



**ANÁLISIS DE SEÑALES EEG CON POTENCIAL USO EN EL CONTROL DE  
UNA MANO ROBÓTICA POR MEDIO DE LA APLICACIÓN DE IMAGINACIÓN  
MOTORA**

**Dany Fernando López Franco  
Jose Manuel Balanta Quintero**

Universidad Antonio Nariño  
Facultad de Ingeniería Mecánica, Electrónica y Biomédica  
Popayán, Colombia  
2021

**ANÁLISIS DE SEÑALES EEG CON POTENCIAL USO EN EL CONTROL DE  
UNA MANO ROBÓTICA POR MEDIO DE LA APLICACIÓN DE IMAGINACIÓN  
MOTORA**

**Dany Fernando López Franco  
Jose Manuel Balanta Quintero**

Proyecto de grado presentado como requisito parcial para optar al título de:

**Ingeniero Biomédico**

Director (a):

Ph.D. Cesar Augusto Quinayas Burgos

Línea de Investigación:

**INGENIERÍA DE REHABILITACIÓN**

Grupo de Investigación:

**BIOINGENIERÍA**

Universidad Antonio Nariño

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Popayán, Colombia

2021

*¡Que nadie se quede afuera, se los dedico a todos!*

*Primero le doy gracias a Dios por permitirme acabar esta etapa de mi vida rodeado de personas maravillosas; le dedico este logro a mi madre Luz Miryam Quintero Bolaños que se con certeza que me mira desde el cielo y ha sido el motor más grande que he tenido durante mi universidad; a mi padre Jose Jhon Balanta Hernández porque con su ejemplo me motivaba a salir adelante siempre, en los momentos difíciles él estuvo empujándome para no dejarme derrotar; a mi hermano Jhon Eduard Balanta Quintero por ser siempre mi apoyo incondicional en todo momento de mi vida; a mi madre terrenal Elizabeth Quintero Bolaños porque ha tomado su rol con una sabiduría inmensa, su sentir de madre nunca me desamparó; a mi novia Stefany Molina Zapata quien fue testigo de los tropiezos, trasnochos y frustradas que trajo esta meta, y finalmente a todos mis familiares que partieron antes de tiempo, todos mis logros serán dedicados a ellos porque si estuvieran aquí desearía hacerles sentir un poco de orgullo familiar.*

*Att: Jose Manuel Balanta Quintero*

*Este trabajo de grado va dedicado principalmente a Dios quien me supo guiar con su bendición y así poder enfrentar todos y cada uno de las adversidades que se presentaron durante mi carrera, enseñándome a no desfallecer en el intento.*

*A Myriam Franco, mi madre y principal cimiento para la construcción de mi vida profesional y personal forjándome en los valores como el respeto, responsabilidad, superación, enseñándome que nunca te debes dar por vencido.*

*A mí familia, que me enseñaron cómo dar mis primeros pasos e irme forjando en una persona con valores y principios.*

*A Valeria Flores, mi novia. Quien creyó en mis capacidades y vivió a mi lado las altas y bajas de esta formación.*

*Esto es para Uds y por Uds.*

*Att: Dany Fernando López Franco*

## **Agradecimientos**

Al finalizar esta etapa de formación vital para nuestras vidas, principalmente queremos agradecer a Dios por guiarnos a lo largo de nuestra carrera, por ser nuestro apoyo y fortaleza en aquellos momentos de dificultad y debilidad para obtener uno de los anhelos más deseados, **SER UN PROFESIONAL**.

Gracias a nuestros padres: **Myriam Amparo Franco, Jhon Balanta Hernandez y Elizabeth Quintero Bolaños** por ser los principales promotores de nuestros sueños, por confiar y creer en nuestras expectativas, por los consejos, valores y principios que nos han inculcado y quienes con su amor, paciencia y esfuerzo nos han permitido llegar a cumplir hoy un sueño más.

De igual manera agradecer a la Universidad Antonio Nariño, a nuestros profesores, pero en especial al **Phd. Cesar Augusto Quinayas**, quien fue el principal colaborador, el cual con su rectitud como docente y director imparte su conocimiento y enseñanza durante todo este proceso permitiéndonos el desarrollo tanto profesional como ser humano.

