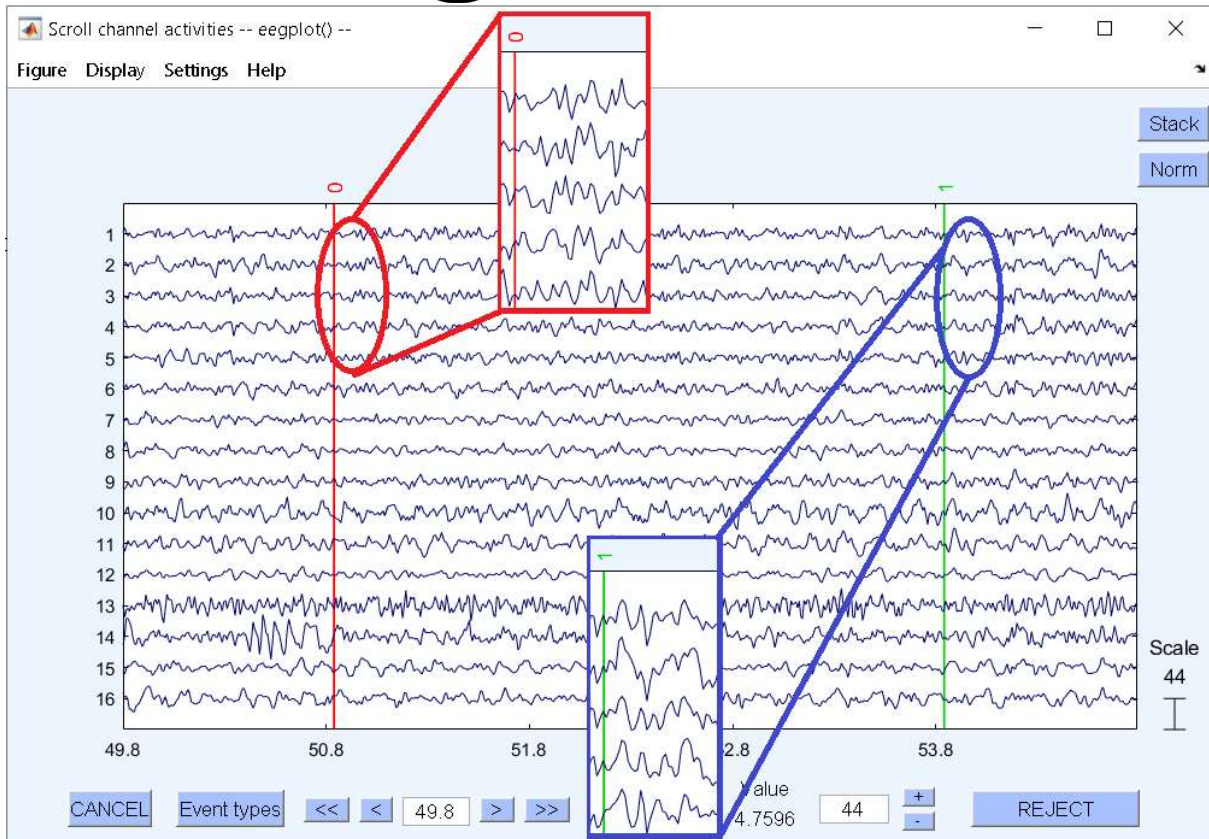


Figure 18 Señales EEG con marcadores de no aparición (línea vertical roja con un cero) y aparición de P300 (línea vertical verde con un uno)

En esta ilustración, en la ausencia de estímulos aleatorios, se observa un pequeño lapso de relajación mental con los voltajes reducidos, durante un muy pequeño tiempo inicial, mientras que con la aparición de un estímulo aleatorio se genera una reacción con un pico de voltaje mayor. Esto es a lo que se deseaba llegar con la primera fase del desarrollo del proyecto, obtener las señales y sus respectivos marcadores. Sin embargo, para obtener una mejor visualización de la señal P300, se requiere realizar un promedio de todas las ventanas de un mismo electrodo con presencia de P300, y otro promedio con ausencia de P300, para compararlos y observar el potencial evocado. Por medio de un programa en Matlab creado por el codirector de esta tesis, es posible realizar el proceso de obtener el potencial evocado real, evaluado en los electrodos O1 y O2, los cuales son

las localizaciones más cercanas a la aparición del potencial evocado P300, dando el siguiente resultado final:

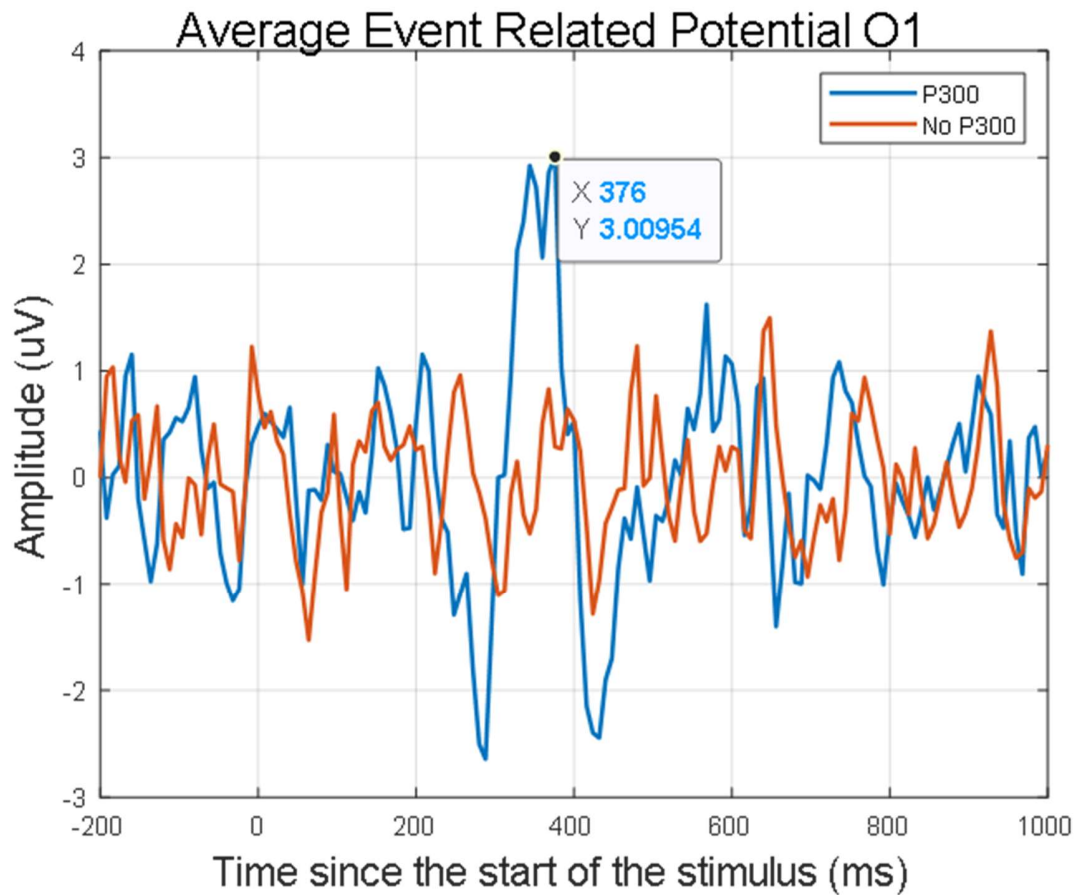


Figure 19 Comparación presencia y ausencia de potencial evocado en electrodo O1

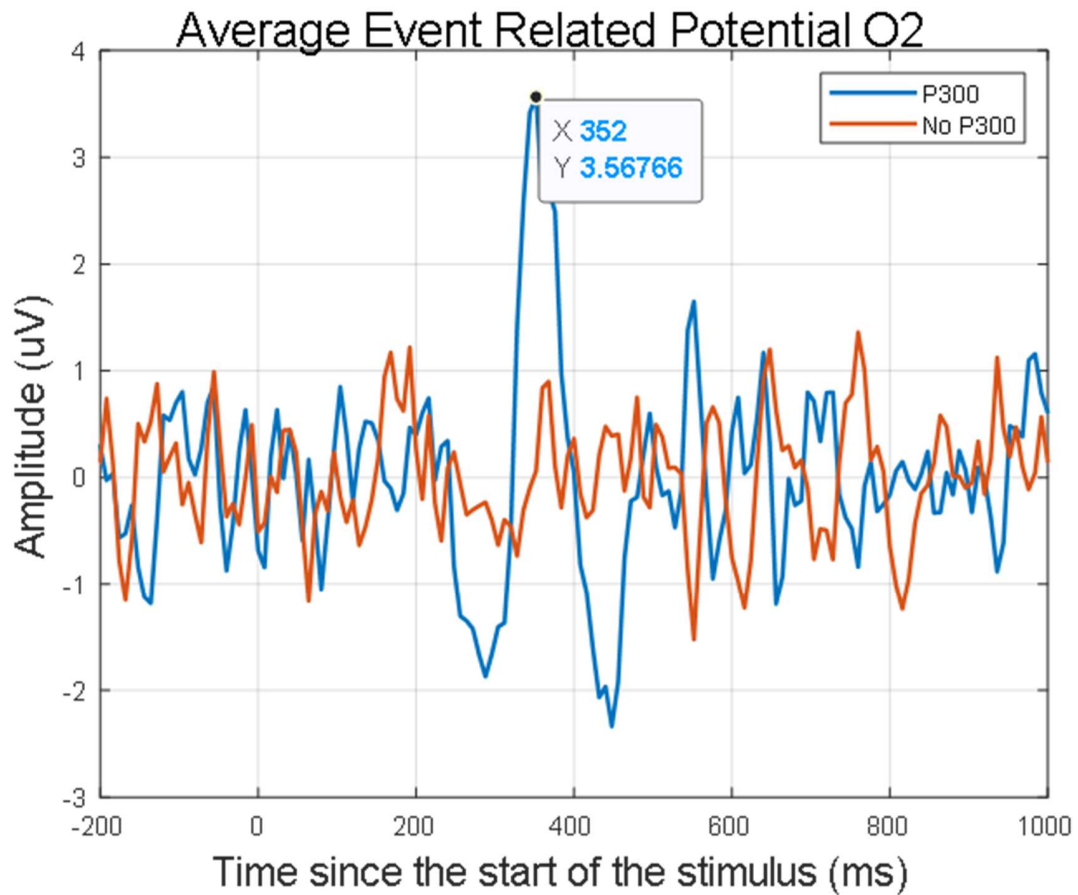


Figure 20 Comparación presencia y ausencia de potencial evocado en electrodo O2

Observando el pico positivo a los cercanos 300 milisegundos después del estímulo para la señal azul, es decir aquella que fue generada con estímulos aleatorios visuales y auditivos, nos permite concluir que las pruebas fueron realizadas de forma satisfactoria, ya que por medio de los potenciales evocados se logra obtener diferencias válidas entre la presencia y ausencia de un estímulo aleatorio visual y auditivo. Es en este punto en el que se da por finalizado la fase 1 del proyecto.

