



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**

**FACULTAD DE INGENIERÍA  
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

Desarrollo de una interfaz cerebro-computadora de adquisición y procesamiento de señales EEG para proporcionar una vía de comunicación básica en personas con síndrome de Angelman mediante el asistente personal inteligente Alexa.

**Trabajo de Titulación para optar al título de Ingeniería en Electrónica y Telecomunicaciones**

**Autores:**

Edison José Brasales Amores

Bryan Fernando Proaño Mariño

**Tutor:**

Mgs. José Luis Jinez Tapia

**Riobamba, Ecuador. 2025**

## DERECHO DE AUTORIA

Yo, Bryan Fernando Proaño Mariño con cédula de ciudadanía **0605832682** y Edison José Brasales Amores con la cédula de ciudadanía **0550022974**, autores del trabajo de investigación titulado: **Desarrollo de una interfaz cerebro-computadora de adquisición y procesamiento de señales EEG para proporcionar una vía de comunicación básica en personas con síndrome de Angelman mediante el asistente personal inteligente Alexa**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Así mismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor de la obra referida será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 16 de enero del 2025.



Bryan Fernando Proaño Mariño

C.I: 0605832682



Edison José Brasales Amores

C.I: 0550022974

## DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Desarrollo de una interfaz cerebro-computadora de adquisición y procesamiento de señales EEG para proporcionar una vía de comunicación básica en personas con síndrome de Angelman mediante el asistente personal inteligente Alexa**, presentado por Edison José Brasales Amores, con cédula de identidad número 0550022974 y Bryan Fernando Proaño Mariño, con cédula de identidad número 0605832682, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 22 de enero de 2025.

**Mgs. Edgar Giovanni Cuzco Silva**

**MIEMBRO DEL TRIBUNAL DE GRADO**

Handwritten signature in blue ink, appearing to read 'Edgar Cuzco', written over a horizontal line.

**Dr. Klever Hernán Torres Rodríguez**

**MIEMBRO DEL TRIBUNAL DE GRADO**

Handwritten signature in blue ink, appearing to read 'Klever', written over a horizontal line.

**Mgs. José Luis Jinez Tapia**

**TUTOR**

Handwritten signature in blue ink, appearing to read 'José Luis Jinez Tapia', written over a horizontal line.

## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Desarrollo de una interfaz cerebro-computadora de adquisición y procesamiento de señales EEG para proporcionar una vía de comunicación básica en personas con síndrome de Angelman mediante el asistente personal inteligente Alexa**, presentado por Edison José Brasales Amores, con cedula de identidad número 0550022974 y Bryan Fernando Proaño Mariño, con cédula de identidad número 0605832682, bajo la tutoría de Mgs. José Luis Jinez Tapia; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 22 de enero del 2025.

**Presidente del Tribunal de Grado**

**PhD. Daniel Antonio Santillan Haro**



**Miembro del Tribunal de Grado**

**Mgs. Edgar Giovanni Cuzco Silva**



**Miembro del Tribunal de Grado**

**Dr. Klever Hernán Torres Rodríguez**





# CERTIFICACIÓN

Que, **BRSALES AMORES EDISON JOSÉ** con CC: **0550022974**, estudiante de la Carrera **ELECTRÓNICA Y TELECOMUNICACIONES, NO VIGENTE**, Facultad de **INGENIERÍA** ha trabajado bajo mi tutoría el trabajo de investigación titulado "**DESARROLLO DE UNA INTERFAZ CEREBRO-COMPUTADORA DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑALES EEG PARA PROPORCIONAR UNA VÍA DE COMUNICACIÓN BÁSICA EN PERSONAS CON SÍNDROME DE ANGELMAN MEDIANTE EL ASISTENTE PERSONAL INTELIGENTE ALEXA**", cumple con el 6 %, de acuerdo al reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 16 de enero de 2025



Escaneé este código QR por:  
**JOSE LUIS JINEZ  
TAPIA**

---

Mgs. José Luis Jinez Tapia  
**TUTOR TRABAJO DE INVESTIGACIÓN**





# CERTIFICACIÓN

Que, **PROAÑO MARIÑO BRYAN FERNANDO** con CC: **0605832682**, estudiante de la Carrera **ELECTRÓNICA Y TELECOMUNICACIONES, NO VIGENTE**, Facultad de **INGENIERÍA** ha trabajado bajo mi tutoría el trabajo de investigación titulado " **DESARROLLO DE UNA INTERFAZ CEREBRO-COMPUTADORA DE ADQUISICIÓN Y PROCESAMIENTO DE SEÑALES EEG PARA PROPORCIONAR UNA VÍA DE COMUNICACIÓN BÁSICA EN PERSONAS CON SÍNDROME DE ANGELMAN MEDIANTE EL ASISTENTE PERSONAL INTELIGENTE ALEXA**", cumple con el 6 %, de acuerdo al reporte del sistema Anti plagio **COMPILATIO**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 16 de enero de 2025



JOSE LUIS JINEZ  
TAPIA

Mgs. José Luis Jínez Tapia  
**TUTOR TRABAJO DE INVESTIGACIÓN**

## **DEDICATORIA**

*Dedico este proyecto de investigación a la memoria de mi querido abuelito, cuya sabiduría, amor y ejemplo de esfuerzo y perseverancia han sido una fuente constante de inspiración en mi vida. A mi familia, especialmente a mis padres, por su apoyo incondicional y por enseñarme el valor del trabajo duro y la dedicación.*

*A mis profesores y asesores, por su guía, paciencia y conocimientos que fueron fundamentales para el desarrollo de esta investigación. A todos aquellos que, de una u otra manera, contribuyeron con su apoyo, inspiración y confianza para hacer posible este proyecto, les estaré siempre agradecido.*

***Edison José Brasales Amores***

## **DEDICATORIA**

*A mi abuelita y a mi madre, quienes con su amor, fortaleza y sabiduría han sido el pilar fundamental de mi vida. A mi abuelita, por enseñarme los valores de la humildad, la paciencia y la perseverancia, y a mi madre, por ser mi guía de superación y mi ejemplo de esfuerzo incondicional. Todo lo que he logrado es gracias a su apoyo constante y su confianza en mí.*

*A mis amigos, compañeros de vida y de sueños, que con su ánimo, comprensión y compañía hicieron de este camino sea un trayecto más llevadero.*

***Bryan Fernando Proaño Mariño***



## **AGRADECIMIENTO**

*En primer lugar, agradezco a Dios por haberme dado la fortaleza, la sabiduría y la perseverancia necesarias para culminar este proyecto de investigación. Su guía y bendiciones fueron fundamentales en cada paso de este proceso, brindándome esperanza y motivación incluso en los momentos más difíciles.*

*Así mismo, expreso mi más sincero agradecimiento a mi familia, especialmente a mi padre, por su apoyo constante. A mi abuelito, cuya memoria ha sido un ejemplo de esfuerzo y dedicación que me inspiró a seguir adelante. A Daya, por su motivación y apoyo inquebrantable, y a mi tutor, José Jinez, por su valiosa guía y conocimientos, que fueron fundamentales para el desarrollo de esta investigación. A todos los que contribuyeron a este logro, les estaré eternamente agradecido.*

***Edison José Brasales Amores***

## **AGRADECIMIENTO**

*A mi abuelita y a mi madre, quienes han sido el pilar fundamental en mi vida. A mi abuelita Leonor Puzma, por su amor incondicional, su sabiduría y las lecciones de vida que siempre llevaré conmigo. A mi madre Cecilia Mariño, por su esfuerzo incansable, sus sacrificios. A mis hermanos que gracias a su fe en mí me ha motivado a superar cada obstáculo y a alcanzar este logro. A Daya por su amor y motivación diaria para poder seguir en las adversidades.*

*A mis amigos, cuya compañía y palabras de aliento fueron esenciales durante este proceso. A mis profesores y sobre todo a mi tutor José Jinez, por su paciencia, su guía y los conocimientos que me brindaron a lo largo de esta experiencia. Su dedicación y apoyo han sido piezas clave para que este proyecto sea una realidad. Finalmente, a todas las personas que, de manera directa o indirecta, contribuyeron a mi formación y al desarrollo de esta investigación.*

*Con todo mi agradecimiento*

***Bryan Fernando Proaño Mariño***

# ÍNDICE GENERAL

DERECHO DE AUTORIA	
DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL	
CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL	
CERTIFICADO ANTIPLAGIO	
DEDICATORIA	
ÍNDICE GENERAL	
ÍNDICE DE TABLAS	
ÍNDICE DE FIGURAS	
RESUMEN	
ABSTRACT	
CAPÍTULO I. INTRODUCCIÓN .....	17
1.1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN.....	19
1.1.1. Planteamiento del Problema .....	19
1.1.2. Justificación .....	20
1.2. OBJETIVOS .....	21
1.2.1. Objetivo General.....	21
1.2.2. Objetivo Específicos.....	21
CAPÍTULO II. MARCO TEÓRICO .....	22
2.1. El cerebro humano .....	22
2.1.1. Las neuronas .....	22
2.1.2. Sustancia gris y Sustancia blanca.....	23
2.1.3. Propiedades de las ondas .....	24
2.2. Electroencefalograma (EEG).....	24
2.2.1. Electroodos .....	25
2.2.2. Generación de las señales EEG.....	26
2.3. Procesamiento digital de señales .....	26
2.3.1. Filtrado Butterworth.....	26
2.3.2. Filtrado de mediana .....	26
2.3.3. Filtrado Pasabanda .....	27
2.4. Extracción de características de señales EEG .....	27
2.4.1. Transformada rápida de Fourier.....	27
2.4.2. Transformada Wavelet.....	27
2.5. Clasificación de características .....	28
2.5.1. Red Neuronal Artificial (ANN).....	28
2.5.2. Máquina de soporte vectorial (SVM).....	30
2.5.3. Análisis discriminante lineal (LDA) .....	30
2.5.4. K-Nearest Neighbors (KNN).....	30
2.6. Interfaz Cerebro-Computadora (BCI).....	30

2.6.1.	Principios de funcionamiento de las BCI .....	31
2.6.2.	Componentes de un sistema BCI .....	31
2.7.	Integración con Asistentes Personales.....	32
2.8.	Machine Learning .....	32
2.8.1.	Tipos de aprendizaje.....	32
2.8.2.	Etapas para el Desarrollo de un Modelo de Machine Learning.....	32
2.8.3.	Lenguajes de desarrollo de Machine Learning .....	33
CAPÍTULO III. METODOLOGÍA .....		35
3.1.	Tipo de investigación .....	35
3.2.	Diseño de investigación.....	35
3.3.	Operacionalización de variables .....	35
3.4.	Técnicas de recolección de datos.....	35
3.5.	Población de estudio y tamaño de muestra.....	36
3.5.1.	Población de estudio.....	36
3.5.2.	Tamaño muestra .....	36
3.6.	Hipótesis .....	37
3.6.1.	Hipótesis nula.....	37
3.6.2.	Hipótesis alternativa.....	37
3.7.	Métodos de análisis.....	37
3.8.	Procedimiento y análisis.....	37
3.8.1.	Fase 1: Paciente y adquisición de señales EEG.....	38
3.8.2.	Fase 2: Procesamiento y filtrado.....	40
3.8.3.	Fase 3: Extracción de características.....	41
3.8.4.	Fase 4: Clasificación de señales.....	42
3.8.5.	Fase 5: Integración con plataforma de Alexa.....	44
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN .....		49
4.1.	Resultados.....	49
4.2.	Discusión .....	50
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....		52
5.1.	Conclusiones .....	52
5.2.	Recomendaciones.....	53
BIBLIOGRAFÍA .....		54
ANEXOS.....		58

## ÍNDICE DE FIGURAS

<b>Figura 1.-</b> Esquema del cerebro humano y sus partes [17].	22
<b>Figura 2.-</b> Estructura general de una neurona [19].	23
<b>Figura 3.-</b> Sustancia Blanca y Sustancia Gris en la Médula Espinal y en el Cerebro [20].	23
<b>Figura 4.-</b> Frecuencias en señales EEG [21].	24
<b>Figura 5.-</b> Electrodo de electroencefalografía [23].	25
<b>Figura 6.-</b> Posicionamiento de electrodos por el sistema internacional 10-20 [24].	25
<b>Figura 7.-</b> Arquitectura de una Red Neuronal: Capas de Entrada, Ocultas y de Salida [31].	28
<b>Figura 8.-</b> Red Neuronal de Capa Única con Conexiones y Pesos Asociados [33].	29
<b>Figura 9.-</b> Red Neuronal Multicapa con Conexiones y Pesos Asociados [33].	29
<b>Figura 10.-</b> Arquitectura de una interfaz cerebro-computadora [38].	31
<b>Figura 11.-</b> Etapas para el Desarrollo de un Modelo de Machine Learning.	33
<b>Figura 12.-</b> Diagrama de bloques.	37
<b>Figura 13.-</b> Detección del dispositivo.	38
<b>Figura 14.-</b> Entorno del Software EmotivPRO.	39
<b>Figura 15.-</b> Resultados de la señal de los 14 sensores del Emotiv EPOC X.	39
<b>Figura 16.-</b> Archivo .csv donde se encuentran los datos de los 14 sensores.	40
<b>Figura 17.-</b> Código utilizado para el filtrado de la señal.	40
<b>Figura 18.-</b> Código utilizado para eliminar los artefactos.	41
<b>Figura 19.-</b> Código utilizado para la normalización de la señal.	41
<b>Figura 20.-</b> Código para detectar los picos más relevantes.	41
<b>Figura 21.-</b> Código para establecer el umbral.	41
<b>Figura 22.-</b> Herramienta de la Red Neuronal Artificial en MATLAB.	42
<b>Figura 23.-</b> Visualización que el modelo fue entrenado correctamente.	43
<b>Figura 24.-</b> Código que convierte las etiquetas numéricas a one-hot encoding.	43
<b>Figura 25.-</b> Interfaz gráfica de la interfaz cerebro-computadora.	44
<b>Figura 26.-</b> Ventana para seleccionar los archivos .csv con los datos del EEG.	44
<b>Figura 27.-</b> Campo de texto donde se ingresó el nombre del sensor y la frecuencia de muestreo.	45
<b>Figura 28.-</b> Resultado de la interfaz.	45
<b>Figura 29.-</b> Función para reproducir la clase predicha.	46
<b>Figura 30.-</b> Configuración de los comandos del asistente virtual Alexa.	47
<b>Figura 31.-</b> Ecosistema final del proyecto de investigación.	47
<b>Figura 32.-</b> Acondicionamiento al BCI.	48
<b>Figura 33.-</b> Resultado de la prueba de Kolmogórov-Smirnov obtenido con RStudio.	50
<b>Figura 34.-</b> Resultado de la prueba de T-Student obtenido con RStudio.	50
<b>Figura 35.-</b> Distribución de datos de las muestras.	51
<b>Figura 36.-</b> Media de los datos con dispositivo y sin dispositivo.	52

## ÍNDICE DE TABLAS

<b>Tabla 1.-</b> Tabla de Variable Independiente .....	35
<b>Tabla 2.-</b> Tabla de Variable Dependientes .....	35
<b>Tabla 3.-</b> Tamaño de la población 1 – sin interfaz cerebro-computadora.....	36
<b>Tabla 4.-</b> Tamaño de la población 1 – con interfaz cerebro-computadora.....	36
<b>Tabla 5.-</b> Tiempos diarios realizando las actividades asignadas con y sin dispositivo.....	49
<b>Tabla 6.-</b> Análisis descriptivo de muestras de las actividades sin y con dispositivo.....	49
<b>Tabla 7.-</b> Nivel de eficiencia del dispositivo.....	51

## RESUMEN

En esta investigación se detalla el desarrollo de una interfaz cerebro-computadora para adquirir y procesar señales de EEG. El asistente personal inteligente Alexa proporciona una vía básica de comunicación para personas con diagnóstico de síndrome de Angelman. Previo al desarrollo del proyecto se utilizaron métodos de última generación para analizar la factibilidad de esta investigación. En el marco teórico se detallan conceptos clave considerados para la creación de esta interfaz, haciendo posible la combinación del cerebro humano con una computadora, cumpliendo así cada uno de los objetivos planteados. Se realizó un análisis estadístico no paramétrico para comparar los tiempos de ejecución de una actividad de la vida diaria (actividades básicas) que realiza el paciente a través de actividades diarias. Las realizadas con el dispositivo concluyeron que se rechaza la hipótesis nula. Se acepta la hipótesis alternativa, dando paso al cálculo del nivel de eficiencia de la interfaz cerebro-computadora. Finalmente, se detallan las conclusiones y recomendaciones para futuras investigaciones.

**Palabras clave:** Síndrome de Angelman, Interfaz Cerebro-Computadora, Machine Learning, Emotiv Epor X.



## ABSTRACT

This research details the development of a brain-computer interface to acquire and process EEG signals. The Alexa intelligent personal assistant provides a basic communication pathway for people diagnosed with Angelman syndrome. Before the project's development, state-of-the-art methods were used to analyze this research's feasibility. The theoretical framework details key concepts considered for creating this interface, making possible the combination of the human brain with a computer, thus fulfilling each of the stated objectives. A non-parametric statistical analysis was made to compare the execution times of an activity of daily living (basic activities) performed by the patient through daily activities. Those performed with the device concluded that the null hypothesis was rejected. The alternative hypothesis is accepted, giving way to the calculation of the level of efficiency of the brain-computer interface. Finally, conclusions and recommendations for future research are detailed.

*Keywords:* Angelman syndrome, brain-computer interface, Machine Learning, Emotiv Epoc X.

**Reviewed by:**



**Lcda. Yesenia Merino Uquillas**

**ENGLISH PROFESSOR**

**0603819871**

## CAPÍTULO I. INTRODUCCIÓN

Una tecnología vanguardista se encargará de la conversión de señales de ondas cerebrales en comandos de telecontrol para dispositivos tecnológicos en el entorno doméstico. Esto se logrará mediante la implementación de un sistema de captación de actividad neuronal a través de electroencefalografía (EEG), lo cual conllevará mejoras significativas en la calidad de vida de individuos con discapacidades y personas de la tercera edad.

Una interfaz cerebro-computadora (BCI) es un sistema de comunicación que supervisa la actividad cerebral, mientras identifica y traduce en comandos específicos las características neuronales que reflejan las intenciones del usuario. Estos comandos permiten el control de dispositivos tecnológicos [1].

Tareas tan simples como activar la radio, encender las luces o regular la calefacción en el hogar pueden resultar insuperables para numerosas personas con discapacidad. Con el propósito de eliminar estas barreras, ha surgido el proyecto denominado "Interfaz cerebro - computadora (BCI) para entrenamiento cognitivo y asistencia domótica contra los efectos de posibles discapacidades o enfermedades.

Uno de los síndromes en los cuales nos enfocaremos es en el síndrome de Angelman el cual, en 1965, el médico inglés Dr. Harry Angelman realizó la primera descripción de tres niños con características similares, las cuales ahora se conocen como el síndrome de Angelman, el doctor observó que todos presentaban rigidez muscular, dificultades en la marcha, ausencia de habla, episodios de risa excesiva y convulsiones [2].

En 1987, la doctora Ellen Magenis, perteneciente al Centro de Ciencia de Salud de Oregón, hizo un hallazgo significativo al identificar a niños con microdeleciones en el cromosoma 15, que inicialmente se creía que padecían el síndrome de Prader-Willi (SPW) sin embargo, estos pacientes presentaban convulsiones y un retraso severo en su desarrollo, características inesperadas en el síndrome de Prader-Willi de manera rápida se determinó que estas microdeleciones en el cromosoma 15 provenían de la madre, a diferencia del síndrome de Prader-Willi donde la deleción siempre se encuentra en el cromosoma 15 de origen paterno dicho descubrimiento facilitó una gran importancia, ya que allanó el camino para el esclarecimiento de diversos mecanismos que causaban el síndrome de Angelman, todos ellos provocados por una alteración en un gen ubicado en el cromosoma 15 [3].