## Resumen

La presente obra a título de proyecto de grado se fundamenta en el análisis de señales de electroencefalografía (EEG) bajo la aplicación de imaginación motora (MI), específicamente en la intención del movimiento de la mano, generando de esta manera una interfaz cerebro-computadora. Las señales a estudiar fueron descargadas de la base de datos gratuita EEG Motor Movement/Imagery Dataset (eegmddb) de PHYSIONET, estas fueron adquiridas por medio del sistema BCI2000 donde los electrodos se ubican de manera superficial en pacientes sanos.

La extracción de características para aplicar la estrategia mental (Imaginación motora), se realizará por medio de la aplicación de la transformada Wavelet discreta (daubechies) y para su clasificación se utilizan dos modelos basados en: máquinas de soporte vectorial y una red neuronal feedforward, visualizando la precisión de cada uno de los modelos empleados en manera de aciertos y fallos a través de una matriz de confusión.

**Palabras clave:** Imaginación motora (MI), mano robótica, electroencefalografía (EEG), interfaz cerebro-computadora, sistema BCI2000, transformada Wavelet continua, physionet.

**Abstract**

This work as a degree project is based on the analysis of electroencephalography (EEG) signals under the application of motor imagination (MI), specifically on the intention of hand movement, thus generating a brain-computer interface. The signals to be studied were downloaded from the free database EEG Motor Movement / Imagery Dataset (eegmmidb) of PHYSIONET, these were acquired through the BCI2000 system where the electrodes are located superficially in healthy patients.

The extraction of characteristics to apply the mental strategy (motor imagination), will be carried out through the application of the discrete Wavelet transform (daubechies) and for its classification two models are used based on: vector support machines and a feedforward neural network, visualizing the precision of each of the models used in the way of successes and failures through a confusion matrix.

**Keywords:** Motor imagery (MI), robotic hand, electroencephalography (EEG), brain-computer interface, BCI2000 system, continuous Wavelet transform, physionet.

## Contenido

	<b>Pág.</b>
<b>1. Planteamiento del problema .....</b>	<b>4</b>
3.1. Justificación .....	6
1.2. Objetivos.....	7
1.1.1. Objetivo General.....	7
1.1.2. Objetivos Específicos .....	7
<b>2. Antecedentes .....</b>	<b>8</b>
2.1. Clasificación de las funciones motoras a partir de señales de electroencefalograma (EEG) basada en un método integrado compuesto por un patrón espacial común y un marco de transformación de ondas .....	8
2.2. Clasificación de señal de EEG de imágenes de motor de cuatro clases utilizando PCA, Wavelet y red neuronal de dos etapas:.....	10
2.3. Clasificación de las señales de EEG de imágenes motoras basadas en la transformada de ondas y la entropía de la muestra .....	12
2.4. Clasificación de las señales de EEG de imágenes motoras mediante el aprendizaje automático .....	12
2.5. Clasificación de las señales de EEG de imágenes motoras con red neuronal convolucional de múltiples entradas mediante el aumento de STFT:.....	13
2.6. Clasificación de señales de eeg de imágenes motoras con filtrado CSP a través de modelos de redes neuronales:.....	14
2.7. Clasificación de la señal de EEG de imágenes motoras multiclase utilizando un enfoque de dispersión .....	15
2.8. Interfaz cerebro computador modular basada en la interpretación del electroencefalograma (EEG) mediante RNA para el control de dispositivos electrónicos: .....	16
<b>3. Marco Conceptual .....</b>	<b>18</b>
3.1. Señales electroencefalográficas (EEG).....	18
3.1.1. Sistemas de ubicación de electrodos (10/20, 10/10 y 10/5).....	19
3.2. Ondas o Ritmos Cerebrales.....	23
3.3. Sistemas BCI.....	25
3.3.1. Cuadro comparativo de cascos de adquisición .....	26
3.4. Imaginación Motora.....	28
3.5. Transformada Wavelet (WT).....	29
3.5.1 Funcionamiento de la Transformada Wavelet .....	29
3.5.2 Familias de las wavelets .....	30
3.5.3 Transformada wavelet continua (CWT) .....	33
3.5.4 Transformada Wavelet Discreta (DWT).....	33
3.6. Transformada de Fourier (FFT) .....	34
3.7. Identificación de Patrones.....	34
3.8. Red neuronal artificial tipo feedforward (RNAf) .....	35
3.9. Métodos de Aprendizaje .....	36
3.10. Matriz de Confusión .....	38
3.11. Máquina de soporte Vectorial .....	39
3.12. Validación Cruzada .....	40
<b>4. Materiales y Métodos .....</b>	<b>42</b>