2.2. FASE 2, ANÁLISIS Y PROCESAMIENTO DE DATOS:

Teniendo las señales electroencefalográficas con los marcadores, suministradas por Psychopy en alianza con OpenBCI, se usó el programa Matlab para poder realizar filtrado y procesamiento de las señales. El esquema general de esta fase es la siguiente:

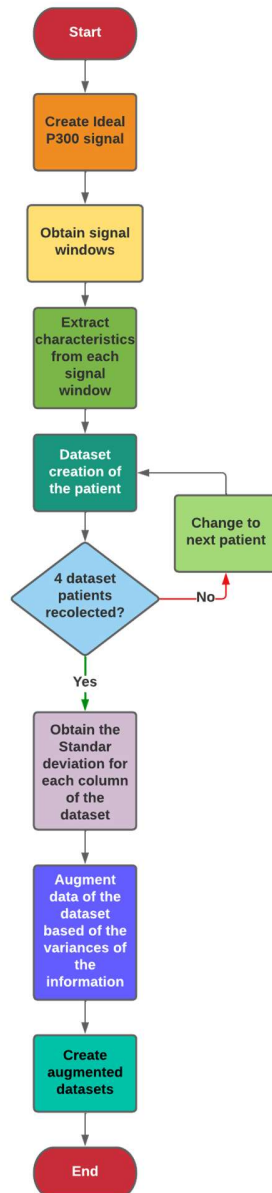


Figure 21 Diagrama de flujo fase 2

Se inició por crear artificialmente en Matlab una señal que contiene la forma muy similar a un potencial evocado P300, basado en sus proporciones de amplitudes y tiempos de subidas y bajadas, dando como resultado la siguiente ventana en muestras:

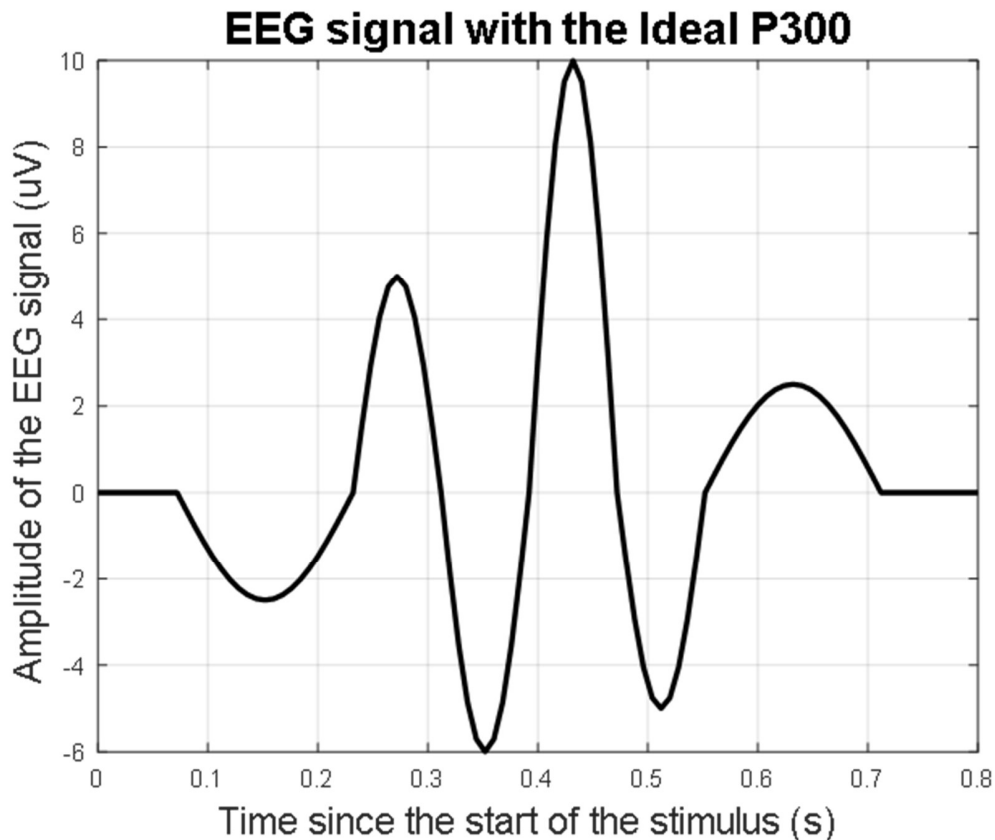


Figure 22 Señal P300 ideal artificial

Los valores de las amplitudes no tendrán mucha importancia, debido a que, en el momento de ser usada, será normalizada, solo es importante las relaciones entre las amplitudes altas y bajas. Esta señal tiene importancia en el aspecto de correlación cruzada, el cual es el proceso en el cual se compara una señal con otra, dando como resultado final otra señal que en cada muestra tendrá un valor en el que, si ambas señales iniciales son muy parecidas en esa muestra, resultará un valor final alto en esa muestra, y entre más diferentes sean las muestras más bajo será el dato final. Esto

es muy útil en el momento de comparar fragmentos de señales con la ideal, para obtener valores altos de correlación cruzada si se asemejan, o valores bajos si son muy diferentes en forma.

Lo siguiente que se procedió fue con la obtención de las diferentes ventanas de señales, los cuales son los fragmentos extraídos del total de la electroencefalografía, donde tienen de referencia los marcadores creados por Psychopy. Una demostración de una ventana sin P300 y una con P300 es presentada a continuación:

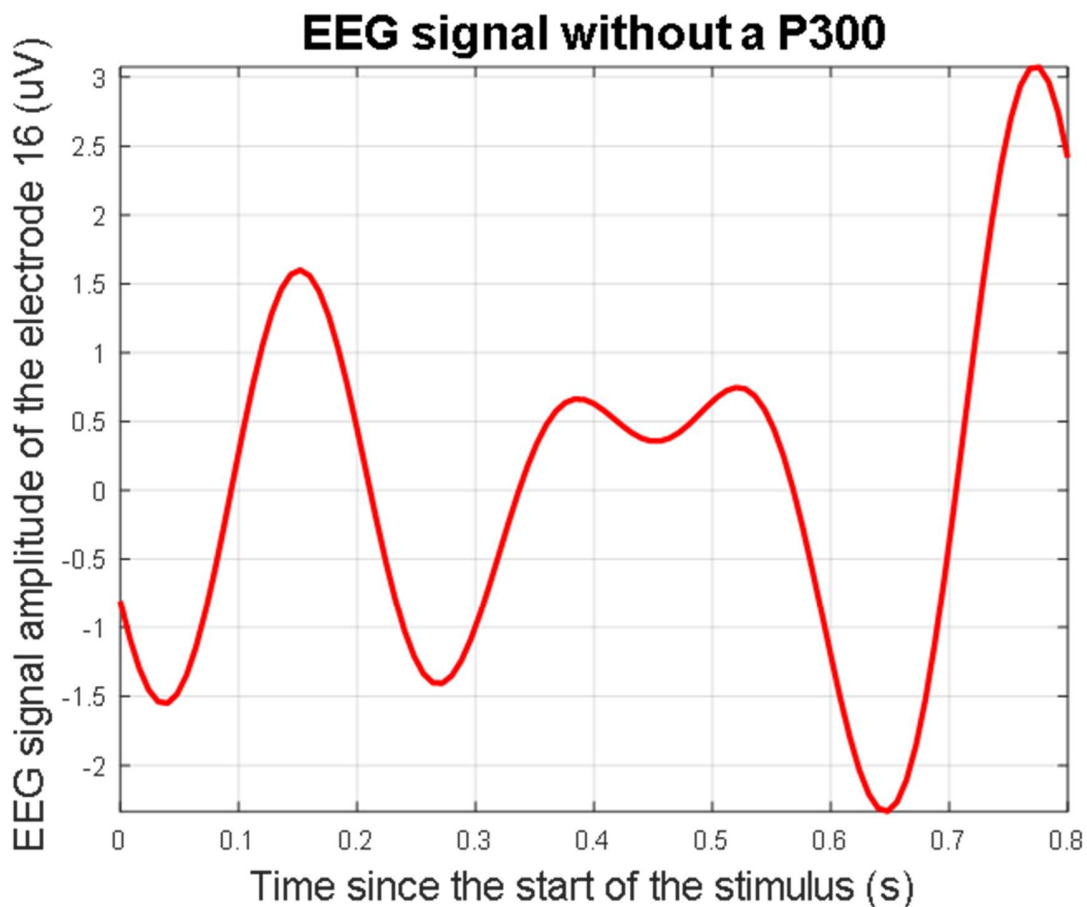


Figure 23 Señal real ejemplificando la ausencia de P300

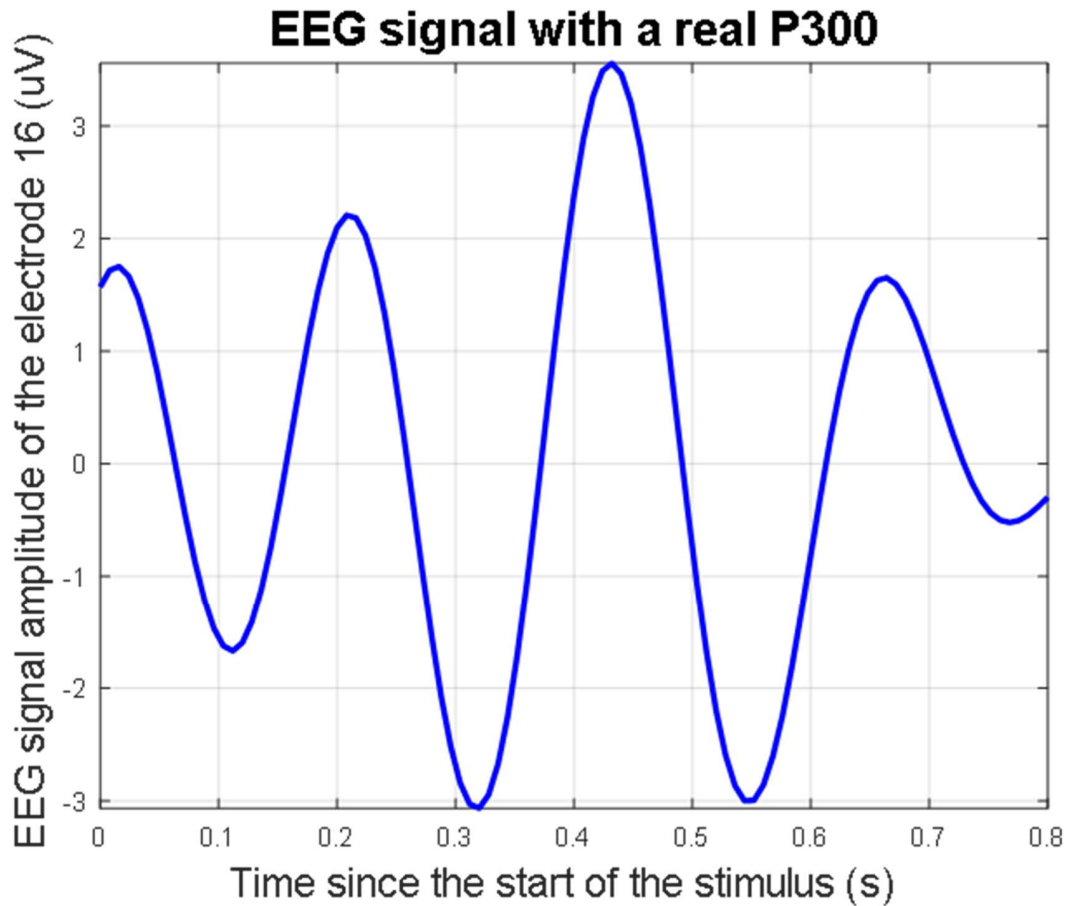


Figure 24 Señal real ejemplificando la presencia de P300

Esto nos permite visualizar la similitud entre la existencia de potencial evocado P300 con la señal ideal creada de forma manual.

Una vez se tienen todas las ventanas de todos los electrodos, se procedió a iniciar la creación del dataset del participante, donde se tendrán los datos de las características extraídas de las ventanas de señales. Se comenzó con encontrar la amplitud máxima de la señal, del cual Matlab usa la función “max” para establecer el valor más alto del conjunto de datos. Si la señal tiene P300, el pico será proporcionalmente más alto, mientras que, si no es una P300, tenderá a tener un valor pico más bajo:

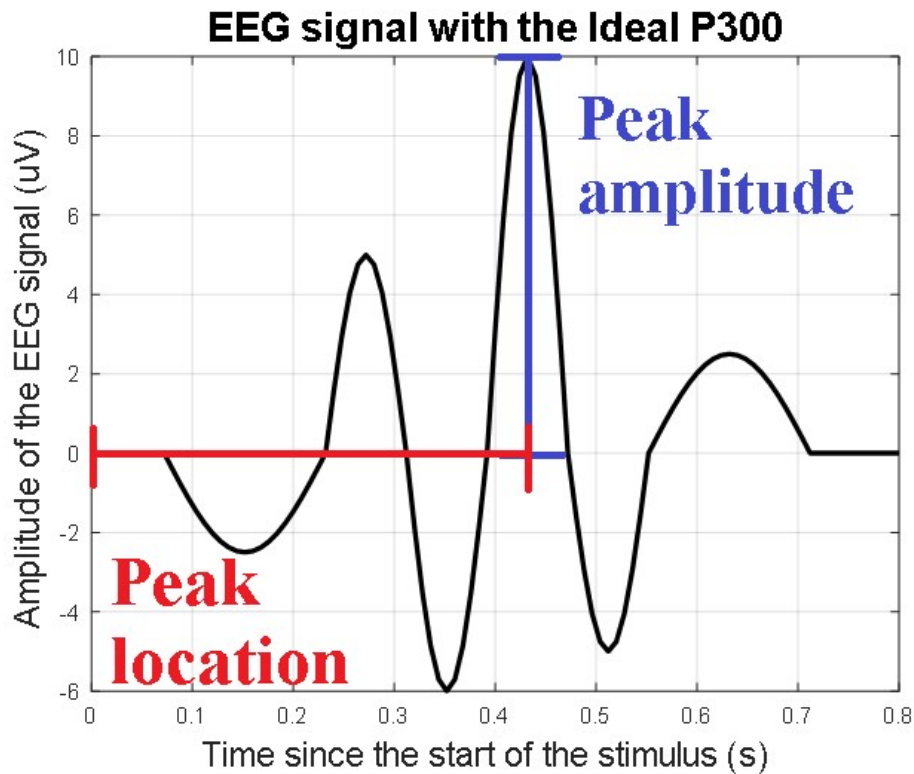


Figure 25 Características 1 y 2 de la extracción

Tal como se ilustra en la figura, la fórmula de Matlab se establece como:

$$V_{peak} = \max(signal)$$

Siguiente a esta característica, está la ubicación del pico máximo, el cual es un valor entero que marca el número de muestra en el que se encuentra localizado el valor de pico máximo de la señal. Si la señal tiene componente de P300, el pico máximo se localizará cerca al centro de las muestras, mientras que si no tiene P300 tenderá a tener el pico más alto en una ubicación alejada del centro de la señal. Matlab tiene la función “find” el cual es el equivalente a encontrar la coordenada X del punto máximo de la señal. Su fórmula se ilustra de la siguiente manera:

$$Loc_{peak} = find(V_{peak})$$

La tercera característica a extraer corresponde a la correlación cruzada entre la señal original y la señal de la ventana, donde se inicia por obtener el valor de correlación cruzada en el centro, debido a que la señal ideal creada de forma artificial tiene el pico más alto en todo el centro de las muestras, así que, si empata lo más cercano con una señal que tiene P300, resultará un valor de correlación cruzada en el centro con un valor alto. Esto se ilustra a continuación:

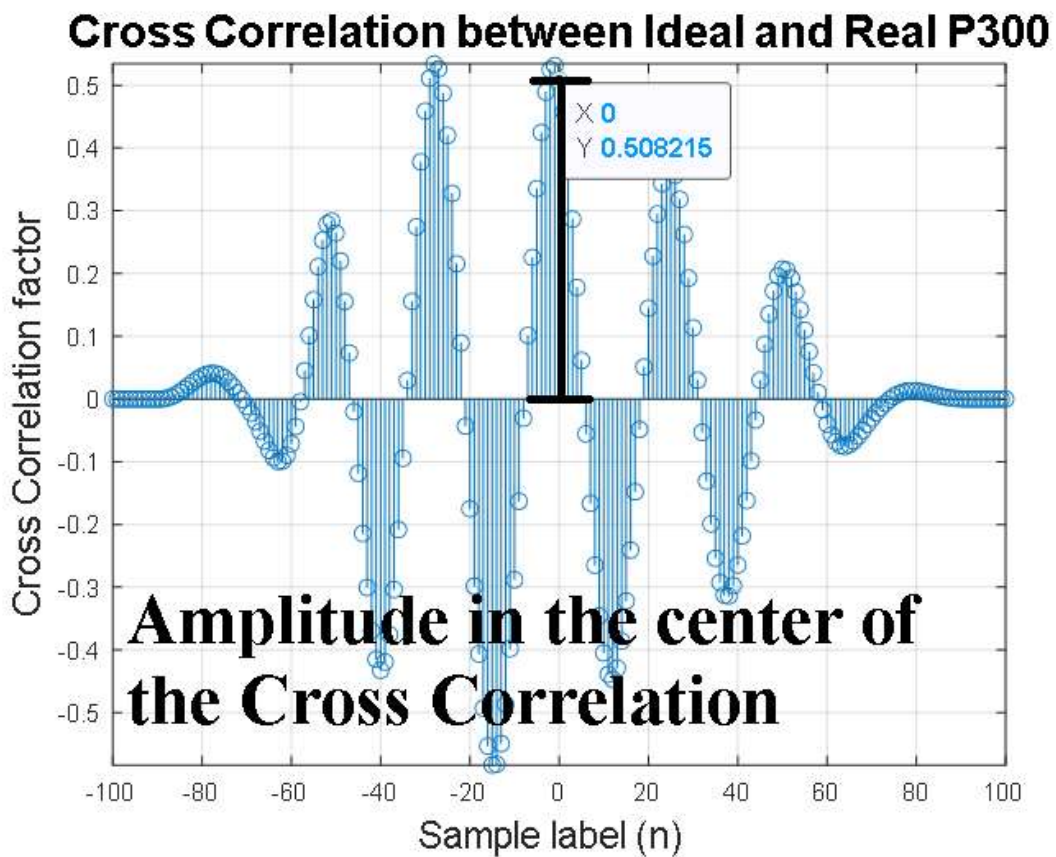


Figure 26 Característica 3 de la extracción

Y su fórmula es dado con la ecuación de Matlab usando “xcorr” que toma el número 50 como el centro del total de 101 muestras, y extrae ese valor de amplitud:

$$A_{centre} = xcorr(50,1)$$

Sacando provecho a la obtención de toda la correlación cruzada, se procedió a calcular el área bajo la curva de la señal final, donde en el caso de parecerse a una P300, debería de obtenerse un área de alto valor, mientras que, si las señales no se parecen, debería arrojar valores de área bajos. Esto se ejemplifica en la siguiente gráfica:

