

Prototipo de mano robótica mediante el uso de una interfaz cerebro máquina (BCI) con un sistema control de movimiento a través de señales EEG

Jonathan José Rincón Cerquera

Asesor

Jairo Luis Gutiérrez Torres

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI
Ingeniería Electrónica

2024

Resumen

Este proyecto tiene como objetivo, emplear señales neuronales para el movimiento de prótesis o manos robóticas con el fin de mejorar la calidad de vida de personas con discapacidad motora o sin manos. Se aborda la problemática de la discriminación que enfrentan estas personas en aspectos de inclusión social y empleabilidad, y la dificultad para acceder a prótesis avanzadas debido a su costo. Para el desarrollo del prototipo, se establece una combinación de ciencia biomédica e ingeniería. Se emplean técnicas de aprendizaje automático para la identificación y clasificación de señales, y se configuran en MATLAB para el procesamiento de datos y el control del movimiento de la mano robótica. El diseño investigativo seguirá un enfoque mixto y se divide en seis fases: documentación y adquisición de señales EEG, identificación de características y clasificación de señales, adquisición de componentes electrónicos, diseño e implementación del prototipo y pruebas de funcionamiento.

Palabras clave: EEG, mano robótica, prototipo, BCI, señales encefalografías.

Abstract

This project aims to use neural signals for the movement of prostheses or robotic hands in order to improve the quality of life of people with motor disabilities or without hands. The problem of discrimination faced by these people in aspects of social inclusion and employability is addressed, as well as the difficulty in accessing advanced prostheses due to their cost. For the development of the prototype, a combination of biomedical science and engineering is established. Machine learning techniques will be used for the identification and classification of signals, and they will be configured in MATLAB for data processing and control of the movement of the robotic hand. The research design will follow a mixed approach and will be divided into six phases: documentation and acquisition of EEG signals, identification of characteristics and classification of signals, acquisition of electronic components, design and implementation of the prototype, and functional tests.

Keywords: EEG, robotic hand, prototype, BCI, encephalography signals.

Tabla de contenido

Introducción	13
Objetivos.....	16
Objetivo general.....	16
Objetivos específicos.....	16
Definición del problema.....	0
Justificación	3
Marco Referencial	5
Electroencefalograma	5
Historia del Electroencefalograma.....	5
Sistema Internacional 10 – 20.....	6
Anatomía Cerebral	7
Hemisferios Cerebrales.....	7
Lóbulos del Cerebro.....	8
Parte Motora del Cerebro.....	9
Ondas o Ritmos Cerebrales.....	10
<i>Ondas Delta</i>	10
<i>Ondas Theta</i>	10
<i>Ondas Alfa</i>	11
<i>Ondas Beta</i>	11
<i>Ondas Gama</i>	12
Artefactos - Problemas en el Análisis EEG	12
Adquisición de Señales - Banco de Señales Fisiológicas	13

<i>Banco de Señales Physionet</i>	13
<i>Banco de señales IEEE</i>	14
Interfaz Cerebro Máquina	14
Partes Importantes de un BCI	15
<i>Sensores</i>	15
<i>Procesamiento de señales</i>	15
<i>Aplicación</i>	15
Mano Robótica.....	15
Antecedentes de Prototipos con EEG	16
<i>Antecedente 1</i>	16
<i>Antecedente 2</i>	17
<i>Antecedente 3</i>	18
<i>Antecedente 4</i>	19
<i>Antecedente 5</i>	19
Metodología	21
Fase 1: Etapa de Planificación y Análisis de Datos	21
Organización de datos basados en Emotiv-based-Dataset.....	23
Análisis de Datos Enfocado a Emotiv	24
Verificación Formato De Datos CSV	27
Visualización Inicial de Datos	27
Inspección de Características	31
Parámetros en el Dominio del Tiempo: Clasificación – Características Temporales	
Estadísticas.....	31

<i>Valor Absoluto Medio</i>	31
<i>Energía Total</i>	31
<i>Parámetro ZRC (Zero Croos Rate)</i>	32
<i>Parámetros de Hjorth</i>	32
<i>Parámetro Actividad (Representa la Potencia de la Señal de EEG)</i>	32
<i>Parámetro Movilidad (Representa la Frecuencia Media – Frecuencia Dominante de la Señal EEG)</i>	33
<i>Parámetro Complejidad (Representa los Cambios en la Frecuencia de la Señal, Compara la Señal EEG con una Señal Sinusoidal)</i>	34
Parámetros en el Dominio de la Frecuencia	34
<i>Espectro de Frecuencias de la Señal</i>	34
<i>Transformada Rápida de Fourier en MATLAB</i>	35
Aplicación Filtro de Mediana	35
Densidad Espectral de la Señal	36
<i>Espectrograma</i>	37
.....	37
Aplicación de Modelos Autorregresivos	38
<i>Aplicación del modelo matemático en MATLAB</i>	38
Matriz de Características.....	39
Uso de la Aplicación Classification Learner de MATLAB	41
Decodificación de Señales Neuronales para el Control de una Mano Robótica (Intención de Movimiento)	44
Diseño del Proyecto EEG Previo a la Implementación	45

Estructura Mecánica Mano Robótica.....	45
Actuadores	45
Sistema de control.....	46
Comunicación con la Mano Robótica.....	48
<i>Configuración Inicial Entre los Softwares</i>	49
<i>Carga y Procesamiento de Datos</i>	49
<i>Decodificación y Predicción</i>	49
<i>Visualización en Tiempo Real</i>	49
<i>Actuación de los Servomotores</i>	49
<i>Interpretación y Actuación en Arduino</i>	50
<i>Retroalimentación</i>	50
Diagrama del Circuito.....	50
Descripción del Diagrama del Circuito.....	50
<i>Microcontrolador Arduino</i>	50
<i>Módulo PCA9685</i>	51
<i>Servomotores</i>	51
<i>Conexión Serie</i>	51
<i>Alimentación y Tierra</i>	51
Implementación en Físico del Prototipo de Mano Robótica.....	52
Pruebas de Funcionamiento	53
Presentación de la Implementación del Sistema.....	53
Comportamiento del Sistema en MATLAB	54
Análisis de Resultados	55

Tiempo de Respuesta del Sistema.....	56
Conclusiones	57
Referencias Bibliográficas	58

Índice de Tablas

Tabla 1 <i>Especificaciones Técnicas del EMOTIV EPOC +</i>	26
Tabla 2 <i>Especificaciones Técnicas del Servodigital LFD-01M</i>	46
Tabla 3 <i>Especificaciones Técnicas del Módulo PCA9685</i>	48
Tabla 4 <i>Tiempo de Respuesta del Sistema</i>	56

Lista de Figuras

Figura 1 “ <i>Encefalograma</i> ”	5
Figura 2 “ <i>Sistema 10 – 20 con Electroencefalografía</i> ”	6
Figura 3 “ <i>Hemisferios Cerebrales</i> ”	8
Figura 4 “ <i>Funciones del Cerebro</i> ”	9
Figura 5 “ <i>Representación Ritmo Delta</i> ”	10
Figura 6 “ <i>Representación Ritmo Theta</i> ”	11
Figura 7 “ <i>Representación Ritmo Alfa</i> ”	11
Figura 8 “ <i>Representación Ritmo Beta</i> ”	12
Figura 9 “ <i>Representación Ritmo Gama</i> ”	12
Figura 10 “ <i>Cajero Automático PhysioBank</i> ”	14
Figura 11 “ <i>Mano Robótica</i> ”	16
Figura 12 “ <i>Diagrama de Bloques del Sistema</i> ”	17
Figura 13 “ <i>Interfaz BCI Cyton Board</i> ”	20
Figura 14 “ <i>Fases del proyecto</i> ”	21
Figura 15 “ <i>Interfaz Banco de Señales Fisiológicas</i> ”	22
Figura 16 “ <i>Análisis de Datos en el Banco de Señales IEEE</i> ”	22
Figura 17 “ <i>Reorganización de Carpetas de la Data</i> ”	24
Figura 18 “ <i>Diadema EMOTIV EPOC+</i> ”	25
Figura 19 “ <i>Cargue de Datos en MATLAB</i> ”	27
Figura 20 “ <i>Registro de Datos de un Participante</i> ”	28
Figura 21 “ <i>Muestras del Participante 1-2 – Derivación Theta</i> ”	28
Figura 22 “ <i>Muestras del Participante 1-2 – Derivación Alpha</i> ”	29

Figura 23 “ <i>Muestras del Participante 1-2 – Derivación Beta</i> ”	29
Figura 24 “ <i>Muestras del Participante 1-2 – Derivación Gamma</i> ”	29
Figura 25 “ <i>Muestras del Participante 1-2 – 4 Derivaciones Anteriores</i> ”	30
Figura 26 “ <i>Filtro de Mediana en Matlab</i> ”	36
Figura 27 “ <i>Densidad Espectral</i> ”	37
Figura 28 “ <i>Espectrograma EEG</i> ”	37
Figura 29 “ <i>Matriz de Características</i> ”	40
Figura 30 “ <i>Análisis Exploratorio de Clasificación</i> ”	42
Figura 31 “ <i>Resultados con el Modelo 1.22 Bagged Trees</i> ”	43
Figura 32 “ <i>Resultado en la Decodificación de Señales EEG</i> ”	44
Figura 33 “ <i>Mano Robótica</i> ”	45
Figura 34 “ <i>Placa Arduino Uno</i> ”	46
Figura 35 “ <i>Modulo PCA9685</i> ”	47
Figura 36 “ <i>Diseño del Circuito</i> ”	50
Figura 37 “ <i>Implementación con los Componentes</i> ”	52
Figura 38 “ <i>Correlación del Circuito – Diseño e Implementación</i> ”	52
Figura 39 “ <i>Línea de Tiempo del Funcionamiento del Circuito EEG</i> ”	53
Figura 40 “ <i>Implementación final</i> ”	53
Figura 41 “ <i>Interfaz de MATLAB</i> ”	54

Lista de Apéndices

Apéndice A <i>Código General en MATLAB</i>	62
Apéndice B <i>Función del Código para Características</i>	68
Apéndice C <i>Código Normalización de la Matriz de Características</i>	74
Apéndice D <i>Código para el Control de la Mano Robótica a Partir de la Decodificación</i>	75
Apéndice E <i>Código de Arduino</i>	80
Apéndice F <i>Vídeo Sustentación Proyecto</i>	83

Introducción

En Colombia existe un registro significativo en la cantidad de personas en condición de discapacidad, una parte de esta población no cuenta con alguna de sus manos, lo que hace complejo su vida cotidiana en cuanto a brechas económicas, empleo, depresión y discriminación por su estado. En este contexto, es importante que se propongan proyectos tecnológicos enfocados a las necesidades de estas personas, especializados en diseños de prototipos de manos artificiales o robóticas; asimismo, vinculando señales de la parte motora del cerebro EEG, con el fin de unificar la ciencia con la ingeniería y basar sus estudios a estas personas. Es por esto que, este trabajo consiste en aportar una pequeña parte a un mundo tecnológico tan amplio, mostrando la importancia de usar y procesar las señales neuronales para el funcionamiento de un prototipo de mano robótica y que tenga la capacidad de funcionar como si fuese una mano real con tan solo pensarlo.

Es motivador profundizar en este tema de la biotecnología, ya que adquiriendo conocimientos de una parte de la medicina; señales EEG, actividad eléctrica del cerebro, comprensión de artefactos, entre otros aspectos, se puede lograr determinar la información relevante para el control de una robótica o prótesis, desarrollando un algoritmo de control adecuado, es allí donde entra la ingeniería electrónica; análisis y diseño de circuitos, programación, aprendizaje automático y robótica, que permitirían diseñar y mejorar el funcionamiento de prototipos de manos robóticas que funcionen con las señales del cerebro, lo que genera la curiosidad de analizar cómo adquirir esas señales fisiológicas de la parte motora del cerebro, para luego procesarlas, descifrarlas, predecir los movimientos, intentar aplicar el aprendizaje automático también conocido como “Machine Learning”, la programación que se requiere para que finalmente sea implementado a un circuito. En este sentido, sabiendo el

proceso que conlleva, podría aportar a futuros proyectos y a esas personas que requieran estos dispositivos, mejorando su condición de vida y más que la mano de un ser humano es importante para que pueda desarrollar sus actividades cotidianas.

Si bien es cierto, existen muchas maneras de aplicar el movimiento de una mano robótica, ya sea con sensores, señales musculares, programadas automáticamente, entre otros. En este proyecto se ha optado por un enfoque diferencial buscando la manera de simular las señales EEG y que estas se acerquen a la realidad. Afortunadamente existen bases o conjuntos de datos de señales fisiológicas del ser humano, en las cuales se pueden evidenciar estudios relacionados con lo que se quiere llegar. Sin embargo, se requiere de conocimientos biomédicos y electrónicos para comprenderlas y programas especializados para efectuar el procesamiento de dichas señales. Logrando esto, se puede acercar al movimiento de una mano robótica con las señales motoras y así identificar patrones para el aprendizaje automático con diversas señales que se le apliquen.

La intención de este proyecto es encontrar el paso a paso de la elaboración de un prototipo de mano robótica, analizando la manera de hacerlo, los materiales necesarios y adecuados a utilizar. En este sentido, para lograrlo, el trabajo cuenta con 6 fases: la primera consiste en la planificación y documentación, investigando y apoyándose con estudios ya realizados o que se acerquen a prototipos con base a señales EEG, la segunda es la simulación de señales EEG, la fase 3 radica en la identificación de características y clasificación de las señales cerebrales, la cuarta en la adquisición de componentes electrónicos, la quinta en el diseño del prototipo robótico y la fase 6 en la implementación del diseño y validación.

Se considera que este tipo de proyecto podría ser de interés para las industrias o clínicas que buscan soluciones para las personas en condición de discapacidad motora o quienes no tengan sus manos y requieran reemplazarla por una robótica. De esta misma manera, en la parte

personal y de la carrera me permitirá crecer de manera profesional adquiriendo conocimientos de ingeniería que logren aportar a la sociedad en futuros prototipos o ideas diferenciales. Por ello este trabajo requerirá de una investigación exhaustiva en el funcionamiento de la parte motora, la estimulación requerida, uso de electrodos; asimismo, la consulta a ingenieros electrónicos y biomédicos que permitan orientar la elaboración y diseño del prototipo.

Finalmente, se busca obtener resultados positivos en la elaboración del prototipo de mano robótica, en cuanto a su implementación y funcionamiento a través de la aplicación de señales EEG, que permita orientar a futuros trabajos, investigaciones o ideas asociadas en el aprovechamiento de las señales neuronales y de esta misma manera que se logre ayudar a las personas en condición de discapacidad motora o que no cuenten con sus manos.

Objetivos

Objetivo general

Diseñar un Prototipo de mano robótica mediante el uso de una interfaz cerebro-máquina (BCI) con un sistema de control de movimientos a través de señales EEG.

Objetivos específicos

Planificar y documentar el procedimiento de cómo desarrollar un prototipo de mano robótica y que su funcionamiento sea con base a señales neuronales.

Simular las señales del cerebro sin la necesidad de experimentar con humanos ni animales.

Seleccionar las variables y los requerimientos que se necesiten para su diseño y adquirir los componentes electrónicos y equipos.

Presentar el diseño del prototipo de mano robótica, el procesamiento y manejo de señales neuronales.

Implementar de manera física el diseño del prototipo y verificar su validación.

Definición del problema

Las personas en condición de discapacidad que presentan ausencia en extremidades como las manos y dificultades en la actividad motriz, algunas han sido discriminadas en actividades sociales y de empleo, sumado a esto problemas psicológicos y depresión debido a su estado y a las brechas de empleo que se limitan. Sin embargo, Colombia cuenta con diversas plataformas de oportunidades laborales para la contratación de personas en esta condición, a pesar de esto, podría existir temores para trabajar debido a la dificultad motriz, funcionalidad, comodidad, estética o no cuentan con una prótesis adecuada para el desarrollo de sus actividades; toda vez que, para acceder a un reemplazo artificial como una mano robótica son de difícil acceso por su costo.

Si bien, Colombia en el Plan de Desarrollo 2018 – 2022 (Desarrollo, 2018), trataba sobre políticas públicas para asumir los desafíos de las personas en condición de discapacidad “Pacto por la inclusión de todas las personas con discapacidad” (Desarrollo, 2018,p.156), mitigando la desventaja en términos de inclusión, desarrollando programas en educación, oferta social, productiva, garantía de derechos, transporte, entre otros. No obstante, carece de políticas enfocadas a la investigación y desarrollo tecnológico que permita nivelar su condición física como por ejemplo crear, subsidiar y proporcionar prótesis adecuadas. Un estudio que se incluyó en el plan de desarrollo.

(DANE, 2011), indica que el 38% se encuentran en condición de pobreza, 32% vulnerables, 27% clase media y el 2% en clase alta.

Según el (DANE, Discapacidad, 2010) Colombia cuenta con 2.018.078 personas en esta condición, de los cuales 928.451 son hombres y 1.089.627 mujeres. De esta población 413.269 presentan falencias en el movimiento del cuerpo, manos, brazos y piernas. Clasificando estos

datos se tiene que, 146.777 personas tienen problemas en llevar, mover y utilizar objetos con las manos. La ciudad de Bogotá registra el 24% con 476.991 personas en condición de discapacidad, de las cuales 35.017 personas presentan dificultad de llevar, mover, utilizar objetos con las manos. Es importante indicar que, la mayor población en esta condición se ubica en localidades de estrato 1, 2, 3 y 4; entre ellas, Usme (2.055), Tunjuelito (1.385), Bosa (3.516), Kennedy (4.128), Fontibón (1.362), Engativá (2.707), Suba (3.489), Mártires (964), Puente Aranda (1.770), Rafael Uribe Uribe (3.805), Ciudad Bolívar (2.935) y Sumapaz (62).

Un factor de atención que se ve reflejado es la empleabilidad de estas personas, teniendo en cuenta que según informa el Plan de Gobierno (Desarrollo, 2018) indica que 167.729 personas que se inscribieron se encontraban trabajando, de este mismo dato el 80% no tiene ningún tipo de contrato, el 11% tienen un contrato a término indefinido y un 7% referencia contar con un contrato a término fijo.

Por otra parte, a pesar de las iniciativas lideradas por diferentes estudiantes de universidades del país como la Universidad Nacional y algunas empresas como Prótesis Avanzadas S.A (Medellín) en diseñar e imprimir manos 3D ya sean cosméticas, mecánicas o mioeléctricas (electrodos que van ligados en un músculo de la persona), Colombia no cuenta con un buen reconocimiento ni apoyo por parte del Gobierno a las microempresas que se dedican a innovar en prótesis médicas modernas que han contado con las iniciativas tecnológicas y el personal adecuado, por lo que es muy mínima la producción y comercialización; además, haciendo difícil donar y subsidiar para las personas que las necesitan.