Contenido	8
4.1. Base de Datos.....	42
4.2. Selección de Canales .....	45
4.3. Diseño y Desarrollo.....	46
4.3.1. Módulo Computacional .....	46
4.3.2. Descripción del algoritmo.....	49
<b>5. Análisis y Resultados.....</b>	<b>64</b>
5.1 Resultados de la Matriz de confusión.....	64
5.2 Resultados de Clasificadores .....	66
5.3 Resultados de Precisión .....	67
<b>5. Conclusiones y recomendaciones .....</b>	<b>69</b>
5.1. Conclusiones.....	69
5.2. Recomendaciones .....	70
<b>A. Anexo: CODIGO IMPLEMENTADO .....</b>	<b>71</b>
<b>Bibliografía .....</b>	<b>84</b>

## I. Lista de Figuras

	<b>Pág.</b>
<b>Figura 1.</b> Marco general para la clasificación de eventos de electroencefalograma motor (EEG) utilizando un filtro de patrón espacial común (CSP) y una transformada de ondícula continua (CWT).....	10
<b>Figura 2.</b> Procedimiento de adquisición de datos de EEG de MI utilizando el dispositivo inalámbrico B-Alert X-10 con un mando asistido por ordenador.....	11
<b>Figura 3.</b> Diagrama de bloques de la metodología propuesta en la clasificación de señales EEG utilizando enfoque de dispersión .....	16
<b>Figura 4.</b> Modelo de referencia para el diseño de interfaces cerebrales .....	17
<b>Figura 5.</b> Diagrama de adquisición de señales cerebrales.....	19
<b>Figura 6.</b> Vista de perfil sistema 10/20.....	20
<b>Figura 7.</b> Vista superior sistema 10/20 .....	21
<b>Figura 8.</b> Relación entre funciones cerebrales y electrodos para EEG .....	22
<b>Figura 9.</b> Áreas de actividad motora en el encéfalo .....	28
<b>Figura 10.</b> Funcionamiento de la Transformada Wavelet.....	30
<b>Figura 11.</b> Familias de las ondículas wavelet.....	31
<b>Figura 12.</b> Familia de ondículas Daubechies en diferentes niveles de descomposición .....	32
<b>Figura 13.</b> Estructura típica de una red neuronal artificial tipo feedforward (RNAf). .....	36
<b>Figura 14.</b> Esquema de una red neuronal feed-forward o MLP.....	38
<b>Figura 15.</b> Matriz de confusión para un clasificador en dos clases.....	39
<b>Figura 16.</b> Hiperplano óptimo obtenido por SVM para dos clases.....	40
<b>Figura 17.</b> Validación cruzada usando iteraciones.....	41
<b>Figura 18.</b> Ubicación de los 64 electrodos. ....	44
<b>Figura 19.</b> Selección de electrodos C3, C4 y Cz, los cuales registran el movimiento de las manos y los pies. ....	45
<b>Figura 20:</b> Descomposición multiresolución de la señal EEG del canal C3.....	48
<b>Figura 21.</b> Diagrama de bloques de la metodología implementada .....	49
<b>Figura 22.</b> Gráficas de señales EEG puras para cada canal (C3,C4 y CZ). ....	54
<b>Figura 23.</b> Transformada de Fourier para cada tarea del canal C3.....	55
<b>Figura 24.</b> Transformada de Fourier para cada tarea del canal C4.....	55
<b>Figura 25.</b> Transformada de Fourier para cada tarea del canal Cz.....	56
<b>Figura 26.</b> Transformada wavelet para cada tarea del canal C3.....	57
<b>Figura 27.</b> Separación de bandas de la tarea T1 aplicando DWT para el canal C3 .....	58

---

<b>Figura 28.</b> Separación de bandas de la tarea T2 aplicando DWT para el canal C3 .....	59
<b>Figura 29.</b> Matriz de confusión de la SVM. ....	63
<b>Figura 30.</b> Clasificación de la Red Neuronal .....	67

**II. Lista de tablas**

	<b>Pág.</b>
<b>Tabla 1.</b> <i>Relación de convenciones entre funciones cerebrales y electrodos para EEG</i> .....	22
<b>Tabla 2.</b> <i>Ondas o Ritmos Cerebrales</i> .....	23
<b>Tabla 3.</b> <i>Comparación entre cascos de adquisición</i> .....	26
<b>Tabla 4.</b> <i>Tabla de resultados de clasificadores SVM y Red Neuronal</i> .....	64
<b>Tabla 5.</b> <i>Tabla de resultados de matriz de confusión</i> .....	65

## **Introducción**

Inicialmente la electroencefalografía ha sido una técnica utilizada para entender el comportamiento y funcionamiento del cerebro a través de la interpretación de la actividad cerebral obtenida en el cuero cabelludo. Por tal razón, las señales electroencefalográficas (EEG) se estudian con el fin de descifrar diferentes intenciones de los usuarios quienes podrían ser capaces de controlar dispositivos externos por medio de su imaginación. Este es un concepto conocido como interfaz cerebro-computador (BCI), es un sistema cuyo proceso de operación es capaz de interpretar los pensamientos de los usuarios y transformarlos en interacciones con el mundo real o virtual (DANE, 2020).