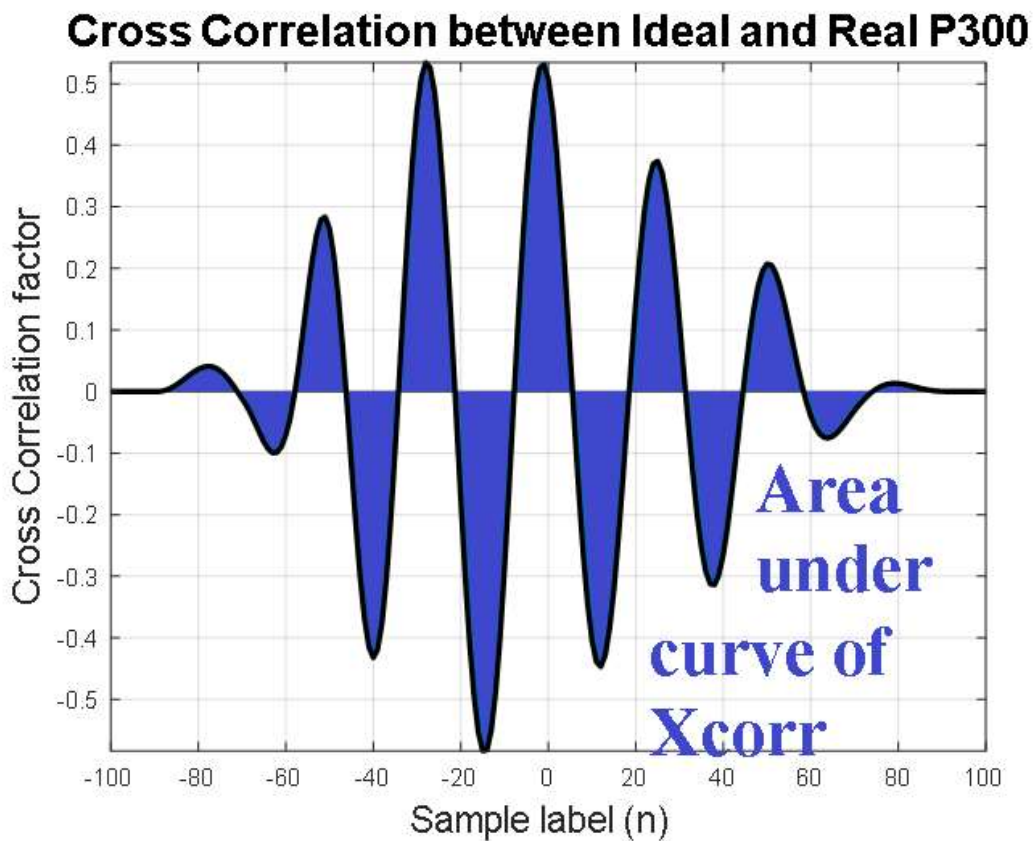


Figure 27 Característica 4 de la extracción

Del cual se tiene su fórmula matemática interpretada como una integral definida, ya que este devuelve el área bajo la curva de una función:

$$Area_{xcorr} = \int_{-100}^{100} xcorr$$

Y aplicado en Matlab, la función `xcorr` devuelve un vector discreto, y para funciones discretas la integral se define como una sumatoria:

$$Area_{xcorr} = \sum_{n=-100}^{100} (|xcorr|)$$

Con esto, la siguiente característica a extraer consiste en calcular el espectro de frecuencias de la ventana de señal, y medir la potencia que presenta en las frecuencias entre 4 y 5 Hertz, ya que estos suelen ser el rango de frecuencias en el que normalmente se encuentra la señal P300, comparado con la ausencia de P300 que tendrá una potencia reducida en ese rango de 4 y 5 Hertz.

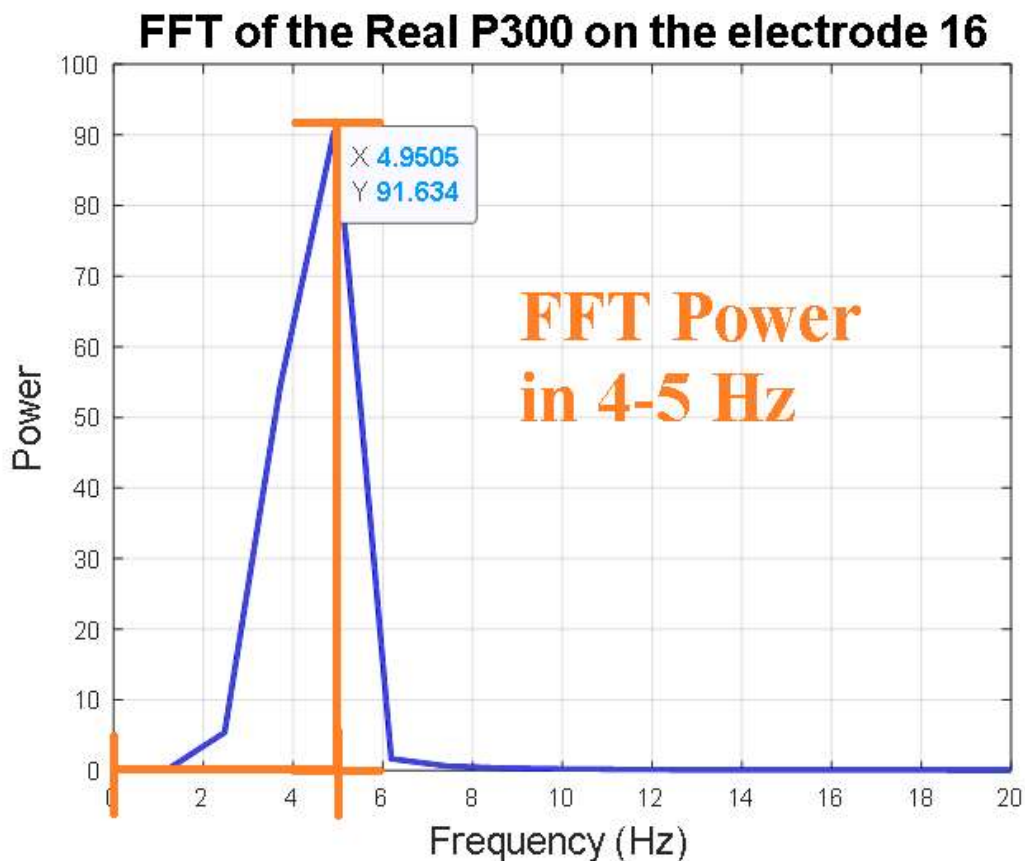


Figure 28 Característica 5 de la extracción

Por lo cual primero se necesita realizar la transformada rápida de Fourier a la señal, siendo matemáticamente de la siguiente forma:

$$g(\varepsilon) = \sqrt{\frac{\beta}{2\pi}} \int_{-\infty}^{\infty} f(x) e^{i\beta\varepsilon x} dx$$

En el cual el enfoque será en el rango de 2 a 6Hz, así que, habiendo obtenido la señal en frecuencia, Matlab extrae la característica con las fórmulas:

$$Y = \text{fft}(\text{signal})$$

$$FFT_{5Hz} = \sum_{n=2}^6 \text{Power}(Y(n, 1))$$

Por último, se obtiene el área bajo la curva de la señal original, en el cual esta área variaría entre una señal con P300 y una sin este potencial evocado, basado en los balances del voltaje, donde la P300 tenga una distribución más constante.

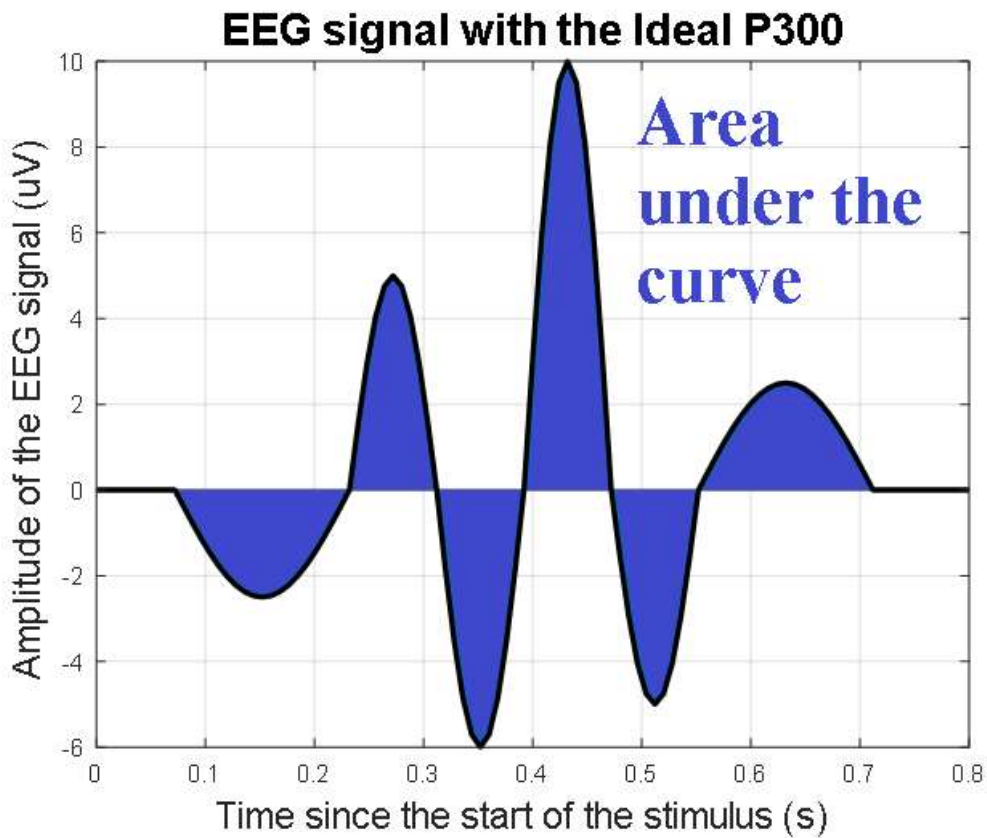


Figure 29 Característica 6 de la extracción

Por lo tanto, al igual que en el área de la correlación cruzada, la fórmula para su obtención sería dada por:

$$Area_{signal} = \int_0^{101} signal$$

Y en Matlab sería a través de la función “trapz”:

$$Area_{signal} = trapz(signal)$$

Una vez obtenidos todas las características de un electrodo, se repite para todos los electrodos, en el que se añade al final del dataset si esos conjuntos de ventanas son determinados a partir de un

marcador de presencia o ausencia de P300. Al final se obtiene un dataset con 97 columnas de características incluídas la presencia o no de P300.