Un aspecto importante es que para el desarrollo de una buena prótesis se vinculan varios factores de implementación tecnológica, que a su vez hace complejo su funcionamiento y diseño, por ejemplo la capacidad de captar las señales motoras del cuerpo humano, ya sean musculares o

cerebrales y luego procesarlas, analizarlas, codificarlas e implementarlas en una interfaz cerebro máquina, para que finalmente sea funcional en una mano robótica; además, lo difícil de adaptarlos a los pacientes y que estos lo puedan asimilar, pero para ello surgen requisitos como la adquisición de permisos y consentimientos informados como la “Declaración de Helsinki”, decretada por la Asociación Médica Mundial, o el “Protocolo de Londres”, estas dos relacionan autorizaciones para investigaciones, principio éticos en la experimentación con seres humanos; asimismo, la Resolución 8430 de 1993 expedida por el Ministerio de Salud de Colombia “Por el cual se establecen las normas científicas, técnicas y administrativas para la investigación en salud”.

En conclusión, la tecnología avanza a pasos agigantados principalmente evidenciada en las comunicaciones, pero sería importante unificarlas con otras ciencias de la medicina, que permitan mejorar en prototipos de prótesis modernizados tecnológicamente y funcionales para estas personas en condición de discapacidad. Sin embargo, para iniciar algo así ¿Se podrá desarrollar un prototipo de prótesis robótica para una mano y que funcione con base a señales neuronales?, ¿Se podrán adquirir o simular las señales del cerebro sin la necesidad de experimentar con humanos y animales? ¿Cómo sería el procedimiento para desarrollar un prototipo de mano robótica y que su funcionamiento sea con base a señales neuronales?

Justificación

Ante las cifras del DANE que indica que Colombia cuenta con 2.018.078 personas en condición de discapacidad, es necesario enfocarse en el 7%; es decir, 146.777 de esta población, quienes presentan dificultades en llevar, mover y utilizar objetos con las manos. Sin embargo, es crucial reducir ese número, para ello se aterrizan con las cifras que tiene la ciudad de Bogotá, revelando en el estudio que se cuentan con 35.017 personas a las que se les dificulta usar las manos ya sea por amputaciones o discapacidad motora.

Con relación a las limitaciones para el desarrollo de sus actividades, que en algunos casos se convierten en desventajas relacionadas con la dificultad de coordinación, manipulación, presión y tacto, es importante buscar una solución tecnológica acompañada con la ingeniería y la medicina, que permitan unificar sus capacidades y desarrollar prototipos que les faciliten su condición de vida.

En este contexto, el presente proyecto busca analizar y encontrar una de las maneras de desarrollar un prototipo de mano robótica en el que su funcionamiento se basa con señales neuronales, investigando trabajos que se acercan a esta propuesta y profundizar en cómo se puede aprovechar las señales cerebrales para mover una mano robótica como si fuese una real con tan solo pensarlo.

Como no se va a experimentar con seres humanos ni animales, se hace necesario encontrar la manera de simular las señales eléctricas del cerebro mediante programas electrónicos, que permitan copiarlas y representarlas teniendo en cuenta las variables de amplitud, frecuencia y fase. Esto con el fin de desarrollar un prototipo funcional; además, de aportar insumos a futuros proyectos y mejorar en el diseño de una prótesis de manos robóticas,

considerando útil para las personas que se encuentren en la condición de discapacidad motora o por amputaciones en sus manos y así mejorar en algo su condición de vida.

Asimismo, se busca mostrar la importancia de enfocar la Ingeniería con otras ramas de la ciencia, para idear maneras o prototipos que sean accesibles a personas que no cuenten con suficientes recursos; ya que, a nivel nacional según en el Plan de Desarrollo 2018 – 2022, basado con un estudio del DANE indica que, el 38% de las personas en condición de discapacidad se encuentran en condición de pobreza, 32% vulnerables, 27% clase media y el 2% en clase alta, por lo que es complejo el acceso económico de prótesis avanzadas con un diseño más estético.

Desde el punto de vista disciplinar se va a trabajar con señales analógicas y digitales; además del análisis de circuitos, por lo tanto se pondrá en práctica las fundamentaciones teóricas y prácticas aprendidas en las asignaturas de electrónica análoga, electrónica digital, electrónica de potencia, señales y sistemas, microprocesadores y microcontroladores, programación, y procesamiento digital de señales, que permitirán buscar una solución adecuada en el desarrollo de la prótesis robótica controlada a través de señales neuronales.

Con este proyecto, permitiría a los demás estudiantes de universidades del país afianzar un mayor conocimiento sobre su elaboración y arriesgarse en desarrollar su intelecto en prototipos funcionales para estas personas en condición de discapacidad, que realmente requieren esta clase de aportes para sus vidas. Frente a esto, la propuesta tiene una utilidad metodológica para futuras investigaciones, sirviendo como insumo en el mejoramiento de prototipos alineados a esta idea.

Marco Referencial

En esta parte se presentan los conceptos, teorías y antecedentes alineados para el desarrollo del proyecto aplicado “Prototipo de mano robótica mediante el uso de una interfaz cerebro-máquina (BCI) con un sistema control de movimientos a través de señales EEG”.

Electroencefalograma

La electroencefalografía o también conocida de manera abreviada como EEG, es una técnica funcional para las clínicas y especialistas dedicados a esta rama de la neurociencia, es una prueba no invasiva, que les permite examinar y registrar las fluctuaciones eléctricas del cerebro. La actividad neuronal es captada por electrodos que se colocan en el cuero cabelludo.

Figura 1

“Encefalograma”



Nota. Ejemplo del uso de electrodos y las señales que genera. Tomada de (Promedic, s.f.)

Historia del Electroencefalograma

La EEG en humanos fue inventada por Hans Berger, jefe de la Unidad de Psiquiatría de la Universidad de Jena (Alemania) en 1924, quien registró las primeras oscilaciones rítmicas del cerebro (Sánchez, 2022), para su logro evidenció que obtenía mejores resultados con pacientes

que sufrían lesiones cerebrales; además, posicionando los electrodos en la frente y en la parte posterior de la cabeza.

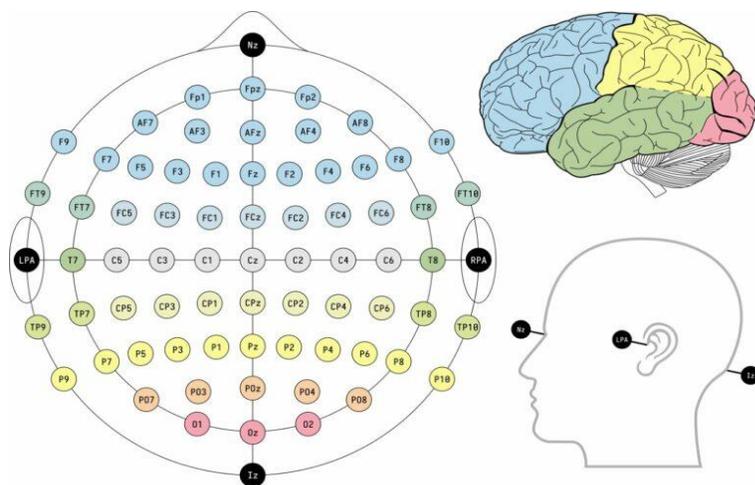
Un aspecto importante de mencionar en los informes realizados por Hans, fue que con el análisis de los diferentes registros EEG que realizó durante varios años (1.133 registros en 76 personas), estableció dos tipos de ondas que visibilizó en ese momento; “las de mayor voltaje y menor frecuencia fueron denominadas ondas alfa, las de menor voltaje y mayor frecuencia, ondas beta” (Sánchez, 2022,p.106).

Sistema Internacional 10 – 20

Para la realización de pruebas de encefalografía existe un sistema internacional 10 – 20, el cual estandariza la ubicación de los electrodos sobre el cuero cabelludo, con el fin de que sea comparable con otros estudios que se les hayan aplicado a diferentes personas en el transcurso del tiempo.

Figura 2

“Sistema 10 – 20 con Electroencefalografía”



Nota. La imagen muestra la posición de electrodos empleada en electroencefalografía (EEG).

Tomada de (Gelimed)

Teniendo en cuenta el Manual básico para enfermeros en electroencefalografía, indica que el sistema 10 – 20 se llama de esta forma, debido a que “los electrodos están espaciados entre el 10% y el 20% de la distancia total entre puntos reconocibles del cráneo” (García, 2011,p.29)

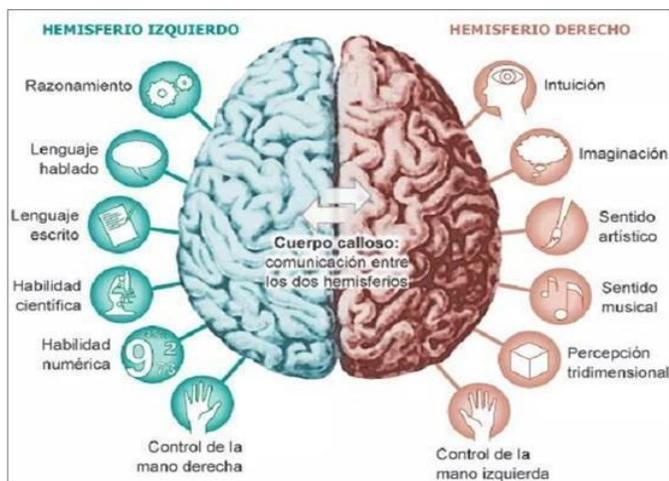
Como se observa en la figura 2, se ubican letras que permiten identificar partes de la corteza cerebral y números para identificar el hemisferio (se constituye en dos mitades izquierda y derecha del cerebro); es decir, con el electrodo en el sitio F se posicionará en el lóbulo frontal, T en el temporal, C en el central, P en el parietal y O en el occipital.

Anatomía Cerebral

Es importante conocer una de las partes más esenciales del ser humano, siendo el cerebro encargado de emitir una de las señales fisiológicas al cuerpo, para la ejecución de acciones del sistema nervioso, respiratorio, parte motora, entre muchas más, que pueden ser de utilidad para conocer su funcionamiento, teniendo en cuenta sus hemisferios y lóbulos cerebrales funcionales para la percepción, conservación, análisis y control.

Hemisferios Cerebrales

El cerebro humano está compuesto por dos hemisferios (derecho e izquierdo), como se aprecia en la figura 3, el izquierdo tiene funciones relacionadas con la lógica, tales como; razonamiento, lenguaje hablado, lenguaje escrito, habilidad científica, habilidad numérica y control de la mano derecha. En cuanto al derecho se ubican habilidades creativas, hallándose la intuición, imaginación, sentido artístico, sentido musical, percepción tridimensional y el control de la mano izquierda (López, 2015)

Figura 3*“Hemisferios Cerebrales”*

Nota. Muestra la composición de los 2 hemisferios del cerebro y sus características. Tomada de (López, 2015).

Lóbulos del Cerebro

La parte del cerebro humano se divide en 4 partes llamadas “lóbulos” (Blog de ITZEL, 2011), guiándose por la Figura 4:

Lóbulo parietal: encargado de procesar la sensibilidad con el tacto (gusto y temperatura); además, se ubica la corteza motora (controlar el movimiento).

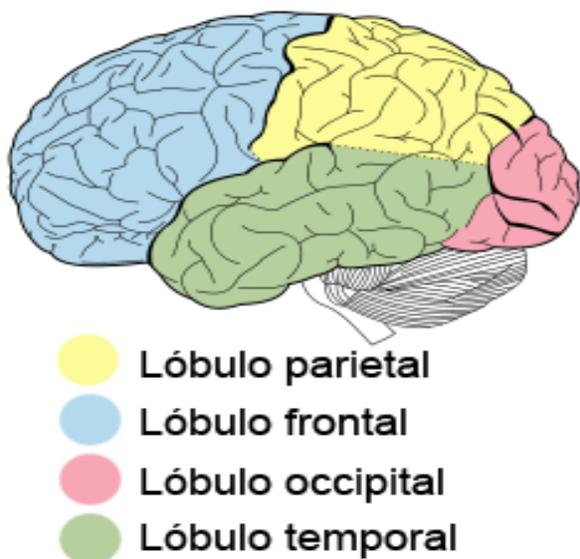
Lóbulo frontal: genera procesos de pensamiento (planificación, imaginación y pensamiento).

Lóbulo occipital: aloja la corteza visual (procesa lo que visualiza el ser humano).

Lóbulo temporal: se ubica cerca de los oídos, por lo que se encarga de procesar la audición; además, hace parte de los recuerdos.

Figura 4

“Funciones del Cerebro”



Nota. Ubicación de los lóbulos del cerebro. Tomada de (Blog de ITZEL, 2011).

Parte Motora del Cerebro

Para que una persona pueda realizar movimientos en diferentes partes del cuerpo, estas tienen que ir integradas a la parte motora del cerebro, las cuales se dividen en cinco áreas: i. **La corteza motora primaria**, que genera impulsos nerviosos, envía órdenes a los músculos haciéndolos que se contraigan o se tensen ii. **Área motora suplementaria**, coordina los movimientos de posturas, iii. **Área de premotoras**, almacenan movimientos de experiencias y coordina movimientos de la actividad motora primaria, iv. **Área motora del lenguaje de Broca**, produce el lenguaje y movimientos musculares y v. **El área de la corteza parietal posterior** transforma la información visual y demás sentidos en movimientos (Arbelaéz, 2020).

Frente a lo anterior, cuando se afecta la parte motora, ya sea por una lesión, puede causar graves consecuencias; tales como, parálisis (pérdida total o parcial del movimiento de alguna

parte del cuerpo), apraxias (no se pueden realizar movimientos por falta de control), disartrias (alteración en el habla), agrafia (dificultad al escribir y lógica) y afasia de broca (alteración del lenguaje expresivo) (Arbelaéz, 2020).

Ondas o Ritmos Cerebrales

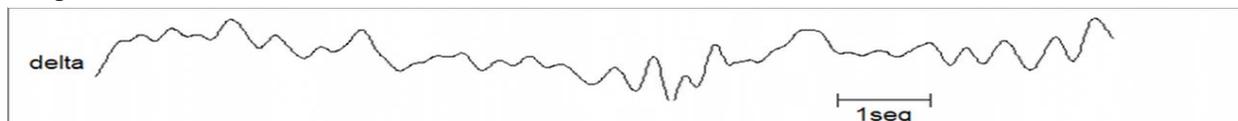
El cerebro contiene cinco tipos de ondas diferentes, las cuales divergen en su banda de frecuencia y amplitud. Según (Sabater, 2022) indica que estas se comportan como notas musicales conformando una sintonía armónica. Es importante indicar que, sus frecuencias oscilan desde los 0.5 Hz hasta los 100 Hz.

Ondas Delta

Este tipo de onda oscila entre los 0.5Hz-3Hz, caracterizándose con tener la menor frecuencia y mayor amplitud (20-200 uV), se visualizan cuando existe un sueño profundo (sin sueños). (Sabater, 2022) explica que estas señales se observan más en los bebés y niños de una menor edad, ya que en los mayores tienden a producir menos esta señal a través de los años.

Figura 5

“Representación Ritmo Delta”



Nota. Representación de una onda Delta. Tomada de (Farfan, 2005)

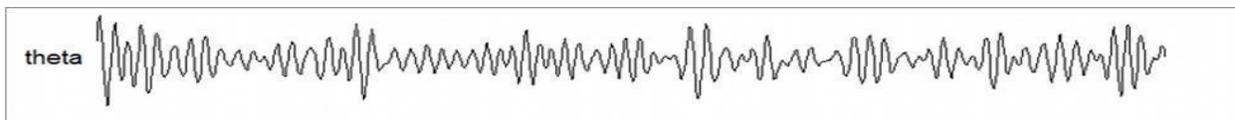
Ondas Theta

Maneja una frecuencia entre los 3.5Hz-8Hz, su amplitud se ubica desde los 20-100 uV, estas señales se relacionan con las habilidades imaginativas. (Sabater, 2022) indica que las ondas Theta muestran mayor actividad al momento de expresar emociones profundas (picos elevados:

trastorno depresivo o falta de atención, picos bajos: ansiedad y estrés y picos adecuados: favorece la creatividad y la intuición).

Figura 6

“Representación Ritmo Theta”



Nota. Representación de una onda Theta. Tomada de (Farfan, 2005)

Ondas Alfa

Su frecuencia oscila entre 8Hz-13Hz, en este tipo de señal se reduce la amplitud, fluctuando entre los (20-60 uV), este ritmo cerebral se visibiliza donde la persona se encuentra en calma, pero no con sueño; es decir, en modo de relajación (nivel elevado: impedirá centrar la concentración y nivel bajo: forja ansiedad, estrés e insomnio) (Sabater, 2022).

Figura 7

“Representación Ritmo Alfa”



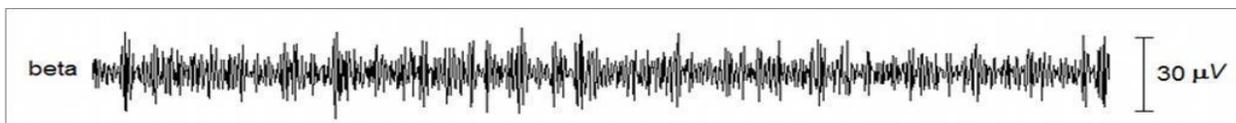
Nota. Representación de una onda Alfa. Tomada de (Farfan, 2005)

Ondas Beta

Como se aprecia en la imagen, las ondas Beta cuentan con una mayor frecuencia oscilando desde los 12Hz-33Hz; además, se reduce su amplitud entre los 2-20 uV, son activadas cuando el ser humano realiza actividades de su vida cotidiana (reuniones, conducir, exámenes, entre otros) (Sabater, 2022).

Figura 8

“Representación Ritmo Beta”



Nota. Representación de una onda Beta. Tomada de (Farfan, 2005)

Ondas Gama

Esta señal Gamma maneja una frecuencia entre los 30Hz-100 Hz. Según (Sabater, 2022) indica que los neurocientíficos están empezando a captar más datos y se hace complejo captarla en los electroencefalogramas, se conoce que la señal se desplaza desde la parte posterior del cerebro hacia adelante. Se relacionan con la capacidad de aprendizaje y de almacenar información nueva; es decir, tareas de alto procesamiento cognitivo.