Existe una gran variedad de limitaciones en el uso de la señal EEG en aplicaciones de la BCI, entre éstas se puede mencionar la cantidad limitada de estados mentales que se pueden clasificar, los artefactos que contaminan la EEG, la baja tasa de transferencia de información (ITR) y el porcentaje limitado de efectividad en la identificación de la intencionalidad del usuario (N. Padfield; Et al., 2019) La intención de movimiento hace referencia a la cognición consciente del usuario en la preparación y ejecución de un movimiento de la extremidad a estudiar (Muller-Putz, Schwarz, Pereira, & Ofner, 2016).

Para el control de sistemas BCI la intención de movimiento puede ser identificada a partir de las señales de EEG, lo cual proporciona una retroalimentación al equipo de entrenamiento, promoviendo la participación de los pacientes y la motivación para la rehabilitación (Ang, Et al., 2014). Para aumentar la eficacia de la rehabilitación, los pacientes deben realizar un entrenamiento con intención de movimiento, principalmente en las personas que han sufrido accidentes cerebrovasculares (Blank, French, Pehlivan, & O'Malley, 2015).

Como resultado de algunas investigaciones, los sistemas BCI han logrado permitir que personas con parálisis en sus extremidades superiores e inferiores puedan desarrollar tareas tales como: escribir en un monitor, manejar cubiertos, desplazarse en una silla de ruedas o manejar prótesis, etc. En resumen, los BCI representan un camino de comunicación directo entre el cerebro y el dispositivo a controlar (Padfield, Et al., 2019). Las características de las señales EEG permiten

---

realizar un análisis profundo para detallar las señales convenientes, es decir aquellas que están relacionadas con las intenciones del usuario y además serán usadas para la traducción en comandos de salida.

En esta etapa los BCI aún presentan una eficiencia restringida, ya que adquirir las señales por métodos no invasivos trae consigo artefactos como el ruido que no son más que señales generadas por movimientos involuntarios como: ocular, parpadeo, respiración, contracción muscular, latidos del corazón, entre otros, limitando al sistema a realizar una correcta caracterización de los fenómenos fisiológicos por su baja resolución espacial derivando esto en un bajo acierto de las acciones que el usuario quiera efectuar con el dispositivo final (Arboleda, Et. Al, 2009).

En este trabajo se propone caracterizar y clasificar señales cognitivas de registros EEG, usando técnicas de caracterización basadas en información espectral, así como de reconocimiento de patrones. Para este propósito es necesario trabajar con registros de señales EEG presentes en una base de datos gratuita y ampliamente usada por la comunidad científica conocida como PhysioNet, fue creada por los desarrolladores de la BCI2000 y está constituida por 1500 registros de 64 canales de EEG muestreados a 160Hz y obtenidas de 109 usuarios sanos, donde cada uno de ellos proporciona 14 registros correspondientes a diferentes tareas motoras e imaginarias.

Para la extracción de características se implementan métodos de análisis espectral como: la transformada de Fourier, transformada de wavelet, modelos autorregresivos, filtrado espacial y métodos no lineales entre otros. La selección y clasificación de las características se realizan haciendo un estudio de reconocimiento de patrones. Posteriormente, se procesan las señales EEG a través de la RNAf que será la encargada de decidir o traducir las intenciones o pensamientos del usuario en salidas directamente aplicables o reconocibles. En cuanto a la clasificación de las señales tratadas se tomaron medidas estadísticas en términos de precisión tanto para las tareas ejecutadas físicamente como para

---

las dos imaginarias que corresponden a apertura/cierre de la mano derecha o izquierda y se establece un acierto final superior al 70% (Hernández González, 2017).