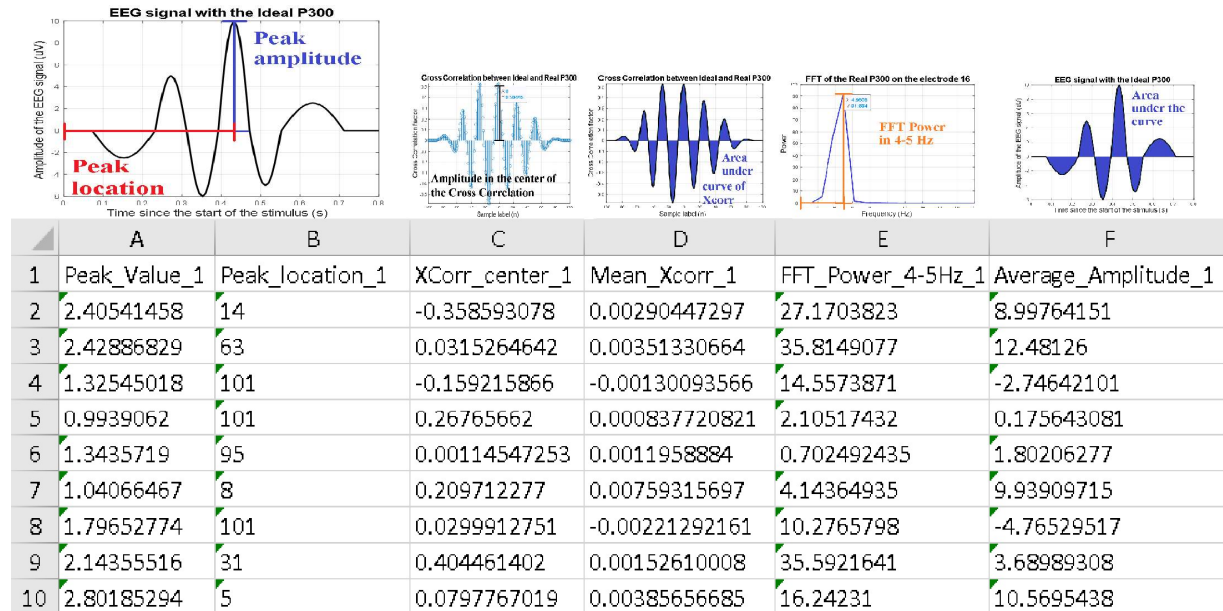


Figure 30 Dataset con sus correspondientes características extraídas

Por lo que esto es por cada participante, y por lo que se realizaron 3 sesiones de 5 minutos cada uno, cada sesión con 50 marcadores de cero y 50 marcadores de 1, se genera un total de 300 filas de datos, 150 con presencia de P300 y 150 con ausencia del mismo.

296	1.15339684,91,0.113986497,0.00235836618,6.2206756,3.94903657,1.40012372,95,-
297	1.0719192,14,-0.278993288,0.00236111936,12.9228158,4.38286949,1.98724926,13,-
298	1.57769144,16,-0.500020147,0.00075002649,37.4394594,2.44379698,1.5128001,74,-
299	1.44746685,17,-0.467135256,-0.000381065979,23.9538853,-0.118117576,1.6343319
300	1.54805124,87,0.235500891,-0.00146612232,5.57507509,-2.1616746,2.37763619,26,
301	0.59696877,47,-0.196999443,0.00190333794,3.52441639,1.52413816,2.53661346,45
302	

Figure 31 Tamaño dataset sin data augmentation

Una vez se tienen los datasets de todos los 4 participantes del estudio, se procedió a realizar un aumento de datos, esto debido a que para entrenar los modelos de Machine Learning es recomendable tener una alta cantidad de datos, para así obtener mejores predicciones.

Por el hecho de que no son imágenes a los que se les debe aumentar datos, sino un conjunto de números, la forma que se procedió a realizar este data augmentation con el dataset original, fue a partir de realizar copias del original, del cual cada copia se le adicionaba un porcentaje de ruido, ya fuera sumando o restando. Esto se logró a partir de las desviaciones estándar y la varianza de cada columna, ya que variaban entre características y entre electrodos, por ejemplo, mientras la desviación estándar de la columna de valor pico del electrodo 1 podría ser de 3, la desviación estándar de la ubicación del valor pico del electrodo 7 podría ser de 25, así que para agregar ruido en un porcentaje que fuera dentro de los rangos de desviación estándar, era necesario obtener las varianzas de cada columna de datos, para así extraer un porcentaje máximo de desviación estándar en el que se modificaría ligeramente de forma aleatoria.

Un ejemplo se describe a continuación, donde se tiene un número en la primera columna del dataset original con valor 4, y toda esa columna tiene una varianza de 0.5. En este caso, la primera réplica de dataset tomará el valor de 4 y le sumará el mismo 4 multiplicado por un valor aleatorio que se encuentre entre -0.5 y 0.5, el cual podría ser 0.22, así que:

$$original = 4$$

$$varianza = 0.5$$

$$replica = original + (original * random(-varianza, varianza))$$

$$random(-varianza, varianza) = 0.22$$

$$replica = 4 + (4 * 0.22)$$

$$replica = 4.88$$

Así que por el hecho de que se garantiza con las varianzas que el valor aleatorio dejará a la réplica dentro de la desviación estándar, se garantiza un nuevo dato dentro del rango aceptable de dato original afectado por ligero ruido.

Al final del proceso, se llega a aumentar el dataset original en 10 veces, el cual se evidencia en el momento de realizar los histogramas de una columna de datos y compararlos entre el dataset original y el aumentado:

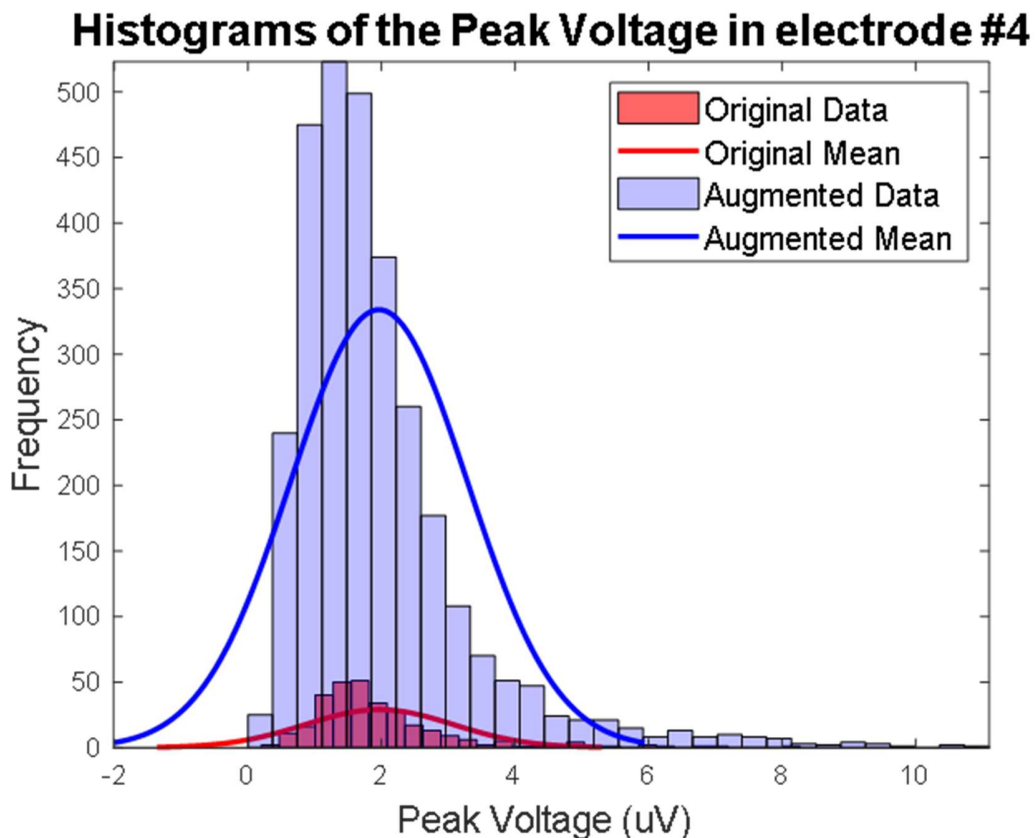


Figure 32 Histogramas comparando entre columna de datos del dataset original (rojo) con el dataset aumentado (azul).

Se observa que el comportamiento con respecto a la mayor aparición de valores se mantiene en la media entre original y aumentado, garantizando un efectivo aumento de datos conservando la varianza y desviación estándar del dataset original. Esto nos permite pasar a la siguiente fase del proyecto, para darle uso a los datasets aumentados.

2.3. FASE 3, CREACIÓN DEL SISTEMA DE MACHINE LEARNING:

Esta fase involucra los datasets creados con anterioridad, donde se usarán los datos sin ampliar y aumentados, con los cuales se realizarán las respectivas comparaciones de resultados. Comienza con la estructura de la fase 3 del proyecto:

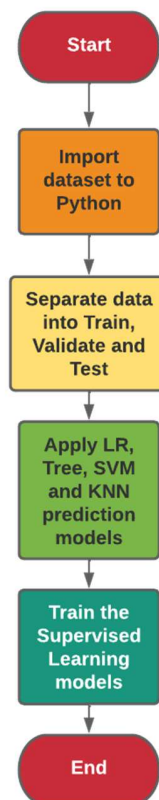


Figure 33 Diagrama de flujo fase 3

Inicialmente se define cada uno de los modelos de Machine Learning de aprendizaje supervisado, de los cuales se usarán LR (Regresión Logística), Árbol o Tree (Árbol de Decisión), SVM (Máquina de Soporte Vectorial) y KNN (K Vecinos más Cercanos).