El síndrome de Angelman (SA) es una condición poco común hereditaria que afecta alrededor de 1 de cada 24 000 recién nacidos [4]. La mayoría de los diagnósticos se realizan típicamente en el rango de edad de 2 a 5 años, ya que es durante este periodo cuando los comportamientos y características distintivas son más visibles [5].

Algunos de los dispositivos electroencefalografía (EEG) más frecuentemente empleados en investigaciones sobre aplicaciones de una interfaz cerebro-computadora (BCI) basadas en electroencefalografía (EEG) son el Emotiv EPOC de Emotiv (San Francisco, CA, EE. UU.), el Quik-Cap de Compumedics Neuroscan (Charlotte, NC, EE. UU.) y el MindWave de

NeuroSky (San José, CA, EE. UU.) [6]. Se registra la actividad eléctrica del cerebro utilizando electrodos que se colocan en el cuero cabelludo o se implantan en la superficie de la corteza cerebral, además la señal capturada se prepara para su procesamiento posterior a través de etapas de amplificación, filtrado y conversión a formato digital [7].

Los métodos utilizados para la selección de características relevantes en señales son las electroencefalografías (EEG) para las interfaces cerebro-computadora (BCI) pueden ser clasificados en los siguientes grupos: los métodos que sustraen información temporal de la señal; métodos que extraen información frecuencial; métodos híbridos, basados en representaciones tiempo-frecuencia, que aprovechan tanto la información temporal y frecuencial [8].

Se pueden identificar ciertos métodos específicos que sobresalen son el análisis en el dominio del tiempo y frecuencia, filtros adaptados, potencia espectral, Common Spatial Patterns (CSP), Transformada Rápida de Fourier (FFT), Modelos Autorregresivos (AR), Transformada Discreta de Wavelet (DWT), Transformada Discreta Wavelet Packet (DWPT), Entropía, Transformada de Hilbert, Redes Neuronales y difusas, y combinaciones entre ellas [9].

En investigaciones y proyectos recientes se evidencia la efectividad en el campo de las interfaces cerebro-computadora como se demostró en Ecuador en el año 2021 el estudiante Eddy Fabián Corrales Bastidas en su proyecto de posgrado con el nombre de “Desarrollo de un Sistema BCI basado en redes Neuronales y Movimientos de la Cabeza para el Manejo de un Ordenador” [10], los resultados fueron eficientes para el objetivo establecido y se debe considerar el número de canales y sensores que capture el dispositivo EEG.

En 2015, estudiantes del Departamento de Eléctrica y Electrónica de la ESPE desarrollaron un proyecto de titulación denominado “Sistema de Control de Brazo Robótico Mediante Ondas Cerebrales Desarrollado en Software Libre para Asistencia a Personas con Capacidades Especiales” [11]. Este proyecto se enfocó en el uso de ondas cerebrales para controlar dispositivos robóticos, facilitando la asistencia a personas con capacidades especiales.

De manera similar, en 2021, Walter David Cunalata Velasco, de la Universidad Técnica de Ambato, presentó su trabajo de titulación “Sistema de Manipulación y Monitoreo del Robot SCORBOT Mediante Señales Electroencefalográficas (EEG)” [12]. En su estudio, el autor destacó que el sistema BCI basado en EEG es capaz de operar con fluidez y un correcto funcionamiento, lo que lo convierte en una opción eficiente y viable.

Aunque las interfaces cerebro-computadora (BCI) han avanzado considerablemente, todavía enfrentan desafíos importantes: la precisión y fiabilidad de estos sistemas son limitadas; además, los dispositivos actuales presentan barreras significativas como el alto costo y la escasa accesibilidad. Por otra parte, aún desconocemos muchos procesos sobre cómo el

cerebro humano interpreta y organiza la información, lo cual dificulta en gran medida el desarrollo de sistemas BCI más efectivos y sofisticados.

Las personas con síndrome de Angelman pueden beneficiarse del uso de dispositivos EEG, cuya función principal es captar y registrar las señales eléctricas producidas por el cerebro; además, permiten transformar esta actividad en comandos interpretables. Mediante un sistema BCI capaz de adquirir y procesar dichas señales, se busca proporcionar a los afectados una herramienta práctica y accesible que les permita comunicar sus pensamientos y necesidades, mejorando su calidad de vida y favoreciendo su integración social.

## **1.1.PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN**

### **1.1.1. Planteamiento del Problema**

El síndrome de Angelman es un trastorno genético que ocasiona un retraso en el desarrollo, dificultades en el habla y el equilibrio, discapacidad intelectual y, ocasionalmente, convulsiones. Las personas que padecen el síndrome de Angelman suelen exhibir una tendencia a sonreír y reír con frecuencia, además de poseer personalidades alegres y entusiastas. Los primeros signos del síndrome de Angelman generalmente se manifiestan como retrasos en el desarrollo, que suelen comenzar entre los 6 y 12 meses de edad [13].

Estudios actuales mencionan que el lenguaje oral, como medio de comunicación, se ve significativamente afectado en su totalidad en el síndrome de Angelman. Aproximadamente a los 4 años, la capacidad de pronunciar palabras básicas se limita a un máximo de 10 palabras, y estas pueden carecer de concordancia con el mensaje deseado [14].

En general, los individuos con síndrome de Angelman pueden comprender señales, pero para lograr una comprensión más básica y frecuente, se utilizan gestos faciales y manuales. Algunos niños con este síndrome pueden presentar comportamientos destructivos debido a su incapacidad para comunicarse o hablar [15].

Es aquí donde surge la necesidad de desarrollar una interfaz cerebro-computadora (BCI) basada en ondas cerebrales para procesar, analizar y transformarlas a señales de audio para personas con síndrome de Angelman así ayudar a su capacidad de interactuar con su entorno y de desarrollar relaciones sociales significativas.

Las interfaces cerebro-computadora (BCI) basadas en ondas cerebrales han demostrado ser una tecnología prometedora para permitir la comunicación y el control de dispositivos en personas con discapacidades motoras y cognitivas. Sin embargo, la implementación de una interfaz cerebro-computadora (BCI) efectiva para personas con síndrome de Angelman requiere de un enfoque específico y adaptado a las necesidades de esta población ya que requieren de otras personas para comunicarse y expresarse.

### **1.1.2. Justificación**

En los últimos años, ha habido notables avances en el campo de las interfaces cerebro-computadora (BCI), aprovechando la capacidad de adquisición y procesamiento de señales electroencefalográficas (EEG). Estos avances tecnológicos han abierto nuevas oportunidades para proporcionar una vía de comunicación alternativa en personas con síndrome de Angelman. La interfaz cerebro-computadora (BCI) basada en señales de electroencefalografías (EEG) ofrece ventajas significativas, ya que permite la comunicación directa entre el cerebro y un dispositivo externo, sin depender de los sistemas motores o del habla, que pueden estar afectados en las personas con este síndrome.

El desarrollo de una interfaz cerebro-computadora (BCI) basada en señales electroencefalográficas (EEG) para personas con síndrome de Angelman tiene el potencial de mejorar significativamente su capacidad de comunicación y su calidad de vida. Al abordar las necesidades de mejora de la comunicación, personalización, accesibilidad y confiabilidad, este proyecto busca ofrecer una solución innovadora y prometedora para un grupo de personas que enfrenta desafíos significativos en su comunicación diaria.

El enfoque principal de este proyecto de investigación es lograr el desarrollo de una interfaz cerebro-computadora (BCI) que permita establecer una vía de comunicación alternativa para personas con síndrome de Angelman. A través de la adquisición y procesamiento de señales electroencefalográficas (EEG), se busca establecer una conexión directa entre el cerebro y un dispositivo externo, como una computadora, para traducir las intenciones y pensamientos de los individuos con este síndrome en comandos y acciones comprensibles. El objetivo final es proporcionar a estas personas una forma efectiva y accesible de comunicarse, superando las barreras impuestas por las limitaciones motoras y del habla asociadas al síndrome de Angelman.

## **1.2.OBJETIVOS**

### **1.2.1. Objetivo General**

- Desarrollar una interfaz cerebro-computadora de adquisición y procesamiento de señales EEG para proporcionar una vía de comunicación básica en personas con síndrome de Angelman mediante el asistente personal inteligente Alexa.

### **1.2.2. Objetivo Específicos**

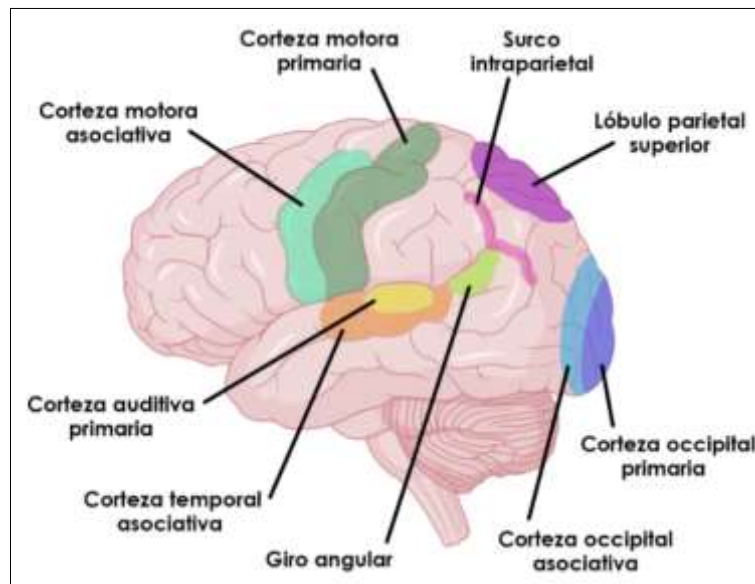
- Investigar y recopilar información sobre los dispositivos de captura de datos de señales electroencefalográficas (EEG) de bajo costo disponibles en el mercado, incluyendo sus características técnicas, precios y proveedores.
- Diseñar el sistema de adquisición de ondas cerebrales (EEG) basados en inteligencia artificial y machine learning.
- Implementar el sistema para analizar las variaciones de respuesta de las señales electroencefalográficas (EEG) que nos ayude a clasificar patrones específicos que correspondan a comandos para comunicación básica auditiva que el paciente requiera mediante el asistente personal Alexa.
- Evaluar el funcionamiento de la interfaz cerebro-computadora en diferentes condiciones y actividades, mediante pruebas de campo y análisis estadístico.

## CAPÍTULO II. MARCO TEÓRICO

A lo largo de este apartado, se dará a conocer las bases tecnológicas y científicas necesarias para poder entender de mejor manera este proyecto. Dado que en este abarcaremos distintas áreas de la ciencia como neurociencia, electrónica y el control por computadora, lo que hace necesario recalcar la teoría básica de este apartado.

### 2.1.El cerebro humano

Los cimientos de este proyecto son la detección de señales cerebrales del usuario a través del casco Emotiv EPOC X. el cerebro humano es el órgano más importante del sistema nervioso, siendo el responsable de recibir todos los impulsos eléctricos de los sentidos a través de la médula espinal, y de enviar respuestas a estos impulsos gestionando las emociones, imaginación, memoria e inteligencia. El cerebro humano posee diferentes zonas dedicadas a tareas específicas [16].



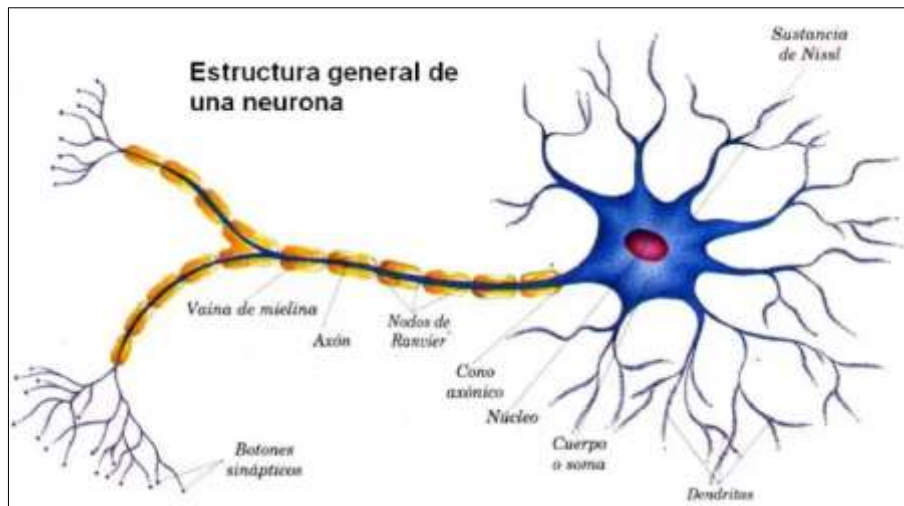
*Figura 1.- Esquema del cerebro humano y sus partes [17].*

#### 2.1.1. Las neuronas

Las neuronas se activan y establecen comunicación con otras a través de señales eléctricas. Esta interacción ocurre debido a la carga negativa presente en los axones y la carga positiva en el cuerpo de la célula piramidal. Sin embargo, el potencial postsináptico generado por una sola neurona es limitado, por lo que este proceso requiere la participación simultánea de un gran número de neuronas, generando así un campo eléctrico más considerable [18].

Estas tareas son realizadas gracias al trabajo coordinado de más de 100.000.000.000 neuronas, estas células son capaces de transmitir información a gran velocidad a través de señales eléctricas, permitiendo así que estas se comuniquen entre sí llamando este mecanismo como sinapsis. Una neurona tiene la siguiente estructura [16]:



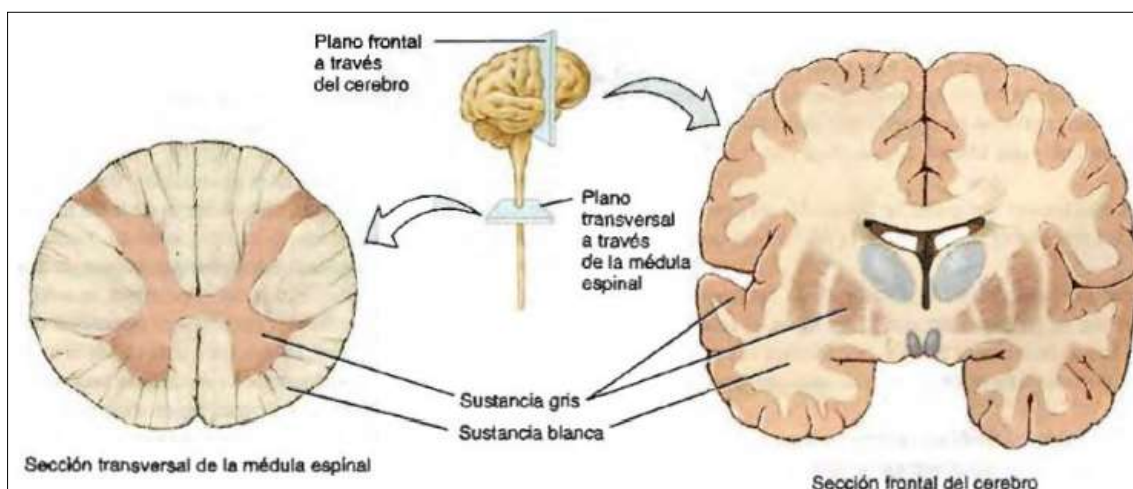


**Figura 2.- Estructura general de una neurona [19].**

- Núcleo
- Cuerpo o soma
- Dendritas
- Axón, encargado de conducir el impulso nervioso desde el soma a otra neurona.

### 2.1.2. Sustancia gris y Sustancia blanca

La vaina de mielina, que recubre gran parte del tejido nervioso, le otorga un color blanco característico, visible en cortes transversales del encéfalo y la médula espinal; por ello, las porciones de las neuronas con axones mielinizados se denominan sustancia blanca. En contraste, las áreas no mielinizadas, como los cuerpos neuronales y las terminaciones axónicas o dendríticas, se denominan sustancia gris. En el sistema nervioso central (SNC) y periférico (SNP), los tractos o vías suelen estar formados mayormente por sustancia blanca, mientras que la sustancia gris se agrupa en masas centrales, como en la médula espinal, donde está rodeada por sustancia blanca, o en núcleos dispersos dentro del encéfalo [20].

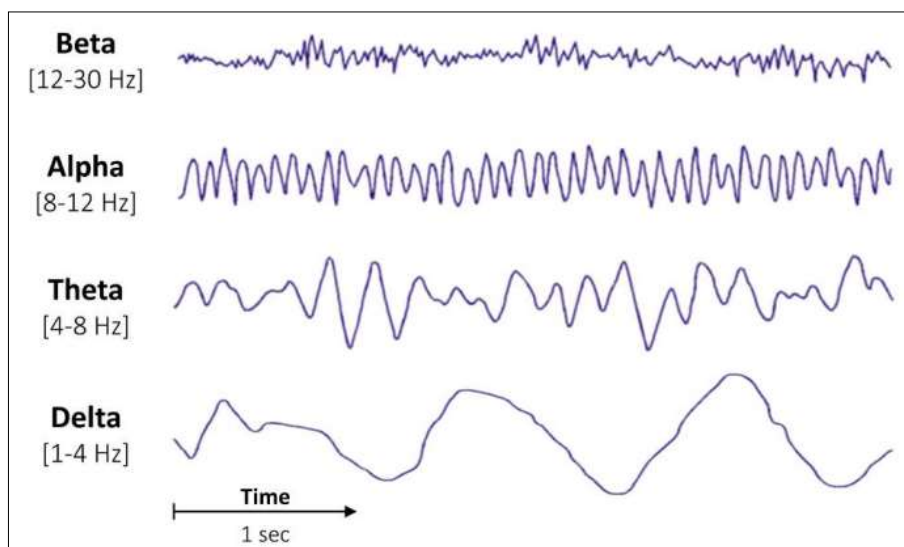


**Figura 3.- Sustancia Blanca y Sustancia Gris en la Médula Espinal y en el Cerebro [20].**

### 2.1.3. Propiedades de las ondas

Las ondas cerebrales, al igual que cualquier tipo de onda, se caracterizan por parámetros físicos como la amplitud y la frecuencia, los cuales son fundamentales para analizar la actividad cerebral. La amplitud, medida en microvoltios ( $\mu\text{V}$ ), indica el valor máximo que alcanza la onda y define el rango de sus valores. Por otro lado, la frecuencia, considerada la variable más relevante en el estudio de las ondas cerebrales, corresponde al número de oscilaciones completadas por unidad de tiempo y es la inversa del período, que representa el tiempo necesario para una oscilación completa [21].

En el análisis de señales EEG, el espectro de la onda se divide en rangos específicos, comúnmente denominados con letras griegas: delta (1 a 4 Hz), theta (4 a 8 Hz), alpha (8 a 12 Hz) y beta (12 a 30 Hz). Las frecuencias superiores a la banda beta, aunque han sido asociadas en estudios con fenómenos como la epilepsia, no son relevantes para este análisis y se excluyen. Del mismo modo, las frecuencias por debajo de 1 Hz se eliminan mediante un filtro de paso alto, ya que suelen estar contaminadas por ruido generado por el offset o desplazamiento [21].