---

## 1. Planteamiento del problema

En la actualidad, Colombia es un país en el que la tasa de muerte por accidentes cerebrovasculares afecta al 7% de la población (DANE, 2020), es necesario aclarar que las personas con este tipo de accidente deben realizar un proceso de rehabilitación mediante un entrenamiento que involucra la intención del movimiento (Blank, French, Pehlivan, & O'Malley, 2015). Según el ministerio de salud, el 2,9% de la población colombiana sufre de alguna discapacidad, donde el 38% de esta tiene serias dificultades al momento de realizar movimientos corporales (manos, piernas y brazos) (Muller-Putz, Et al., 2016).

La implementación de los sistemas de Interfaz Cerebro-Ordenador, BCI, por sus siglas en inglés *Brain-Computer Interface*, se han incrementado en las últimas décadas (Arboleda, Et al., 2009), estos son una herramienta de la ingeniería de rehabilitación que permite restaurar las funciones de comunicación y control de personas que presentan una discapacidad. El sistema descrito previamente, presenta un problema que puede afectar a la hora de captar las señales ya que el largo periodo de entrenamiento (algunas semanas o meses) requerido para que un usuario pueda utilizar una BCI este produce señales de forma poco consciente afectando las muestras, dificultando de una manera mínima la clasificación, identificación y extracción de patrones.

El uso de sistemas BCI, permite clasificar eventos relacionados a tareas específicas de EEG, procesar las señales y registrar la actividad cerebral (Arboleda, y otros, 2009). Cuando el sistema BCI está enfocado en una sola modalidad de aplicación genera solo un comando de control el cual puede ser extraído de múltiples clases en un estudio, facilitando de esta manera su decodificación, pero arrojando un resultado en el movimiento de menor calidad y eficiencia (Pfurtscheller, 2004). Como consecuencia a la variedad de usuarios y a que algunos de estos presentan dificultad para generar las señales necesarias por lograr una tarea motora; el enfoque del sistema BCI basado en una sola

---

modalidad no permite satisfacer los requerimientos de las tareas EEG establecidas (Kandel, 2012)

Los sistemas BCI basados exclusivamente en la implementación de señales EEG, aún presentan una baja precisión en la transferencia de información y aceptabilidad del usuario para ser incluidos en terapias de rehabilitación desarrolladas con intención de movimiento (Ma, Et al., 2017) o para la restauración de alguna función motora perdida. Para esta razón, se han propuesto diversos sistemas (hBCI) los cuales emplean características de múltiples bioseñales que a través de comandos mejoran la clasificación en la identificación de tareas motoras específicas (Amiri, Fazel-Rezai, & Asadpour, 2013), actualmente se han reportado algunas mejoras en la decodificación de EEG utilizado sistemas (hBCI) que una sola modalidad de BCI (Ma, Et al., 2017), sin embargo este campo de aplicación sigue bastante abierto, ya que esta es una tecnología reciente.

En este proyecto de grado se propone un estudio que permita establecer la integración neuronal de la conectividad de las señales de EEG obtenidas de la base datos (PhysioNet) con la imaginación motora (MI), el cual consiste en la evocación, por parte del usuario, de un movimiento o gesto para aprender o mejorar su ejecución activando patrones neuronales similares a aprender o mejorar las acciones o gestos realizados. La imaginación motora combinada con terapia convencional (fisioterapia o terapia ocupacional), ha demostrado tener efectos positivos en la rehabilitación motora de las extremidades superiores entre los pacientes puesto a prueba. (Zhang, Et al., 2019)

### **Pregunta de investigación**

¿Cuál es el procedimiento necesario para la implementación de un clasificador basado en redes neuronales facultado para extraer características y clasificar movimientos que puedan ser asociados al control de una mano robótica

---

provenientes de las señales electroencefalográficas (EEG) obtenidas de la aplicación de la técnica de imaginación motora (MI)?

### **3.1. Justificación**

Es importante desarrollar este proyecto ya que, según el DANE, actualmente en Colombia y en todo el departamento del Cauca existe un total de 13.915 personas que presentan discapacidad. Por otra parte, se encontró que existe un total de 2.772 personas que han sufrido alteraciones en funciones o estructuras corporales (brazos, manos, piernas). (DANE, 2020)

No se tiene conocimiento de estadísticas exactas que puedan llevar a identificar la población mínima que presenta discapacidad en estas funciones corporales; por otra parte, es necesario profundizar en la técnica de imaginación motora a través de señales EEG en aras de identificar las intenciones potenciales que permitan controlar una prótesis de mano, siendo esto un aporte significativo desde la Ingeniería Biomédica para que las personas que presentan esta condición de discapacidad puedan avanzar en el rango de movimiento de los miembros afectados, proporcionando funcionalidad y mejoramiento de la calidad de vida, logrando así una solución y un apoyo a la reintegración social, familiar y laboral. (Fernández-Gómez & Sánchez-Cabeza, 2018)