Ya con los modelos definidos, se procede a ingresar los datasets a JupyterLab, el cual es una derivación de Python en el cual se puede ejecutar por segmentos la programación, mientras permite visualizar gráficas de diferentes clases. El dataset a ser analizado es importado a JupyterLab como se ilustra a continuación:

```
mainpath = "../EEG_python/"  
filename = "Dataset.csv"  
fullpath = os.path.join(mainpath, filename)  
df_original = pd.read_csv(fullpath)
```

Figure 34 Importación del dataset a JupyterLab

En el cual se definió la ruta del archivo, se le hizo la lectura y se almacenó en la variable “df_original”. Una vez hecho esto, se realizó una ligera limpieza de datos, tales como remover una columna que se genera automáticamente que contiene la enumeración de las columnas, al igual que tomar la columna de P300 que tiene valores de 0 para ausencia y 1 para presencia, y convertirla en dos columnas con valores lógicos con dummies, que son columnas de una característica booleana, y se descartó la columna “dummy_P300_0” ya que no se iba a necesitar por el hecho de ya tener su opuesta llamada “dummy_P300_1”. Hecho esto, se procede a tomar el total del dataset, y se separan los datos en un 70% del total para entrenamiento, 20% para validación y 10% para pruebas, como se observa en este fragmento de código:

```

from sklearn.model_selection import train_test_split
X_train, X_test_inicial, y_train, y_test_inicial = train_test_split(X, y, test_size=0.3, random_state=4)
X_validation, X_test, y_validation, y_test = train_test_split(X_test_inicial, y_test_inicial, test_size=0.33, random_state=4)
print ('Train set:', X_train.shape, y_train.shape)
print ('Validation set:', X_validation.shape, y_validation.shape)
print ('Test set:', X_test.shape, y_test.shape)

```

```

Train set: (2100, 96) (2100,)
Validation set: (603, 96) (603,)
Test set: (297, 96) (297,)

```

Figure 35 Separación de dataset entre entrenamiento, validación y prueba

Con esto, se procede a crear cada modelo de aprendizaje supervisado, e ingresarles los datos de entrenamiento, para tener un modelo predictor de señal P300:

<pre> from sklearn.linear_model import LogisticRegression from sklearn.metrics import confusion_matrix LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train) LR </pre> <pre> LogisticRegression(C=0.01, solver='liblinear') </pre>	<pre> from sklearn.tree import DecisionTreeClassifier from sklearn.metrics import confusion_matrix algoritmo=DecisionTreeClassifier() algoritmo.fit(X2_train,y2_train) </pre> <pre> DecisionTreeClassifier() </pre>
<pre> from sklearn import svm from sklearn.metrics import confusion_matrix algoritmo2=svm.SVC() algoritmo2.fit(X3_train,y3_train) </pre> <pre> SVC() </pre>	<pre> from sklearn.neighbors import KNeighborsClassifier from sklearn.metrics import confusion_matrix algoritmo3=KNeighborsClassifier(n_neighbors=3) algoritmo3.fit(X4_train,y4_train) </pre> <pre> KNeighborsClassifier(n_neighbors=3) </pre>

Figure 36 Modelos de Machine Learning con los datos de entrenamiento: LR (izquierda superior), Tree (derecha superior), SVM (izquierda inferior) y KNN (derecha inferior)

Con esto ya se tienen los 4 modelos de aprendizaje supervisado entrenados con el 70% del total de datos del dataset suministrado, donde se observa que cada uno se entrena con variables de terminación de nombre "..._train". Cabe resaltar que el modelo de KNN requiere definirse desde el inicio el número de vecinos cercanos que tendrá en cuenta para el entrenamiento.

Esto nos permite continuar con la última fase del desarrollo del proyecto.

2.4. FASE 4, VALIDACIÓN Y MÉTRICAS DE EVALUACIÓN:

Esta última fase permitió obtener los resultados finales del estudio, del cual se tiene el siguiente esquema:

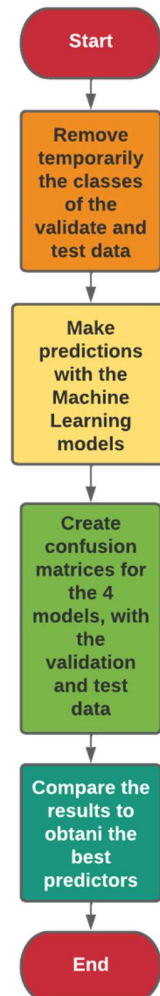


Figure 37 Diagrama de flujo fase 4

Se inició por remover temporalmente las clases de los conjuntos de datos de validación y prueba, con el propósito de ser ingresados a los diferentes modelos de Machine Learning, y así que determinen qué predicciones se establecen, si se encuentra o no la P300. Una vez hecho esto, se

obtuvieron los resultados de predicciones, y con esto se pudo comparar con los verdaderos valores de las clases, ya fuera presencia o ausencia de señal P300 basados en las características extraídas de las señales.

Lo que se iba obteniendo, eran las matrices de confusión, los cuales son cuadros que comparan los valores predichos con los reales, en el que las predicciones verdaderas de presencia y ausencia de P300 se conocen como “Verdaderos Positivos” y “Verdaderos Negativos” respectivamente, mientras que las predicciones falsas en las que el sistema considera que hay P300 y en realidad no había se conoce como “Falsos Positivos” y la inversa son los “Falsos Negativos”. Con esto, una vez sometidos los grupos de datos de validación y prueba a los modelos de Machine Learning, se obtenían los siguientes resultados:

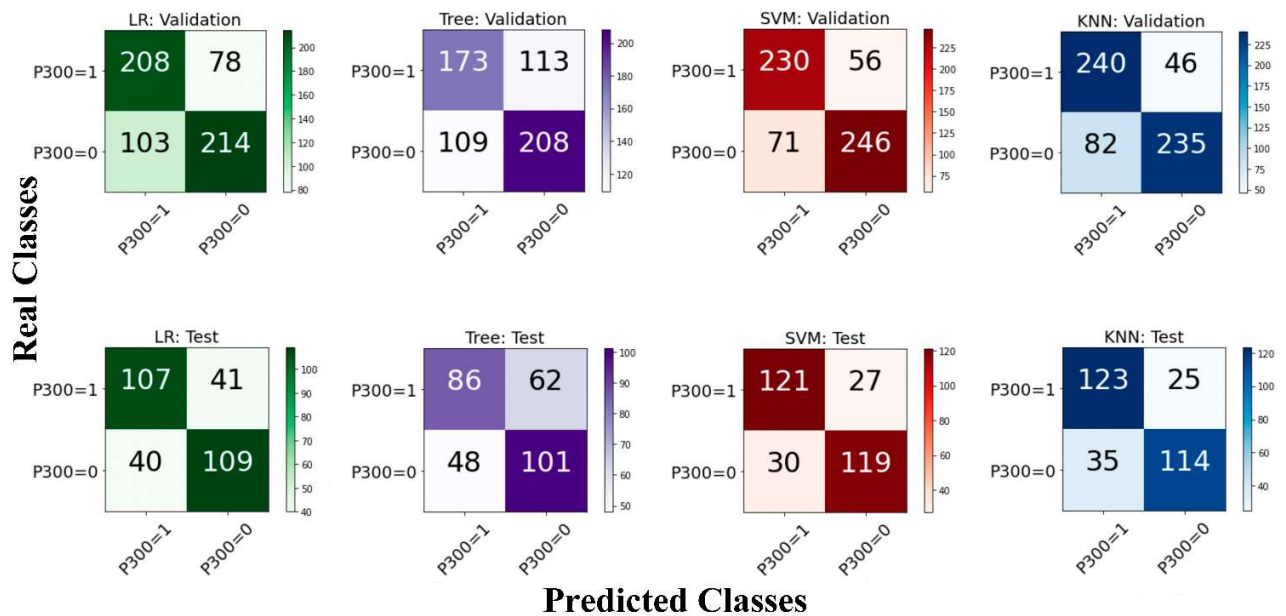


Figure 38 Matrices de confusión datos de validación (superiores) y datos de prueba (inferiores) con Regresión Logística (Verdes), Árbol de decisión (Morados), Máquina de Soporte Vectorial (Rojos) y K Vecinos más Cercanos (Azules) de participante I/A.



Se realizaron las matrices de confusión de todos los 4 participantes, con ambos datasets, los originales y los aumentados, permitiendo así realizar las respectivas comparativas y los análisis de datos deseado.

3. ANÁLISIS DE RESULTADOS Y DISCUSIÓN:

Ya habiendo llegado a la obtención de las matrices de confusión de los participantes, involucrando tanto las de validación como las de prueba, para los cuatro tipos de modelos, se observó que las matrices de confusión relacionados a SVM y KNN tienen un más alto número de verdaderos positivos y negativos, comparados con los valores de LR y árbol, los cuales tienen muchos más valores en los falsos.

Con estos datos de las matrices de confusión, se pueden realizar los cálculos de la exactitud, la precisión, la sensibilidad, la especificidad, y el más importante para las conclusiones, el puntaje F1. Cada uno de las métricas tienen las siguientes fórmulas matemáticas:

$$\text{Exactitud} = \frac{VP + VN}{VP + FP + FN + VN}$$

Donde:

$$VP = \text{Verdaderos Positivos}$$

$$VN = \text{Verdaderos Negativos}$$

$$FP = \text{Falsos Positivos}$$

$$FN = \text{Falsos Negativos}$$

Estas nomenclaturas será la misma para las demás ecuaciones.

$$\text{Precisión} = \frac{VP}{VP + FP}$$

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

$$\text{F1 Score} = \frac{2 * \text{precisión} * \text{sensibilidad}}{\text{precisión} + \text{sensibilidad}}$$

Una ventaja que ofrecen las librerías usadas en JupyterLab, es la posibilidad de calcular automáticamente los valores de estas métricas, por lo tanto, se procedió a generar los valores de Puntaje F1 de cada modelo, por cada participante, dando los siguientes resultados basados en validación y en prueba.