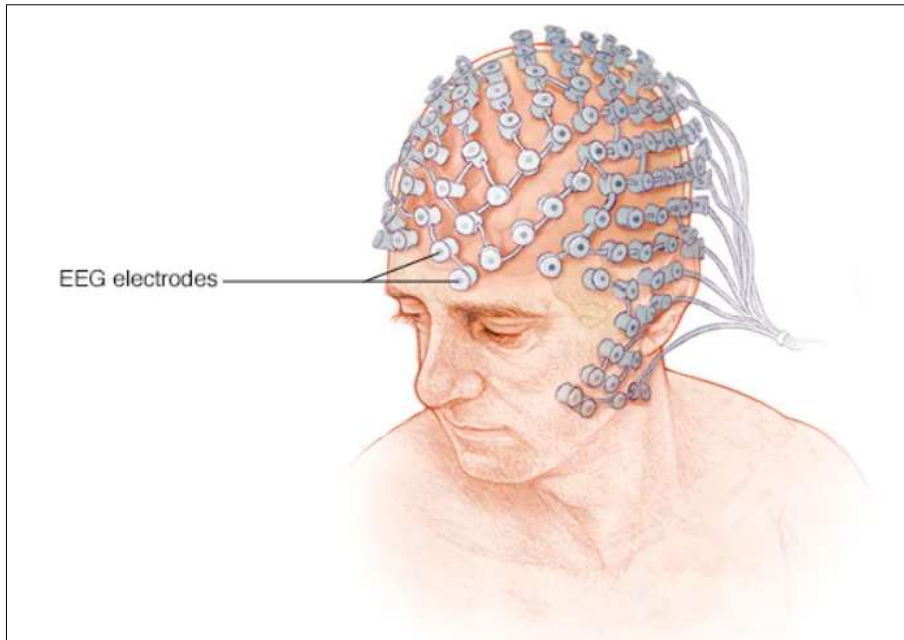
*Figura 4.- Frecuencias en señales EEG [21].*

## 2.2. Electroencefalograma (EEG)

La actividad eléctrica del cerebro se registra en diferentes escalas, cada una de las cuales refleja patrones oscilatorios relacionados con la activación funcional. En las escalas más pequeñas, los registros con microelectrodos profundos, implantados directamente en la corteza cerebral, permiten detectar tanto los picos de actividad de neuronas individuales como la actividad sincronizada de poblaciones neuronales, conocida como potencial de campo local (LFP), que representa la actividad de miles de células nerviosas [22].

En una escala mayor, los registros subdurales mediante electrocorticograma (ECoG) utilizan macroelectrodos clínicos para medir la actividad de millones de células. A nivel no invasivo, el electroencefalograma (EEG) y el magnetoencefalograma (MEG) registran la actividad

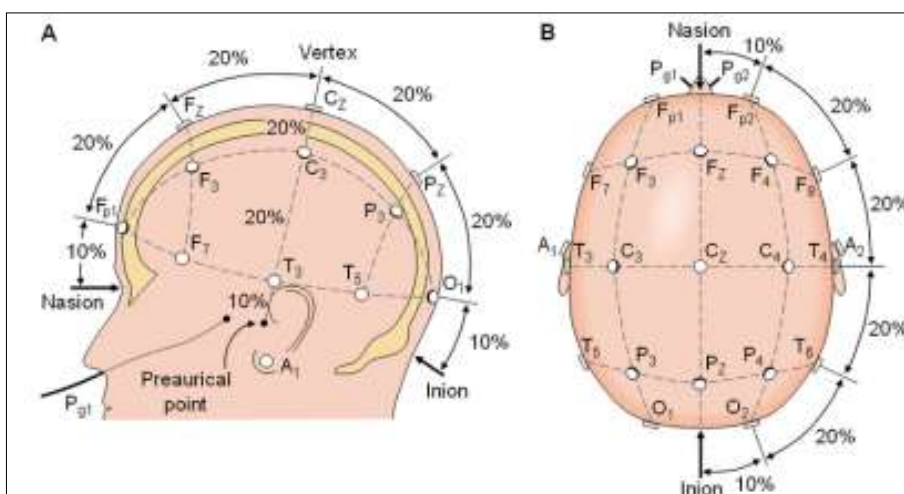
eléctrica a gran escala, capturando de manera simultánea la actividad de múltiples áreas corticales [22].



*Figura 5.- Electrodo de electroencefalografía [23].*

### 2.2.1. Electrodo

Para la extracción de datos, los electrodo deben colocarse siguiendo el sistema 10-20, como se muestra en la Figura 6. Estos electrodo, generalmente fabricados de plata, oro o acero inoxidable con recubrimiento de cloruro de plata, permiten estandarizar las mediciones realizadas por clínicos e investigadores. Sin embargo, el número de electrodo utilizados puede variar dependiendo de la especificidad de las mediciones o del diseño experimental [24].



*Figura 6.- Posicionamiento de electrodo por el sistema internacional 10-20 [24].*

### 2.2.2. Generación de las señales EEG

La corteza cerebral está compuesta principalmente por neuronas piramidales, y las señales EEG se generan por la sincronización local de miles de estas neuronas debido a la suma de sinapsis excitadoras en sus dendritas, lo que produce un dipolo entre la carga del soma y la de las dendritas. Esta actividad eléctrica surge del movimiento de iones como  $\text{Na}^+$ ,  $\text{K}^+$ ,  $\text{Ca}^{2+}$  y  $\text{Cl}^-$  a través de las membranas neuronales [25].

Las señales EEG, que son ondas eléctricas generadas por la corteza cerebral, presentan voltajes entre 0 y 300  $\mu\text{V}$  y frecuencias de 0.5 a 100 Hz cuando se miden con electrodos superficiales no invasivos. Entre las ventajas de este tipo de señales destacan su buena resolución temporal en milisegundos, la disponibilidad continua para su captura, la facilidad de grabación y el bajo costo de los equipos necesarios para su detección [25].

## 2.3. Procesamiento digital de señales

### 2.3.1. Filtrado Butterworth

Un filtro digital consiste en una serie de operaciones aplicadas a una señal de entrada para producir una señal de salida, reduciendo ciertas frecuencias y permitiendo el paso de aquellas que son de interés. En la sección denominada "Filtrado" de SigProU, se dispone de una variedad de filtros (pasa-bajas, pasa-altas, pasa-banda y rechaza-banda) con diferentes aproximaciones (Butterworth, Chebyshev I, Chebyshev II o inverso y Elíptico). Estos filtros se utilizan para procesar las señales EEG seleccionadas (ya sea una sola o varias), eliminando diversos tipos de artefactos o permitiendo únicamente el paso del rango de frecuencias deseado para su análisis posterior [26].

$$H(s) = \frac{1}{\prod_{k=1}^N (s - p_k)}$$

Donde  $N$ , es el orden del filtro y  $p_k$ , son los polos de la función de transferencia.

### 2.3.2. Filtrado de mediana

El filtro de mediana es un instrumento eficiente para el procesamiento de señales EEG, en particular para eliminar ruido impulsivo o artefactos, como los producidos por movimientos oculares o parpadeos. En contraste con otros filtros, el filtro de mediana es una técnica no lineal que funciona en una ventana de valores próximos a cada punto de la señal, organizando los valores vecinales y sustituyendo el valor central por el valor mediano. Este procedimiento posibilita eliminar el ruido sin alterar los rasgos relevantes de la señal EEG, tales como los picos y las oscilaciones, que resultan esenciales en el estudio. Al utilizar este filtro, se potencia la calidad de la señal, preservando los datos pertinentes, lo que simplifica su análisis clínico o de investigación.

$$y[n] = \text{mediana}(x[n - k], x[n - (k - 1)], \dots, x[n + k])$$

Donde  $x[n]$ , es la señal original,  $y[n]$ , es la señal filtrada (resultado del filtro),  $k$ , define el tamaño de la ventana del filtro, es decir, el número de puntos alrededor de  $x[n]$  que se usan para calcular la mediana.

### 2.3.3. Filtrado Pasabanda

Para eliminar las componentes de frecuencia no deseadas en las señales mediante un filtro pasabanda, se empleará el filtro digital de fase cero de MATLAB y el filtro de EEGLAB, que es un filtro de respuesta finita al impulso (FIR). La elección de estos dos tipos de filtros se debe a que el filtro de EEGLAB puede determinar automáticamente el orden del filtrado mediante métodos heurísticos, logrando así mejores resultados. Sin embargo, esto puede conllevar un mayor costo computacional y, por lo tanto, un tiempo de procesamiento más largo debido al mayor orden del filtro [27].

## 2.4.Extracción de características de señales EEG

La relevancia de las características en las señales EEG radica en la necesidad de identificar y utilizar atributos específicos para aplicaciones efectivas de Machine Learning. Estas características son patrones distintivos dentro de las señales EEG que pueden indicar estados mentales, respuestas neuronales o condiciones de salud. La selección adecuada de estas características es crucial, ya que permite que los modelos de Machine Learning se centren en la información más relevante y discriminativa. Este enfoque mejora la precisión de los modelos al reducir la complejidad y el ruido de los datos, facilitando la detección de patrones significativos y permitiendo predicciones más precisas [28].

### 2.4.1. Transformada rápida de Fourier

La serie de Fourier simboliza una función periódica como una secuencia infinita de funciones senoidales, cuyas frecuencias superan la frecuencia básica de la misma. Por lo tanto, al multiplicar la señal a procesar por una serie de señales senoidales de diferentes frecuencias, podemos reconocer las frecuencias que se encuentran en la señal en análisis. Teniendo en cuenta que la señal es discreta y que la frecuencia posee una magnitud limitada, se establece matemáticamente la transformada de Fourier [26]:

$$F(t) = \int_{-\infty}^{\infty} y(t) \cdot e^{-j2\pi ft} dt$$

Donde  $y(t)$  la señal en el dominio del tiempo,  $F(t)$  la señal en el dominio de la frecuencia,  $i$  la unidad imaginaria equivalente a  $\sqrt{-1}$ , y  $e^{-j2\pi ft}$  la función exponencial compleja.

### 2.4.2. Transformada Wavelet

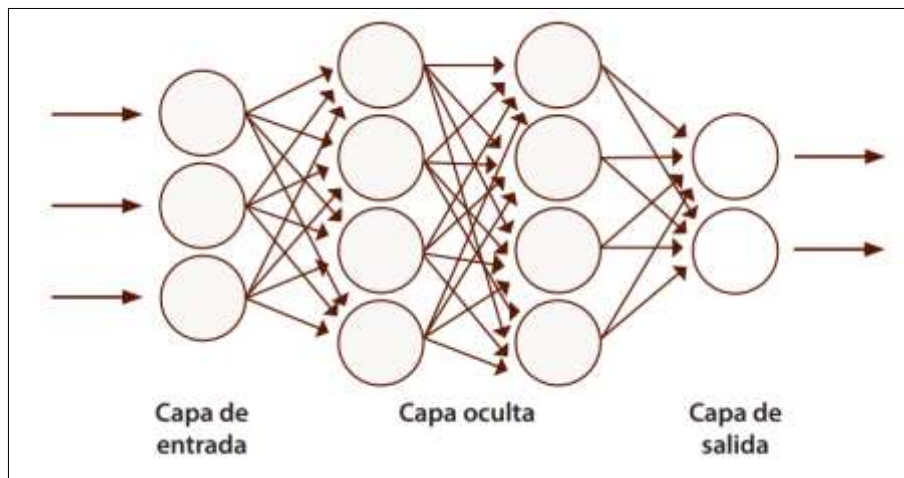
La Transformada Wavelet (WT) se utiliza para analizar señales y fenómenos en el dominio del tiempo y la frecuencia de manera simultánea. A diferencia de la Transformada de Fourier, que descompone una señal en sinusoides de diferentes frecuencias, la WT emplea funciones de onda conocidas como "wavelets", que tienen una duración finita tanto en el dominio del tiempo como en el de la frecuencia. Este método se usa comúnmente para separar la señal

de EEG en diferentes bandas de frecuencia y luego calcular descriptores de la señal, como componentes estadísticos, DE, PSD, entre otros [29].

## 2.5. Clasificación de características

### 2.5.1. Red Neuronal Artificial (ANN)

Una red neuronal artificial (ANN) es un modelo matemático computacional que consta de una capa de entrada, capas de salida y capas ocultas. Estas capas están compuestas por neuronas artificiales interconectadas a través de sinapsis, cada una con su propio peso. El funcionamiento de una ANN imita el comportamiento del cerebro humano, siendo capaz de aprender a través de la experiencia, precedida por un entrenamiento repetitivo [30].



*Figura 7.- Arquitectura de una Red Neuronal: Capas de Entrada, Ocultas y de Salida [31].*

La potencia de cálculo y la capacidad de procesamiento de las redes neuronales artificiales (ANN) se derivan de las numerosas conexiones entre las neuronas que las componen [32].

#### 2.5.1.1. Red neuronal de una capa

Cada entrada está vinculada a cada neurona artificial mediante un peso específico. En la práctica, algunas conexiones pueden eliminarse, e incluso pueden existir conexiones entre las entradas y salidas de las neuronas dentro de una misma capa. Generalmente, las redes más grandes y complejas ofrecen un rendimiento superior en términos de capacidad de cálculo en comparación con las redes más simples. Las configuraciones de estas redes pueden variar considerablemente, pero comparten una característica común: el ordenamiento de las neuronas en capas o niveles, similar a la estructura de capas que se observa en ciertas áreas del cerebro [32].



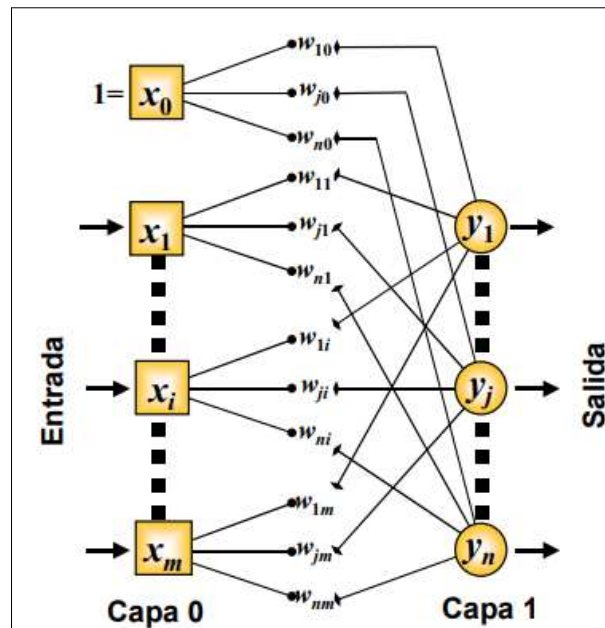


Figura 8.- Red Neuronal de Capa Única con Conexiones y Pesos Asociados [33].

### 2.5.1.2. Red neuronal multicapa

Las redes multicapa se componen de varias capas simples dispuestas en secuencia, donde la salida de una capa se convierte en la entrada de la siguiente. Se ha demostrado que las redes multicapa superan en rendimiento a las redes de una sola capa. Es importante destacar que la mejora que ofrecen las redes multicapa se debe a la función de activación no lineal entre las capas. Sin el uso de esta función no lineal, sería posible diseñar una red de una sola capa que fuera equivalente a una red multicapa [32].

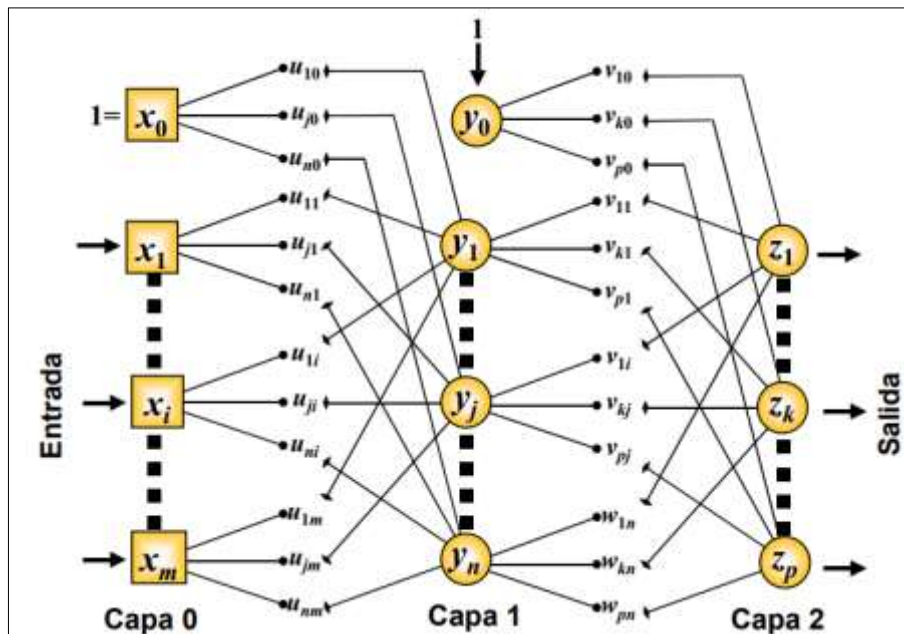


Figura 9.- Red Neuronal Multicapa con Conexiones y Pesos Asociados [33].



### 2.5.2. Máquina de soporte vectorial (SVM)

La máquina de soporte vectorial (SVM), cuyo nombre proviene de las siglas en inglés, se basan en la teoría del aprendizaje estadístico. Estos clasificadores funcionan discriminando información mediante la búsqueda de hiperplanos óptimos que logran separar puntos de datos pertenecientes a una clase dentro de un espacio que puede tener un número infinito de dimensiones (lo que se conoce como hiperespacio) [34].

El modelo tiene como objetivo separar una clase de otro conjunto de características correspondientes a una clase distinta. Una vez que se define el hiperplano, se le denomina plano de decisión, cuyo propósito es dividir un grupo de características de distintas clases y agruparlas mediante la definición de límites de decisión. Los clasificadores SVM buscan el hiperplano más adecuado para garantizar que el margen de separación entre las clases sea lo más amplio posible [34].

### 2.5.3. Análisis discriminante lineal (LDA)

Uno de los clasificadores más utilizado es el análisis discriminante lineal (LDA), es un clasificador supervisado que busca una combinación lineal de características que maximice la separación entre las clases. El LDA utiliza la información obtenida de la matriz de covarianza de cada clase y la matriz de covarianza combinada de todas las clases para calcular la combinación lineal óptima. Luego, se emplea esta combinación lineal para clasificar la señal EEG en una de las dos clases [35].

### 2.5.4. K-Nearest Neighbors (KNN)

Es un algoritmo de clasificación supervisada sencillo que asigna una etiqueta de clase a una nueva observación, basándose en las etiquetas de clase de sus  $k$  vecinos más cercanos en el espacio de características. Este algoritmo almacena todos los casos disponibles y clasifica los nuevos datos según una medida de similitud. En esencia, KNN clasifica una observación de acuerdo con la clasificación de sus vecinos más cercanos. Para un  $K \in \mathbb{N}$ , una observación se asigna a la clase  $j$  con la mayor frecuencia entre sus  $k$  vecinos más cercanos [36].

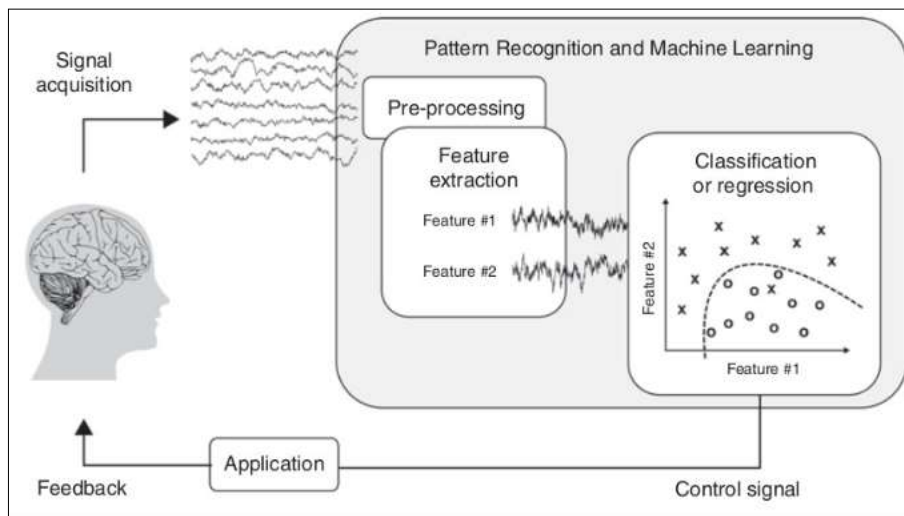
$$p_j(x) = \frac{K_j}{K}$$

Donde  $K_j$ , es el número de observaciones de la clase  $j$  dentro de los  $K$  vecinos más cercanos. Los KNN se obtienen calculando la distancia entre la observación a predecir y todas las observaciones del conjunto de datos de entrenamiento. Esta distancia puede calcularse utilizando diferentes métodos, como Euclídea, Manhattan, Minkowski, entre otros. Sin embargo, no existe un método estructurado para encontrar el mejor valor de  $K$ . Este se determina a través de un proceso de ensayo y error, asumiendo que los datos de validación son desconocidos [36].