Figura 9

“Representación Ritmo Gama”



Nota. Representación de una onda Gama. Tomada de (Farfan, 2005)

Artefactos - Problemas en el Análisis EEG

Se generan dificultades al momento de analizar las señales EEG adquiridas, teniendo en cuenta diversos factores que inciden, alteran y las distorsionan, son conocidas como artefactos (señales indeseadas) y se dividen en tres; problemas en los electrodos (ubicación inapropiada, conexiones desgastadas y falta de gel), interferencia fisiológica (acción muscular, movimiento de la lengua, respiración, pulso, escalofríos e impedancia del cuero cabelludo) y interferencia

electromagnética (dispositivos eléctricos cercanos, interferencia típica a 50 - 60 Hz de la red de alimentación, marcapasos, entre otras).

Adquisición de Señales - Banco de Señales Fisiológicas

Existen diversos bancos de señales fisiológicas que permiten explorar y descargar señales biomédicas como el electroencefalograma (EEG), electrocardiograma (ECG), entre otros, permitiendo acceder y conocer en detalle las muestras tomadas a diversos pacientes por parte de comunidades de investigadores, logrando observar su reacción por medio de ondas. Su descarga puede ser convertida en archivos de señal para ser trabajados en diferentes aplicaciones de programación y procesamiento de señales. Entre los bancos fisiológicos se destaca Physionet y la base de datos de IEEE.

Banco de Señales Physionet

De acuerdo con (Physionet, s.f.), este banco de señales se fundó formalmente en el año 1999 como “Componente de divulgación del recurso de investigación para señales fisiológicas complejas”, el cual está compuesto por profesionales de diferentes disciplinas; científicos informáticos, físicos, matemáticos, especialistas biomédicos, otros. Los recursos que ofrecen son las grabaciones digitales de señales fisiológicas (cardiopulmonares, neuronales y otras señales biomédicas) con la participación de pacientes sanos y sujetos con problemas de salud, incluso incluyen muerte cardíaca súbita, epilepsia, trastornos, envejecimiento, etc. Para visualizarlas se compone de una caja de herramientas para su observación, descritos así: trazar forma de onda, trazar.

Figura 10*“Cajero Automático PhysioBank”*

CAJERO AUTOMÁTICO PHYSIOBANK

Entrada

Base de datos:

Grabar:

Señales:

Anotaciones:

Salida

Longitud: 10 segundos 1 minuto 1 hora 12 horas para terminar

Formato de tiempo:
 hora Fecha tiempo transcurrido horas minutos segundos muestras

Formato de datos: estándar alta precisión unidades de ADC sin procesar

Caja de herramientas

Navegación

Nota. Interfaz banco de señales de Physionet. Tomada de (Physionet, s.f.)

Banco de señales IEEE

IEEE Advancing Technology for Humanity “La organización profesional técnica más grande del mundo para el avance de la tecnología”, cuenta con una variedad de avances de innovación y bases de datos que permiten explorar y analizar, tales como; inteligencia artificial, astronomía, ciencias biomédicas y de salud, señales fisiológicas, comunicaciones, computación, entre muchas más. En este sentido, es útil para la adquisición de bases de datos EEG confiables para desarrollar el presente proyecto.

Interfaz Cerebro Máquina

Se describe comúnmente como Brain - Computer Interface (BCI) por sus siglas en inglés, su función es traducir las intenciones que tiene el humano para interactuar en el mundo real o virtual. Es allí donde entra una BCI, la cual mide la actividad cerebral, procesa los datos y remite

unas características, permitiéndole interactuar en un entorno específico con base a lo que desea la persona (hombre máquina) - (Minguez).

Partes Importantes de un BCI

Para que una BCI pueda funcionar manteniendo una comunicación natural con el humano y permita acceder a la información se requiere de tres aspectos importantes (Minguez), así:

Sensores

Para un mayor acceso en la parte cerebral del ser humano, se necesita de elementos clínicos y de instrumentación como electrodos al momento de la electroencefalografía, esto con el objetivo de captar la actividad cerebral.

Procesamiento de señales

Es fundamental decodificar las señales neuronales importantes que infieren en la intención del usuario, por lo cual, requerirá de filtros después de recoger la actividad cerebral.

Aplicación

Se inicia con la interacción con el entorno aplicando la BCI, permitiendo mover dispositivos electrónicos (silla de ruedas, manos robóticas, control de ordenadores, entre otros).

Mano Robótica

En el mundo actual existen diversos prototipos de manos robóticas que son empleadas para diferentes necesidades del ser humano, que incluso son modificadas por prótesis biomédicas, con el fin de que las personas puedan reemplazar su mano por una robótica, teniendo en cuenta anomalías o amputaciones.

Por lo general, las manos robóticas utilizadas por estudiantes para efectuar diversos experimentos se caracterizan por contar con cinco servomotores efectuando movimientos a través de hilos de nailon que van conectados a cada uno de los cinco dedos de una mano.

Figura 11

“Mano Robótica”



Nota. Representación mano robótica. Tomada de (Alvarez, s.f.)

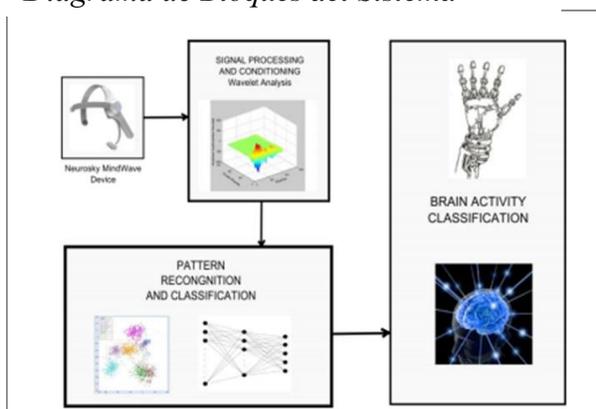
Antecedentes de Prototipos con EEG***Antecedente 1***

Existe una investigación por parte de estudiantes de la Universidad de Cuenca (Ecuador), titulada “Adquisición de señales cerebrales utilizando el dispositivo MindWave” (Felipe Torres - Christian Sánchez - Kenneth Palacio, 2014) que fue desarrollado por el fabricante Neurosky, que permite verificar la actividad cerebral de manera no invasiva con la aplicación de estímulos, utilizando una diadema en la cabeza que contiene electrodos; además, es inalámbrica y puede mostrar las señales a un computador. Para su procesamiento propusieron el uso de la transformada Wavelet para el acondicionamiento de señales y que puede ser desarrollada a través de Matlab. Dentro de los experimentos adquirieron señales cerebrales del movimiento de las muñecas de las manos, analizando que son diferentes de la mano izquierda y la derecha; de la misma manera, experimentaron con varios sujetos voluntarios y concluyeron que no existe una

correlación en las señales, son diferentes para cada uno. Por otra parte, el proyecto da a conocer las 5 señales que se pueden conseguir a través de un electroencefalograma como la señal delta (oscilan entre 0,5 a 4 Hz), theta (de 5 y 7,5 Hz), Alfa (8 a 13 Hz), Beta (14 y 26 Hz) y Gamma (mayores a 30 Hz).

Figura 12

“Diagrama de Bloques del Sistema”



Nota. Se visualiza los bloques que se requiere para el movimiento de una mano robótica.

Tomada de (Felipe Torres - Christian Sánchez - Kenneth Palacio, 2014)

Antecedente 2

Un trabajo de investigación de un estudiante de la Escuela Superior Politécnica de Chimborazo de Ecuador, titulado “Implementación de un prototipo de prótesis de mano usando el método BCI no invasivo a través de estímulos endógenos” (Torres, 2018), menciona el término interfaz cerebro computador BCI, el cual permite interpretar las señales para futuros controles de ordenadores externos como “neuro prótesis robóticas”, mostrando 5 etapas de esta interfaz, así:

Etapa 1. Corresponde a la adquisición de señales del cerebro y se realiza la reducción del ruido.

Etapa 2. Preprocesamiento y realce de la señal que consiste en preparar y acondicionar las señales para el procesamiento.

Etapa 3. Extracción de características específicas de las señales cerebrales y que se graban para su estudio.

Etapa 4. Clasificación de las señales aplicando un algoritmo.

Etapa 5. Control de interfaz que después de un filtro los traducen en comandos para ser interpretados por un ordenador o dispositivo electrónico.

En este trabajo utiliza una diadema BCI desarrollada por la empresa Emotiv Systems empleada para la adquisición de señales del cerebro de manera no invasiva, la cual utiliza 5 electrodos con una capacidad de 128 muestras por segundo, una frecuencia de 1 a 43 Hz y una conectividad por Bluetooth, posterior a eso da a conocer los materiales y el proceso de elaboración de una mano robótica a través de una impresión 3D, profundizando en las dimensiones de la mano humana y tipos de movimientos de los dedos. Concluye en su trabajo que se consiguió la apertura y cierre de la mano, siendo controlada por un gesto facial por medio de la diadema anteriormente mencionada.

Antecedente 3

Otro trabajo de investigación de la Universidad La Salle de la Ciudad de México titulado “Decodificación de imaginación motora en la señal de electroencefalograma mediante mapas auto organizados” (Guittierez, 2015), usaron pacientes que fueron sentados en un sillón, quienes imaginaron el movimiento de pinza gruesa de la mano derecha e izquierda. Para la adquisición de las señales colocaron 11 electrodos y registrándose con un amplificador a una frecuencia de muestreo de 256 Hz, dando a conocer que los movimientos de las manos son registrados en la corteza sensorial motora que se ubica en la zona central de la corteza cerebral, indicaron que para

minimizar las interferencias usaron electrodos en las orejas; además, colocaron dos canales de electromiografía en los brazos de los sujetos de prueba, con el fin de comprobar que no se realizaba un movimiento real. Asimismo, informaron la importancia de aplicar filtros pasa-banda tipo Butterworth para un rango de frecuencia de 8 a 30 Hz.

Antecedente 4

Un trabajo de investigación de estudiantes de la Universidad del Cauca – Colombia, desarrollaron una “Prótesis de mano virtual movida por señales encefalográficas EEG” (Karin Correa Arana, 2016), adquiriendo las señales a través de un casco de tecnología Emotiv, el cual cuenta con 14 electrodos no invasivos, empleado para capturar las señales emitidas por una persona en su parte motora, para luego ser utilizado en una prótesis virtual capaz de realizar los agarres tipo pinza, cilíndrico y esférico; además, de cerrar y abrirse completamente. Todo esto a través del pensamiento de la persona denominado “intención de movimiento”, esto es logrado por un entrenamiento el cual se muestra e imagina imágenes relacionadas con los movimientos de las manos, lo que permite guardar patrones de señales relacionados a las acciones de control para la manipulación de la prótesis.

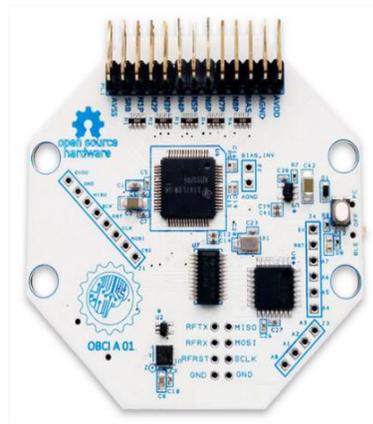
Antecedente 5

Una comunidad de investigadores quienes cuentan con una página “Interfaces cerebro-computadora de código abierto” (OpenBCI, s.f.), la cual, tienen componentes, guías y tutoriales que permiten explicar el cómo usar el equipo OpenBCI con el dispositivo Cyton Board, que es una interfaz neuronal de 8 canales compatibles con Arduino y puede ser usado para tomar muestras de la actividad cerebral (EEG), muscular (EMG) y cardíaca (ECG) en tiempo real. Para el uso de electrodos explican la configuración para procesar datos EEG y transmitirlos desde la placa a un ordenador. Este grupo de investigación ha desarrollado proyectos de cómo controlar

cosas electrónicas usando las señales neuronales, entre ellos movimiento de brazo robótico, control de juegos, computadores, entre otros, basándose en herramientas de biodetección.

Figura 13

“Interfaz BCI Cyton Board”



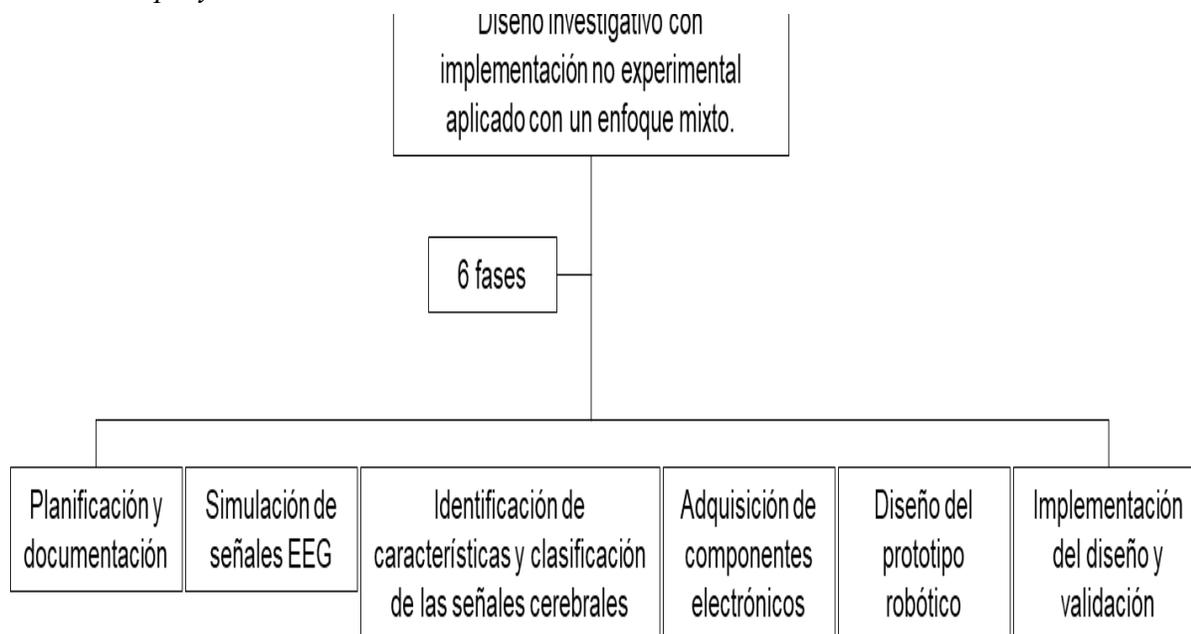
Nota. Interfaz BCI Cyton Board, con su sistema de control. Tomada de (OpenBCI, s.f.)

Metodología

Para el desarrollo de los objetivos propuestos se formuló una metodología integrada por seis fases:

Figura 14

“Fases del proyecto”



Nota. Se evidencia la metodología aplicada mediante fases. Tomada de (Imagen propia)

Fase 1: Etapa de Planificación y Análisis de Datos

Inicialmente, fue necesario encontrar un banco de señales fisiológicas adecuado, reconocido y confiable para este proyecto. Por lo cual, fue necesario contactar al ingeniero biomédico Juan Sebastián Malagón colombiano de la Universidad El Bosque, quien recomendó usar la base de datos IEEE Advancing Technology for Humanity.

En este sentido, se usa una base de datos de la página IEEE Data Port (Puerto de datos IEEE) en la que se ubica un conjunto de datos de señales cerebrales y señales de movimiento de la mano correspondiente (BS-HMS-DATASET).

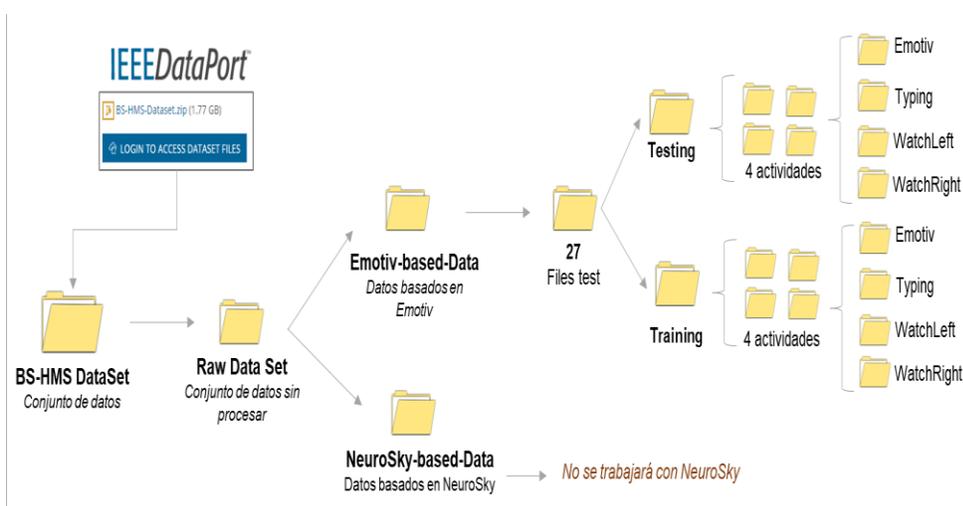
Figura 15

“Interfaz Banco de Señales Fisiológicas”

Nota. Banco de señales EEG de IEEE Data Port. Tomada de (IEEE DataPort, s.f.)

Figura 16

“Análisis de Datos en el Banco de Señales IEEE”



Nota. Se observa cómo está organizado el DATASET original sin ningún procesamiento.

Tomada de (imagen propia)

La DATASET proporciona dos subconjuntos de información; uno basado en NeuroSky con 32 participantes y el otro en Emotiv con 27 voluntarios. En la figura 14 se visualiza, como se encuentran organizadas las carpetas, profundizando en Emotiv-based-Dataset (datos a trabajar en el proyecto), la cual se componen por una subcarpeta para cada usuario con el nombre “Testing” (pruebas), cada “Test User” (usuario de prueba) cuenta con 2 carpetas de sesiones una llamada “Training” (entrenamiento) y la otra para datos “Testing” (pruebas), cada una de estas dos sesiones cuenta con cuatro carpetas (actividad 1, 2, 3 y 4) y cada una de estas cuentan con 3 subcarpetas (Emotive EEG Data y Smartwatch).

Es importante indicar que, las actividades 1, 2, 3, y 4 de cada uno de los participantes, se caracterizan de acuerdo con lo siguiente (IEEE DataPort, s.f.):

Actividad 1: miran un video de emoción feliz.

Actividad 2: escriben respuestas de preguntas dadas en la actividad 1.