3.1. F1 SCORE DATOS DE VALIDACIÓN:

Realizando los cálculos del puntaje F1 de los cuatro participantes, y de ambos datasets por participante, siendo el original y el que tiene data augmentation, se obtuvo la siguiente gráfica:

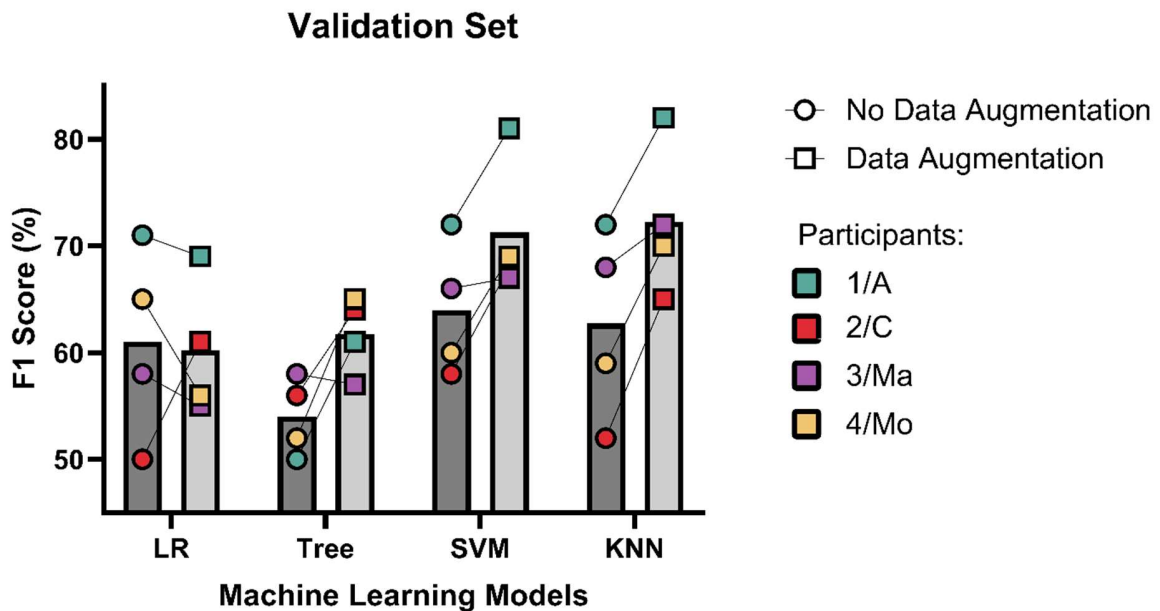


Figure 39 Porcentaje Puntaje F1 de los cuatro participantes para datos de validación

En el cual se observa que los promedios de puntaje F1, establecidos por las barras, muestran que los modelos de SVM y KNN tienen mejores predicciones, y de los cuales se ven mejorados aún más en el momento en el que se realiza data augmentation, incrementando en más de un 6% ese puntaje F1, comparado con los modelos de LR y árbol, los cuales se evidencia una disminución de porcentaje para el primer modelo, y un aumento mucho menor para el segundo modelo.

Otro análisis que se obtiene con esta gráfica, es los altos porcentajes para el participante 1/A, del cual superan en gran medida la media de porcentajes F1. Esto se debe a que el participante 1/A, quien es el mismo autor de este proyecto de grado, aplicó una mayor presión del casco sobre la cabeza, mientras se ajustaban los electrodos de tal forma que hicieran el mejor contacto posible sobre el cuero cabelludo, por ende, se obtendrían mejores datos de actividad cerebral, comparados con los otros participantes que se les aplicaba presión, pero menor, viéndose afectado por el cabello. Esto es evidencia en el participante 4/Mo en términos de puntaje, quien también tenía poco cabello en el momento de realizar las mediciones cerebrales, y obtuvo altos puntajes F1 también comparados con la media.

Por último, se observa en la gráfica que la regresión logística no se ve afectada de forma positiva en el momento de realizar un aumento de datos, ya que se evidencia que 3 de los 4 participantes presentaron disminuciones en su puntaje F1 en el momento de aplicar data augmentation, esto posiblemente por el hecho de que el sistema de regresión logística detectaría una mayor dispersión de datos, y por ende una reducción en la capacidad de separar los datos en dos grupos de manera efectiva.

3.2. F1 SCORE DATOS DE PRUEBA:

Este conjunto corresponde a hacer las pruebas con el 10% del total de los datasets, dando como resultados los siguientes valores:

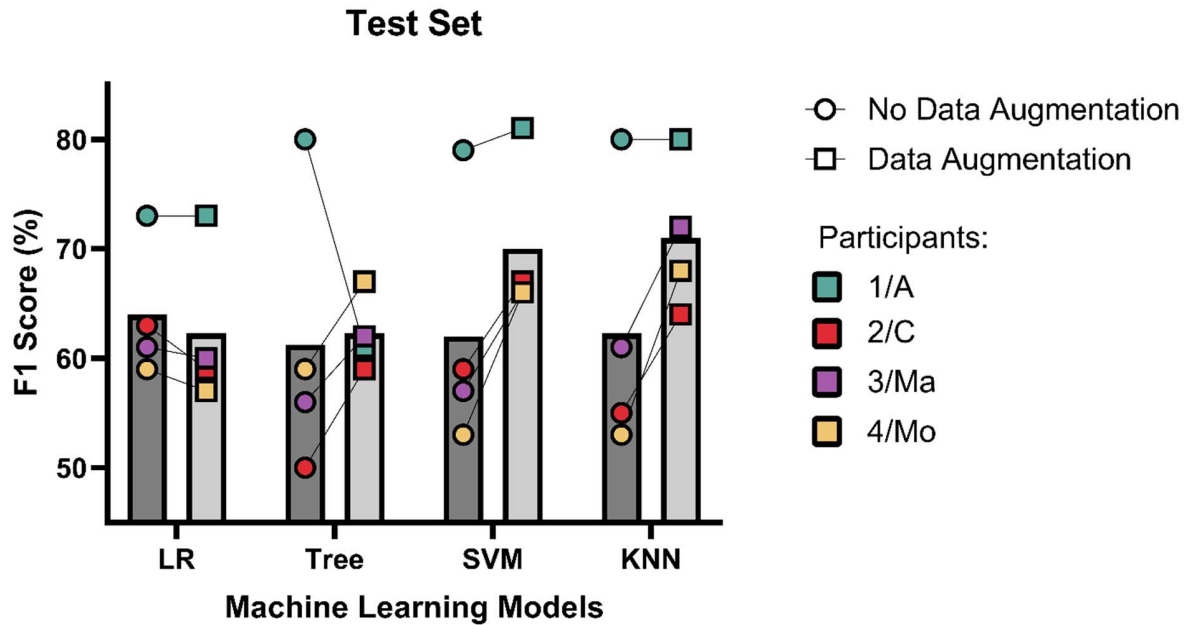


Figure 40 Porcentaje Puntaje F1 de los cuatro participantes para datos de prueba

En el que también se presenta comportamientos semejantes a los datos de validación. El puntaje F1 se reduce en 3 de 4 participantes durante el aumento de datos en la regresión logística, se presenta un aumento de porcentaje en SVM y KNN al aplicar data augmentation, y los puntajes son mayores en SVM y KNN comparados con LR y árbol. Con esto se visualiza que en el momento de realizar pruebas en participantes es recomendable usar los modelos de SVM o KNN para esa predicción de señales P300.

4. CONCLUSIONES Y TRABAJOS FUTUROS:

Con estos resultados finales, fue posible culminar el proyecto de grado, donde se pueden redactar las siguientes conclusiones:

- Con ayuda de los programas de OpenBCI, Psychopy y Matlab, fue posible obtener los resultados más favorables en términos de hardware con el casco, y de software con la comunicación de información por medio de LSL. Psychopy resultó ser muy efectivo en términos de establecimiento de marcadores coordinados con la aparición de estímulos visuales y auditivos aleatorios, y con el uso de LabRecorder se ancló de manera satisfactoria las señales EEG con los marcadores, siendo evidencia el potencial evocado visto en los electrodos O1 y O2. Matlab permitió obtener las señales y marcadores para extraerles sus características y crear así el dataset deseado para el estudio, junto con el aumento de datos para mejorar los resultados de predicciones de P300.
- El casco de electroencefalografía tiene un alto rendimiento siempre y cuando se puedan ajustar correctamente los electrodos sobre el cuero cabelludo, ya que se ve afectado por la cantidad de cabello que la persona tenga en el momento de realizar el estudio. De ser posible, para personas de abundante cabello, es recomendable ajustar los electrodos lo más profundo al casco posible, y de tenerse disponibilidad se puede llegar a usar gel conductor que aumente la conductividad de los electrodos sobre la cabeza.
- Las características extraídas de cada electrodo durante el estudio fueron suficientes para entrenar los modelos de aprendizaje supervisado, ya que se aplicó el método de análisis por cada instante en el que aparecía un estímulo, dando paso a realizar predicciones con cada marcador de P300 que iría apareciendo en las señales cerebrales.

- Los modelos de Machine Learning en su mayoría presentaban una mejoría en términos de predicciones de señal P300 en el momento en el que se les generaba un aumento de datos al dataset original, por el hecho de que teniendo más datos para entrenar permitía llegar a acertar con mayor precisión los nuevos datos que inicialmente recibía sin las clases.
- Los modelos de aprendizaje supervisado “Máquina de Soporte Vectorial” y “K Vecinos más Cercanos” resultaron ser los mejores predictores de la aparición de la señal P300 en este estudio, el cual se veía menos afectado por circunstancias como ruido, mientras se veía más favorecido por el aumento de datos.

Y, por último, como trabajos que se pueden proponer a futuro, es posible enfatizarse más en la modificación de los hiperparámetros de los modelos de aprendizaje supervisado, con el fin de poder aumentar la precisión de predicciones. También, se puede trabajar en la implementación de redes neuronales que sean capaces de aprender sobre los comportamientos cerebrales durante la visualización de estímulos aleatorios. Otro ítem posible a desarrollar, es el crear un sistema que permita realizar las extracciones de características en tiempo real, junto con hacer predicciones de la señal P300 en tiempo real, con el fin de darle uso en aplicaciones más avanzadas como ayudas para personas discapacitadas.

5. REFERENCIAS BIBLIOGRÁFICAS:

Amarillo, M. (2022). *Componentes Comunes del ERP, Parte 1*. Bogotá.

Biarnés Rabella, C. (2018). Diseño, caracterización y evaluación de electrodos capacitivos para la medida de ECG y EEG. *Universitat Politècnica de Catalunya*, 12-15.

Blanco Díaz, C. F., & Ruiz Olaya, A. F. (2019). Caracterización de señales de EEG relacionadas a potenciales evocados visuales en estado estacionario. *Ontare*, 18-20.

Broggi Angulo, O. A., Koc Gonzáles, D. G., & Martínez Esteban, P. C. (2022). *Guía de procedimiento de electroencefalografía y videoelectroencefalografía*. San Borja: Ministerio de Salud de la República del Perú.

Chandra Poonia, R., Singh, V., & Ranjan Nayak, S. (2022). *Deep Learning for Sustainable Agriculture, A volume in Cognitive Data Science in Sustainable Computing*. India: Elsevier.

Cohen, M. X. (2014). Advantages and Limitations of Time- and Time-Frequency-Domain Analyses. In M. X. Cohen, *Analyzing Neural Time Series Data, Theory and Practice* (pp. 15-30). London: The MIT Press.