## 2.6. Interfaz Cerebro-Computadora (BCI)

La Interfaz Cerebro-Computadora, conocida como BCI por sus siglas en inglés, es una interfaz asistida por computadora que facilita la interacción directa entre el cerebro y el

entorno de un individuo mediante actuadores conectados a la computadora. Este tipo de interfaz ha surgido para abordar la necesidad de establecer un canal de comunicación novedoso entre un sujeto y su entorno, sin depender de sus vías nerviosas o musculares [37].



*Figura 10.- Arquitectura de una interfaz cerebro-computadora [38].*

### 2.6.1. Principios de funcionamiento de las BCI

Las BCI funcionan mediante la detección y clasificación de patrones específicos en la actividad cerebral, los cuales se asocian con intenciones o pensamientos del usuario. Estos patrones pueden ser expuestos por estímulos externos (como el parpadeo o la imaginación motora) o ritmos cerebrales espontáneos (como los ritmos alfa o beta). Una vez que los patrones son detectados y clasificados, se traducen en comandos para controlar dispositivos externos, como computadoras, prótesis o sistemas de comunicación [38].

### 2.6.2. Componentes de un sistema BCI

Un sistema BCI común consta de varios componentes clave:

- **Adquisición de señales:** Consiste con la colocación de electrodos en el cuero cabelludo para registrar la actividad cerebral eléctrica [38].
- **Acondicionamiento de señales:** Incluye la amplificación, filtración y digitalización de las señales EEG para su posterior procesamiento [38].
- **Procesamiento de señales:** Involucra técnicas de procesamiento digital de señales para extraer características de los datos EEG, como cambios en la potencia espectral [38].
- **Clasificación:** Utiliza algoritmos de aprendizaje automático para interpretar las características extraídas y asociarlas con intenciones o comandos específicos del usuario [38].
- **Interfaz de control:** Traduce los comandos clasificados en acciones concretas para controlar dispositivos externos, como cursores en pantalla, prótesis o sistemas de comunicación [38].

## 2.7. Integración con Asistentes Personales

La integración de BCI con asistentes personales inteligentes, como Alexa, representa una innovación significativa. Esto no solo facilita la comunicación, sino que también permite a los usuarios realizar tareas cotidianas, como controlar dispositivos del hogar, mediante su actividad cerebral. La implementación de algoritmos de aprendizaje automático en el procesamiento de señales EEG es crucial para mejorar la precisión y la velocidad de respuesta de estas interfaces [39].

## 2.8. Machine Learning

El machine learning es la disciplina que se enfoca en crear algoritmos capaces de permitir a las computadoras llevar a cabo diversas tareas mediante el reconocimiento de patrones en grandes conjuntos de datos. Además, estos sistemas tienen la capacidad de optimizar su desempeño conforme reciben más información [40].

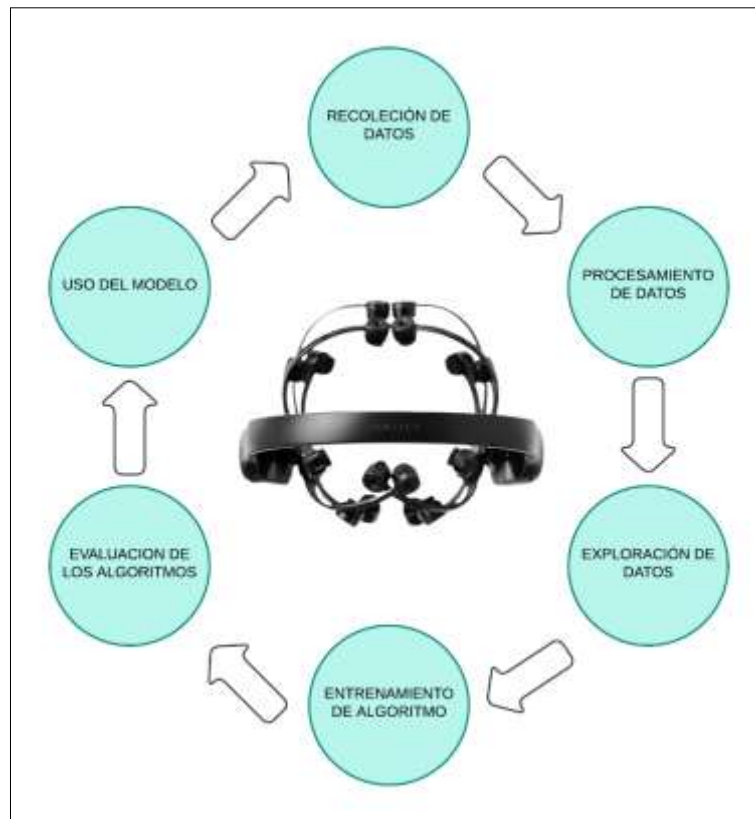
### 2.8.1. Tipos de aprendizaje

Los algoritmos de aprendizaje automático se dividen en tres categorías principales:

- **Aprendizaje supervisado:** Este enfoque emplea datos etiquetados para entrenar modelos, donde cada entrada está asociada a una salida conocida. Algunos ejemplos comunes son la clasificación y la regresión [41].
- **Aprendizaje no supervisado:** Trabaja con datos sin etiquetas, enfocándose en descubrir patrones ocultos dentro de ellos. Es ampliamente utilizado en tareas como el clustering y la reducción de dimensionalidad [41].
- **Aprendizaje semi supervisado:** Este enfoque combina un pequeño conjunto de datos etiquetados con un conjunto más amplio de datos sin etiquetar, utilizando los primeros para orientar la clasificación y la identificación de características en los segundos [41].
- **Aprendizaje por refuerzo:** Este modelo de aprendizaje automático que, aunque guarda similitudes con el aprendizaje supervisado, no utiliza datos de muestra para su entrenamiento. En su lugar, aprende de manera progresiva a través de un proceso de prueba y error [41].

### 2.8.2. Etapas para el Desarrollo de un Modelo de Machine Learning

La creación de un modelo de ML no se limita a usar un algoritmo de aprendizaje o una biblioteca de ML; es un proceso que generalmente implica al menos 6 pasos como se muestra en la Figura 11.



*Figura 11.- Etapas para el Desarrollo de un Modelo de Machine Learning.*

- **Recopilación de datos:** Los datos pueden ser obtenidos desde diversas fuentes, como sitios web, a través de una API o de una base de datos. Este proceso suele ser uno de los más complejos y requiere un tiempo determinado para completarse [41].
- **Preprocesamiento de los datos:** Una vez que se tiene acceso a los datos, es necesario asegurarse de que todos estén en un formato adecuado para ser utilizados por el algoritmo de aprendizaje. Generalmente, se deben realizar varias tareas de limpieza y ajuste antes de poder utilizarlos [41].
- **Exploración de los datos:** Se lleva a cabo un análisis preliminar para identificar problemas como valores faltantes o para detectar patrones que puedan facilitar la construcción del modelo. En esta fase, se deben identificar valores atípicos y las características más relevantes que influirán en la predicción [41].
- **Entrenamiento del algoritmo:** Los algoritmos de aprendizaje son alimentados con los datos procesados en las etapas anteriores. El objetivo es que los algoritmos extraigan patrones y relaciones útiles para luego generar predicciones precisas
- **Evaluación del algoritmo:** Se realizan pruebas sobre los resultados obtenidos durante el entrenamiento del algoritmo para verificar su desempeño y la validez del conocimiento generado a partir de los datos procesados [41].

### 2.8.3. Lenguajes de desarrollo de Machine Learning

Hay diversos lenguajes de programación utilizados para desarrollar aplicaciones de IA y ML. Cada tipo de aplicación tiene sus necesidades y limitaciones particulares, por lo que algunos lenguajes pueden ser más adecuados que otros dependiendo de los problemas que se busquen

resolver. A lo largo del tiempo, los lenguajes de programación han evolucionado, y algunos han sido diseñados específicamente para cumplir con los requisitos particulares de las aplicaciones de IA.

### **2.8.3.1. Lenguaje MatLab**

La plataforma MATLAB está diseñada para abordar problemas científicos y de ingeniería de manera eficiente. Su lenguaje, basado en matrices, es una de las formas más naturales de representar las matemáticas computacionales. Además, los gráficos integrados facilitan la visualización de datos y la extracción de información relevante. MATLAB cuenta con una extensa biblioteca de herramientas integradas (Toolboxes) que permiten comenzar a trabajar de inmediato con algoritmos esenciales para cada campo [41].

MATLAB ofrece diversas funcionalidades que facilitan el trabajo en aprendizaje automático (MathWorks for Machine Learning, SF):

- Compara diferentes enfoques, como regresión logística, árboles de clasificación, máquinas de vectores de soporte, métodos combinados y aprendizaje profundo.
- Aplica técnicas de reducción y refinamiento de modelos para crear modelos más precisos que mejoren el poder predictivo de los datos.
- Integra modelos de aprendizaje automático en sistemas, clústeres y nubes empresariales, y permite dirigirlos hacia hardware integrado en tiempo real.
- Genera automáticamente código para el análisis de sensores integrados.
- Ofrece flujos de trabajo integrados que abarcan desde el análisis de datos hasta la implementación.

Para el análisis de datos multidimensionales, la herramienta Statistics and Machine Learning Toolbox proporciona técnicas como selección de características, regresión gradual, análisis de componentes principales (PCA), regularización y otros métodos de reducción de dimensionalidad, los cuales ayudan a identificar las variables o características que afectan el modelo [41].

Este conjunto de herramientas incluye algoritmos de aprendizaje automático supervisados y no supervisados, como máquinas de vectores de soporte (MVS), árboles de decisión potenciados y en bolsas, vecinos más cercanos (k-NN), medios k, medoides k, agrupamiento jerárquico, modelos de mezcla gaussiana y modelos de Markov ocultos (MathWorks, Comparación entre MATLAB y Python, SF) [41].

### CAPÍTULO III. METODOLOGÍA

Este capítulo describe los aspectos metodológicos que orientaron la realización de la investigación, incluyendo el tipo y método de investigación, el procedimiento y análisis, la población de estudio y el tamaño de la muestra, además del método utilizado para el análisis y procesamiento de los datos.

#### 3.1. Tipo de investigación

El presente proyecto de investigación se enfocó en la creación de una interfaz cerebro computadora siendo así de tipo aplicativo y desarrollo, con propósitos prácticos y específicos para brindar soluciones a problemas de un paciente con dificultades para interactuar con su entorno y desarrollar relaciones sociales significativas.

#### 3.2. Diseño de investigación

El diseño de esta investigación es experimental controlado, por el paciente seleccionado que presenta el síndrome de Angelman.

#### 3.3. Operacionalización de variables

*Tabla 1.- Tabla de Variable Independiente*

Variable	Concepto	Indicador
Cantidad de muestras obtenidas en el registro de EEG.	Cantidad de muestras, tiempo, frecuencia.	Segundos Hz

*Tabla 2.- Tabla de Variable Dependientes*

Variable	Concepto	Indicador
Tiempo del paciente que tarda en expresarse tradicionalmente.	Duración en el cual el paciente no utiliza el dispositivo.	Minutos, segundos
Tiempo del paciente que tarda en expresarse con el dispositivo.	Duración en el cual el paciente utiliza el dispositivo.	Minutos, segundos

#### 3.4. Técnicas de recolección de datos

Para la recolección de datos se consideraron dos escenarios. En el primer escenario, el paciente intentó comunicarse de manera tradicional. En el segundo escenario, se utilizó una interfaz cerebro-computadora, integrado con el asistente virtual Alexa, para facilitar la comunicación del paciente. En ambos casos, se registraron los tiempos requeridos por el paciente para expresar una necesidad básica de comunicación.

### 3.5. Población de estudio y tamaño de muestra

#### 3.5.1. Población de estudio

La población de estudio está definida por los datos que fueron recolectados con el paciente, que representan el tiempo de ejecución de la actividad asignada en el paciente tarda en expresar una necesidad básica y la misma actividad en la que tarda en expresar una necesidad básica con la interfaz cerebro-computadora, integrado el asistente virtual Alexa.

*Tabla 3.- Tamaño de la población 1 – sin interfaz cerebro-computadora*

Número de pacientes	Toma de tiempos por día	Días	Tamaño de la población
1	5	15	75

*Tabla 4.- Tamaño de la población 1 – con interfaz cerebro-computadora*

Número de pacientes	Toma de tiempos por día	Días	Tamaño de la población
1	5	15	75

#### 3.5.2. Tamaño muestra

El presente estudio se realizó con un único paciente, seleccionado bajo criterios específicos que responden a los objetivos de la investigación. La elección del paciente se fundamentó en su diagnóstico confirmado de síndrome de Angelman, así como en su disposición y viabilidad para participar en las pruebas experimentales.

Al basarse en un solo paciente, la principal limitación es que no se pueden hacer afirmaciones definitivas sobre cómo los resultados se aplicarían a otros individuos con la misma condición. Los hallazgos pueden estar influenciados por variables individuales, como características genéticas, el contexto social, o la historia médica del paciente, lo que restringe la capacidad para establecer patrones o conclusiones universales.

Aunque el uso de un solo paciente limita la generalización de los resultados, este enfoque permite concentrar los esfuerzos en la adquisición precisa de datos y en la validez inicial de los procesos técnicos involucrados. De esta forma, el estudio establece una base sólida para futuras investigaciones que podrán ampliar la muestra y validar los hallazgos obtenidos.

Esto permite un análisis exhaustivo y una comprensión más detallada de los efectos, comportamientos y respuestas de un individuo específico, lo cual puede proporcionar información valiosa que luego pueda ser generalizada o ampliada a través de estudios con muestras más grandes.

Dado que el propósito principal de este trabajo es el desarrollo y la validación técnica de una interfaz cerebro-computadora, no fue necesario determinar un tamaño de muestra mediante procedimientos estadísticos. La naturaleza exploratoria de la investigación permitió trabajar con un único caso, lo cual resultó exhaustivo y evaluar la factibilidad del sistema propuesto.

### 3.6.Hipótesis

#### 3.6.1. Hipótesis nula

El tiempo que tarda el paciente en completar la actividad asignada sin la interfaz cerebro-computadora es menor o igual que el tiempo que lleva con la interfaz cerebro-computadora.

#### 3.6.2. Hipótesis alternativa

El tiempo que tarda el paciente en completar la actividad asignada sin la interfaz cerebro-computadora es mayor que el tiempo que lleva con la interfaz cerebro-computadora.

### 3.7.Métodos de análisis

Para comprobar si la diferencia entre las muestras recolectadas del paciente seguía una distribución normal, se aplicó la prueba de Kolmogórov-Smirnov. Una vez confirmado que dicha diferencia se ajustaba a una distribución normal, se utilizó la prueba paramétrica de t de Student para extraer conclusiones sobre la hipótesis planteada. Además, se elaboraron diagramas de cajas, lo que permitió facilitar tanto la interpretación como el análisis de los datos obtenidos.

### 3.8.Procedimiento y análisis

En la Figura 12, se detallan las fases que se utilizaron para la adquisición y procesamiento las señales EEG adquiridas mediante la diadema Emotiv EPOC X, con el fin de identificar patrones relevantes que permitan la clasificación de comandos específicos con el objetivo de proporcionar al paciente una vía de comunicación básica.

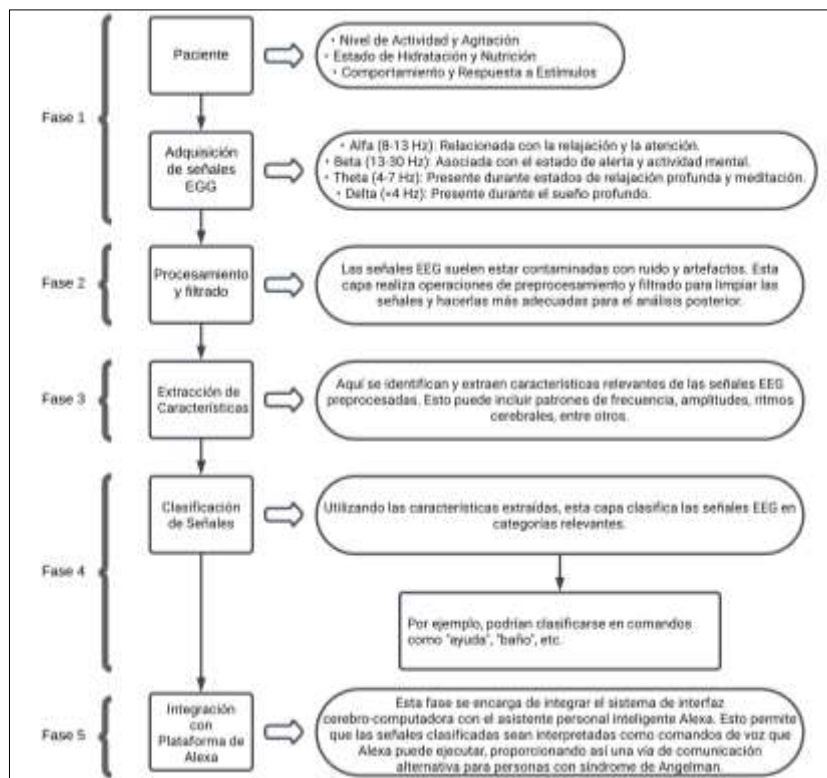


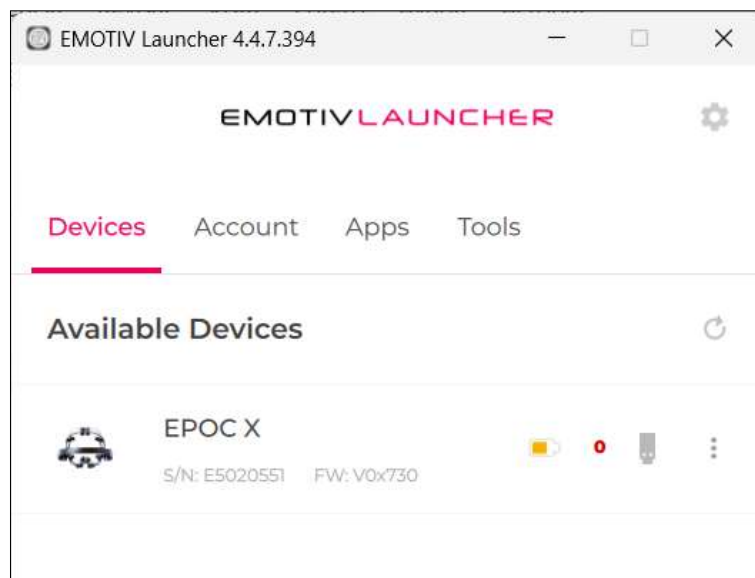
Figura 12.- Diagrama de bloques.



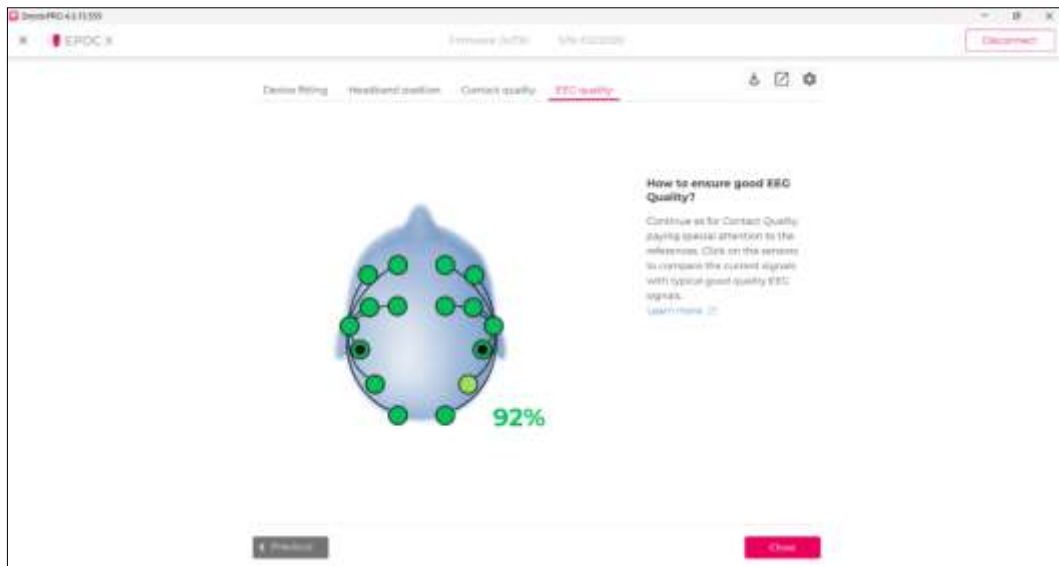
### 3.8.1. Fase 1: Paciente y adquisición de señales EEG.