Por otro lado, esta técnica de Análisis de señales EEG (imaginación motora) recupera parte de la función perdida, se necesita el apoyo de algún tipo de herramienta de asistencia o tecnología a fin de mejorar la recuperación o de restaurar la capacidad de la anatomía que ya no está presente, explica el Dr. Nick Langhals, quien supervisa la investigación de ingeniería protésica respaldada por los NIH (National Institutes of Health). (Quiroga & Calla, 2019) Tiene como ventaja un dispositivo protésico biónico funcional, el cual es capaz de interpretar impulsos electroencefalográficos disminuyendo en gran medida la discapacidad que trae consigo la falta de una extremidad.

---

La herramienta desarrollada para la sustitución de miembros superiores, consiste en analizar los impulsos generados por la señal EEG (electroencefalografía) para realizar actividades de una persona con miembros sanos, esto con el fin de encontrar características que definan movimientos importantes y básicos en la vida, como por ejemplo, flexión, extensión, abducción aducción de la mano, abducción aducción del pulgar, desviación radial, pronación, supinación, aprehensión, pinza, hiperextensión de los dedos, oposición del pulgar, compresión digital, compresión palmar y rotación. Es así como esta propuesta permite un aporte significativo al campo de la biomédica desde la rehabilitación robótica logrando una mejor calidad de vida en dicha población.

## **1.2. Objetivos**

### **1.1.1. Objetivo General**

- Implementar un clasificador basado en redes neuronales que permita identificar movimientos que puedan ser asociados al control de una mano robótica provenientes de las señales electroencefalográficas (EEG) obtenidas de la aplicación de la técnica de imaginación motora (MI).

### **1.1.2. Objetivos Específicos**

- Procesar las señales EEG de la base de datos physionet y encontrar las características frecuenciales y temporales con el fin de extraer información de la intención del movimiento aplicando la técnica de imaginación motora.
- Implementar a través de redes neuronales un algoritmo de clasificación que utiliza como entradas características obtenidas de las transformadas wavelets aplicadas a los canales C3, C4 y Cz de las señales de la base de datos EEG Motor Movement/Imagery Dataset (eegmmidb) de physionet, para clasificar el

---

movimiento imaginado por el usuario y que se puede asociar para controlar una mano robótica, las tareas tomadas como referencia son las siguientes:

**Tarea 1:** El usuario abre y cierra el puño izquierdo o derecho.

**Tarea 2:** El usuario imagina abrir y cerrar el puño izquierdo o derecho.

- Validar el desempeño de la técnica de clasificación utilizando una matriz de confusión y las métricas de desempeño.

## 2. Antecedentes

En este capítulo se propone realizar un levantamiento bibliográfico sobre el estado del arte de los estudios que fueron realizados con la implementación de sistemas BCI, imaginación motora, con el fin de identificar patrones de movimientos encontrados en los registros de las señales EEG comprobando su efectividad en el procesamiento y clasificación de las señales obtenidas.

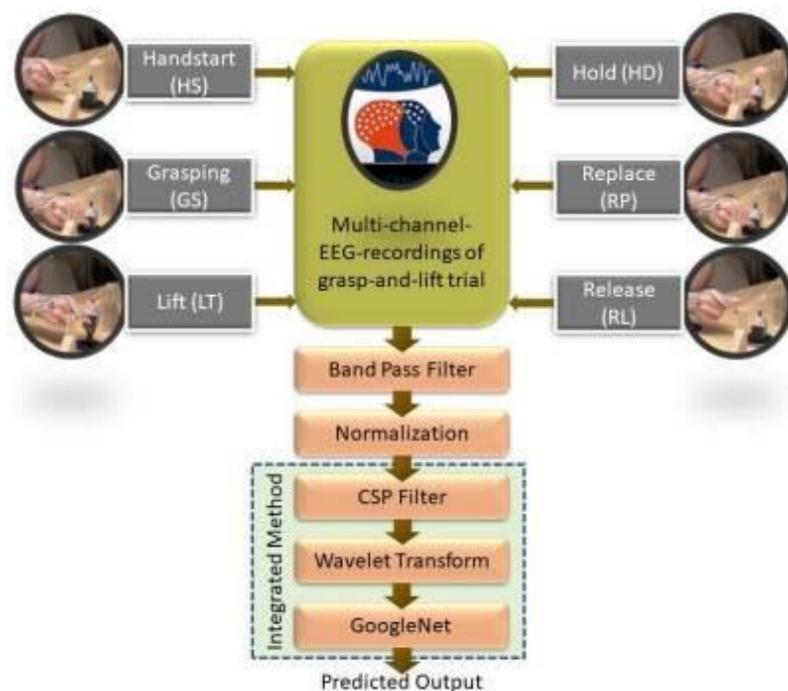
A continuación, se presenta una breve descripción de los trabajos más relevantes en el Campo de la implementación de Imaginación Motora los cuales se tomaron como base para esta investigación:

### **2.1. Clasificación de las funciones motoras a partir de señales de electroencefalograma (EEG) basada en un método integrado compuesto por un patrón espacial común y un marco de transformación de ondas**

En este trabajo, se investiga un algoritmo para la categorización de 6 funciones motoras desde una señal de electroencefalograma (EEG) que combina un filtro de patrón espacial común (CSP) y una transformada de wavelet de tipo continua (CWT). Los datos de EEG entienden 6 eventos de agarre y levantamiento, que se aplican para averiguar el potencial de utilizar EEG como señales de ingreso con dispositivos de interfaz cerebro-computadora para el control de dispositivos

protésicos para el desplazamiento de las extremidades mejores. Los canales de EEG seleccionados son los que se hallan sobre la corteza motora, C3, Cz y C4, así como en la zona parietal, P3, Pz y P4. El algoritmo planteado incluye 3 fases primordiales, filtrado de paso de banda, filtrado de CSP y transformación y entrenamiento de wavelet en GoogLeNet para sustracción de propiedades, aprendizaje de propiedades y categorización.

El filtrado de paso de banda se hace para elegir la señal de EEG en la banda de 7 Hz a 30 Hz a medida que se eliminan los aparatos involucrados con el parpadeo, los latidos del corazón y el desplazamiento de los músculos. El funcionamiento del algoritmo de categorización postulado se evalúa en términos de exactitud, sensibilidad, especificidad, precisión con valores promedio de 94.8%, 93.5%, 94.7%, 94.1%, respectivamente los cuales indican un óptimo funcionamiento del algoritmo postulado en la categorización de eventos de agarre y elevación desde señales de EEG (Yahya, Musa, Ong, & Elamvazuthi, 2019)



---

**Figura 1.** Marco general para la clasificación de eventos de electroencefalograma motor (EEG) utilizando un filtro de patrón espacial común (CSP) y una transformada de ondícula continua (CWT).

**Adaptado de:** *Classification of Motor Functions from Electroencephalogram (EEG) Signals Based on an Integrated Method Comprised of Common Spatial Pattern and Wavelet Transform Framework.* Sensors, por Yahya N, Musa H, Ong ZY, Elamvazuthi I. 2019; 19(22):4878.

## **2.2. Clasificación de señal de EEG de imágenes de motor de cuatro clases utilizando PCA, Wavelet y red neuronal de dos etapas:**

En este trabajo se propone un procedimiento novedoso con base en la implementación combinada del estudio de componentes principales (PCA), la transformación de ondas (WPT) en conjunto con un algoritmo de aprendizaje automático de 2 fases para clasificar la señal de EEG de MI compuesta por 4 clases. Para su desarrollo se incluyeron eventos de 4 clases: imaginar levantar la mano izquierda, la mano derecha, el pie izquierdo y el pie derecho.

Uno de los desafíos a resolver es separar el patrón de la señal EEG del lóbulo similar al de la señal, como el pie izquierdo VS la mano izquierda. Otro problema crítico es detectar los movimientos MI de 2 pies diferentes ya que su grado de activación es bastante bajo y presentan un jefe casi parecido.