El sistema 10-20 se utiliza en electroencefalografía (EEG). (s.f.). Obtenido de GVB geliMED GmbH: <https://gvb-gelimed.com/es/10-20-elektrodensystem-in-der-elektroenzephalografie-eeg/>

Electroencefalografía (EEG). (2018). Obtenido de BrainSigns srl: <https://brainsigns.com/es/science/s2/technologies/eeg>

- Gannouni, S., Aledaily, A., Belwafi, K., & Aboalsamh, H. (2021). Emotion detection using electroencephalography signals and a zero-time windowing-based epoch estimation and relevant electrode identification. *Nature Portfolio*, 5-7.
- García, T. T. (2011). Manual básico para enfermeros en electroencefalografía. *Especial*, 29-33.
- Gonzales, L. (19 de Julio de 2019). *K Vecinos más Cercanos – Teoría*. Obtenido de aprendeIA: <https://aprendeia.com/algorithmo-k-vecinos-mas-cercanos-teoria-machine-learning/>
- Hinrichs, H. (2004). Electroencephalography. In J. Moore, & G. Zouridakis, *Biomedical Technology and Devices Handbook* (pp. 153-173). Boca Raton: CRC Press LLC.
- Hojas, I. M. (s.f.). *Regresión Logística en Python*. Obtenido de Stat Developer: <https://www.statdeveloper.com/regresion-logistica-en-python/>
- Kumar, S. (19 de Julio de 2019). *Data Augmentation Increases Accuracy of your model — But how ?* Obtenido de Medium: <https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722>
- Macías Macías, J. M., Ramirez Quintana, J. A., Méndez Aguirre, J. S., Chacón Murgia, M. I., & Corral Sáenz, A. D. (2020). Procesamiento Embebido de P300 Basado en Red Neuronal Convolutiva para Interfaz Cerebro-Computadora Ubicua. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, vol. 9, núm. 2, 1-24. Obtenido de <https://www.redalyc.org/journal/5122/512267931005/html/>
- Morillo, L. E. (s.f.). ANÁLISIS VISUAL DEL ELECTROENCEFALOGRAMA. 145-153.
- N.N. (27 de Septiembre de 2023). *Desviación Típica*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Desviación_típica

- Nunez, P. L., & Srinivasan, R. (2006). Recording Strategies, Reference Issues, and Dipole Localization. In P. L. Nunez, & R. Srinivasan, *Electric Fields of the Brain, The Neurophysics of EEG* (pp. 275-313). New York: Oxford University Press.
- Peirce, J. G. (2019). *PsychoPy2: Experiments in behavior made easy*. Obtenido de Behavior research methods: <https://doi.org/10.3758/s13428-018-01193-y>
- Razavi, M., Janfaza, V., Yamauchi, T., Leontyev, A., Longmire-Monford, S., & Orr, J. (2021). OpenSync: An opensource platform for synchronizing multiple measures in neuroscience experiments. 3-7.
- Ríos P., L., & Álvarez D., C. (2013). Epilepsy diagnostic: update in eeg contribution. *Revista Médica Clínica Las Condes*, 953-957.
- Romo, R. (s.f.). *Árboles de Decisión / Decision Trees con python*. Obtenido de Rubén J. Romo: <https://rubenjromo.com/decision-trees/>
- Sherman, D., & Walterspacher, D. (2006). Electroencephalography. In J. G. Webster, *Encyclopedia of Medical Devices and Instrumentation* (pp. 62-83). Hoboken: John Wiley & Sons, Inc.
- Urgilés Cárdenas, D. F., & Vásquez Rodríguez, G. J. (2017). Implementación de un Sistema BCI para el Análisis del Comportamiento de Bioseñales Neurológicas. 14-32.
- Wallisch, P., Lusignan, M. E., Benayoun, M. D., Baker, T. I., Dickey, A. S., & Hatsopoulos, N. G. (2014). Matlab Tutorial. In P. Wallisch, M. E. Lusignan, M. D. Benayoun, T. I. Baker, A. S. Dickey, & N. G. Hatsopoulos, *Matlab for Neuroscientists* (pp. 38-53). Oxford: Elsevier.



Wood, C. C., Truett, A., Goff, W. R., Williamson, P. D., & Spencer, D. D. (2008). On the Neural Origin of P300 in Man. *Science Direct*, 51-56.

Zhuang, P., Toro, C., Grafman, J., Manganotti, P., Leocani, L., & Hallett, M. (1996). Event-related desynchronization (ERD) in the alpha frequency during development of implicit and explicit learning. *Elsevier*, 1-8.

6. ANEXOS:

6.1. PROGRAMAS MATLAB:

La carpeta contiene los programas de Matlab usados para el proyecto, de los cuales se describirá brevemente la función de cada uno, en el orden de ejecución en la tesis:

6.1.1. EEGLAB:

La subcarpeta con el nombre de “eeglab2024.0” tiene internamente el archivo llamado “eeglab.m” el cual es el primero que se ejecuta en Matlab. Es aquel que permite importar la información de la unión de las señales EEG junto con los marcadores, a partir del archivo de extensión .xdf creado con LabRecorder.

6.1.2. CLASIFICACIÓN P300:

El archivo “Clasificacion_P300.m” es aquel que extrae las características de las señales electroencefalográficas, basadas en las ventanas que genera con los marcadores de cada porción de señal electroencefalográfica para los 16 electrodos.

6.1.3. ACOPLADOR DE DATOS:

El archivo con nombre “Acoplador_datos.m” permite unir múltiples datasets de un mismo participante para formar uno solo general.

6.1.4. AUMENTO DE DATOS:

El programa “Augment_data.m” es aquel que genera el aumento de datos de un dataset, basado en las desviaciones estándar de cada columna de información, al igual que porcentajes aleatorios para replicar el dataset original con ruido en los datos. Es posible alterar el tamaño final del dataset aumentado, cambiando el número de iteraciones del proceso de aleatoriedad.

6.1.5. FILTRADO DE SEÑAL:

Este programa denominado “filtersignal.m” no es directamente ejecutado en el proyecto, sino que es una función usada en el programa “Clasificacion_P300.m” para la parte de filtrado de señal a través del uso de un pasabandas, el cual permitía establecer un rango específico de frecuencias a permitir y descartar las demás.

6.1.6. CREACIÓN DE VENTANAS:

El programa “windowing.m” tampoco es directamente ejecutado en el proyecto, solo es requerido de encontrarse también en la misma dirección que el programa “Clasificacion_P300.m” ya que es usado en la parte de creación de ventanas de presencia o ausencia de señal P300, teniendo de funcionamiento el establecer el centro de la ventana, el número de muestras que se desea a cada lado desde el centro, y si se desea normalizar o no la señal.

6.1.7. VISUALIZACIÓN DE POTENCIALES EVOCADOS:

Por último, el programa de nombre “ERP.m” permite importar cualquier archivo de formato inicial de tipo .xdf, del cual se requiere guardar, con ayuda de EEGLab, en un formato .mat, con ejemplo aquel archivo de nombre “EEG.mat” para obtener los potenciales evocados de un electrodo en específico. El programa fue creado por el codirector de esta tesis, con comentarios añadidos por el autor de este proyecto de grado.

6.2. PROGRAMA JUPYTERLAB:

La carpeta consiste en los siguientes archivos:

6.2.1. DATASET A ANALIZAR:

El Excel con nombre “Dataset.csv” es aquel que contiene el archivo a analizar y procesar en el programa de Machine Learning. Se debe dejar este nombre para cualquier dataset con el fin de ser recibido correctamente por el programa de JupyterLab.

6.2.2. DETECTOR DE LA SEÑAL P300:

El programa “Detector_P300.ipynb” es el archivo de JupyterLab que ejecuta todo el proceso de Machine Learning del proyecto, entre ellos la importación, limpieza y procesamiento de los datos, en los 4 modelos de aprendizaje supervisado implementados.

6.2.3. DATASETS LIMPIOS:

Los archivos de Excel de nombres “Dataset_final_c.csv” y “Dataset_final_x.xlsx” son los generados por el programa de JupyterLab, donde se han realizado limpieza de datos y se producen específicamente para ser utilizados por el archivo de detección de P300 en JupyterLab.

6.2.4. RESPALDO AUTOMÁTICO:

La subcarpeta con nombre “.ipynb_checkpoints” tiene un respaldo automático de última ejecución del archivo principal de JupyterLab.

6.3. PROGRAMA PSYCHOPY:

El programa de Psychopy de nombre “Generador_P300.psyexp” es el archivo que muestra los cuadrados en pantalla cambiando de color, mientras crea los marcadores para las señales EEG. Este programa depende de la hoja de cálculo con nombre “Colores.xlsx” para generar los colores de los cuadrados a presentar en el experimento, al igual que el tono de sonido y su respectivo volumen.



La carpeta lleva incluida un archivo llamado “Generador_P300_lastrun.py” que se genera automáticamente al ejecutar el programa de Psychopy inicial.

6.4. DATASETS GENERADOS:

Los datasets creados a partir de los datos recolectados de los participantes tienen dos referencias de nombres, los que inician con “Dataset_total_...” son los datos recolectados de todo un participante, sin aumento de datos, mientras que los que inician con “Dataset_augmented_...” son los datasets aumentados de los participantes. Cabe resaltar que cada dataset finaliza en el nombre de las referencias de cada participante del estudio, participante 1/A, participante 2/C, participante 3/Ma y participante 4/Mo.

6.5. CONSENTIMIENTOS INFORMADOS:

Se presentan los consentimientos informados firmados por los participantes 2/C, 3/Ma y 4/Mo, para la autorización legal del uso de las señales electroencefalográficas extraídas durante el estudio. Para tener en cuenta, no se tiene el consentimiento informado del participante 1/A debido a que es el mismo autor del proyecto.

6.6. VIDEOS:

Se tienen demostraciones del funcionamiento real del programa OpenBCI de nombre “Interfaz_OpenBCI.mp4” en el que se observará una toma de muestras de señales cerebrales en tiempo real del participante 1/A; y el funcionamiento de Psychopy de nombre “Interfaz_psychopy.mp4” donde se visualiza la pantalla que genera el programa para crear los estímulos visuales.



6.7. MANUAL DE USUARIO:

Por último, se entrega el manual de usuario para que cualquier persona pueda poner en operación el casco EEG y obtener sus propios datos, desde los programas preliminares a instalar en su respectivo computador, hasta la obtención en Matlab de los datos de electroencefalografía y posicionamiento de los marcadores, para el uso que se le desee dar académicamente.