En la fase inicial del proyecto, se priorizó la preparación del entorno y la familiarización del paciente con el equipo de adquisición de señales EEG. Para ello, se estableció un ambiente tranquilo y cómodo que minimizara distracciones y ayudara al paciente a sentirse relajado. Posteriormente, se introdujo al paciente al casco Emotiv, explicando su funcionamiento y ajustándolo cuidadosamente para garantizar un contacto adecuado de los sensores con el cuero cabelludo, empleando lubricante conductor para optimizar la captación de señales.

Una vez colocado el dispositivo, se utilizó el software Emotiv Launcher para registrar las señales EEG bajo condiciones controladas. Los datos obtenidos se exportaron en un formato compatible y se transfirieron a MATLAB, donde se realizó un procesamiento inicial que incluyó filtrado de artefactos y segmentación. Este procedimiento aseguró la calidad de los datos, permitiendo su uso en las etapas posteriores de desarrollo de la interfaz cerebro-computadora.



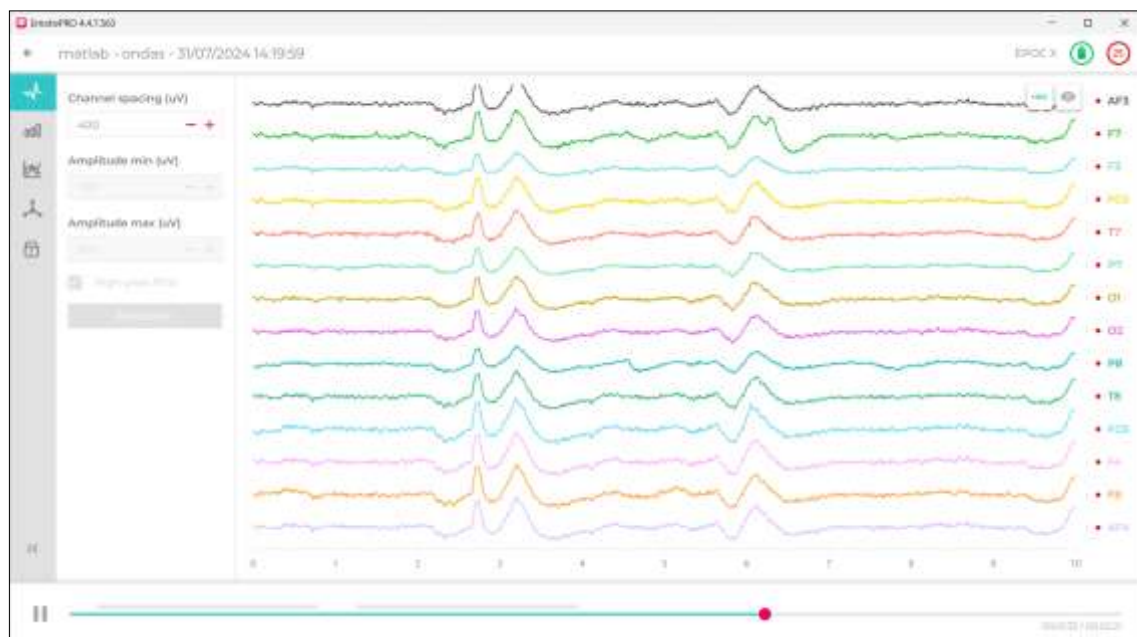
*Figura 13.- Detección del dispositivo.*



**Figura 14.-** Entorno del Software EmotivPRO.

Al momento de utilizar el casco, se estableció que el paciente debía encontrarse en un estado de total relajación. Esta condición fue fundamental para garantizar la correcta adquisición de las señales EEG, ya que cualquier tensión física o emocional podía generar interferencias y artefactos no deseados en los datos registrados. Se consideró que el ambiente debía ser tranquilo y libre de distracciones, permitiendo que el paciente alcanzara un estado de reposo adecuado antes de iniciar el proceso de medición.

De esta manera, se aseguraron registros más precisos y confiables de las señales eléctricas cerebrales, lo cual fue crucial para el análisis, filtrado y posterior procesamiento de los datos en el desarrollo del proyecto.



**Figura 15.-** Resultados de la señal de los 14 sensores del Emotiv EPOC X.

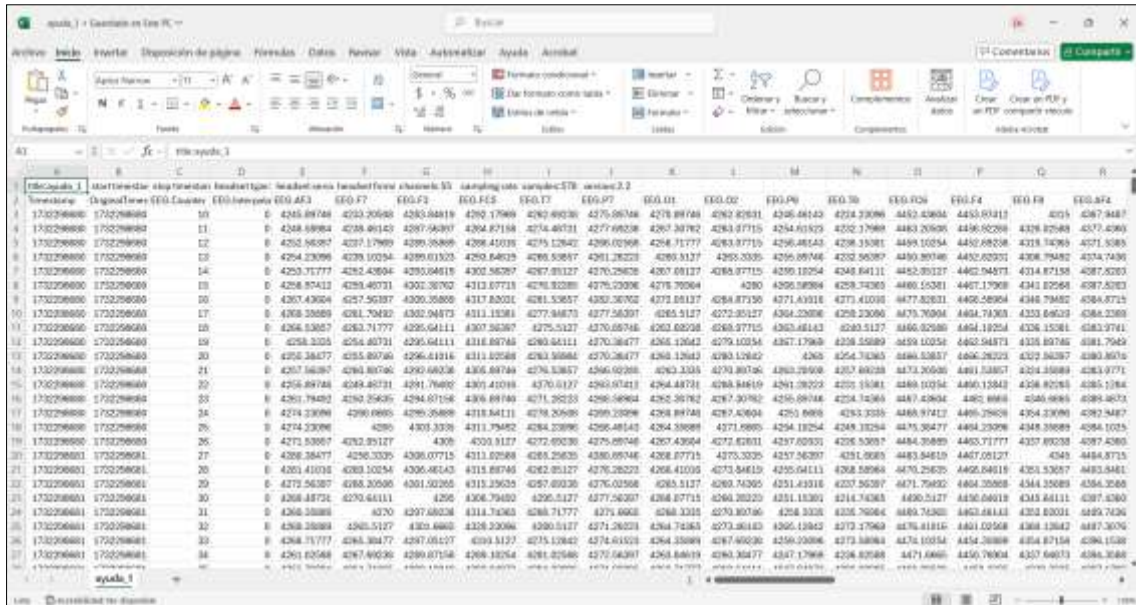


Figura 16.- Archivo .csv donde se encuentran los datos de los 14 sensores.

### 3.8.2. Fase 2: Procesamiento y filtrado

#### 3.8.2.1. Filtrado de señales

El filtrado de señales fue un proceso esencial para eliminar las frecuencias no deseadas en las señales EEG. Se utilizó un filtro pasabanda con frecuencias de corte de 1 a 30 Hz, diseñado para permitir el paso de las frecuencias relevantes dentro de ese rango, como las ondas alfa, beta y gamma, mientras filtraba las frecuencias fuera de ese rango. El filtro empleado fue un filtro de Butterworth de cuarto orden; este tipo de filtro es conocido por su capacidad para suprimir las frecuencias no deseadas de manera eficiente, sin causar distorsiones significativas en las señales relevantes.

```
% Parámetros del filtro
fs = 128; % Frecuencia de muestreo
lowCutoff = 1; % Frecuencia de corte baja
highCutoff = 30; % Frecuencia de corte alta
[b, a] = butter(4, [lowCutoff, highCutoff] / (fs / 2), 'bandpass');
```

Figura 17.- Código utilizado para el filtrado de la señal.

#### 3.8.2.2. Eliminación de artefactos.

La eliminación de artefactos fue necesaria debido a la presencia de ruidos generados por factores como los movimientos musculares, los parpadeos o un mal contacto de los electrodos. Para lograr esto, se utilizó un filtro de mediana, una técnica no lineal eficaz para suprimir artefactos impulsivos, sin afectar las características esenciales de la señal. Se aplicó un filtro de mediana con una ventana de tres puntos; este filtro permitió suavizar las señales y reducir la interferencia de los artefactos sin eliminar las oscilaciones relevantes de la actividad cerebral.

```
% Filtrado adicional: Filtro de mediana para eliminar artefactos
medianFilteredData = medfilt1(filteredDataArray, 3); % Ventana de 3 puntos
```

*Figura 18.- Código utilizado para eliminar los artefactos.*

### 3.8.2.3. Normalización de señales.

La normalización de señales fue un paso necesario para garantizar que las amplitudes de las señales EEG fueran comparables entre sí. Este proceso ajustó los valores de las señales para que se encontraran dentro de un rango específico, generalmente entre 0 y 1, lo que evitó que las variaciones de amplitud entre diferentes señales influyeran en el análisis. En este caso, la normalización se realizó restando el valor mínimo de la señal y dividiendo el resultado entre el rango de la señal, es decir, la diferencia entre el valor máximo y el valor mínimo de la misma.

```
% Normalización de la señal
normalizedDataArray = (medianFilteredData - min(medianFilteredData)) ./ ...
    (max(medianFilteredData) - min(medianFilteredData));
```

*Figura 19.- Código utilizado para la normalización de la señal.*

### 3.8.3. Fase 3: Extracción de características

La extracción de características se enfocó en identificar las propiedades más relevantes de cada segmento de señal. Entre los métodos empleados, se utilizó la identificación de picos máximos, lo que permitió detectar los picos más significativos en cada ventana de tiempo. Además, se empleó la transformada de Fourier para analizar las componentes de frecuencia de las señales EEG. Estos métodos ayudaron a extraer información sobre las frecuencias dominantes y las características dinámicas de la actividad cerebral, lo que resultó crucial para el análisis posterior y la clasificación de las señales.

En la Figura 20, se utilizó la función `findpeaks()` para detectar picos en las señales normalizadas. Este proceso busca los picos más altos que cumplen con un umbral mínimo de altura `minHeight` y distancia mínima entre picos `MinPeakDistance`.

```
[pks, locs] = findpeaks(normalizedDataArray(:, j), 'MinPeakHeight', minHeight, 'MinPeakDistance', fs/2);
```

*Figura 20.- Código para detectar los picos más relevantes.*

Se extrajeron solo los picos que fueron mayores que un umbral basado en el pico máximo detectado. Los picos relevantes fueron aquellos que superaron el umbral del 70% del pico máximo.

```
% Establecer umbral para los picos relevantes
threshold = max(pks) * 0.7; % Umbral del 70% del máximo
relevantPeaks = pks(pks >= threshold); % Picos por encima del umbral
```

*Figura 21.- Código para establecer el umbral.*

Si no se encuentran picos relevantes, se agrega NaN al vector de características para indicar que no se encontraron picos significativos en la señal.

Las características extraídas, incluyendo la media y la desviación estándar de la señal, junto con las características derivadas de los picos relevantes, fueron almacenadas en un vector de características, para utilizarse con Inteligencia Artificial y Machine Learning.

### 3.8.4. Fase 4: Clasificación de señales

#### 3.8.4.1. Inteligencia Artificial.

La inteligencia artificial (IA) estuvo representada por el uso de una Red Neuronal Artificial (ANN); en este caso, fue el modelo utilizado para clasificar las señales EEG en diferentes categorías (por ejemplo: 'ayuda', 'baño', etc.).

- **Red Neuronal Artificial (ANN):** Fue implementada como una herramienta de clasificación; su estructura, definida por el modelo `patternnet(hiddenLayerSize)`, contó con una capa oculta compuesta por 20 neuronas como se muestra en la Figura 22. Este modelo procesó las señales EEG, transformando los datos de entrada en predicciones categorizadas (por ejemplo: 'ayuda', 'baño', etc.), lo que permitió asignar cada señal a su respectiva categoría de salida.

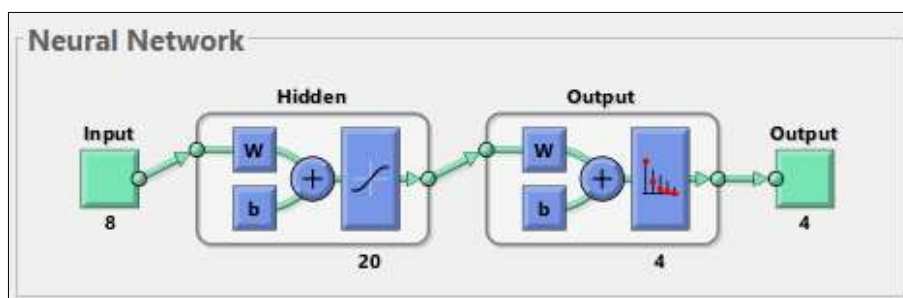


Figura 22.- Herramienta de la Red Neuronal Artificial en MATLAB.

- **División de datos en entrenamiento, validación y prueba:** Los datos fueron divididos en tres conjuntos: entrenamiento (80%), validación (10%) y prueba (10%), utilizando la función `dividerand`; esta división permitió entrenar el modelo, evaluar su rendimiento durante el entrenamiento y probar su capacidad de generalización sobre datos no vistos.
- **Entrenamiento de la Red Neuronal:** El modelo de la red neuronal fue entrenado mediante la función `train`, utilizando los datos de entrenamiento (características y etiquetas correspondientes) para ajustar los pesos (son los valores numéricos que se asignan a las conexiones entre las neuronas) y parámetros de la red, minimizando el error de predicción en cada iteración.

Una vez completado el entrenamiento, se mostró un mensaje indicando que el modelo de la Red Neuronal fue entrenado correctamente como se muestra en la Figura 23, lo que señalaba que el modelo estaba listo para ser utilizado en predicciones y evaluaciones.

```

Command Window
Warning: Variable names were modified to make them valid MATLAB identifiers. The original names
are saved in the VariableDescriptions property.
Clases únicas en las etiquetas:
    ayuda
    baño
    comer
    dormir

Entrenando el modelo...
Modelo de Red Neuronal entrenado.

```

*Figura 23.- Visualización que el modelo fue entrenado correctamente.*

### 3.8.4.2. Machine Learning.

El aprendizaje automático (ML) estuvo representado en el código mediante el uso del enfoque de aprendizaje supervisado. El modelo fue entrenado utilizando un conjunto de datos etiquetados (las señales EEG con sus respectivas categorías) y posteriormente se utilizó para predecir las categorías en nuevos datos.

- **Aprendizaje supervisado:** El modelo fue entrenado con datos etiquetados; en este caso, las etiquetas fueron las categorías ("ayuda", "baño", etc.) asociadas a las señales EEG. El código convirtió las etiquetas a one-hot encoding mediante la función `ind2vec()`: una forma de representar las categorías en un formato numérico adecuado para el modelo de red neuronal.

```

% === CONVERTIR LAS ETIQUETAS A UNA MATRIZ ONE-HOT ===
% Convertir etiquetas categóricas a números
labelsNumeric = double(etiquetas);

% Convertir las etiquetas numéricas a one-hot encoding
numClases = numel(clasesUnicas);
labelsOneHot = full(ind2vec(labelsNumeric', numClases)); % Usar ind2vec para convertir a one-hot encoding

```

*Figura 24.- Código que convierte las etiquetas numéricas a one-hot encoding.*

- **Predicción de Nuevos Datos:** Después de que el modelo fue entrenado, el código utilizó este modelo entrenado para predecir la categoría de un nuevo archivo de señales EEG. Para hacerlo, se extrajeron primero las características del archivo de prueba ("prueba\_ayuda\_1.csv") utilizando la misma técnica aplicada durante el entrenamiento. Estas características extraídas fueron luego alimentadas al modelo entrenado, permitiendo que la red neuronal generara una predicción sobre cuál sería la categoría del nuevo archivo (por ejemplo, "ayuda", "baño", etc.). Este proceso de predicción es esencial en el aprendizaje automático, ya que permite al modelo generalizar lo aprendido a nuevos datos no vistos durante el entrenamiento.
- **Conversión de la salida:** La salida de la red neuronal fue en formato one-hot encoding, representando las categorías posibles con valores binarios. Para obtener la clase predicha, se utilizó la función `max()` para identificar el valor máximo del vector de salida, lo que indicó la categoría más probable. Luego, este valor se mapeó a la clase correspondiente, permitiendo así obtener la categoría final predicha para el nuevo archivo. Este proceso fue esencial para interpretar correctamente la predicción del modelo.



### 3.8.5. Fase 5: Integración con plataforma de Alexa

#### 3.8.5.1. Interfaz cerebro-computadora.

En el desarrollo de la interfaz cerebro-computadora, se utilizó MATLAB GUIDE para crear una interfaz gráfica de usuario (GUI) que permitiera interactuar de manera eficiente con los datos obtenidos de las señales EEG. Se diseñaron diferentes elementos en la interfaz, como cuadros de texto, botones y figuras como se muestra en la Figura 25, para facilitar la visualización y el procesamiento de las señales. La GUI permitió cargar archivos de EEG, procesarlos mediante los filtros y técnicas de preprocesamiento, y mostrar los resultados de manera intuitiva. Además, se implementaron funciones para realizar comparaciones entre las señales y extraer características, lo que facilitó la interpretación de los datos y la toma de decisiones en tiempo real.

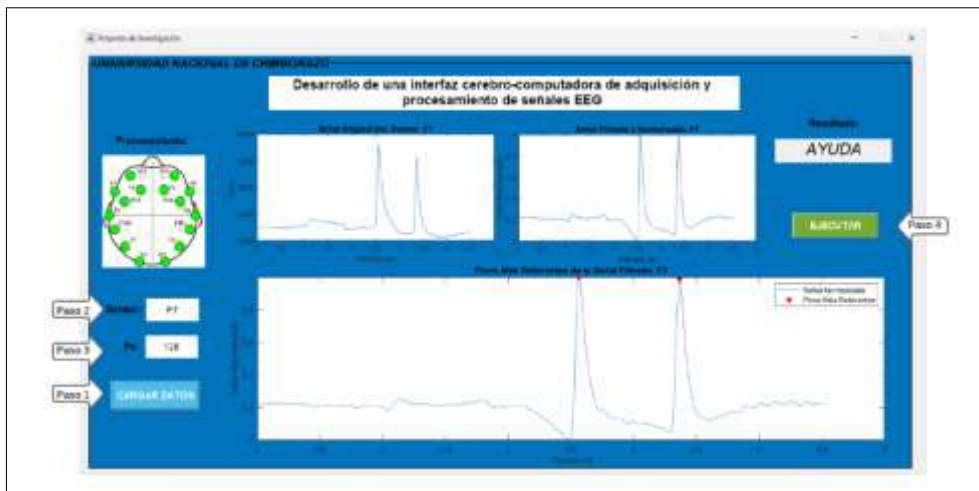


Figura 25.- Interfaz gráfica de la interfaz cerebro-computadora.

Pasos seguidos para el funcionamiento de la interfaz del proyecto de investigación:

- **Paso 1:** Se implementó un botón que permitía cargar un archivo .csv con los datos del EEG obtenidos del dispositivo.