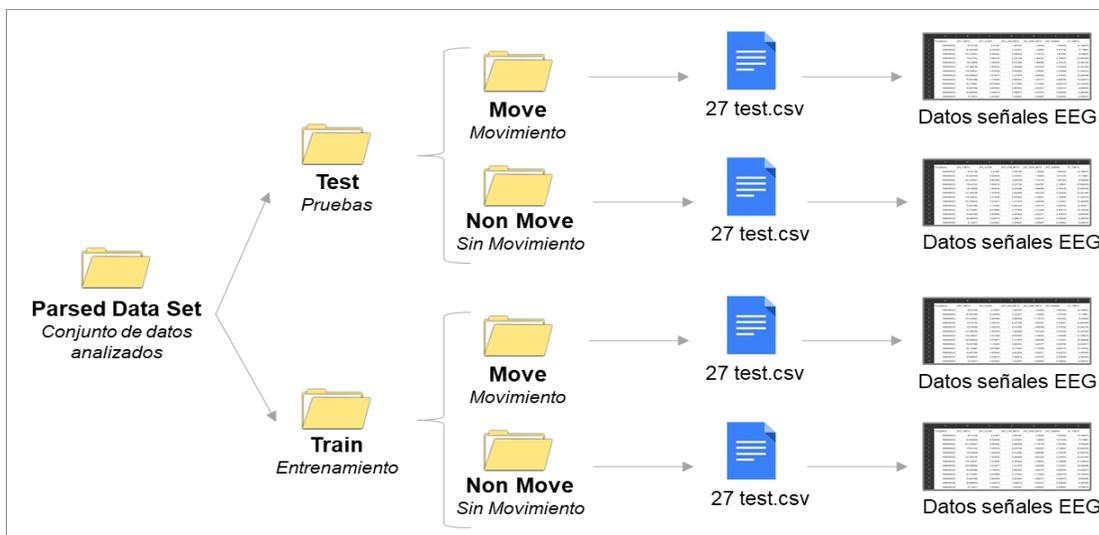
Actividad 3: miran un video de emoción triste.

Actividad 4: escriben respuestas de preguntas dadas en la actividad

Organización de datos basados en Emotiv-based-Dataset

De acuerdo con el Ingeniero Biomédico Juan Sebastián Malagón, recomendó la necesidad de organizar los datos de Emotiv en movimiento y no movimiento. En este sentido, se realizó un algoritmo en Python con la herramienta Google Colab, con el fin de hacer más ágil la organización, teniendo en cuenta la cantidad de datos de IEEE.

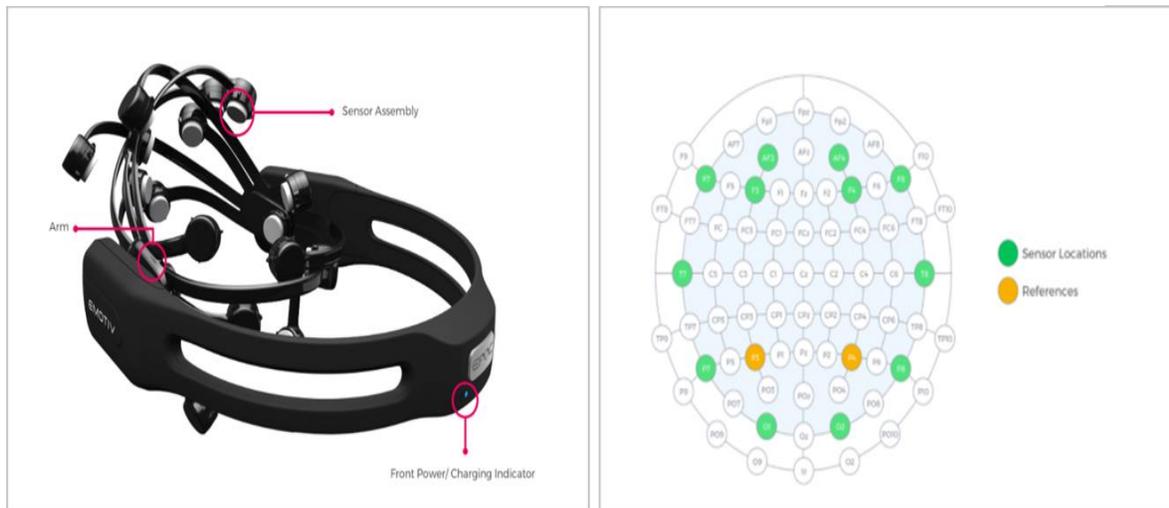
El algoritmo permitió tratar el conjunto de datos, concatenando las carpetas de los 27 usuario de pruebas, unificando las actividades; 2 con 4 (movimiento) y las actividades 1 con 3 (sin movimiento), quedando en cada una de ellas las 27 pruebas como se observa en la figura 17.

Figura 17*"Reorganización de Carpetas de la Data"*

Nota. Se ubican las carpetas de las que se van a analizar los datos tanto de pruebas como de entrenamiento del DATASET. Tomada de (imagen propia)

Análisis de Datos Enfocado a Emotiv

Gracias al conjunto de datos de IEEE Data Port, se puede tener acceso a las señales EEG, específicamente relacionadas al movimiento de las manos con actividades determinadas. Para este proyecto, se usa el conjunto de datos de Emotiv, el cual experimentan con un neuro auricular llamada EMOTIV EPOC +, que cuenta con 14 canales de detección cerebral, electrodos a base de solución salina (sensores húmedos, sin el empleo de geles), conexión inalámbrica y sensores de movimiento de ejes (detectar movimiento de la cabeza).

Figura 18*“Diadema EMOTIV EPOC+”*

Nota. Diadema EMOTIV EPOC + con algunas características. Tomada de (Emotiv, s.f.)

En la figura se puede apreciar la diadema Emotiv, la cual cuenta con (2) brazos con electrodos, diseñado para proporcionar cobertura de los lóbulos frontal, prefrontal, temporal, parietal y occipital con base al sistema internacional 10-20.

Tabla 1*Especificaciones Técnicas del EMOTIV EPOC +*

EMOTIV EPOC +	Especificaciones Técnicas
Número de canales	14 (más referencias CMS – electrodo activo que es colocado en el centro de los electrodos de medición, DRL – electrodo pasivo colocado lejos de los electrodos de medición).
Nombres de canales (10 – 20 ubicaciones internacionales)	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AFA
Método de muestreo	Muestreo secuencial ADC único
Tasa de muestreo	128 muestras por segundo (SPS)
Banda ancha	0.2 – 45 Hz, filtros de muesca digitales a 50Hz y 60 Hz
Filtración	Filtro Sinc digital integrado de quinto orden
Conectividad	Inalámbrico patentado de 2.4 Ghz
Muestreo de Movimiento	128 Hz

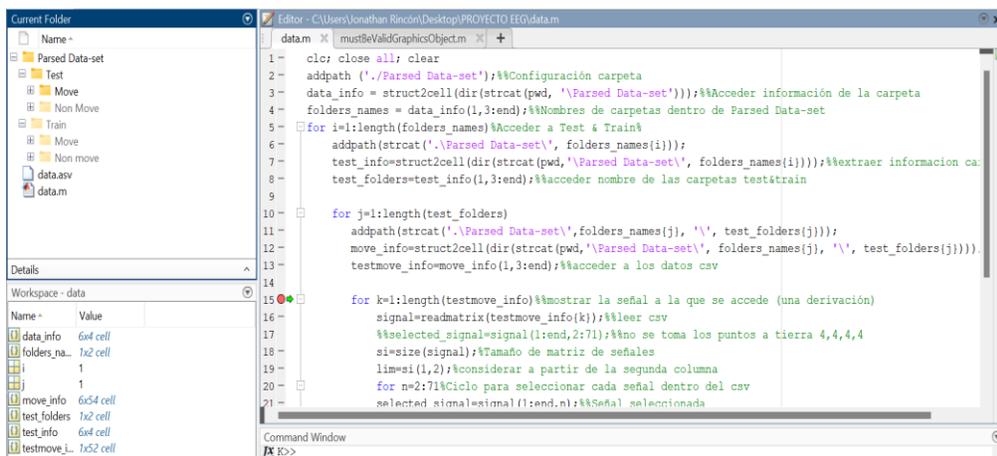
Nota. Se visualiza las características del EMOTIV. Fuente: (Emotiv, s.f.)

Verificación Formato De Datos CSV

La matriz de formato csv cuenta con 3182 datos en filas, correspondientes al tiempo de muestras por segundo y 85 posiciones en columnas, donde aparecen los tipos de ritmos cerebrales o derivaciones Theta, Alpha, Beta y Gamma.

Figura 19

“Cargue de Datos en MATLAB”



```

1- clc; close all; clear
2- addpath ('./Parsed Data-set'); %%Configuración carpeta
3- data_info = struct2cell(dir(strcat(pwd, '\Parsed Data-set'))); %%Acceder información de la carpeta
4- folders_names = data_info(1,3:end); %%Nombres de carpetas dentro de Parsed Data-set
5- for i=1:length(folders_names) %%Acceder a Test & Train
6-     addpath(strcat('\Parsed Data-set\', folders_names{i}));
7-     test_info=struct2cell(dir(strcat(pwd, '\Parsed Data-set\', folders_names{i}))); %%extraer informacion ca
8-     test_folders=test_info(1,3:end); %%acceder nombre de las carpetas test&train
9-
10- for j=1:length(test_folders)
11-     addpath(strcat('\Parsed Data-set\', folders_names{j}, '\', test_folders{j}));
12-     move_info=struct2cell(dir(strcat(pwd, '\Parsed Data-set\', folders_names{j}, '\', test_folders{j})));
13-     testmove_info=move_info(1,3:end); %%acceder a los datos csv
14-
15-     for k=1:length(testmove_info) %%mostrar la señal a la que se accede (una derivación)
16-         signal=readmatrix(testmove_info{k}); %%leer csv
17-         %%selected_signal=signal(1:end,2:71); %%no se toma los puntos a tierra 4,4,4,4
18-         si=size(signal); %%Tamaño de matriz de señales
19-         lim=si(1,2); %%considerar a partir de la segunda columna
20-         for n=2:71 %%Ciclo para seleccionar cada señal dentro del csv
21-             selected_signal=signal(1:end,n); %%Señal seleccionada

```

Nota. cargue de datos Emotiv en MATLAB. Tomada de (imagen propia)

Visualización Inicial de Datos

Para visualizar las señales, se toma como referente la frecuencia de muestreo $F = 128 \text{ SMP}$, la longitud de la señal y el vector de tiempo $T_i = \text{Señal} * 1\left(\frac{1}{FS}\right)$ periodo de muestro.

$1\left(\frac{1}{FS}\right)$ periodo de muestro.

Figura 20

“Registro de Datos de un Participante”

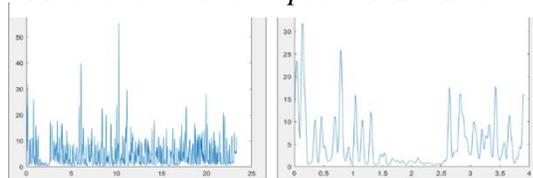
Untitled															
TimeStamp	AF3_THETA	AF3_ALPHA	AF3_LOW_BETA	AF3_HIGH_BETA	AF3_GAMMA	F7_THETA	F7_ALPHA	F7_LOW_BETA	F7_HIGH_BETA	F7_GAMMA	F3_THETA	F3_ALPHA	F3_LOW_BETA	F3_HIGH_BETA	F3_GAMMA
Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
156659002	63.411580	5.519470	1.453187	1.295420	1.092443	14.188973	1.540523	0.414399	0.359444	0.304057	1.525679	1.419775	0.425131	0.622268	0.645459
156659002	82.802448	6.534928	2.325301	1.498290	1.472126	17.786610	1.790340	0.531886	0.456768	0.396640	8.440625	1.620597	0.515312	0.774069	0.985775
156659002	101.225941	6.981862	3.680508	1.719115	1.851552	20.698260	1.861838	0.733954	0.586901	0.496978	11.119743	1.674737	0.688539	0.951810	1.363871
156659002	116.527420	7.054016	5.337738	1.904767	2.129831	22.684059	1.778427	0.990514	0.717121	0.586464	12.375511	1.589467	0.942845	1.102186	1.671571
156659002	125.980800	7.094434	6.913386	1.988366	2.216235	23.606726	1.631126	1.235719	0.805369	0.643215	12.885278	1.416362	1.235700	1.169565	1.805934
156659002	127.046188	7.303918	7.944969	1.932225	2.074024	23.261464	1.505540	1.390826	0.822992	0.651232	12.782417	1.216143	1.486382	1.122927	1.717177
156659002	118.338541	7.574206	8.093892	1.748481	1.748086	21.538233	1.439873	1.402429	0.768235	0.607104	11.690863	1.027534	1.609098	0.972089	1.435624
156659002	100.589502	7.619377	7.311879	1.495594	1.313541	18.469888	1.422721	1.267432	0.667169	0.522579	8.829220	0.866842	1.556321	0.765675	1.052662
156659002	76.947986	7.176345	5.864542	1.263177	0.906764	14.402817	1.395671	1.038625	0.560628	0.419790	7.471346	0.731603	1.345427	0.570896	0.683649
156659002	52.733661	6.075898	4.177404	1.112829	0.604175	10.144354	1.297153	0.790728	0.481592	0.322300	5.131168	0.609224	1.043180	0.444248	0.411990
156659002	34.687596	4.693954	2.681604	1.052011	0.430374	6.950089	1.154809	0.592182	0.441296	0.246593	3.509961	0.517243	0.735502	0.404018	0.262350
156659002	28.886503	3.538219	1.696613	1.042437	0.358268	6.286385	1.043689	0.492906	0.431720	0.198464	3.385166	0.493756	0.499314	0.428426	0.206419
156659002	37.256710	3.033922	1.329287	1.028667	0.344842	8.936019	1.035505	0.500731	0.436238	0.175180	5.081743	0.566992	0.375398	0.477144	0.197374
156659002	56.955698	3.256362	1.481687	0.975542	0.368295	14.608272	1.150719	0.590208	0.441433	0.171614	8.383715	0.725776	0.356853	0.515695	0.206739
156659002	81.320128	3.913347	1.921185	0.878502	0.423384	21.802682	1.343399	0.711130	0.439543	0.182833	12.416869	0.920178	0.402181	0.524751	0.228132
156659002	101.537686	4.561360	2.383899	0.747851	0.497781	28.185425	1.527278	0.809911	0.424444	0.204217	15.926426	1.091535	0.458893	0.495339	0.255975
156659002	111.328842	5.056222	2.724457	0.618474	0.578888	31.725275	1.666009	0.859749	0.396483	0.234623	17.796079	1.223025	0.503186	0.433105	0.283749
156659002	109.519113	5.571917	2.934358	0.527693	0.660393	31.745673	1.774377	0.858660	0.360997	0.276648	17.656503	1.343461	0.543642	0.354414	0.307771
156659002	99.329293	6.279622	3.037817	0.497994	0.737947	28.777262	1.875171	0.814682	0.324309	0.332064	15.746683	1.480618	0.588882	0.279655	0.328043
156659002	86.975781	7.169825	3.056100	0.529131	0.805098	24.545293	1.982399	0.739157	0.293890	0.397736	13.075851	1.634035	0.635733	0.228216	0.346524
156659002	77.082974	7.976080	2.990451	0.598598	0.851009	20.589687	2.081615	0.648671	0.277039	0.462980	10.509197	1.768593	0.675696	0.211643	0.363843

Nota. Datos del participante 1-2 – etapa de prueba señal de movimiento MATLAB. Tomada de (imagen propia)

De acuerdo con los datos csv, se tomó en cuenta al participante 1-2 en la etapa de prueba con señal de movimiento, tomando como referente para su visualización las derivaciones THETA, ALPHA, BETA y GAMA del electrodo ubicado en la posición F7 del sistema 10 – 20, teniendo en cuenta que se ubica en el lóbulo frontal, una zona de movimiento o motora, visualizando los siguientes registros:

Figura 21

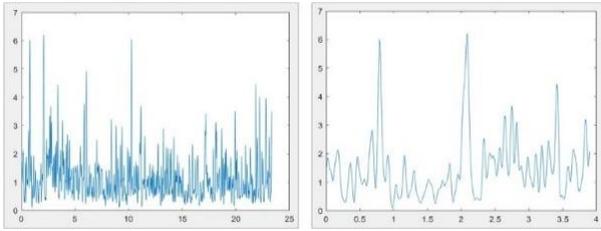
“Muestras del Participante 1-2 – Derivación Theta”



Nota. participante 1-2 csv – todas las muestras y 500 muestras de movimiento MATLAB, señal THETA. Tomada de (imagen propia)

Figura 22

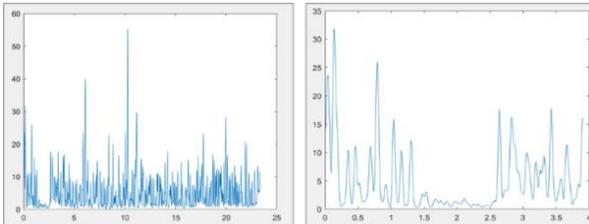
“Muestras del Participante 1-2 – Derivación Alpha”



Nota. participante 1-2 csv – todas las muestras y 500 muestras de movimiento MATLAB, señal ALPHA. Tomada de (imagen propia)

Figura 23

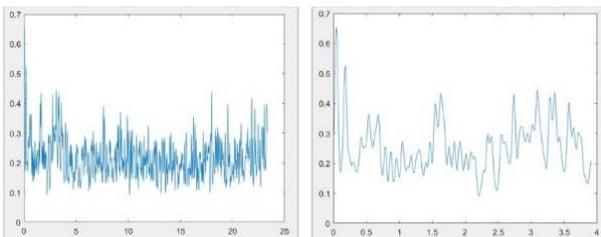
“Muestras del Participante 1-2 – Derivación



Nota. participante 1-2 csv – todas las muestras y 500 muestras de movimiento MATLAB, señal Beta. Tomada de (imagen propia)

Figura 24

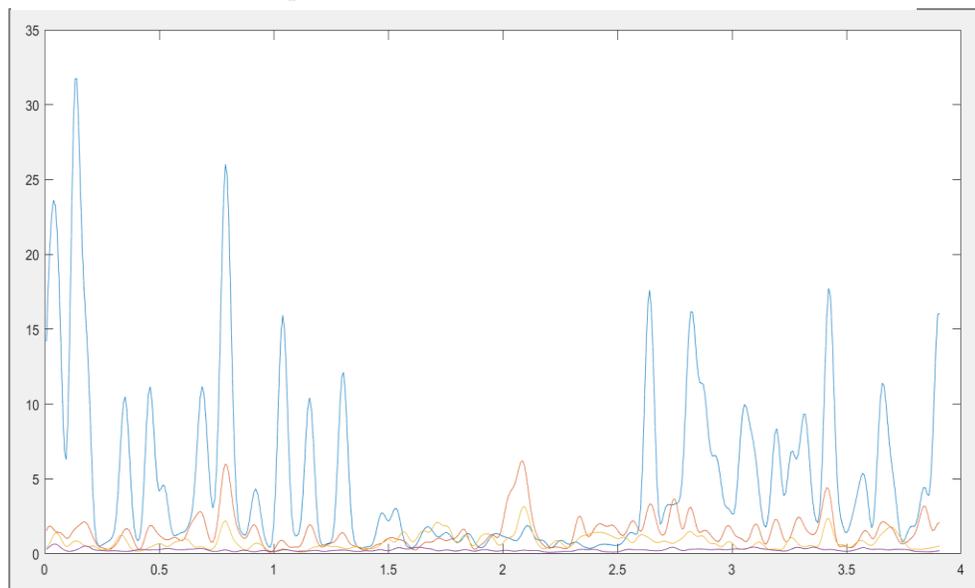
“Muestras del Participante 1-2 – Derivación Gamma”



Nota. participante 1-2 csv – todas las muestras y 500 muestras de movimiento MATLAB, señal GAMMA. Tomada de (imagen propia)

Figura 25

“Muestras del Participante 1-2 – 4 Derivaciones Anteriores”



Nota. participante 1-2 csv – todas las muestras y 500 muestras de movimiento MATLAB, 4 derivaciones Theta, Alfa, Beta y Gamma. Tomada de (imagen propia)

En esta figura se visualizan los 4 ritmos de señales EEG del electrodo F7, observando las muestras de 0 a 500, en algunas de ellas se detallan picos en frecuencias y amplitudes que podría ser generado por algún tipo de artefacto.

Inspección de Características

Parámetros en el Dominio del Tiempo: Clasificación – Características Temporales

Estadísticas

Valor Absoluto Medio

$$MAV = \frac{1}{N} \sum_{j=1}^N |x_j|$$

A partir del modelo matemático del valor absoluto medio, la suma de j (número de muestras) hasta N (número total de muestras) del valor absoluto de cada muestra x_j (cada una de las muestras de la señal), todo se divide entre N , con el fin de promediar el valor absoluto, por lo tanto se convierte el modelo matemático a MATLAB.

```
%% Valor absoluto medio
vam=0; %% Inicializar una variable para asignarle posteriormente el valor obtenido
for i=1:length(signal)
    vas=abs(signal(i)); %% Variable que cambia con cada ciclo dependiendo del valor de
    la muestra
    vam=vam+vas; %% Sumamos vas+vam
end
vam=vam/length(signal); %% Dividir entre el número de muestras
```

Energía Total

Parámetro que permite conocer la energía de la señal, es fundamental en el momento de la aplicación del filtrado, toda vez que permite visualizar que previa y posteriormente la existencia de un cambio en su energía (deber ser lo mínimo posible); es decir que, si se presenta un cambio abrupto en la energía de la señal significa que se está perdiendo información de la señal.

$$SSI_k = \sum_{i=1}^N (|x_i^2|)$$

El modelo matemático del parámetro de la energía es la suma del valor absoluto del cuadrado de lo que hay en cada muestra, con el fin de obtener la energía de la señal.

```
%% Energía total
en=0; %% Inicializar energía en 0

for i=1:length(signal)
    enm(i)=signal(i)^2; %% Cada muestra al cuadrado }
    en=en+enm(i); %% Sumar a la variable inicializada en cero
end
```

Parámetro ZRC (Zero Croos Rate)

Este parámetro permite indicar que tan seguida cruza por cero el registro de la señal, el cual puede estar relacionado con la frecuencia; es decir, a mayor frecuencia se puede inferir que existe repetidamente cruces en cero.

```
%% ZCR (Zero croos rate)
w=diff(signal>0);
zcr=sum(abs(diff(signal>0)))/length(signal); %% Obtener indicador de cruces por cero
```

Parámetros de Hjorth

Existen 3 parámetros estadísticos para identificar propiedades de actividad, movilidad y complejidad en el dominio del tiempo, así:

Parámetro Actividad (Representa la Potencia de la Señal de EEG)

$$A_x = \frac{\sum_{t=1}^T (x(t) - \mu)^2}{T}$$

El modelo matemático indica la sumatoria de todas las muestras, $x(t)$ cada muestra $-\mu$ media de todas las muestras, todo elevado al cuadrado y dividido sobre todo el número de muestras, igualmente, se adapta el modelo en MATLAB.

```
%% Activity (representa la potencia de la señal de EEG)

% Obtener media

S=sum(signal);

med=S/length(signal);

act=0;% Inicializar activity

for i=1:length(signal)

    val=(signal(i)-med)^2; % Variable que cambia con cada ciclo dependiendo del número de
muestra

    act=act+val; % Act = valor previo + valor actual

end

activity=act/length(signal);
```

Parámetro Movilidad (Representa la Frecuencia Media – Frecuencia Dominante de la Señal EEG).

$$A_x = \sqrt{\frac{\text{var}(x'(t))}{\text{var}(x(t))}}$$

Se tiene en cuenta la varianza de la señal a partir de la primera derivada, dividida por la varianza de la señal.

```
%% Movility

var1=var(signal); %% Varianza de señal original

var2=var(diff(signal)); %% Varianza de primera derivada de la señal con diff

divar=var2/var1; %% Cociente de las derivadas

movility=sqrt(divar); %% Obtener la raíz cuadrada del cociente
```

% Obtener frecuencia dominante

fremedia=movility*(fs/(2*pi)); %Multiplicar moviliti por la frecuencia de muestreo dividida entre
2 pi

Parámetro Complejidad (Representa los Cambios en la Frecuencia de la Señal, Compara la Señal EEG con una Señal Sinusoidal).

$$c_x = \frac{M'(x(t))}{M(x(t))}$$

El modelo matemático obtiene del cociente de movilidad de la derivada, dividida de la movilidad de la señal original.

%% Complexity

%Movility x'(t

s1=diff(signal); %Primera derivada de la señal

s2=diff(s1); %Segunda derivada

var1=var(s1); %VArianza de primera derivada

var2=var(s2); %Varianza de segunda derivada

divar=var2/var1; %Cociente de varianzas

movility2=sqrt(divar);%Mvility de la primera derivada

complexity=movility2/movility;

Parámetros en el Dominio de la Frecuencia

Espectro de Frecuencias de la Señal

$$(f) \int_{-\infty}^{\infty} X(t) e^{-j2\pi ft} dt$$

Se usa el modelo matemático de la transformada de Fourier (es una transformada bipolar que cuenta con un lado positivo y negativo que son exactamente iguales, por ende, la mitad de

cero a ceros positivos y de cero a negativos) para generar un espectro de frecuencias a partir de cada una de las muestras.

Transformada Rápida de Fourier en MATLAB

```
Y=fft(signal);%%FFT - transformada rápida de Fourier
P2 = abs(Y/L);%obtener únicamente la parte real de la transformada
P1 = P2(1:L/2+1);%Elimina espectro reflejado
P1(2:end-1) = 2*P1(2:end-1);%eliminar espectro reflejado
f= fs*(0:(L/2))/L;% Vector de frecuencias
```

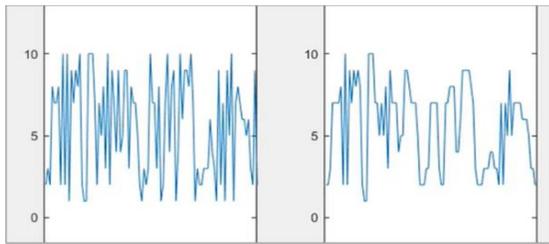
Se observó el contenido frecuencial, detallado la aplicación del teorema de Nyquist el cual indica que, esas frecuencias tienen que ir hasta la mitad de la frecuencia de muestreo, en este caso, la frecuencia es de 128 muestras por segundo; por lo tanto, tienen que ir hasta 64 Hz.

Analizando el espectro de Fourier con el teorema de Nyquist, se logró verificar que el filtrado corresponde desde 0 hasta el 45 con relación al ancho de banda, siendo congruente el filtrado de las señales del Data Set.

Aplicación Filtro de Mediana

Se aplica el filtro temporal para atenuar artefactos detectados, con el fin de suavizar la señal, comprobando no es recomendable este filtro, toda vez que se comprobó que existe una pérdida de información en los componentes en frecuencia debido a su atenuación.

En este sentido, por parte del Ingeniero Biomédico Oscar Aguilar Mejía, recomienda no emplear para este tipo de biopotenciales utilizar filtros temporales (mediana, Savitsky Golay).

Figura 26*“Filtro de Mediana en Matlab”*

Nota. Se observa que al aplicar el filtro de mediana sus componentes en frecuencia se ven afectados. Tomada de (imagen propia)

En este contexto, al evidenciar el espectro de Fourier da un indicio que los filtros del Data Set se encuentran bien con respecto a la información que se tiene, se establece que las frecuencias son las adecuadas. Es de anotar que, el filtro del DataSet corresponde a un filtro Pasabanda (deja pasar un determinado rango de frecuencias y atenúa el paso de las demás), por lo tanto, se omite la parte de filtrado y se continúa con las características temporales estadísticas en tiempo y frecuencia.

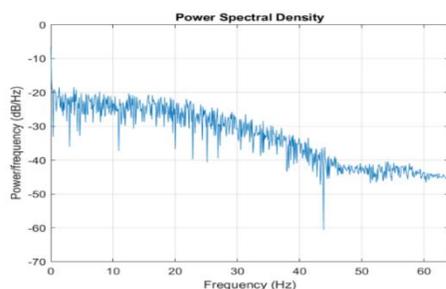
Densidad Espectral de la Señal

Es distribución de la potencia del EEG sobre las distintas frecuencias. La densidad espectral permite conocer el rango de frecuencias donde se concentran las variaciones de potencia.

```
Hpsd = dspdata.psd(P1(1:length(P1)/2),'fs',fs); %(espectro(1:mitad de la longitud del
espectro,'fs',fs)
```

Figura 27

“Densidad Espectral”



Nota. Se visualiza la densidad espectral, permitiendo observar el rango de frecuencias en MATLAB. Tomada de (imagen propia)

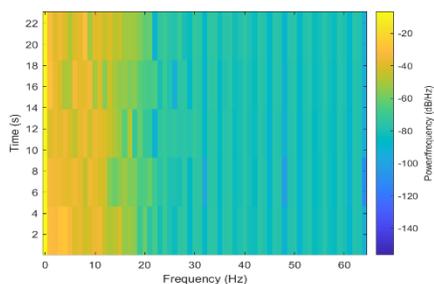
Espectrograma

```
spectrogram(signal, 600, 10, fs, fs, 'xaxis')
```

Este método representa los datos EEG como un mapa de calor, donde la intensidad de color indica la potencia de la señal en una frecuencia específica. Muestra el contenido frecuencia de la señal EEG con respecto al tiempo.

Figura 28

“Espectrograma EEG”



Nota. Se visualiza la representación gráfica que muestra la actividad eléctrica del cerebro en el tiempo y la frecuencia. Tomada de (imagen propia)

Aplicación de Modelos Autorregresivos

Ajusta la secuencia de datos EEG, mediante un proceso aleatorio el cual tiene en cuenta las muestras pasadas (valor actual al que se quiere predecir dependiendo del mismo valor, pero en el pasado), teniendo en cuenta la ecuación del proceso autorregresivo (modelo AR de orden p) - (Medina, Sierra, & Barrios, 2018).

$$X_k = a_1x_{k-1} + a_2x_{k-2} + a_px_{k-p} + r_k$$

a = coeficientes para clasificar

p = cantidad de muestras que se quieren modelar

x = valores observados de la muestra

rk = termino de erro

Aplicación del modelo matemático en MATLAB

```
%% Autoregresión
signal=double(signal);%Convertir a double para implementar Varm
j=varm(1,5);%Crear vector de coeficientes
y=estimate(j,signal);%estimar los coeficientes de la
autorregresion estimate(vector creado con varm, señal)
co=y.AR;%Obtener coeficiente guardados en la celda ar de la
estructura y
co=cell2mat(co);%Convertir en vectro de variable double
(conveniencia)
```

Matriz de Características

Por medio de los pasos anteriores se genera una tabla a través de MATLAB que contiene información cuantitativa extraída de cada canal en los datos EEG, en esta sección se selecciona un conjunto de datos con los canales deseados, en este caso, las derivaciones que se acercan o influyen de alguna manera en la corteza motora o premotora, las cuales podrían proporcionar información relevante a la intención de movimientos para el control de la mano robótica.

Frente a lo anterior, de los 14 canales que usa la diadema Emotiv, se tienen en cuenta 6 canales; FC5 y FC6 (ubicados en la corteza premotora y corteza motora suplementaria – asociadas a la planificación de movimientos) y F3, F4, F7, y F8 (tomadas desde el lóbulo frontal – señales relacionadas con la actividad motora, pero en el ámbito del control y toma de decisiones).

En esta parte previa antes del entrenamiento del modelo, es importante tener en cuenta las potencias de las bandas de frecuencia (theta, alfa, baja beta, alta beta y gamma) para cada uno de los electrodos de interés (F3, F4, F7, F8, FC5, FC6), proporcionando una visión amplia de la actividad cerebral en estas áreas; las ondas alfa están asociadas con estados de relajación y ociosidad, las ondas beta con la concentración y la actividad mental, y las ondas gamma se han relacionado con la percepción y el procesamiento de información de alto nivel.

Como aspecto de interés, para reducir los puntos en el tiempo, se procesan las señales seleccionadas en ventanas de 7 segundos, en cada una de ellas, se calculan varias características que incluyen medidas de la frecuencia y la complejidad de la señal. Para concluir, la matriz de características sirve como resumen cuantitativo de los datos EEG originales, permitiendo un análisis detallado y la aplicación de un algoritmo de machine learning.

Figura 29

“Matriz de Características”

1	nms	valor_absoluto_medio	energia	activity	movility	frecuencia_medio	complexity	potencia	car1	car2	car3	car4	car5	category
2	1-2_AF3_THETAw1	1,50457E+14	6,42025E+12	4,90172E+14	2,76524E+14	5,63329E+14	1,70685E+14	5,59739E+14	4,11455E+14	-7,26155E+14	6,85888E+14	-3,46694E+14	7,5163E+14	1
3	1-2_AF3_THETAw2	1,71836E+14	7,35264E+14	5,2533E+14	3,0651E+14	6,24417E+14	1,57478E+14	4,68024E+14	4,04985E+14	-7,04643E+14	6,56921E+14	-3,27851E+14	7,00529E+13	1
4	1-2_AF3_ALPHAw1	3,28681E+14	1,79229E+14	9,20003E+14	2,91954E+14	5,94764E+14	1,59315E+14	1,60293E+14	4,06168E+14	-7,07771E+14	6,60748E+14	-3,30646E+13	7,10534E+13	1
5	1-2_AF3_ALPHAw2	3,00586E+14	1,61257E+14	8,96224E+14	3,1022E+14	6,31976E+14	1,55783E+14	1,0611E+14	4,09979E+13	-7,21837E+14	6,80687E+13	-3,436E+14	7,43308E+14	1
6	1-2_AF3_LOW_BETAw1	1,35483E+14	2,72919E+14	1,21042E+14	2,63058E+14	5,35897E+14	1,77567E+14	2,11E+14	4,06145E+14	-7,08161E+14	6,62231E+14	-3,32224E+13	7,16905E+14	1
7	1-2_AF3_LOW_BETAw2	1,15879E+13	1,81378E+14	6,81527E+14	2,88078E+14	5,86869E+14	1,69412E+14	1,86726E+14	4,09154E+13	-7,1922E+13	6,77702E+14	-3,42499E+14	7,45679E+14	1
8	1-2_AF3_HIGH_BETAw1	5,78805E+14	3,4834E+14	5,37573E+13	2,49514E+14	5,08305E+14	2,05029E+14	3,03717E+14	4,02756E+14	-6,99712E+14	6,54314E+14	-3,29008E+14	7,13148E+14	1
9	1-2_AF3_HIGH_BETAw2	5,5817E+14	3,20251E+14	4,58689E+14	2,8979E+14	5,90356E+14	1,68751E+13	2,62714E+14	4,10585E+14	-7,25849E+14	6,89624E+14	-3,52062E+14	7,73667E+14	1
10	1-2_AF3_GAMMAw1	3,89378E+14	1,68371E+14	3,62988E+14	2,43226E+14	4,95496E+14	1,87914E+14	1,41642E+14	4,0879E+14	-7,18233E+13	6,77955E+14	-3,43872E+14	7,51035E+14	1
11	1-2_AF3_GAMMAw2	3,4925E+14	1,18743E+14	1,05495E+13	2,89346E+14	5,89451E+14	1,71902E+14	9,86366E+14	4,16159E+14	-7,44371E+13	7,13963E+14	-3,6704E+14	8,1038E+14	1
12	1-2_F3_THETAw1	4,23912E+14	8,83255E+14	9,67805E+14	2,93908E+14	5,98745E+14	1,64323E+14	2,96531E+14	4,03003E+14	-6,9705E+14	6,45594E+14	-3,19866E+13	6,8026E+14	1
13	1-2_F3_THETAw2	2,69483E+14	1,36908E+14	8,01785E+14	3,0823E+14	6,27922E+14	1,5753E+14	8,96231E+14	4,00946E+14	-6,89987E+14	6,35282E+14	-3,12482E+14	6,56208E+14	1
14	1-2_F3_ALPHAw1	1,08585E+14	4,88894E+14	5,33849E+14	2,88765E+14	5,88268E+14	1,65357E+14	2,39487E+14	4,1943E+14	-7,54645E+14	7,27134E+14	-3,75045E+14	8,28659E+14	1
15	1-2_F3_ALPHAw2	8,22651E+14	8,50189E+14	2,72118E+14	3,1488E+14	6,41468E+14	1,51781E+14	6,0721E+14	4,02913E+14	-6,96378E+14	6,43568E+14	-3,1765E+13	6,69081E+14	1
16	1-2_F3_LOW_BETAw1	6,96167E+14	2,43518E+14	2,66937E+14	2,9324E+14	5,97384E+14	1,64791E+14	5,76427E+14	4,17678E+14	-7,4942E+14	7,21027E+14	-3,71807E+13	8,22406E+14	1
17	1-2_F3_LOW_BETAw2	4,59876E+14	2,45392E+14	6,23886E+14	2,67589E+14	5,45128E+14	1,83507E+14	2,43563E+14	4,02286E+14	-6,94557E+14	6,42502E+14	-3,18724E+14	6,81929E+13	1
18	1-2_F3_HIGH_BETAw1	2,36207E+14	2,38642E+14	2,60763E+13	2,93761E+14	5,98444E+14	1,64418E+14	7,90635E+14	4,1735E+14	-7,48369E+14	7,19701E+14	-3,71028E+14	8,20589E+14	1
19	1-2_F3_HIGH_BETAw2	2,44126E+14	5,96535E+13	6,98012E+14	2,80644E+14	5,71724E+14	1,72418E+14	4,95773E+14	4,06551E+13	-7,11568E+14	6,69653E+14	-3,3885E+14	7,38455E+14	1
20	1-2_F3_GAMMAw1	8,44383E+14	2,41186E+14	2,6205E+14	2,92932E+14	5,96756E+14	1,64323E+14	6,09882E+14	4,18291E+14	-7,51354E+14	7,23462E+14	-3,73246E+14	8,25718E+14	1
21	1-2_F3_GAMMAw2	1,6172E+14	2,50472E+13	1,80113E+13	2,75465E+14	5,61173E+13	1,76842E+14	2,18454E+14	4,13239E+14	-7,35484E+14	7,04166E+14	-3,62535E+14	8,03474E+14	1
22	1-2_F4_THETAw1	3,36419E+14	1,80269E+14	8,8015E+14	2,84578E+14	5,79738E+14	1,68134E+14	1,65928E+14	3,80409E+13	-6,16602E+14	5,30867E+14	-2,42675E+14	4,73215E+14	1
23	1-2_F4_THETAw2	4,16582E+14	2,86384E+14	1,46085E+14	3,18559E+14	6,48963E+14	1,5203E+14	1,86104E+14	3,9905E+14	-6,84261E+14	6,28394E+14	-3,08954E+14	6,51438E+13	1
24	1-2_F4_ALPHAw1	1,79654E+14	4,45038E+14	1,7394E+14	2,23061E+14	4,54417E+14	2,01449E+14	4,95172E+14	4,08449E+14	-7,13164E+14	6,66242E+13	-3,32754E+14	7,09469E+14	1
25	1-2_F4_ALPHAw2	1,61593E+14	3,11041E+14	8,60202E+14	3,07196E+14	6,25814E+13	1,53065E+14	2,57146E+14	4,06767E+14	-7,07603E+14	6,56507E+14	-3,24764E+14	6,85997E+14	1
26	1-2_F4_LOW_BETAw1	1,11369E+14	1,64745E+14	5,98361E+12	2,25516E+14	4,59418E+14	2,08695E+14	1,67789E+14	4,05559E+14	-7,04449E+14	6,55552E+14	-3,26663E+14	6,97479E+14	1