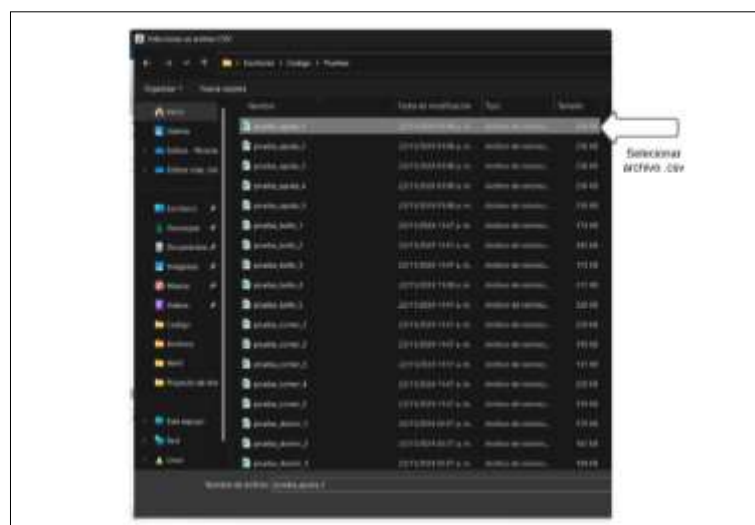
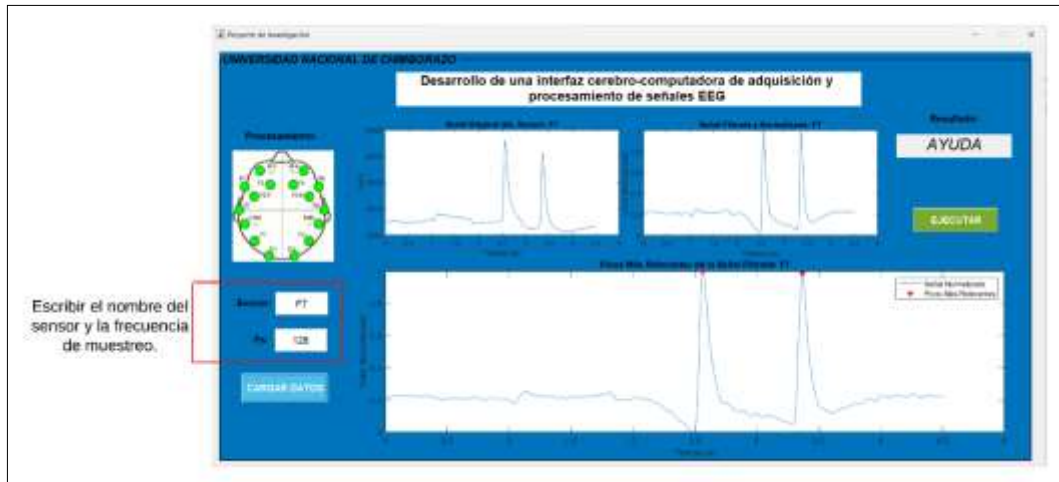


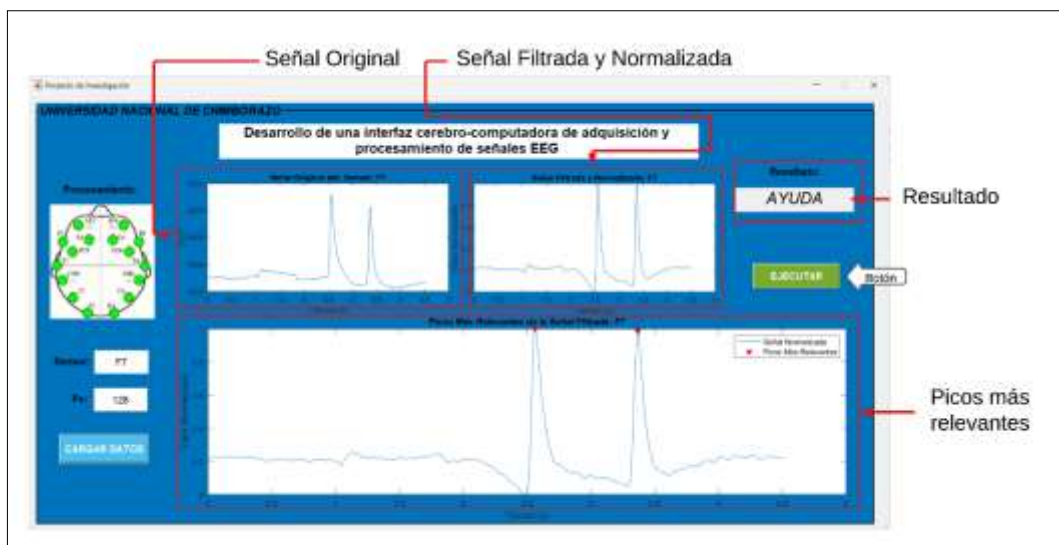
Figura 26.- Ventana para seleccionar los archivos .csv con los datos del EEG.

- **Paso 2:** Se incluyó un campo de texto donde se ingresaba el nombre del sensor, ya que el dispositivo disponía de 14 sensores: AF3, AF4, F3, F4, F7, F8, FC5, FC6, T7, T8, P7, P8, O1 y O2.
- **Paso 3:** Se habilitó otro campo de texto para ingresar la frecuencia de muestreo del dispositivo Emotiv Epoc X.



*Figura 27.- Campo de texto donde se ingresó el nombre del sensor y la frecuencia de muestreo.*

- **Paso 4:** Una vez completados los pasos anteriores, se hizo clic en el botón EJECUTAR, lo que permitió obtener la señal original del sensor seleccionado, la señal filtrada y normalizada, así como identificar los picos más relevantes. Estos picos se utilizaron como características clave para entrenar el modelo de Machine Learning del proyecto. Como resultado, se predijo una de las necesidades básicas del paciente, como “AYUDA” o “BAÑO”, entre otras opciones.



*Figura 28.- Resultado de la interfaz.*



Al ejecutar la interfaz, se genera un mensaje con respecto al resultado predicho a través de una función que activa al asistente virtual Alexa para ejecutar los comandos configurados. Los mensajes fueron grabados con la voz humana en formato .mp4, como se puede visualizar en la Figura 29.

```
function reproducirMensaje(clasePredicha)
    switch clasePredicha
        case 'ayuda'
            audioFile = 'ayuda.mp4';
        case 'baño'
            audioFile = 'baño.mp4';
        case 'comer'
            audioFile = 'comer.mp4';
        case 'dormir'
            audioFile = 'dormir.mp4';
        otherwise
            disp('No hay mensaje asociado para la clase predicha.');
```

```
            return;
        end

    % Reproducir el archivo de audio
    if isfile(audioFile)
        [y, Fs] = audioread(audioFile);
        sound(y, Fs);
    else
        disp(['Archivo de audio no encontrado: ', audioFile]);
    end
end
```

*Figura 29.- Función para reproducir la clase predicha.*

Este sistema está diseñado para un paciente con síndrome de Angelman, con el objetivo de mejorar su calidad de vida al proporcionarle una vía de comunicación efectiva y accesible. Los mensajes están vinculados a rutinas personalizadas que permiten al paciente expresar sus necesidades, como pedir ayuda, solicitar atención, o interactuar con su entorno mediante dispositivos inteligentes.

### **3.8.5.2. Personalización de comandos activados por voz**

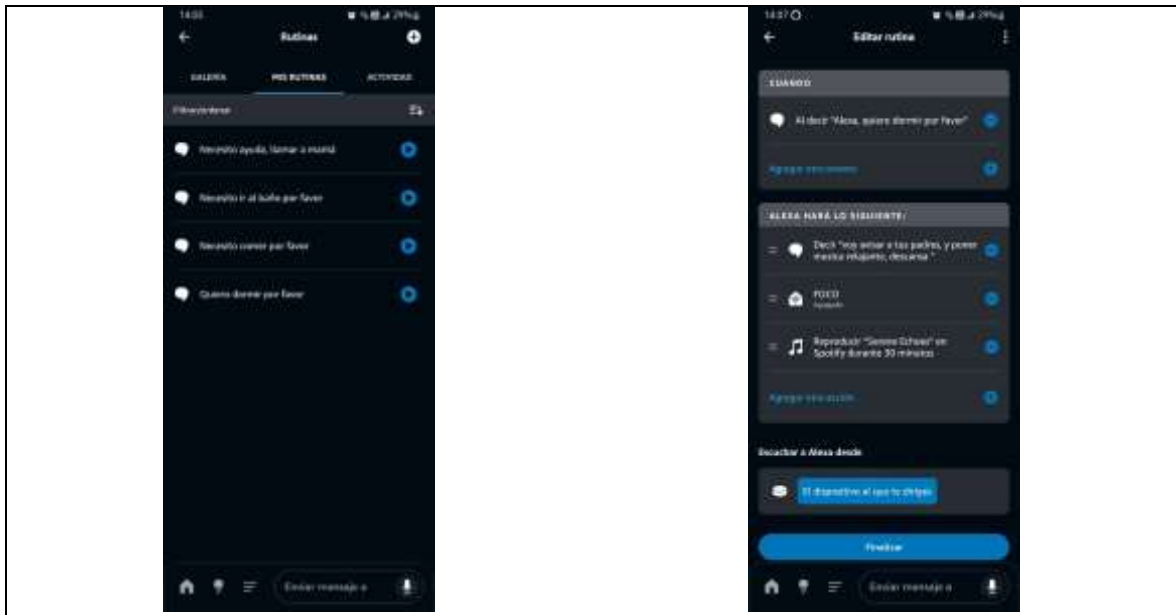
La interfaz desarrollada permite configurar comandos específicos en el asistente virtual Alexa, los cuales se activan mediante mensajes previamente programados. Estos comandos están diseñados para facilitar la comunicación y mejorar la autonomía de personas con necesidades especiales, como en el caso de un paciente con síndrome de Angelman. Entre los comandos personalizados se incluyen:

- “Necesito ayuda, llamar a mamá”: Este comando activa una rutina que realiza una llamada a un contacto de emergencia, proporcionando una respuesta inmediata en situaciones críticas.
- “Necesito ir al baño por favor”: Al detectar este comando, se encienden las luces del baño durante un período de tiempo específico, facilitando el acceso seguro en horarios nocturnos o en situaciones de baja visibilidad.
- “Necesito comer por favor”: Este mensaje activa una rutina que puede enviar un recordatorio a otra persona en el hogar o activar un enchufe inteligente conectado a

una cafetera o dispensador automático de alimentos, agilizando el acceso a insumos básicos.

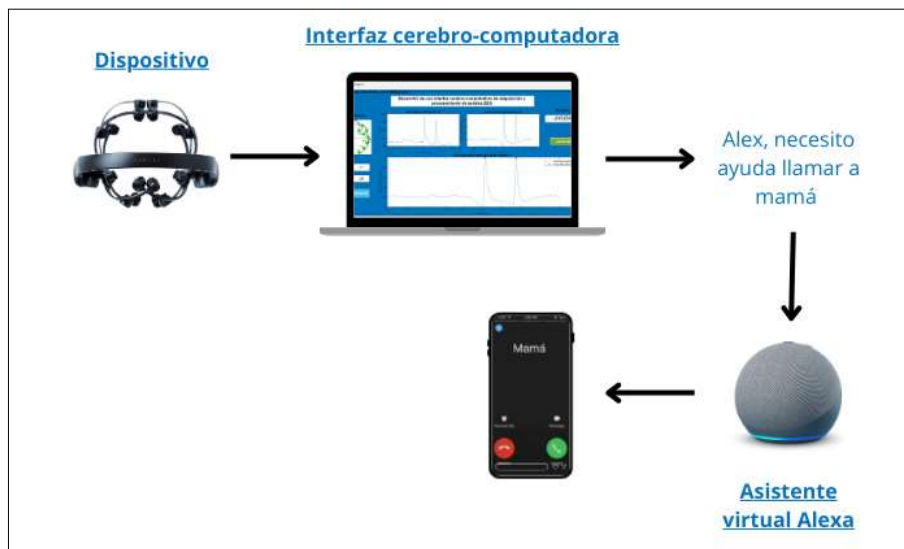
- “Quiero dormir por favor”: Este comando inicia una rutina destinada a generar un ambiente propicio para el descanso. Las acciones incluyen apagar luces y reproducir sonidos relajantes.

Estas funcionalidades no solo amplían las capacidades del asistente virtual, sino que también proporcionan al paciente un mayor control sobre su entorno de manera intuitiva y accesible.



*Figura 30.- Configuración de los comandos del asistente virtual Alexa.*

La integración de la interfaz con Alexa no solo facilita la comunicación, sino que también fomenta la autonomía del paciente en actividades diarias, promoviendo su bienestar y reduciendo las barreras de interacción en su entorno.

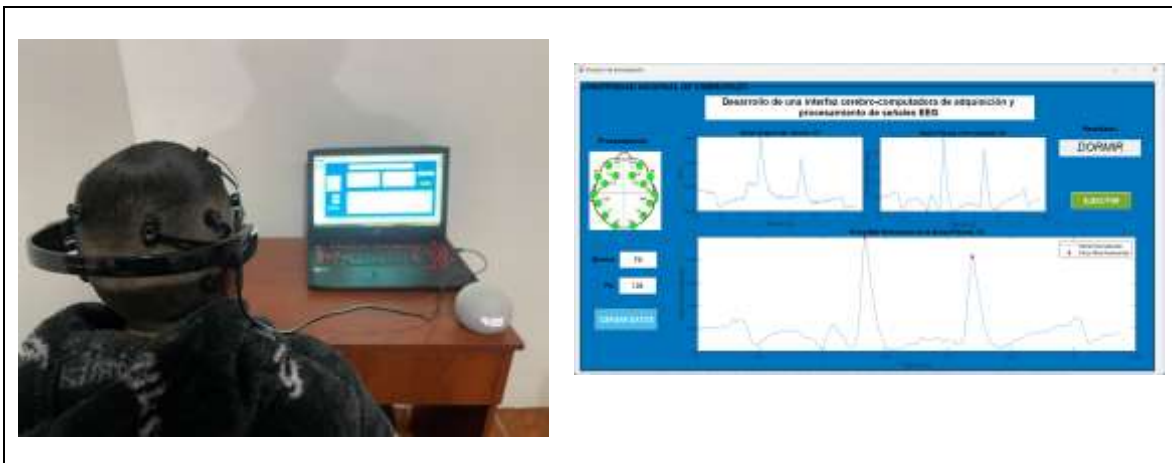


*Figura 31.- Ecosistema final del proyecto de investigación.*

### 3.8.5.3. Acondicionamiento

El proyecto se desarrolló con el propósito de proporcionar una vía de comunicación alternativa para un paciente con síndrome de Angelman, utilizando la diadema Emotiv Epoc X y comandos de voz a través de Alexa. Inicialmente, el paciente debía estar en un estado de relajación, lo que permitió alcanzar una condición adecuada para la adquisición de señales EEG. Para lograrlo, se realizó la colocación precisa de la diadema siguiendo las recomendaciones del fabricante. Se humedecieron los sensores con solución salina para optimizar el contacto con el cuero cabelludo, asegurando la alineación correcta en puntos anatómicos clave como el hueso mastoideo y el centro del cráneo. Además, se verificó la conectividad de los electrodos mediante el software de Emotiv, que mostró la calidad de la señal en tiempo real. La calibración inicial se realizó mientras el paciente permanecía en reposo, estableciendo así una línea base confiable para las mediciones posteriores.

Se realizaron registros preliminares durante un periodo controlado, aplicando filtros pasabanda entre 1 y 30 Hz para eliminar interferencias y mejorar la precisión de los datos. Posteriormente, se extrajeron características clave como picos máximos y su distribución temporal, asegurando que los datos procesados fueran representativos de patrones cerebrales específicos. Para mantener la calidad de las señales, se monitoreó la humedad de los sensores durante el proceso y se realizaron ajustes cuando fue necesario. Dichos patrones fueron utilizados como base para entrenar modelos de Machine Learning, los cuales facilitaron la clasificación automática de comandos asociados a necesidades básicas del paciente. Finalmente, el dispositivo fue almacenado y limpiado correctamente tras cada uso para prolongar su durabilidad, asegurando así un rendimiento confiable en futuras sesiones.



*Figura 32.- Acondicionamiento al BCI.*

## CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

### 4.1.Resultados

En la Tabla 5, se detalla los tiempos obtenidos en segundos de cada día que fueron realizados con el paciente. Estos tiempos reflejan cuanto se demora el paciente en completar la actividad requerida ya sea sin el dispositivo y con el dispositivo.

*Tabla 5.- Tiempos diarios realizando las actividades asignadas con y sin dispositivo.*

Días	Paciente sin dispositivo					Paciente con dispositivo				
	RE(s)	A1(s)	A2(s)	A3(s)	A4(s)	RE(s)	A1(s)	A2(s)	A3(s)	A4(s)
<b>Día 1</b>	60	24.36	27.68	28.47	27.21	60	09.14	11.57	10.82	07.95
<b>Día 2</b>	60	25.78	29.47	29.47	29.14	60	08.54	10.96	10.47	07.58
<b>Día 3</b>	60	32.41	29.36	31.24	28.47	60	09.25	09.23	10.52	08.69
<b>Día 4</b>	60	29.77	26.14	29.14	26.15	60	10.28	08.63	11.47	09.74
<b>Día 5</b>	60	23.91	24.78	38.74	24.85	60	08.96	08.89	09.21	07.46
<b>Día 6</b>	60	27.41	28.69	22.34	24.93	60	11.58	09.87	09.36	09.47
<b>Día 7</b>	60	28.74	26.37	25.74	24.78	60	08.96	08.33	08.98	09.47
<b>Día 8</b>	60	26.47	27.98	29.14	26.58	60	08.78	09.75	11.30	09.34
<b>Día 9</b>	60	30.12	27.14	27.69	29.17	60	09.58	08.92	10.96	07.75
<b>Día 10</b>	60	26.78	29.87	28.47	28.96	60	09.46	09.71	09.96	07.69
<b>Día 11</b>	60	24.87	27.58	29.47	26.56	60	08.59	09.36	09.23	07.87
<b>Día 12</b>	60	24.87	29.56	32.74	29.75	60	09.21	08.45	08.96	09.85
<b>Día 13</b>	60	25.69	27.47	30.07	26.47	60	10.85	11.68	11.92	10.74
<b>Día 14</b>	60	38.74	37.09	37.98	39.95	60	11.49	11.23	11.13	12.74
<b>Día 15</b>	60	27.56	32.11	29.89	27.98	60	09.35	09.15	09.74	09.77

En la Tabla 6, se observa el análisis descriptivo del conjunto de datos, donde existe una diferencia significativa entre las medias de ambas muestras.