Nota. Se visualiza la tabla de características de los 6 canales con sus respectivas potencias de

bandas de frecuencia con un total de 16.960 datos. Tomada de (imagen propia)

Uso de la Aplicación Classification Learner de MATLAB

En el siguiente paso para el análisis, se emplea la aplicación Classification Learner, es una herramienta útil para entrenar modelos de clasificación de machine learning utilizando datos tabulados; es decir, la matriz de características generada anteriormente.

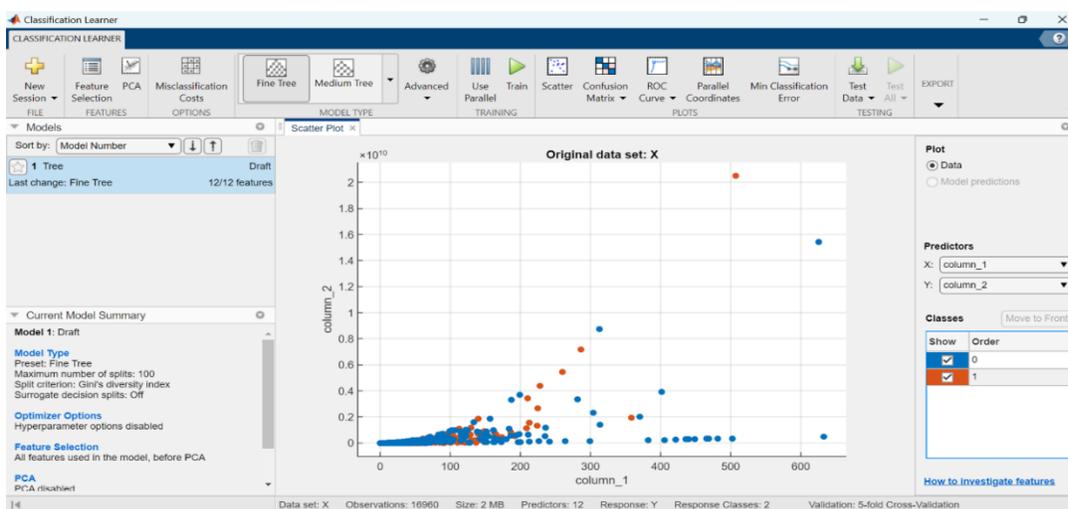
La matriz de características denominada “X”, representa las entradas del modelo (los datos a partir de los cuales el modelo aprenderá a hacer sus predicciones). En este proyecto las características (X) están compuestas por las características extraídas de los canales de EEG seleccionados. Cada una característica como el valor absoluto medio, la energía, la actividad, la movilidad, la frecuencia media, la complejidad, la potencia, entre otras, representando una dimensión en el espacio de características. Estos valores sirven como la base sobre la cual el modelo de aprendizaje automático construirá su capacidad para distinguir entre diferentes clases de datos.

Vector objetivo denominado “Y”, representa las etiquetas o clases de datos, en este proyecto las clases pueden ser diferentes tipos de movimientos o estados mentales (movimiento y no movimiento). El vector objetivo es lo que el modelo de aprendizaje automático intentará predecir a partir de las características de la matriz “X”. Durante el entrenamiento el modelo aprende a asociar patrones en la matriz X con clases específicas en “Y”.

Frente a lo anterior, por medio de Classification Learner, se importa la matriz de características como las entradas X y los objetivos como salida Y. Es importante mencionar que, la aplicación proporciona una interfaz fácil de usar, permitiendo seleccionar el tipo de modelo de clasificación a entrenar, configurar sus parámetros y evaluar su rendimiento. En este proceso se puede iterar y optimizar el modelo hasta obtener los resultados sin un sobre ajuste.

Figura 30

“Análisis Exploratorio de Clasificación”



Nota. Se puede observar los patrones al momento de comparar características o la relación existente entre ellas, valores atípicos en el modelo; visibilizando en color azul las de no movimiento y naranja de movimiento. Tomada de (imagen propia)

Se validaron todas las señales de los 27 participantes en los diferentes modelos de Classification Learner, obteniendo como resultado en el entrenamiento, una precisión del 70%. Sin embargo, con el fin de perfeccionar este valor, se analizaron las señales de 2 pacientes, tanto en la etapa de entrenamiento y prueba, para las actividades de movimiento y no movimiento, arrojando como resultado un 81,2%.

Asimismo, se analizó la matriz de confusión que permite dar una lectura que tan bueno fue el clasificador del algoritmo, indicando los tipos de clases, aciertos (verdaderos positivos) y desaciertos, permitiendo observar la cantidad de veces que se equivocó, frente a los dos participantes en entrenamiento y prueba.

Es importante indicar que, al analizar los datos de los 3 primeros participantes en solo en la etapa de entrenamiento, se obtuvo como resultado un 87,5% en los diferentes modelos. Sin embargo, para evitar un sobre ajuste, se anexaron los dos tipos de datos, prueba y entrenamiento.

Figura 31

“Resultados con el Modelo 1.22 Bagged Trees”



Nota. En las dos imágenes se observa el resultado con los dos participantes en las dos etapas (entrenamiento y prueba), evidenciando un 81,2% de efectividad. Tomada de (imagen propia)

Decodificación de Señales Neuronales para el Control de una Mano Robótica (Intención de Movimiento)

Este paso de decodificación es importante en la interfaz cerebro – máquina (BCI), teniendo en cuenta que se refiere a la interpretación de señales neuronales para producir una salida deseada, como el movimiento de una mano robótica. En este sentido, en el modelo de decodificación al momento de extraer las características, se entrena un modelo usando un algoritmo de aprendizaje automática para interpretar las características y producir una salida. Para este proyecto la salida es una predicción de si hay un movimiento o sin movimiento. Es de indicar que, para utilizar el modelo en situaciones del mundo real, es necesario hacer pruebas en datos no vistos anteriormente que permita validar su precisión y robustez. Posterior a ello, ya validado el modelo puede ser empleado en tiempo real para interpretar señales neuronales y controlar dispositivos externos, como una mano robótica, en respuesta a la actividad cerebral del usuario.

Figura 32

“Resultado en la Decodificación de Señales EEG”



```
Command Window
Sin movimiento
Movimiento detectado
Sin movimiento
Movimiento detectado
Sin movimiento
Movimiento detectado
Sin movimiento
fx >> |
```

Nota. Los mensajes de "Sin movimiento" y "Movimiento detectado" son el resultado esperado en función de los datos. Tomada de (imagen propia)

Diseño del Proyecto EEG Previo a la Implementación

Estructura Mecánica Mano Robótica

Se utilizó la mano robótica "uHand Robotic Hand", fabricada por Hiwonder, y hecha de una aleación de aluminio y acrílico. Esta mano está diseñada para simular el movimiento de dedos biónicos. Sus dimensiones son 165*90 mm y tiene un peso de 185g. Incorpora servos antibloqueo LFD 01. La adquisición de esta mano robótica se realizó a través de Amazon.com, bajo el nombre "Robot Mano Cinco Dedos Solo Movimiento Robot Biónico Brazo Mecánico DIY (Mano Derecha)". Se eligió este modelo de robot debido a que es visualmente llamativo y cumple con los requerimientos del proyecto, en particular, su capacidad para mover los dedos en respuesta a señales EEG de movimiento y no movimiento.

Figura 33

“Mano Robótica”



Nota. Mano robótica para la implementación EEG. Tomada de (Hiwonder, s.f.)

Actuadores

Como se indicó anteriormente, se emplearon 5 servomotores para el funcionamiento de la mano robótica denominado “LFD-01M Robot Servo 6V 9g Micro Servo antibloqueo”, teniendo las siguientes especificaciones técnicas:

Tabla 2*Especificaciones Técnicas del Servodigital LFD-01M*

Actuador: servomotor LFD-01M	ESPECIFICACIONES TÉCNICAS
Servodigital LFD-01M	50mA
Corriente de servo	700mA
Peso	14g
Método de control	PWM
Tamaño	32,5*12*29,85mm
Pulso pwm	500~2500us(0-180°)
Tensión de trabajo	4,8-6V
Velocidad de rotación	0,12 segundos/60°4,8 V; 0,10 s/60°6 V

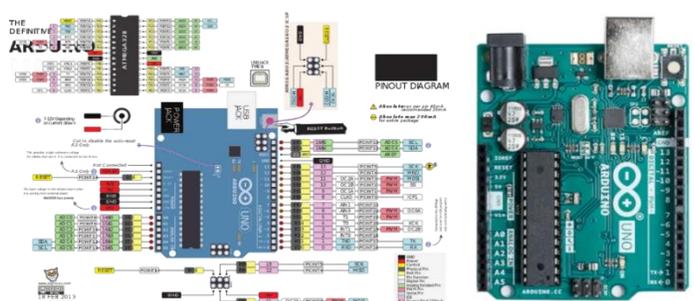
Nota. Se observan las características del servomotor. Fuente: (Hiwonder, s.f.)

Sistema de control

Se empleó ARDUINO UNO como sistema de control, teniendo en cuenta su facilidad de programación y su compatibilidad con diversos módulos y sensores. Para este proyecto, es fundamental este microcontrolador, en el sentido que recibe las señales EEG controladas desde MATLAB y programadas en Arduino determinado cómo actuar en respuesta de estas señales.

Figura 34

“Placa Arduino Uno”

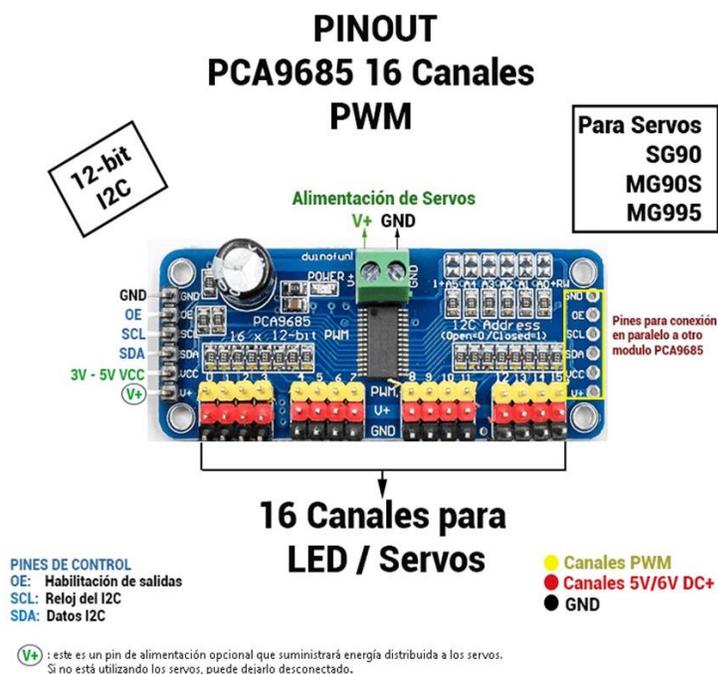


Nota. Diagrama de la placa Arduino y del ATmega328PU, cuenta con 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM). Tomada de (Arduino.cl, s.f.)

Asimismo, para un control preciso y eficiente de los servomotores de la mano robótica, se integró el PCA9586, un controlador de PWM de 16 canales, permitiendo generar señales PWM en múltiples canales de manera simultánea, proporcionando una modulación adecuada del ancho de pulso; es decir, traduciéndolo en movimientos controlados y suaves a los servos. Es importante resaltar que, este módulo se comunica con el Arduino a través del protocolo I2C, reduciendo la cantidad de pines necesarios del Arduino.

Figura 35

“Modulo PCA9685”



Nota. Servocontrolador LED que a través del bus I2C de 16 canales. Tomada de (UNIT ELECTRONICS, s.f.)

Tabla 3*Especificaciones Técnicas del Módulo PCA9685*

Modulo PCA9685	ESPECIFICACIONES TÉCNICAS
Voltaje de alimentación	5V DC
Máxima corriente por pin	25mA
Interfaz	I2C
Resolución	12 bits
Número de canales	16
Frecuencia de operación	40 Hz a 1000 Hz
Dimensiones	62 mm x 26 mm
Protección:	Inversión de Voltaje

Nota. Se detallan las características del servo controlador. Fuente: (UNIT ELECTRONICS, s.f.)

Comunicación con la Mano Robótica

Para lograr que desde MATLAB se controle los movimientos de acuerdo con la decodificación de señales neuronales para el control de una mano robótica (intención de movimiento), fue necesario instalar el paquete de MATLAB Support Package for Arduino, con el fin de lograr la inicialización del Arduino en MATLAB y comunicarse con el microcontrolador. De igual forma en Arduino, se tuvo que instalar la librería Adafruit_PWMServoDriver para la comunicación con el módulo PCA9685.

Configuración Inicial Entre los Softwares

Se establece una comunicación entre MATLAB y Arduino a través del puerto "COM5" a una velocidad de 9600 baudios. De igual forma, se configuran los ángulos iniciales de apertura y cierre para los servomotores.

Carga y Procesamiento de Datos

Se cargan las características previamente extraídas de las señales EEG desde featuresMatrix.mat (código completo en los anexos "Decodificación_simulación"). De esta matriz, solo se toman las primeras 12 columnas, que contienen las características relevantes, posteriormente, se convierten en una tabla en MATLAB para su manipulación y procesamiento.

Decodificación y Predicción

En este procedimiento se carga el modelo entrenado, verificando que dicho modelo sea adecuado, posteriormente, se utiliza para hacer predicciones basadas en las características cargadas. Las predicciones indicarán si se detecta movimiento o no.

Visualización en Tiempo Real

A través de MATLAB se crea el gráfico donde se muestran las predicciones a medida que se generan; es decir, si se detecta un movimiento (predicción=1), se muestra en el gráfico y se envía una orden al Arduino para abrir la mano robótica. Si no detecta movimiento (predicción=0), la mano robótica se cierra.

Actuación de los Servomotores

En base a la predicción, se envía un comando específico a Arduino a través del puerto serie, el cual indica a qué canal del servomotor (específico para cada dedo o parte de la mano) debe moverse y cuál debe ser su ángulo.

Interpretación y Actuación en Arduino

El microcontrolador espera y escucha las instrucciones que llegan a través del puerto serie. Una vez que la recibe válida (delimitada por los caracteres '<' y '>'), extrae el canal del servomotor y el ángulo al que debe moverse. Utilizando el controlador PCA9685, se genera la señal PWM correspondiente para mover el servomotor al ángulo deseado.

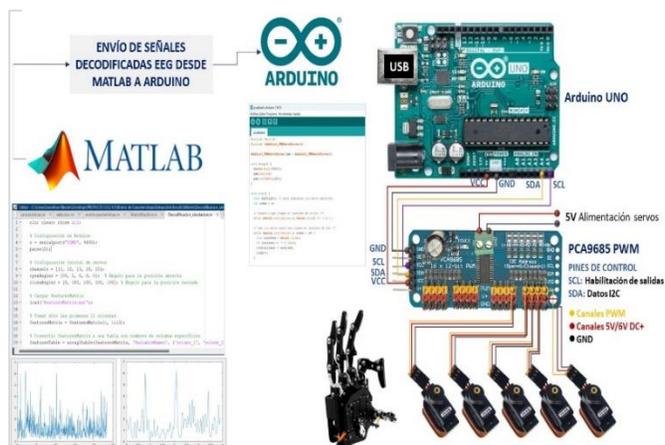
Retroalimentación

Después de actuar en base a la instrucción, Arduino envía un mensaje de confirmación a MATLAB a través del puerto serie, indicando que el comando fue ejecutado.

Diagrama del Circuito

Figura 36

“Diseño del Circuito”



Nota. Diseño del diagrama del circuito para el control EEG. Tomada de (imagen propia)

Descripción del Diagrama del Circuito.

Microcontrolador Arduino

Actúa como el cerebro del sistema, establece comunicación con MATLAB a través del puerto serie y Controla el módulo PCA9685 para gestionar el movimiento de los servomotores.