*Tabla 6.- Análisis descriptivo de muestras de las actividades sin y con dispositivo.*

Muestras	Resultados Descriptivos	
<b>Actividades sin dispositivo</b>	Media	28.6718
	Mediana	28.2250
	Varianza	13.526
	Desviación Estándar	3.67783
	Mínimo	22.34
	Máximo	39.95
	Rango	17.61
	Rango Inter cuartil	3.31
	Asimetría	1.449
	Curtosis	2.393
<b>Actividades con dispositivo</b>	Media	9.6648

	Mediana	9.4100
	Varianza	1.449
	Desviación Estándar	1.20393
	Mínimo	7.46
	Máximo	12.74
	Rango	5.28
	Rango Inter cuartil	1.76
	Asimetría	0.384
	Curtosis	-0.364

Dado que cada muestra contenía más de 50 datos, se utilizó la prueba de normalidad de Kolmogórov-Smirnov para evaluar su distribución. Posteriormente, se analizó la diferencia entre ambas muestras con el objetivo de verificar si seguían una distribución normal, obteniendo un valor p de 0.3418.

```

Asymptotic one-sample Kolmogorov-Smirnov test
data: Diferencia
D = 0.12219, p-value = 0.3418
alternative hypothesis: two-sided

```

**Figura 33.-** Resultado de la prueba de Kolmogórov-Smirnov obtenido con RStudio.

El valor p, al ser mayor que el nivel de significancia, indicó que los datos siguen una distribución normal. Esto sugiere que los datos tienen un enfoque paramétrico, por lo que, para comparar las muestras definidas, se decidió emplear la prueba de t de Student. Como resultado, esta prueba arrojó un valor p de  $2.2 \times 10^{-16}$ , lo que llevó a rechazar la hipótesis nula y aceptar la hipótesis alternativa.

```

welch Two Sample t-test
data: Actividad_SD and Actividad_CD
t = 37.461, df = 70.294, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 18.17418      Inf
sample estimates:
mean of x mean of y
28.690678  9.670169

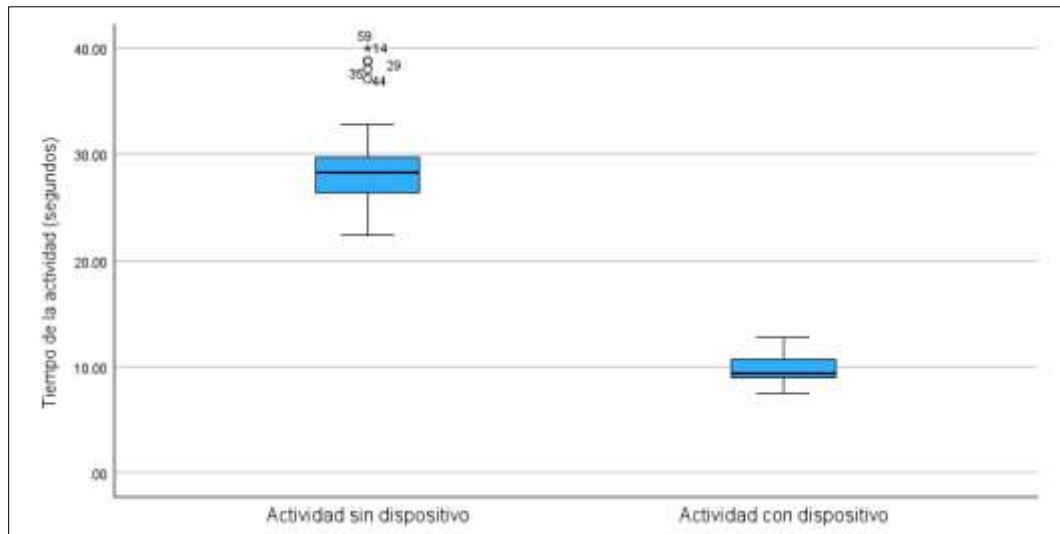
```

**Figura 34.-** Resultado de la prueba de T-Student obtenido con RStudio.

#### 4.2. Discusión

Se puede notar que la mediana de la muestra correspondiente a la "Actividad sin dispositivo" es superior a la de la "Actividad con dispositivo", lo que indica que hay diferencias significativas entre los tiempos registrados en ambas muestras. En resumen, esto demuestra

que los pacientes completan la actividad asignada en menos tiempo al utilizar el prototipo, lo que respalda la validez de la hipótesis alternativa.



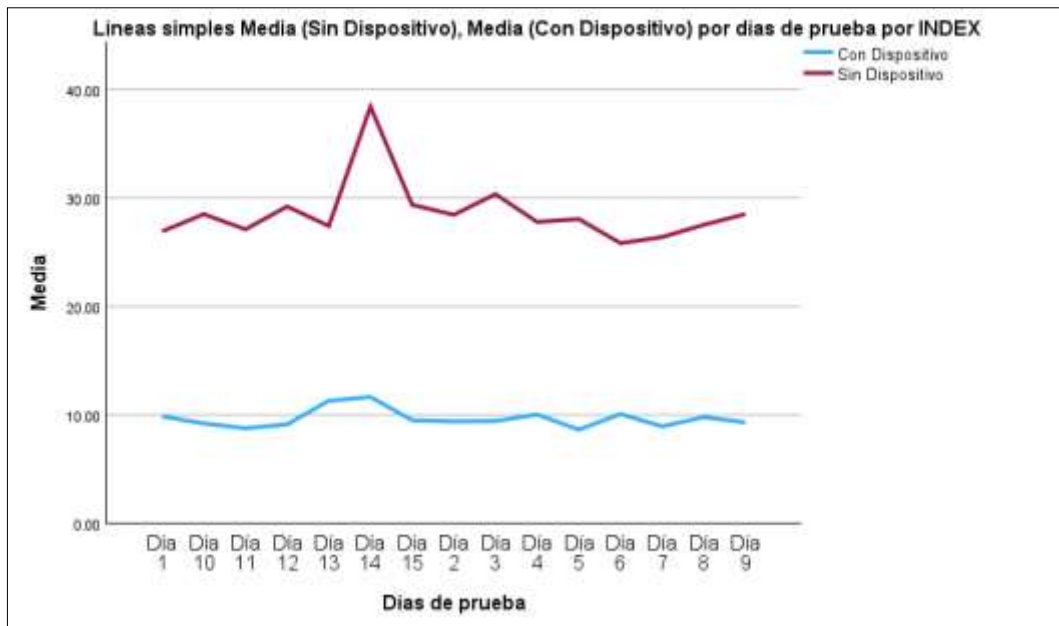
**Figura 35.-Distribución de datos de las muestras.**

Al realizar las actividades asignadas con el dispositivo nos podemos dar cuenta que hay una mejora significativa para realizar las actividades asignadas, viendo como el proyecto se acopla a los resultados esperados al realizar este proyecto. También podemos observar que hubo tiempos incongruentes con los datos obtenidos esto se debe a que el paciente tuvo dificultades de concentración y malestar con la toma de datos del día 14 viendo estos reflejados en el diagrama de cajas.

Al observar que con el prototipo se reducen los tiempos de actividad, se calculó el nivel de eficiencia utilizando la media de cada muestra. Los resultados se detallan en la siguiente tabla.

**Tabla 7.- Nivel de eficiencia del dispositivo.**

Media-Sin dispositivo	Media-Con dispositivo	Nivel de eficiencia
28.6718 segundos	9.6648 segundos	66.27 %



**Figura 36.-** Media de los datos con dispositivo y sin dispositivo.

La Figura 36, muestra claramente que, a lo largo de los días de prueba, los datos correspondientes al paciente "Con Dispositivo" (línea azul) se mantuvieron consistentes y con variaciones mínimas, indicando una tendencia hacia la estabilidad o un posible acoplamiento en el uso del dispositivo. Este comportamiento contrasta con los datos del paciente "Sin Dispositivo" (línea roja), donde las medias presentan fluctuaciones más pronunciadas, incluyendo un pico destacado en el día 14. Este resultado sugiere que el dispositivo podría haber contribuido a mejorar la consistencia en el desempeño del grupo, favoreciendo una adaptación progresiva o una respuesta más uniforme a lo largo del tiempo.

## CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

- El uso de la interfaz cerebro-computadora permitió una comunicación básica más rápida y efectiva en comparación con los métodos tradicionales, reduciendo significativamente los tiempos de respuesta del paciente.
- La implementación del filtro Butterworth y la normalización de señales EEG garantizó una mayor precisión en el procesamiento de datos, eliminando artefactos y ruido no deseado.
- El modelo de red neuronal artificial demostró ser eficaz en la clasificación de señales EEG, con una precisión alta para necesidades básicas como "ayuda" y "baño".
- La integración de la interfaz con Alexa mejora la calidad de vida de pacientes con síndrome de Angelman al ofrecer una comunicación accesible y personalizada.
- Mediante comandos específicos, permite expresar necesidades y activar rutinas automatizadas.
- La tecnología EEG utilizada en el proyecto, como la diadema Emotiv Epoc X, es una solución viable para la adquisición de señales cerebrales en entornos no invasivos.

- Se demostró que la prueba de t de Student es una herramienta estadística adecuada para validar la hipótesis de que el uso del dispositivo mejora la eficiencia en la comunicación.
- El sistema desarrollado alcanzó un nivel de eficiencia del 66.27 %, lo que confirma su potencial como alternativa en la mejora de la calidad de vida de personas con síndrome de Angelman.
- Los resultados obtenidos reflejan la importancia de las tecnologías asistidas en la inclusión social y la independencia de personas con discapacidades severas.
- Aunque los resultados son prometedores, se identificaron limitaciones relacionadas con el uso de un único paciente, lo que restringe la generalización de los hallazgos.

## **5.2.Recomendaciones**

- Ampliar el estudio a una población mayor para validar la efectividad y fiabilidad del sistema en diferentes casos y escenarios.
- Investigar el uso de otros modelos de machine learning para mejorar la precisión y adaptabilidad del sistema a patrones EEG más complejos.
- Explorar dispositivos EEG más avanzados o con mayor número de electrodos para captar señales de forma más detallada.
- Integrar funcionalidades adicionales al sistema, como detección de emociones o control de dispositivos domóticos avanzados para ayudar al paciente a desenvolverse mejor en su entorno.
- Realizar pruebas longitudinales para evaluar la estabilidad del sistema en el tiempo y su impacto a largo plazo en la calidad de vida del paciente.
- Implementar mejoras en la interfaz gráfica de usuario para facilitar el uso del sistema por parte de cuidadores o familiares.
- Evaluar el impacto emocional y social del dispositivo en el paciente y su entorno, con el objetivo de identificar posibles áreas de mejora.
- Realizar estudios comparativos con otras tecnologías asistidas para determinar las ventajas y desventajas del enfoque basado en BCI.
- Diseñar estrategias de bajo costo para hacer el sistema más accesible a familias con recursos limitados.
- Colaborar con instituciones médicas y de investigación para optimizar el uso del sistema en contextos clínicos y educativos.



## BIBLIOGRAFÍA

- [1] «Las ondas cerebrales, una tecnología para controlar la robótica en el hogar - El Día - Hemeroteca 06-01-2012», eldia.es. Accedido: 5 de agosto de 2024. [En línea]. Disponible en: <https://www.eldia.es/sociedad/2012-01-06/41-ondas-cerebrales-tecnologia-controlar-robotica-hogar.htm>
- [2] L. C. Faife Abril y I. V. Mayo Chirino, «Síndrome de Angelman», *Revista Cubana de Medicina General Integral*, vol. 28, n.º 3, pp. 331-339, sep. 2012, Accedido: 5 de agosto de 2024. [En línea]. Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_abstract&pid=S0864-21252012000300013&lng=es&nrm=iso&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S0864-21252012000300013&lng=es&nrm=iso&tlng=es)
- [3] Charles A. Williams, Sarika U. Peters, y Stephen N. Calculator, «Síndrome de angelman 3 29 10 - INFORMACION ACERCA DEL SINDROME DE ANGELMAN 7 ma Edición Enero 1, - Studocu». Accedido: 5 de agosto de 2024. [En línea]. Disponible en: <https://www.studocu.com/latam/document/universidad-politecnica-de-nicaragua/economia-de-empresas/sindrome-de-angelman-3-29-10/64414085>
- [4] D. A. Cobo-Álvarez, A. C. Veloz-Pico, y A. S. Samaniego-Vargas, «Síndrome de Angelman», *Revista Información Científica*, vol. 101, n.º 5, Art. n.º 5, dic. 2022, Accedido: 6 de agosto de 2024. [En línea]. Disponible en: <https://revinfcientifica.sld.cu/index.php/ric/article/view/4047>
- [5] M. F. Cordero-Molina, G. Escribano-Röber, M. Carvajal-Game, y R. Gaete, «Síndrome de Angelman: Correlación y seguimiento electroencefalográfico. | EBSCOhost», vol. 17, n.º 1, p. 3, 1 de enero de 2016. Accedido: 6 de agosto de 2024. [En línea]. Disponible en: <https://openurl.ebsco.com/contentitem/gcd:118323767?sid=ebsco:plink:crawler&id=ebsco:gcd:118323767>
- [6] Kaido Värbu, Naveed Muhammad, y Yar Muhammad, «Past, Present, and Future of EEG-Based BCI Applications». Accedido: 7 de agosto de 2024. [En línea]. Disponible en: <https://www.mdpi.com/1424-8220/22/9/3331>
- [7] J. Gutiérrez-Martínez, J. Cantillo-Negrete, R. I. Cariño-Escobar, y D. Elías-Viñas, «Los sistemas de interfaz cerebro-computadora: una herramienta para apoyar la rehabilitación de pacientes con discapacidad motora», *Investigación en Discapacidad*, vol. 2, n.º 2, pp. 62-69, 2013, Accedido: 6 de agosto de 2024. [En línea]. Disponible en: <https://www.medigraphic.com/cgi-bin/new/resumen.cgi?IDARTICULO=44426>
- [8] B. Medina, J. E. Sierra, y A. B. Ulloa, «Técnicas de extracción de características de señales EEG en la imaginación de movimiento para sistemas BCI», *Revista ESPACIOS*, vol. 39, n.º 22, may 2018, Accedido: 7 de agosto de 2024. [En línea]. Disponible en: <https://www.revistaespacios.com/a18v39n22/18392236.html>
- [9] I. Ortiz Ramírez, «Machine Learning para MI-BCI orientada al procesado de las señales EEG en tiempo real», 2021, Accedido: 7 de agosto de 2024. [En línea]. Disponible en: <https://hdl.handle.net/11441/126983>

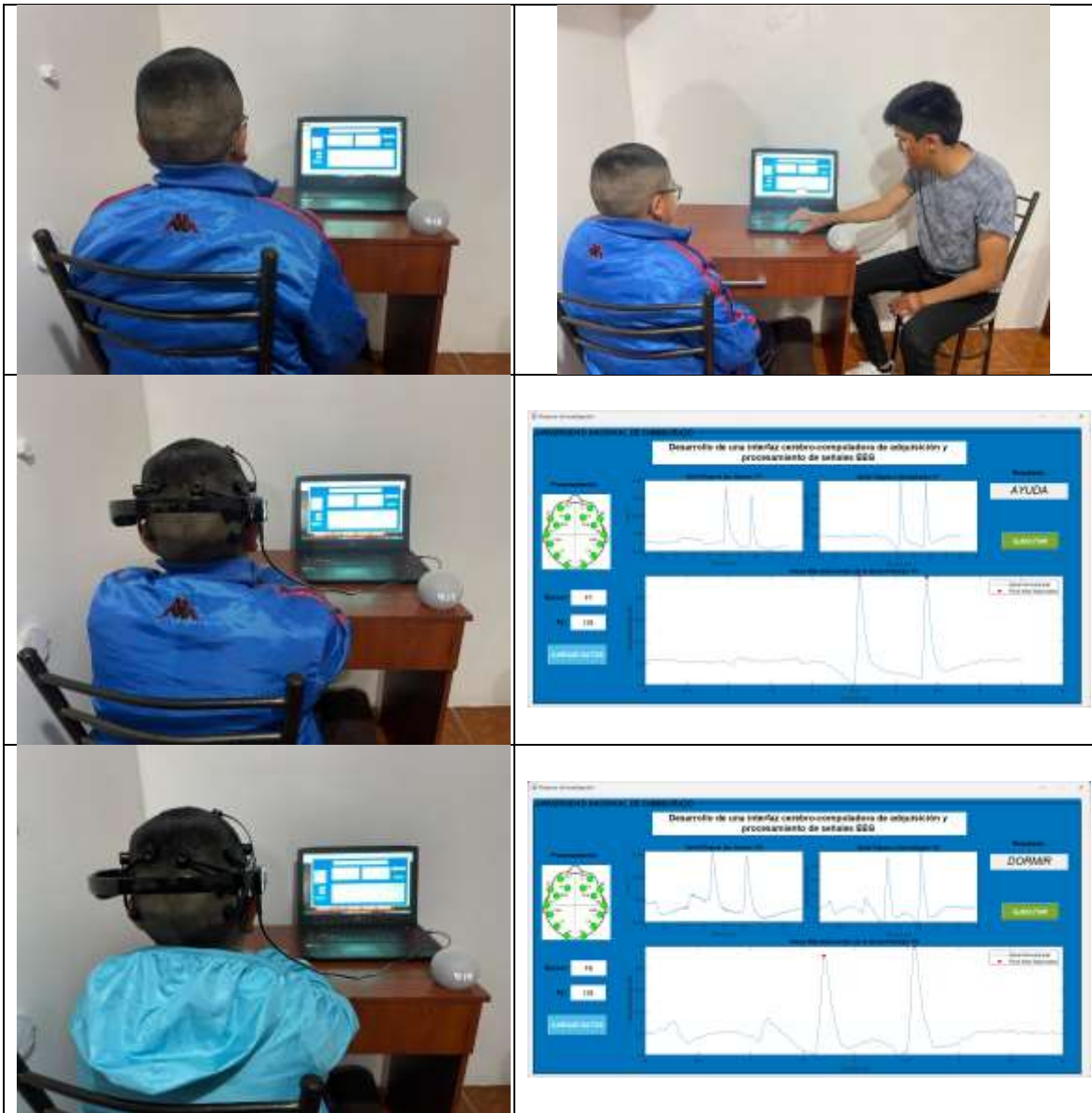
- [10] E. F. Corrales Bastidas, «Desarrollo de un sistema BCI basado en redes neuronales y movimientos de la cabeza para el manejo de un ordenador», bachelorThesis, 2021. Accedido: 8 de agosto de 2024. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/20288>
- [11] Solís L y Tapia J, «Sistema de control de brazo robótico mediante ondas cerebrales desarrollado en software libre para asistencia a personas con capacidades especiales». Accedido: 8 de agosto de 2024. [En línea]. Disponible en: <https://repositorio.espe.edu.ec/xmlui/handle/21000/10588?show=full>
- [12] W. D. Cunalata Velasco, «Sistema de manipulación y monitoreo del robot Scorbot mediante señales electroencefalográficas (EEG)», ago. 2021, Accedido: 12 de agosto de 2024. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/handle/123456789/33856>
- [13] C. Maranga, T. G. Fernandes, E. Bekman, y S. T. Da Rocha, «Angelman syndrome: a journey through the brain», *The FEBS Journal*, vol. 287, n.º 11, pp. 2154-2175, jun. 2020, doi: 10.1111/febs.15258.
- [14] E. M. Sepúlveda, P. L. Resa, S. D. Matute, R. Frontera, y E. C. Camacho, «Calidad de vida en familias con personas con síndrome de Angelman», *Revista INFAD de Psicología. International Journal of Developmental and Educational Psychology.*, vol. 1, n.º 1, Art. n.º 1, ago. 2021, doi: 10.17060/ijodaep.2021.n1.v1.2064.
- [15] J. R. Romero Guanoluisa y L. M. Moyolema Torres, «Intervención fisioterapéutica en niños con síndrome de Angelman», bachelorThesis, Universidad Ncional de Chimborazo, 2023. Accedido: 12 de agosto de 2024. [En línea]. Disponible en: <http://dspace.unach.edu.ec/handle/51000/10469>
- [16] P. Pindado Herráez, «Integración del dispositivo Emotiv EPOC en una aplicación Brain Computer Interfaze (BCI) asociada al control de un exoesqueleto», abr. 2021, Accedido: 2 de septiembre de 2024. [En línea]. Disponible en: <http://hdl.handle.net/10317/9324>
- [17] M. P. Rojas, «El cerebro que escribe: Aspectos cognitivos y estructurales», NeuroClass. Accedido: 4 de septiembre de 2024. [En línea]. Disponible en: <https://neuro-class.com/el-cerebro-que-escribe-aspectos-cognitivos-y-estructurales/>
- [18] C. Moratilla y J. Javier, «Análisis de emociones de sujetos a partir de señales de EEG, una aproximación con modelos de aprendizaje automático», *Analysis of emotions of subjects from EEG signals, an approach using machine learning models*, oct. 2022, Accedido: 5 de septiembre de 2024. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/187024>
- [19] «La Neuronas Tipos | PDF | Neurona | Axon», Scribd. Accedido: 10 de septiembre de 2024. [En línea]. Disponible en: <https://es.scribd.com/document/371090249/La-Neuronas-tipos>
- [20] B. E. H. Corona, «Introducción al Sistema Nervioso», Universidad de Guanajuato, 2021.
- [21] M. Herrero Arnedo, «Obtención de métricas sobre la actividad cerebral de usuarios a través de señales EEG y Machine Learning», Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València, 2023. Accedido: 14 de septiembre de 2024. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/198725>

- [22] C. Amo Usanos, «Actividad gamma del electroencefalograma. Métodos de análisis con objetivos clínicos», <http://purl.org/dc/dcmitype/Text>, Universidad de Alcalá, 2021. Accedido: 14 de septiembre de 2024. [En línea]. Disponible en: <https://produccioncientifica.ucm.es/documentos/61bd8499ddfffc606267d822>
- [23] «Electroencefalografía (EEG) - Mayo Clinic». Accedido: 15 de septiembre de 2024. [En línea]. Disponible en: <https://www.mayoclinic.org/es/tests-procedures/eeg/about/pac-20393875>
- [24] Carolina Coronado Alderete, «ANÁLISIS DE DATOS ELECTROENCEFALOGRÁFICOS DE RECUPERACIÓN EN PACIENTES CON TRAUMATISMO CRANEOENCEFÁLICO LEVE», Instituto Nacional de Astrofísica, Óptica y Electrónica, 2022. Accedido: 17 de septiembre de 2024. [En línea]. Disponible en: <http://inaoe.repositorioinstitucional.mx/jspui/handle/1009/2334>
- [25] A. M. Cruz-Rodríguez y H. Sánchez-Machet, «Prótesis de mano controlada con señales EEG», *Momento*, n.º 63, pp. 34-51, jul. 2021, doi: 10.15446/mo.n63.96407.
- [26] A. S. Guzmán y A. C. Ramírez, «Introduciendo SigProU, una interfaz gráfica de usuario para el análisis y procesamiento de señales EEG», *Ciencia Latina Revista Científica Multidisciplinar*, vol. 5, n.º 6, Art. n.º 6, dic. 2021, doi: 10.37811/cl\_rcm.v5i6.1307.
- [27] L. de Luis y M. Javier, «Procesamiento digital de señales de EEG para clasificación de trastornos psicóticos mediante aprendizaje de máquinas», 2021.
- [28] C. M. D. Mejía, «Institución Universitaria», Instituto Tecnológico Metropolitano, Colombia, 2024.
- [29] A. Valderrama Hickmann, «Clasificación de emociones en electroencefalograma con Wavelet Scattering y algoritmos de machine learning.», Universidad de Concepción, 2024. Accedido: 12 de octubre de 2024. [En línea]. Disponible en: <https://repositorio.udec.cl/handle/11594/10099>
- [30] F. A. I. Flores, D. L. C. Sanchez, R. O. E. Urbina, J. A. D. Soto, y S. E. V. Medrano, «Diseño e implementación de una red neuronal artificial para predecir el rendimiento académico en estudiantes de Ingeniería Civil de la UNIFSLB», *REVISTA VERITAS ET SCIENTIA - UPT*, vol. 10, n.º 1, Art. n.º 1, may 2021, doi: 10.47796/ves.v10i1.464.
- [31] F. A. Incio-Flores, D. L. Capuñay-Sanchez, R. O. Estela-Urbina, F. A. Incio-Flores, D. L. Capuñay-Sanchez, y R. O. Estela-Urbina, «Modelo de red neuronal artificial para predecir resultados académicos en la asignatura Matemática II», *Revista Electrónica Educare*, vol. 27, n.º 1, pp. 338-359, abr. 2023, doi: 10.15359/ree.27-1.14516.
- [32] X. B. Olabe, «REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES», 2021.
- [33] Fernando Izaurieta y Carlos Saavedra, «Redes Neuronales Artificiales», *Departamento de Física, Universidad de Concepción Chile*, 2022.
- [34] S. G. Pena, «Comparacion de metricas lineales y no lineales de las senales de EEG en la deteccion de situaciones de emergencia», Universidad de las Américas Puebla,

2022. Accedido: 19 de septiembre de 2024. [En línea]. Disponible en: [https://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lbm/garcia\\_pena\\_s/](https://catarina.udlap.mx/u_dl_a/tales/documentos/lbm/garcia_pena_s/)
- [35] D. S. González, J. Barragán, O. Mendoza-Montoya, y J. M. Antelis, «Análisis y clasificación de señales electroencefalográficas para el control de una órtesis robótica de mano con una interfaz cerebro computador basada en el paradigma de imaginación motora», 2023.
- [36] B. Clariana Riera, «Detección de períodos de crisis epilépticas en EEG: una comparación de algoritmos de clasificación supervisada de aprendizaje automático multiclase», Bachelor thesis, Universitat Politècnica de Catalunya, 2024. Accedido: 15 de enero de 2025. [En línea]. Disponible en: <https://upcommons.upc.edu/handle/2117/419598>
- [37] I. Moreno *et al.*, «Los sistemas de interfaz cerebro-computadora basado en EEG: características y aplicaciones», *I+D Tecnológico*, vol. 15, n.º 2, Art. n.º 2, jul. 2019, doi: 10.33412/idt.v15.2.2230.
- [38] «Interfaz Cerebro Computadora (BCI) - Libro online de IAAR». Accedido: 25 de septiembre de 2024. [En línea]. Disponible en: <https://iaarbook.github.io/interfaz-cerebro-computadora-BCI/>
- [39] B. Graimann, B. Z. Allison, y G. Pfurtscheller, *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. Springer Science & Business Media, 2022.
- [40] Daniel, «Machine Learning: definición, funcionamiento, usos», Formación en ciencia de datos | DataScientest.com. Accedido: 26 de septiembre de 2024. [En línea]. Disponible en: <https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos>
- [41] Esperanza Manrique Rojas, «Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo - ProQuest», Universidad Autónoma Baja California. [En línea]. Disponible en: <https://www.proquest.com/docview/2388304894?pq-origsite=gscholar&fromopenview=true&sourcetype=Scholarly%20Journals>

# ANEXOS

## Anexo I. Acondicionamiento a la interfaz cerebro-computadora.





## Anexo II. Programación de interfaz cerebro-computadora.

```

function varargout = procesamiento(varargin)
% PROCESAMIENTO MATLAB code for procesamiento.fig
%   PROCESAMIENTO, by itself, creates a new PROCESAMIENTO or raises
the existing
%   singleton*.
%
%   H = PROCESAMIENTO returns the handle to a new PROCESAMIENTO or
the handle to
%   the existing singleton*.
%
%   PROCESAMIENTO('CALLBACK', hObject,eventData,handles,...) calls
the local
%   function named CALLBACK in PROCESAMIENTO.M with the given input
arguments.
%
%   PROCESAMIENTO('Property','Value',...) creates a new
PROCESAMIENTO or raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before procesamiento_OpeningFcn gets called.
An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to procesamiento_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help procesamiento

```

```

% Last Modified by GUIDE v2.5 05-Nov-2024 11:39:09

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @procesamiento_OpeningFcn, ...
                  'gui_OutputFcn',  @procesamiento_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before procesamiento is made visible.
function procesamiento_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to procesamiento (see VARARGIN)

% Choose default command line output for procesamiento
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes procesamiento wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = procesamiento_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% Muestra la imagen en axes1 al iniciar la interfaz
axes(handles.axes2); % Activa el eje llamado axes1
img = imread('emotiv.png'); % Lee la imagen .jpg
imshow(img); % Muestra la imagen en el eje activo

% --- Executes on button press in procesar.
function procesar_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to procesar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Verifica si el archivo ya fue cargado
if ~isfield(handles, 'data')
    errordlg('Por favor, carga primero un archivo .csv.', 'Error');
    return;
end

% Obtén el nombre de la columna desde el Edit Text
columnName = get(handles.sensores, 'String');
prefixedColumnName = ['EEG_', columnName];

% Verifica si la columna con el prefijo existe en el archivo
cargado
if ismember(prefixedColumnName,
handles.data.Properties.VariableNames)
    % Extrae los datos de la columna
    columnData = handles.data.(prefixedColumnName);

    % Obtén la frecuencia de muestreo desde el Edit Text "fs"
    fs = str2double(get(handles.fs, 'String'));

    % Verifica si la frecuencia de muestreo es válida
    if isnan(fs) || fs <= 0
        errordlg('Por favor, ingresa un valor válido para la
frecuencia de muestreo.', 'Error');
        return;
    end

    % Limita la duración de la señal a 20 segundos
    samplesLimit = min(length(columnData), 20 * fs); % 20 segundos
    columnData = columnData(1:samplesLimit);

    % Calcula el vector de tiempo
    t = (0:length(columnData)-1) / fs;

    % Grafica la señal original en axes3
    axes(handles.axes3);
    plot(t, columnData);
    title(['Señal Original deL Sensor: ', columnName]); % Muestra
solo el nombre sin "EEG_"
    xlabel('Tiempo (s)');
    ylabel('Valor');

    % Filtrado pasabanda (ejemplo de 1-30 Hz)
    d = designfilt('bandpassiir', 'FilterOrder', 4, ...
        'HalfPowerFrequency1', 1, 'HalfPowerFrequency2',
30, ...
        'SampleRate', fs);
    filteredData = filtfilt(d, columnData);

    % Filtrado adicional: Filtro de mediana para eliminar
artefactos
    medianFilteredData = medfilt1(filteredData, 3); % Ventana de 3
puntos

    % Normalización de la señal
    normalizedData = (medianFilteredData - min(medianFilteredData))
/ (max(medianFilteredData) - min(medianFilteredData));

```



```

        % Grafica la señal filtrada y normalizada en axes5
        axes(handles.axes5);
        plot(t, normalizedData);
        title(['Señal Filtrada y Normalizada: ', columnName]); %
Muestra solo el nombre sin "EEG_"
        xlabel('Tiempo (s)');
        ylabel('Valor Normalizado');

        % Encontrar picos más relevantes en la señal filtrada
        minHeight = 0.5; % Ajusta este valor según sea necesario para
seleccionar solo picos relevantes
        [picos, locs] = findpeaks(normalizedData, 'MinPeakHeight',
minHeight, 'MinPeakDistance', fs/2);

        % Graficar los picos relevantes en el axes4
        axes(handles.axes4);
        plot(t, normalizedData); hold on;
        plot(t(locs), picos, 'rv', 'MarkerFaceColor', 'r'); % Marca los
picos con triángulos rojos
        title(['Picos Más Relevantes de la Señal Filtrada: ',
columnName]);
        xlabel('Tiempo (s)');
        ylabel('Valor Normalizado');
        legend('Señal Normalizada', 'Picos Más Relevantes');
        hold off;

        % Mostrar la cantidad de picos relevantes encontrados en un
cuadro de diálogo
        % msgbox(['Total de picos relevantes encontrados: ',
num2str(length(picos))], 'Conteo de Picos Relevantes');

        else
            errordlg('La columna especificada no existe en el archivo.',
'Error');
        end

        % === PARTE 4: ENTRENAMIENTO Y PREDICCIÓN ===
        % Ruta a la carpeta con los archivos
        carpeta = 'C:\Users\edi_o\OneDrive\Escritorio\Codigo\Archivos';
        etiquetas = [repelem("ayuda", 15), repelem("baño", 15),
repelem("comer", 15), repelem("dormir", 15)];
        columnasInteres = [6, 17]; % Columnas relevantes

        % Convertir etiquetas
        etiquetas = categorical(etiquetas);

        % Procesar los archivos
        [features, labels] = procesarArchivos(carpeta, etiquetas,
columnasInteres);

        % Verificar dimensiones
        if size(features, 1) ~= numel(etiquetas)
            error('Las dimensiones no coinciden entre características y
etiquetas.');
```

```

% Entrenar modelo ANN
hiddenLayerSize = 20;
mdlANN = patternnet(hiddenLayerSize);
[trainInd, ~, ~] = dividerand(size(features, 1), 0.80, 0.10, 0.10);
mdlANN = train(mdlANN, features(trainInd, :)', labelsOneHot(:,
trainInd));

% Predicción del nuevo archivo
nuevoArchivo = handles.data; % Archivo cargado desde GUIDE
[nuevoFeatureVector, ~, ~] = extraerCaracteristicas(nuevoArchivo,
columnasInteres);

% Validar dimensiones
if size(nuevoFeatureVector, 2) ~= size(features, 2)
    error('Dimensiones no coinciden con las características del
modelo.');
```

```

end

% Realizar predicción
prediccion = mdlANN(nuevoFeatureVector)';
[~, clasePredicha] = max(prediccion, [], 2);
clasesUnicas = unique(etiquetas);
clasePredicha = clasesUnicas(clasePredicha);

% Mostrar la clase predicha
% msgbox(['La mejor coincidencia para el archivo es: ',
char(clasePredicha)], 'Predicción de Clase');
```

```

set(handles.coincidencia, 'String', upper(char(clasePredicha)));

% Reproducir el mensaje asociado a la clase
reproducirMensaje(clasePredicha);

function sensores_Callback(hObject, eventdata, handles)
% hObject    handle to sensores (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sensores as text
%         str2double(get(hObject,'String')) returns contents of sensores
as a double

% --- Executes during object creation, after setting all properties.
function sensores_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sensores (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function fs_Callback(hObject, eventdata, handles)
% hObject    handle to fs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fs as text
%        str2double(get(hObject,'String')) returns contents of fs as a
double

% --- Executes during object creation, after setting all properties.
function fs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cargar.
function cargar_Callback(hObject, eventdata, handles)
% hObject    handle to cargar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Abre un cuadro de diálogo para seleccionar el archivo .csv
[file, path] = uigetfile('*.csv', 'Selecciona un archivo CSV');

% Verifica si el usuario seleccionó un archivo
if isequal(file, 0)
    disp('No se seleccionó ningún archivo.');
```

```

else
    % Lee el archivo CSV solo una vez y guárdalo en handles
    fullFileName = fullfile(path, file);
    handles.data = readtable(fullFileName); % Guarda los datos en
handles

    % Actualiza los handles para almacenar la información
    guidata(hObject, handles);

%
    msgbox('Archivo cargado exitosamente. Puedes ingresar el
nombre del sensor.', 'Archivo Cargado');
end

```

### Anexo III. Función para extraer las características.

```
function [featureVector, originalData, filteredDataArray,
normalizedDataArray] = extraerCaracteristicas(dataTable,
columnasInteres)
    % Parámetros del filtro
    fs = 128; % Frecuencia de muestreo
    lowCutoff = 1; % Frecuencia de corte baja
    highCutoff = 30; % Frecuencia de corte alta
    [b, a] = butter(4, [lowCutoff, highCutoff] / (fs / 2), 'bandpass');
    % Filtro pasa-banda

    % Extraer los datos de las columnas de interés
    originalData = table2array(dataTable(:, columnasInteres)); % Extrae
los datos de las columnas de interés

    % Filtrar datos con filtfilt (filtrado pasabanda)
    filteredDataArray = filtfilt(b, a, originalData);

    % Filtrado adicional: Filtro de mediana para eliminar artefactos
medianFilteredData = medfilt1(filteredDataArray, 3); % Ventana de 3
puntos

    % Normalización de la señal
normalizedDataArray = (medianFilteredData -
min(medianFilteredData)) ./ ...
(max(medianFilteredData) -
min(medianFilteredData));

    % Inicializar el vector de características
numColumns = size(filteredDataArray, 2);
featureVector = [];

    % Extraer características de cada columna de EEG
for j = 1:numColumns
    % Detectar los picos relevantes
    minHeight = 0.5; % Ajustar este valor según sea necesario para
seleccionar solo picos relevantes

    [pks, locs] = findpeaks(normalizedDataArray(:, j),
'MinPeakHeight', minHeight, 'MinPeakDistance', fs/2);

    % Verificar si se encontraron picos
    if isempty(pks)
        featureVector = [featureVector, NaN, NaN]; % Si no se
encuentran picos, añadir NaN
    else
        % Establecer umbral para los picos relevantes
        threshold = max(pks) * 0.7; % Umbral del 70% del máximo
relevantPeaks = pks(pks >= threshold); % Picos por encima
del umbral

        % Si no hay picos relevantes, agregar valores NaN
        if isempty(relevantPeaks)
            featureVector = [featureVector, NaN, NaN];
        else
            % Extraer características de los picos relevantes
            featureVector = [featureVector, mean(relevantPeaks),
std(relevantPeaks)];
        end
    end
end
end
```

```
        % Solo extraer estadísticas de la señal normalizada
        meanEEG = mean(normalizedDataArray(:, j)); % Promedio
        stdEEG = std(normalizedDataArray(:, j)); % Desviación
estándar

        % Añadir las características estadísticas al vector
        featureVector = [featureVector, meanEEG, stdEEG];

    end
end
```

#### Anexo IV. Función para procesar los archivos,

```
function [features, labels] = procesarArchivos(carpeta, etiquetas,
columnasInteres)
    % Obtener una lista de todos los archivos en la carpeta
    archivos = dir(fullfile(carpeta, '*.csv'));
    numArchivos = numel(archivos);

    % Inicializar las variables de salida
    features = [];
    labels = [];

    for i = 1:numArchivos
        % Leer el archivo CSV
        archivoPath = fullfile(carpeta, archivos(i).name);
        data = readtable(archivoPath);

        % Filtrar las columnas de interés
        eegData = data(:, columnasInteres); % Asumiendo que las
columnas de EEG están en las posiciones 5:18

        % Extraer características, por ejemplo, estadísticas básicas de
las señales
        % Esto se puede cambiar según las características que desees
extraer
        meanEEG = mean(eegData, 1); % Promedio de cada columna
        stdEEG = std(eegData, 0, 1); % Desviación estándar de cada
columna
        maxEEG = max(eegData, [], 1); % Máximo de cada columna
        minEEG = min(eegData, [], 1); % Mínimo de cada columna

        % Combinar las características
        archivoFeatures = [meanEEG, stdEEG, maxEEG, minEEG];

        % Almacenar las características y las etiquetas
correspondientes
        features = [features; archivoFeatures];
        labels = [labels; etiquetas(i)];
    end

    % Imprimir las dimensiones de las características y las etiquetas
    % disp(['Dimensiones de features: ', num2str(size(features))]);
    % disp(['Número de muestras (filas de features): ',
num2str(size(features, 1))]);
    % disp(['Número de etiquetas: ', num2str(numel(labels))]);

    % Verificar si el número de etiquetas coincide con el número de
muestras
    if numel(labels) ~= size(features, 1)
        error('El número de etiquetas no coincide con el número de
muestras en features.');
```

## Anexo V. Función para reproducir el mensaje.

```
function reproducirMensaje(clasePredicha)
    switch clasePredicha
        case 'ayuda'
            % Mensaje y audio para Ayuda
            msg = 'Necesito ayuda, por favor';
            audioFile = 'ayuda.wav';

        case 'baño'
            % Mensaje y audio para Baño
            msg = 'Necesito ir al baño, por favor';
            audioFile = 'bano.wav';

        case 'comer'
            % Mensaje y audio para Comer
            msg = '¿Podrías traerme algo de comida?, por favor';
            audioFile = 'comer.wav';

        case 'dormir'
            % Mensaje y audio para Dormir
            msg = 'Quiero dormir, por favor';
            audioFile = 'dormir.wav';

        otherwise
            % Caso sin mensaje asociado
            msg = 'Sin mensaje asociado';
            audioFile = '';
    end

    % Generar el archivo de audio con PowerShell si existe un mensaje
    if ~isempty(audioFile)
        system(['PowerShell -Command "Add-Type -AssemblyName
System.Speech; ', ...
            '$synthesizer = New-Object
System.Speech.Synthesis.SpeechSynthesizer; ', ...
            '$synthesizer.SelectVoice('Microsoft Sabina Desktop'); ',
...
            '$synthesizer.SetOutputToWaveFile('', audioFile, ''); ',
...
            '$synthesizer.Speak('', msg, '');"']);

        % Esperar un poco para la generación del archivo
        pause(2);

        % Reproducir el archivo de audio si existe
        if isfile(audioFile)
            [y, Fs] = audioread(audioFile);
            sound(y, Fs);
        end
    else
        disp('No hay mensaje asociado para la clase predicha.');
```