Módulo PCA9685

El controlador basado en señales PWM, se comunica con Arduino a través del protocolo I2C, para el control de los 5 servomotores, utilizando los puertos 11 (control dedo pulgar), 12 (dedo índice), 13 (medio), 14 (anular) y 15 (meñique).

Servomotores

Son los actuadores que se mueven en el ángulo específico basado en la señal PWM recibida, representando el movimiento de los dedos, como se indicó anteriormente, en la conexión del módulo PCA9685. Es importante mencionar, que el cable de color naranja es donde ingresa la señal, el rojo es VCC y el café GND.

Conexión Serie

Se efectúa la comunicación bidireccional entre MATLAB y Arduino a través del puerto serie (COM5). En este sentido, desde MATLAB envía los comandos al microcontrolador para mover los servomotores basado en las predicciones. Arduino confirma la ejecución de los comandos enviados.

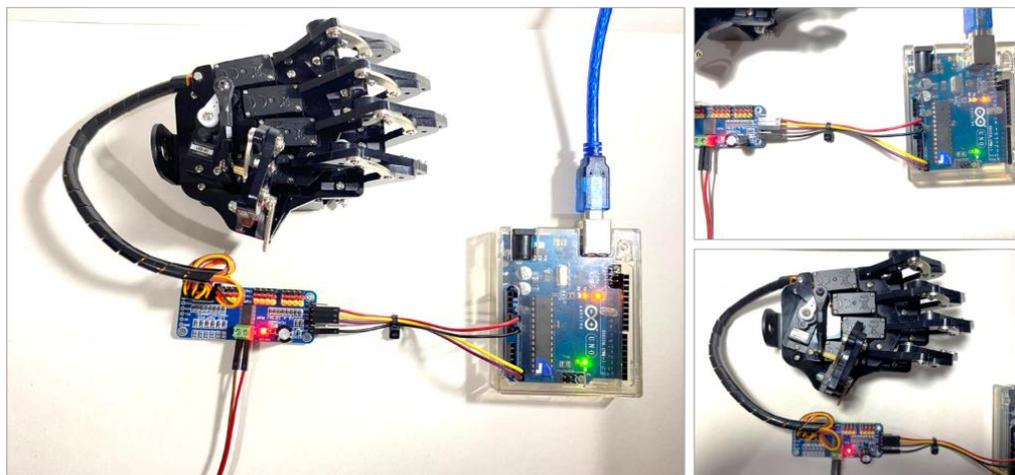
Alimentación y Tierra

El Arduino, el módulo PCA9685 y los servomotores se conectan a tierra. Asimismo, se emplea el uso de una batería de 5 voltios, que permita dar una alimentación adecuada a los servomotores para su funcionamiento.

Implementación en Físico del Prototipo de Mano Robótica

Figura 37

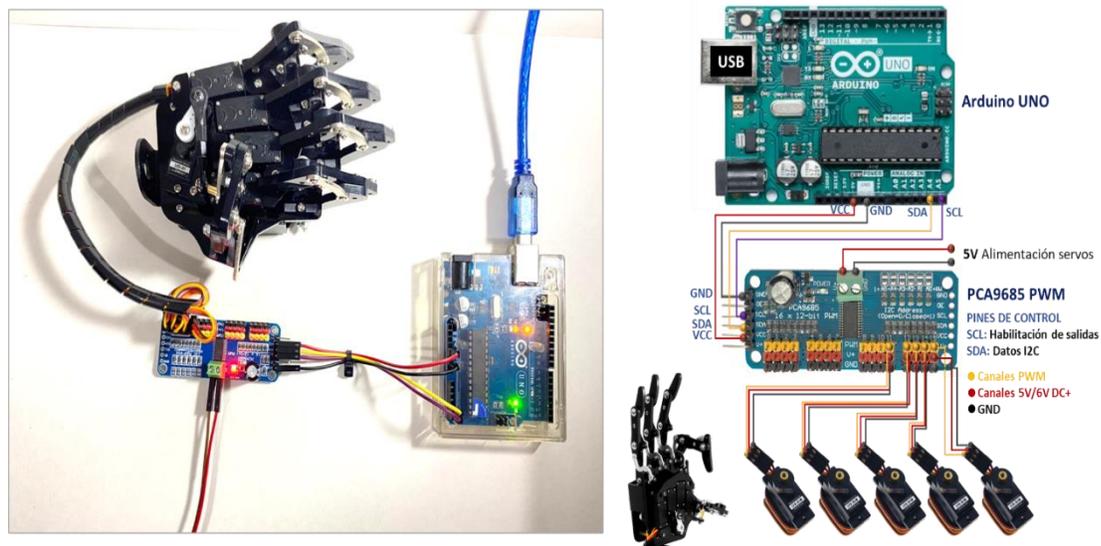
“Implementación con los Componentes”



Nota. Implementación del circuito de acuerdo con el diseño. Tomada de (imagen propia)

Figura 38

“Correlación del Circuito – Diseño e Implementación”

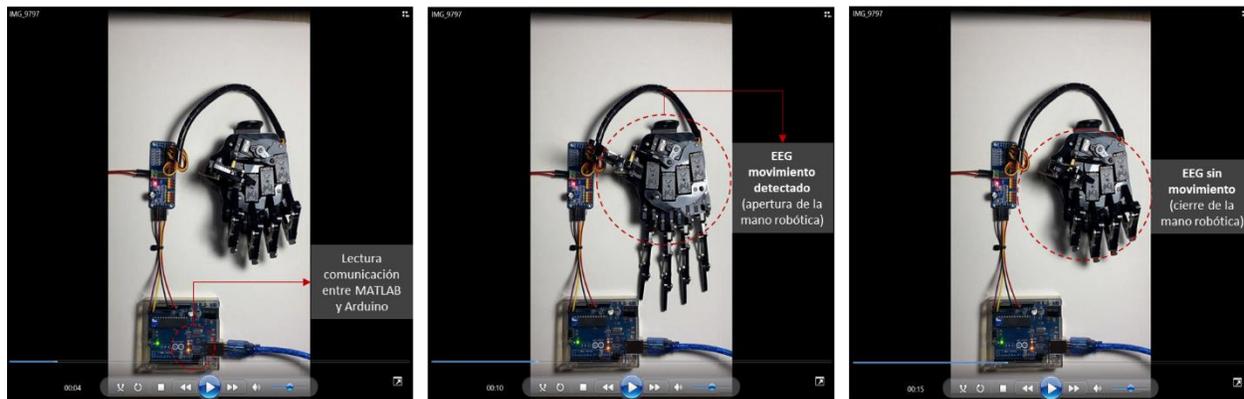


Nota. Comparación circuito. Tomada de (imagen propia)

Pruebas de Funcionamiento

Figura 39

“Línea de Tiempo del Funcionamiento del Circuito EEG”



Nota. Se evidencia la comunicación que llega al puerto COM5 del Arduino desde MATLAB y el reconocimiento de la señal EEG decodificada de movimiento detectado y sin movimiento, para la apertura o cierre la mano robótica. Tomada de (imagen propia)

Presentación de la Implementación del Sistema

Figura 40

“Implementación final”



Nota. Se evidencia la implementación final en maqueta. Tomada de (imagen propia)

Análisis de Resultados

El principal objetivo de este proyecto fue utilizar señales EEG para controlar de manera efectiva una mano robótica. A través de la implementación del sistema se pudo observar los siguientes aspectos:

A través de la combinación de MATLAB y el modelo de machine learning, el sistema fue capaz de decodificar con éxito las señales EEG pregrabadas (DATA banco de señales fisiológicas), distinguiendo entre las intenciones de movimiento y no movimiento del usuario. Es importante indicar que, si bien no se hicieron pruebas con electrodos en tiempo real, la utilización de un banco de señales previamente recolectadas y categorizadas permitió simular con precisión la interacción del usuario con el sistema.

La conexión entre MATLAB y Arduino resultó ser efectiva, a pesar de los desafíos iniciales relacionados con la alimentación de los servomotores, la solución de usar una batería externa de 5 voltios garantizó una respuesta adecuada y sincronizada de la mano robótica.

La mano robótica respondió adecuadamente a las predicciones obtenidas a partir de las señales decodificadas. Cuando el sistema predijo una intención de movimiento, la mano robótica abrió los dedos y cuando no hubo intención de movimiento, los dedos se cerraron. Esta respuesta validó aún más la precisión del sistema de decodificación.

A partir de los resultados obtenidos, se concluye que el proyecto ha demostrado ser viable y tiene un gran potencial en aplicaciones futuras. Si bien esta etapa inicial se utilizó un conjunto de datos pregrabado, los resultados indican que el sistema podría adaptarse para funcionar en tiempo real, abriendo el camino hacia aplicaciones avanzadas y prácticas.

Tiempo de Respuesta del Sistema

Tabla 4

Tiempo de Respuesta del Sistema

Pruebas	Tiempo en segundos (desde Matlab a la mano robótica)		
	Comunicación a Arduino	Cierre de dedos	Apertura de dedos
Prueba 1	0,5	0,1	0,1
Prueba 2	0,3	0,1	0,1
Prueba 3	0,3	0,1	0,1

Nota. Elaboración propia.

Se realizó la evaluación del tiempo de respuesta del sistema, especialmente en cuanto a la comunicación del entre MATLAB y Arduino, así como la actuación de la mano robótica.

En las pruebas realizadas, se observó que la comunicación entre MATLAB y Arduino experimentó mejora, como se evidencia en la tabla, evidenciado una optimización en la transmisión de los datos. Es importante resaltar que, el tiempo de respuesta para la actuación de la mano robótica, tanto el cierre como la apertura, fue de manera inmediata, registrándose en un 0,1 segundo. Frente a esto, se puede decir que, estos tiempos mostrarían una eficiencia y eficacia del sistema en la interpretación y ejecución de comandos basados en señales EEG.

Conclusiones

Durante el desarrollo del proyecto, se demostró que es viable usar señales EEG para controlar un sistema robótico, en este caso, una mano robótica. Las pruebas realizadas mostraron una respuesta adecuada a los servomotores ante las señales decodificadas. En este sentido, se puede decir que, este proyecto indica un potencial importante para futuras aplicaciones en áreas como rehabilitación o asistencia a personas con movilidad limitada.

Uno de los desafíos presentados en este proyecto fue garantizar que el algoritmo de machine learning alcanzara un nivel de aprendizaje superior al 80% antes de la decodificación, teniendo en cuenta que, este nivel de precisión era esencial para garantizar una interpretación adecuada de las señales EEG, y, por ende, una respuesta efectiva en la mano robótica.

El uso de MATLAB y Arduino demostró ser una combinación eficiente para el procesamiento y decodificación de las señales EEG. No obstante, se reconoce que hay un margen para optimizar y simplificar el código, lo que podría aumentar la eficacia del sistema.

En futuros trabajos, sería interesante explorar la integración de sensores EEG más sofisticados y preciso, lo que podría permitir una decodificación aún más exacta de las intenciones del usuario.

La investigación presentada sienta las bases para el desarrollo de sistemas robóticos controlados por el cerebro, que podrían tener impacto en campos como la medicina, la rehabilitación y el cuidado asistido. Si bien este proyecto se centró en una mano robótica, las técnicas y metodologías empleadas podrían adaptarse para controlar otros sistemas robóticos.

Referencias Bibliográficas

- Alvarez, J. (s.f.). *Mano robotica*. <https://www.pinterest.com.mx/pin/295830269279520022/>
- Arbelaéz, M. A. (30 de julio de 2020). *La corteza motora: características y funciones*. La Mente es Maravillosa: <https://lamenteesmaravillosa.com/la-corteza-motora-caracteristicas-y-funciones/>
- Arduino.cl. (s.f.). *Arduino UNO [fotografía]*. cl: <https://arduino.cl/arduino-uno/>
- Blog de ITZEL. (12 de octubre de 2011). *Funciones del cerebro*. <https://blogdeitzel.wordpress.com/ciencia-cognitiva/funciones-del-cerebro/>
- DANE. (marzo de 2010). *Discapacidad*. <https://www.dane.gov.co/index.php/estadisticas-por-tema/salud/discapacidad>
- DANE. (2011). *En cuesta Nacional de Calidad de Vida*.
- Desarrollo, P. N. (2018). *Pacto por Colombia, pacto por la equidad*. <https://id.presidencia.gov.co/especiales/190523-PlanNacionalDesarrollo/documentos/BasesPND2018-2022.pdf>.
- Emotiv. (s.f.). *Validación de EMOTIV EPOC+ para extraer correlatos ERP del procesamiento emocional de rostros*. <https://www.emotiv.com/independent-studies/validation-of-emotiv-epoc-for-extracting-erp-correlates-of-emotional-face-processing/>
- Farfan, F. (enero de 2005). *Ejemplos de ritmos alfa, beta, theta y delta*. https://www.researchgate.net/publication/304349022_Control_Cerebral_de_Interfases_Analisis_Exploratorio_de_Tecnicas_Parametricas_Digitales_para_la_Deteccion_y_Cuantificacion_de_Estados_Mentales

Felipe Torres - Christian Sánchez - Kenneth Palacio. (21 de septiembre de 2014). Adquisición y análisis de señales cerebrales utilizando el dispositivo MindWave:

https://dspace.ucuenca.edu.ec/bitstream/123456789/21351/1/MATCH%2714_09_Torres%20et%20al.pdf

García, T. T. (2011). *Manual básico para enfermeros en electroencefalografía.* Enfermería

Docente:

<http://www.sspa.juntadeandalucia.es/servicioandaluzdesalud/huvvsites/default/files/revistas/ED-094-07.pdf>

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

Guittierez, R. C.-C.-J. (mayo de 2015). *Decodificación de imaginación motora en la señal de electroencefalografía mediante mapas auto-organizados.*

<https://www.redalyc.org/pdf/342/34247483006.pdf>

Hiwonder. (s.f.). *uHand: Los dedos de la mano robótica Hiwonder se mueven individualmente para el bricolaje del robot [fotografía].*

https://www.hiwonder.com/products/uhand?variant=32473888587863&_pos=4&_sid=7323aaed4&_ss=r

IEEE DataPort. (s.f.). *Señales cerebrales y el conjunto de datos de señales de movimiento de la mano correspondiente (BS-HMS-DATASET).* [//iee-dataport.org/open-access/brain-signals-and-corresponding-hand-movement-signals-dataset-bs-hms-dataset](https://iee-dataport.org/open-access/brain-signals-and-corresponding-hand-movement-signals-dataset-bs-hms-dataset)

Karin Correa Arana, A. V. (diciembre de 2016). *Prótesis de mano virtual movida por señales encefalograficas – EEG.*

https://www.researchgate.net/publication/307547691_Protesis_de_Mano_Virtual_Movida_Por_Senales_Encefalograficas_-_EEG/fulltext/57c837cd08ae28c01d51bbea/Protesis-de-Mano-Virtual-Movida-Por-Senales-Encefalograficas-EEG.pdf

López, A. (2015 de marzo de 2015). *¿Cuáles son las habilidades que se desarrollan en cada uno de los hemisferios de nuestro cerebro?* Blogs 20 minutos:

<https://blogs.20minutos.es/yaestaellistoquetodolosabe/cuales-son-las-habilidades-que-se-desarrollan-en-cada-uno-de-los-hemisferios-de-nuestro-cerebro/>

Medina, B., Sierra, J. E., & Barrios, A. (20 de febrero de 2018). *Técnicas de extracción de características de señales EEG en la imaginación de movimiento para el sistema BCI.*

Revista Espacios: <https://www.revistaespacios.com/a18v39n22/a18v39n22p36.pdf>

Minguez, J. (s.f.). *Tecnología de Interfaz Cerebro Computador.*

https://webdiis.unizar.es/~jminguez/Sesion001_UJI.pdf

OpenBCI. (s.f.). *Interfaces cerebro - computadora de código abierto.* <https://openbci.com/>

Physionet. (s.f.). <https://archive.physionet.org/cgi-bin/atm/ATM>

Promedic. (s.f.). *Electroencefalograma [Fotografía].* Neurología:

<https://promedic.com.gt/centro-diagnostico-guatemala/electroencefalograma/>

Sabater, V. (25 de mayo de 2022). *Tipos de ondas cerebrales: Delta, Theta, Alfa, Beta y Gamma.*

La mente es maravillosa: <https://lamenteesmaravillosa.com/tipos-de-ondas-cerebrales/>

Sánchez, L. P. (2022). *Breve historia de la electroencefalografía.* Acta Neurológica Colombiana:

http://www.acnweb.org/acta/2002_18_2_104.pdf

Torres, L. M. (2018). *Implementación de un prototipo de prótesis de mano usando el método BCI no invasivo a través de estímulos endógenos.*

<http://dspace.esPOCH.edu.ec/bitstream/123456789/9202/1/108T0248.pdf>

UNIT ELECTRONICS. (s.f.). *PCA9685 16 Canales PWM I2C 12-Bit [fotografía].* uelectronics:

<https://uelectronics.com/producto/pca9685-16-canales-pwm-12-bit-i2c-servo-sg90-mg90s-mg995/>

Apéndices

Apéndice A

Código General en MATLAB

```

clc; close all; clear;

addpath ('./Parsed Data-set'); %Configuración carpeta

deseadas =

[2,3,4,5,6,12,13,14,15,16,57,58,59,60,61,67,68,69,70,71,7,8,9,10
,11,62,63,64,65,66,17,18,19,20,21,52,53,54,55,56]; %Modificar
para seleccionar derivaciones

fs = 128; %Frecuencia de muestreo

data_info = struct2cell(dir(strcat(pwd, '\Parsed Data-set')));
%Acceder información de la carpeta

folders_names = data_info(1,3:end); %Nombres de carpetas dentro
de Parsed Data-set

counter = 1;

nms = {};

for i = 1:length(folders_names) %Acceder a Test & Train%
    addpath(strcat('./Parsed Data-set\', folders_names{i}));
    test_info = struct2cell(dir(strcat(pwd, '\Parsed Data-set\',
folders_names{i}))); %extraer informacion carpeta
    test_folders = test_info(1,3:end); %acceder nombre de las
carpetas test&train

```

```

for j = 1:length(test_folders)
    addpath(strcat('\Parsed Data-set\' , folders_names{i},
'\', test_folders{j}));
    move_info = struct2cell(dir(strcat(pwd, '\Parsed Data-
set\' , folders_names{i}, '\', test_folders{j})));
    testmove_info = move_info(1,3:end); %acceder a los datos
csv

    for k = 1:length(testmove_info) %mostrar la señal a la
que se accede (una derivación)
        signal = readmatrix(testmove_info{k}); %leer csv
        name = testmove_info{k}; %Nombre del registro para
posteriormente crear archivo de características
        name = name(1:end-4);
        opts = detectImportOptions(name);
        variableNames = opts.SelectedVariableNames(2:71);
        seleccionadas = seleccion(signal, deseadas);
%Función para seleccionar derivaciones
        cs = length(deseadas); %Cantidad de derivaciones
seleccionadas
        allFeatures1 = [];
        allFeatures2 = [];

```

```

        for s = 1:cs %Ciclo para extraer características de
cada derivación

            senal = seleccionadas(:, s); %Señal seleccionada
            senal1 = senal(1:fs*9); %Primera ventana de 9
segundos

            senal2 = senal((fs*9)+1:fs*14); %Segunda ventana
de 5 segundos

            features1 = caracteristicasmx(senal1, j);
%Función que devuelve las características de cada derivación
            features2 = caracteristicasmx(senal2, j);

            allFeatures1 = [allFeatures1 features1]; %
Concatena características de todos los canales para ventana 1
            allFeatures2 = [allFeatures2 features2]; %
Concatena características de todos los canales para ventana 2
        end

        if i == 1 && j == 1 && k == 1

            Mx = allFeatures1;

            ta = size(Mx);

            ta = ta(1);

            Mx(ta+1, :) = allFeatures2;

        else

```

```
        ta = size(Mx);
        ta = ta(1);
        Mx(ta+1, :) = allFeatures1;
        ta = size(Mx);
        ta = ta(1);
        Mx(ta+1, :) = allFeatures2;

    end

    nms{counter, 1} = strcat(name, '_w1');
    nms{counter + 1, 1} = strcat(name, '_w2');
    counter = counter + 2;

end

end

end

valor_absoluto_medio = Mx(:, 1);
energia = Mx(:, 2);
activity = Mx(:, 3);
movility = Mx(:, 4);
frecuencia_media = Mx(:, 5);
complexity = Mx(:, 6);
potencia = Mx(:, 7);
car1 = Mx(:, 8);
car2 = Mx(:, 9);
car3 = Mx(:, 10);
```

```

car4 = Mx(:, 11);
car5 = Mx(:, 12);
category = Mx(:, 13);
T = table(nms, valor_absoluto_medio, energia, activity,
movility, frecuencia_media, complexity, potencia, car1, car2,
car3, car4, car5, category);
path_of_signal = ('\Results\');
mkdir(path_of_signal);
path_of_signal = strcat('\Results\', 'Matrix');
mkdir(path_of_signal);
path_of_signal = strcat('\Results\', 'Matrix',
'\Matriz_de_caracteristicas.csv');
writetable(T, path_of_signal); %Variable donde guardamos la
tabla, nombre del archivo que se va a crear
% Mx ahora contiene las características de tus datos de
decodificación.
featuresMatrix = Mx(:, 1:end-1); % Suponiendo que la última
columna es la etiqueta/category.

```

Explicación general

Inicialización: Limpia la consola, cierra todas las figuras y borra todas las variables.

Configuración de la Ruta: Añade la carpeta 'Parsed Data-set' al camino para acceder a sus archivos.

Selección de Derivaciones: Define las derivaciones (canales) de interés en el vector 'deseadas'.

Frecuencia de Muestreo: Establece una frecuencia de muestreo de 128 Hz.

Acceso a Datos: Explora y accede a las subcarpetas y archivos dentro de 'Parsed Dataset'. Las estructuras de bucle for permiten navegar por la jerarquía de carpetas y extraer datos de archivos CSV.

Extracción de Características: Para cada archivo CSV, se seleccionan derivaciones de interés y se extraen características de dos ventanas temporales (una de 9 segundos y otra de 5 segundos) para cada derivación.

Almacenamiento de Características: Las características extraídas se almacenan en una matriz llamada 'Mx', donde cada fila representa un conjunto de características para una ventana y derivación específicas.

Creación de Tabla: Se crea una tabla con las características extraídas, etiquetadas por derivación y ventana. Esta tabla es posteriormente guardada en un archivo CSV llamado 'Matriz_de_caracteristicas.csv' en el directorio 'Results\Matrix'.

Extracción de Matriz de Características: Finalmente, se extrae una matriz de características sin la columna de etiquetas.

Apéndice B

Función del Código para Características

```

function [features]=caracteristicas(signal)

%% Caracteristicas temporales (estadisticas)

fs=128;%Frecuencia de muestreo

L=length(signal);%Longitud de la señal

%% Valor absoluto medio

vam=0;%%Inicializar una variable para asignarle posteriormente
el valor obtenido

for i=1:length(signal)

    vas=abs(signal(i)); %%Variable que cambia con cada ciclo
dependiendo del valor de la muestra

    vam=vam+vas; %Sumamos vas+vam

end

vam=vam/length(signal);%%Dividir entre el número de muestras

%% Energía total

en=0; %%Inicializar energía en 0

for i=1:length(signal)

    enm(i)=signal(i)^2;%%Cada muestra al cuadrado}

    en=en+enm(i); %%Sumar a la variable inicializada en cero

end

%% ZCR (Zero croos rate)

w=diff(signal>0);

```

```

zcr=sum(abs(diff(signal>0)))/length(signal); %%Obtener indicador
de cruces por cero

%% PARÁMETROS DE HJORTH

    %% Activity (representa la potencia de la señal de EEG)

%Obtener media
S=sum(signal);
med=S/length(signal);
act=0; %%Inicializar activity
for i=1:length(signal)
    val=(signal(i)-med)^2; %%Variable que cambia con cada ciclo
dependiendo del número de muestra
    act=act+val; %%Act = valor previo + valor actual
end
activity=act/length(signal);

%% Movility (representa la frecuencia media (frecuencia
dominante) de la señal de EEG)
var1=var(signal); %%Varianza de señal original
var2=var(diff(signal)); %%Varianza de primera derivada de la
señal con diff
divar=var2/var1; %%Cociente de las derivadas
movility=sqrt(divar); %%Obtener la raíz cuadrada del cociente

% Obtener frecuencia dominante

```

```

fremedia=movility*(fs/(2*pi)); %Multiplicar moviliti por la
frecuencia de muestreo dividida entre 2 pi
%% Complexity (Representa los cambios en la frecuencia de la
señal, compara la señal de eeg con una señal sinusoidal)
%Movility x'(t)
s1=diff(signal); %Primera derivada de la señal
s2=diff(s1); %Segunda derivada
var1=var(s1); %VArianza de primera derivada
var2=var(s2); %Varianza de segunda derivada
divar=var2/var1; %Cociente de varianzas
movility2=sqrt(divar);%Mvility de la primera derivada
complexity=movility2/movility;
%% Caracteristicas frecuenciales
%% Transformada de fourier (espectro de frecuencias)
Y=fft(signal); %%Obtener fft (tranfromada rápida de fourier)
P2 = abs(Y/L);%Obtener la parte absoluta
%Eliminar la dualidad o gráfica en espejo
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = fs*(0:(L/2))/L;%Obtener el vector de frecuencias
figure (1)
plot(f,P1) %Mostrar espectro
title('Single-Sided Amplitude Spectrum of X(t)')

```

```

xlabel('f (Hz)')
ylabel('|P1(f)|')

%% PSD

%La Densidad Espectral de Potencia (siglas DEP, en inglés Power
Spectral Density) de una señal
%es una función matemática que da a conocer la distribución de
la potencia de dicha señal sobre
%las distintas frecuencias en donde está formada. Así, se puede
establecer el rango de frecuencias
%donde se concentran las variaciones de potencia. La observación
del comportamiento de señales en el
%dominio de la frecuencia resulta de gran ayuda, ya que se
pueden discriminar las variaciones más
%fácilmente que en el dominio del tiempo, esto permite la
comparación entre dos grupos poblacionales,
%y detectar una variación en el comportamiento del parámetro
estudiado.

figure(2)

Hpsd = dspdata.psd(P1(1:length(P1)/2), 'fs', fs);

%(espectro(1:mitad de la longitud del espectro, 'fs', fs)

plot(Hpsd)

%% Espectrograma

%

figure(3)

```

```

spectrogram(signal,600,10,fs,fs,'xaxis')%%Espectrograma
(señal,longitud de ventana (tiempo),noverlap,cuantas frecuencias
va a considerar el gráfico,frecuencia de muestreo
%% Autoregresión
signal=double(signal);%Convertir a double para implementar Varm
%Obtener señal transpuesta
%%Para varm, datos tipo double y vector columna
% figure
% plot(signal)
j=varm(1,5);%Crear vector de coeficientes
y=estimate(j,signal);%estimar los coeficientes de la
autorregresion estimate(vector creado con varm, señal)
co=y.AR; %Obtener coeficiente guardados en la celda ar de la
estructura y
co=cell2mat(co);%Convertir en vector de variable double
(conveniencia)
features=[vam,en,zcr,activity,movility,fremedia,complexity,co];%
Vector de características de cada derivación

```

Explicación de la función en MATLAB

Inicialización: La función inicia estableciendo la frecuencia de muestreo (128 Hz) y calculando la longitud de la señal.

Características Temporales:

Valor Absoluto Medio (VAM): Calcula el promedio de los valores absolutos de las muestras en la señal.

Energía Total: Calcula la suma de los cuadrados de las muestras.

Zero Cross Rate (ZCR): Mide la tasa a la que la señal cambia de positivo a negativo y viceversa.

Parámetros de Hjorth:

Activity: Representa la potencia de la señal de EEG.

Movility: Indica la frecuencia dominante de la señal de EEG.

Complexity: Compara los cambios en la frecuencia de la señal con una señal sinusoidal.

Características Frecuenciales:

Transformada de Fourier (FFT): Representa el espectro de frecuencias de la señal.

Densidad Espectral de Potencia (PSD): Muestra cómo la potencia de una señal se distribuye con diferentes frecuencias.

Espectrograma: Representa una visualización del espectro de frecuencias de una señal a medida que cambia con el tiempo.

Autorregresión: Estima los coeficientes de un modelo autorregresivo para la señal.

```
function [seleccionadas]=seleccion(grupo,deseadas)
a=length(deseadas);%Número de derivaciones deseadas
for i=1:a
    seleccionadas(:,i)=grupo(:,deseadas(i));%Crear una matriz
que contenga las derivaciones deseadas
end
```

Apéndice C

Código Normalización de la Matriz de Características

```

clc;close all;clear

%%Cargue de los datos

load('Features.mat')

%%Normalización de la matriz

X_max = max(X);

X_min = min(X);

X_norm = (X - X_min)./(X_max - X_min);

```

Explicación del código de normalización

Carga de Datos:

`load('Features.mat')`: Carga el archivo `Features.mat` en el espacio de trabajo. Este archivo debería contener una o más variables, una de las cuales se espera que sea `X`, dado que es utilizada en los siguientes pasos del código.

Normalización de la Matriz:

`X_max = max(X)`: Calcula el valor máximo de cada columna en la matriz `X` y lo almacena en `X_max`.

`X_min = min(X)`: Calcula el valor mínimo de cada columna en la matriz `X` y lo almacena en `X_min`.

`X_norm = (X - X_min)./(X_max - X_min)`: Esta línea normaliza cada valor en la matriz `X` según su columna. El resultado es una matriz `X_norm` donde los valores de cada columna están escalados en el rango `[0, 1]`.

Esta es una técnica común de preprocesamiento en análisis de datos y aprendizaje automático para asegurar que todas las características tengan el mismo rango y peso.

Apéndice D

Código para el Control de la Mano Robótica a Partir de la Decodificación

```
clc; clear; close all;

% Configuración de Arduino

s = serialport("COM5", 9600);

pause(2);

% Configuración inicial de servos

channels = [11, 12, 13, 14, 15];

openAngles = [90, 0, 0, 0, 0]; % Ángulo para la posición
abierta

closeAngles = [0, 180, 180, 180, 180]; % Ángulo para la posición
cerrada

% Cargar featuresMatrix

load('featuresMatrix.mat');

% Tomar sólo las primeras 12 columnas

featuresMatrix = featuresMatrix(:, 1:12);

% Convertir featuresMatrix a una tabla con nombres de columna
específicos

featuresTable = array2table(featuresMatrix, 'VariableNames',
{'column_1', 'column_2', 'column_3', 'column_4', 'column_5',
'column_6', 'column_7', 'column_8', 'column_9', 'column_10',
'column_11', 'column_12'});

% Cargar el Modelo

loadedData = load('compactModel.mat');
```

```
compactModel = loadedData.compactModel;

if isfield(compactModel, 'ClassificationEnsemble')
    model = compactModel.ClassificationEnsemble;
    predictions = predict(model, featuresTable);
else
    error('No se encontró el modelo de clasificación en
compactModel.mat');
end

% Visualización gráfica
figure;
h = animatedline;
xlabel('Tiempo');
ylabel('Predicción');
title('Predicción de Movimiento en Tiempo Real');
grid on;

% Visualización o simulación
lastPrediction = -1; % Variable para almacenar la última
predicción
for i = 1:length(predictions)
    % Actualización de la visualización gráfica
    addpoints(h, i, predictions(i));
    drawnow limitrate;
```

```

    if predictions(i) == 1 && lastPrediction ~= 1 % 1 es
movimiento

        disp('EEG: Movimiento detectado');

        for j = 1:length(channels)
            setServoAngle(s, channels(j), closeAngles(j),
false); % Ahora se cierra la mano

        end

        pause(5); % Pausa para que los servos se muevan

    elseif predictions(i) == 0 && lastPrediction ~= 0

        disp('EEG: Sin movimiento');

        for j = 1:length(channels)
            setServoAngle(s, channels(j), openAngles(j), false);

% Ahora se abre la mano

        end

        pause(5); % Pausa para que los servos se muevan

    end

    lastPrediction = predictions(i);

end

% Función para enviar ángulo a Arduino

function setServoAngle(serialObj, channel, angle, withPause)

    if nargin < 4 % Si no se proporciona el argumento
'withPause'

        withPause = true; % Valor predeterminado

    end

```

```
cmd = sprintf('<%d,%d>', channel, angle);  
write(serialObj, cmd, "char");  
  
if withPause  
    pause(0.5);  
  
end  
  
end
```

Explicación del código de decodificación

Configuración de Arduino:

Configura un puerto serial para comunicarse con un Arduino conectado al puerto "COM5" a una tasa de baudios de 9600. Luego, pausa durante 2 segundos para darle tiempo al puerto para inicializar.

Configuración de Servos:

Define los canales de los servos y sus ángulos para las posiciones "abierta" y "cerrada".

Cargar y Preparar Datos:

Carga una matriz llamada featuresMatrix y toma solamente las primeras 12 columnas.

Convierte esa matriz en una tabla con nombres de columnas específicos para el procesamiento posterior.

Cargar y Utilizar Modelo:

Carga un modelo previamente entrenado (llamado compactModel) desde un archivo.

Verifica si el modelo cargado es del tipo 'ClassificationEnsemble'. Si lo es, utiliza el modelo para predecir en base a la tabla de características. Si no, muestra un error.

Visualización en Tiempo Real:

Crea una figura para mostrar la predicción en tiempo real.

Simula o visualiza la predicción en tiempo real. Si el modelo predice un "1", considera que hay un movimiento y cierra la mano (o activa los servos en la posición "cerrada"). Si predice un "0", abre la mano.

Función Auxiliar para Comunicarse con Arduino:

setServoAngle: Esta función envía comandos al Arduino para cambiar el ángulo de un servo específico. Utiliza el formato <canal,ángulo> para enviar el comando al Arduino.

Apéndice E

Código de Arduino

```
#include <Wire.h>

#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

void setup() {

  Serial.begin(9600);

  pwm.begin();

  pwm.setPWMFreq(60);

}

void loop() {

  char inData[20]; // Para almacenar los datos entrantes

  int index = 0;

  // Espera a que llegue el carácter de inicio '<'

  while (Serial.available() && Serial.read() != '<') { }

  // Lee los datos hasta que llegue el carácter de fin '>'

  while (Serial.available() && index < 19) {

    char received = Serial.read();

    if (received == '>') break;

    inData[index] = received;

    index++;

  }

  inData[index] = '\0'; // Termina la cadena
```

```

// Extrae el canal y el ángulo de la cadena

int channel, angle;

sscanf(inData, "%d,%d", &channel, &angle);

if (channel >= 0 && channel <= 15 && angle >= 0 && angle <= 180) {

    setServoAngle(channel, angle);

    Serial.println("Command executed");

} else {

    Serial.print("Unknown command: ");

    Serial.println(inData);

}

}

void setServoAngle(int channel, int angle) {

    int pulse = map(angle, 0, 180, 150, 600);

    pwm.setPWM(channel, 0, pulse);

}

```

Explicación código Arduino

Inclusión de Bibliotecas:

Wire.h: Es la biblioteca para comunicarse con dispositivos I2C.

Adafruit_PWMServoDriver.h: Es la biblioteca para el driver PWM de Adafruit que permite controlar servos y otros dispositivos.

Inicialización de Objetos:

Se crea una instancia pwm del objeto Adafruit_PWMServoDriver.

Función setup():

Inicia la comunicación serial a 9600 baudios.

Inicia el driver PWM con el método `begin()`.

Establece la frecuencia del PWM a 60 Hz usando el método `setPWMFreq(60)`.

Función `loop()`:

Se define un array de caracteres `inData` para almacenar los datos que se recibirán a través de la comunicación serial.

Se espera a que llegue un carácter de inicio `<` desde la comunicación serial.

Luego, se leen los caracteres entrantes hasta que se encuentre el carácter de fin `>` o se alcance el límite del array (19 caracteres).

Se termina la cadena con un carácter nulo `\0`.

Se utiliza la función `sscanf` para extraer dos números enteros (canal y ángulo) de la cadena recibida.

Se verifica que los valores del canal y el ángulo estén dentro de los límites permitidos (0-15 para el canal y 0-180 para el ángulo).

Si los valores son válidos, se llama a la función `setServoAngle` para mover el servo al ángulo deseado y se envía un mensaje "Command executed" a través de la comunicación serial.

Si los valores no son válidos, se envía un mensaje "Unknown command" junto con la cadena recibida.

Función `setServoAngle(int channel, int angle)`:

Convierte el ángulo proporcionado (0-180 grados) a un valor de pulso (150-600) usando la función `map`.

Usa el método `setPWM` del objeto `pwm` para establecer el pulso PWM en el canal y ángulo especificados.

Apéndice F

Vídeo Sustentación Proyecto

<https://www.youtube.com/watch?v=N0cACaqpC1I&t=1s>