Se utiliza el PCA para minimizar las magnitudes de la señal del lóbulo izquierdo y derecho del cerebro, posteriormente, se usa WPT para sustraer la característica de la clase distinto de la señal EEG y al final, la red neuronal artificial se entrena en 2 fases - la primera fase identifica el lóbulo desde el jefe de la señal y la 2 fase identifica si la señal es de desplazamiento de la mano MI o del pie MI desplazamiento del pie. El procedimiento planteado se aplica a las señales de EEG en relación con el desplazamiento MI de 15 competidores y se localizó una maravillosa exactitud de categorización (>74% de media). Los resultados del

procedimiento postulado demuestran su efectividad en la aplicación práctica de la BCI. (Rahman, Et al., 2019)



**Figura 2.** Procedimiento de adquisición de datos de EEG de MI utilizando el dispositivo inalámbrico B-Alert X-10 con un mando asistido por ordenador

**Adaptado de:** Clasificación de señales de EEG de imágenes motoras de cuatro clases utilizando PCA, Wavelet y red neuronal de dos etapas por Md. Asadur Rahman, Farzana Khanam, Md. Kazem Hossain, Mohammad Khurshed Alam y Mohiuddin Ahmad, 2019, publicado en Revista internacional de aplicaciones y ciencias informáticas avanzadas (IJACSA).

---

### **2.3. Clasificación de las señales de EEG de imágenes motoras basadas en la transformada de ondas y la entropía de la muestra:**

El presente artículo presenta un método de extracción de características de EEG basado en la transformada wavelet y la entropía de la muestra. Se analizaron los cambios dinámicos de la entropía de la muestra de la señal de EEG imaginaria del movimiento de la mano izquierda y derecha y su importancia neurofisiológica. Finalmente, se utilizó el discriminante lineal de Fisher para clasificar el EEG imaginario del movimiento de la mano izquierda-derecha. Los experimentos muestran que la tasa promedio de clasificación correcta fue del 88,9%.

Los experimentos muestran que hay un claro significado fisiológico, pero también hay diferencias significativas en las imágenes motoras de la mano izquierda-derecha, lo que favorece la mejora de la precisión de la clasificación BCI, y también es un valor muy útil de los parámetros característicos (Guo, Su, & Liang., 2017)

### **2.4. Clasificación de las señales de EEG de imágenes motoras mediante el aprendizaje automático:**

En este trabajo, se explora la clasificación de señales de electroencefalografía (EEG), específicamente para tareas de imaginación motora (IM) las cuales pueden llevarse a cabo mediante el uso de modelos de aprendizaje automático y de aprendizaje profundo, pero hay que mantener un equilibrio entre la precisión y el tiempo de cálculo.

Este proyecto tiene como objetivo crear un modelo de aprendizaje automático que pueda optimizarse para la clasificación en tiempo real y que tenga una precisión de clasificación relativamente aceptable. El modelo propuesto se basa en el patrón espacial común (CSP) para la extracción de características, así como en el análisis discriminante lineal (LDA) para la clasificación. Con una etapa de preprocesamiento sencilla y una selección adecuada de los datos para el entrenamiento, el modelo demostró tener una precisión equilibrada de +80%,

---

manteniendo un tiempo de ejecución pequeño (milisegundos) que se opta por las clasificaciones en tiempo real. (Abdeltawab & Ahmad, 2020)

### **2.5. Clasificación de las señales de EEG de imágenes motoras con red neuronal convolucional de múltiples entradas mediante el aumento de STFT:**

En este artículo se propone un marco de clasificación de señales EEG de imágenes motoras basada en una red neuronal convolucional (CNN) para mejorar la precisión de su clasificación. Se generaron 1000 imágenes aumentadas adicionales las cuales se dividieron en imágenes de entrenamiento y de prueba para cada sujeto. El rendimiento de cada sujeto se evaluó con cuatro métricas de rendimiento: Exactitud, Precisión, Sensibilidad y Especificidad.

La clasificación de imágenes motoras se divide en 2 clases, primero aplicamos la transformada de Fourier de (STFT) en las señales de series temporales de EEG para transformar las señales en imágenes 2D. A continuación, se entrena una red neuronal convolucional multi entrada con concatenación de características para lograr una clasificación robusta de las imágenes, para regularizar la red se añade una normalización por lotes. El aumento de datos se utiliza para aumentar muestras y como regularizador secundario.

Se propuso una CNN de tres entradas de tres entradas, tomadas de un conjunto de datos de la señal EEG recogidos del conjunto de datos BCI Competition IV 2b y el conjunto de datos de la Competición BCI. Los resultados experimentales muestran que la precisión media de la clasificación alcanzada fue del 89,19% en el conjunto de datos 2b, mientras que nuestro modelo alcanzó el mejor de 97,7% de precisión para el sujeto 7.

También se amplió nuestro enfoque y exploramos un esquema basado en el aprendizaje de transferencia con el modelo ResNet-50 pre entrenado, que mostró resultados prometedores. En general, este enfoque mostró un rendimiento

---

competitivo en comparación con otros métodos. (Shovon, Nazi, Dash, & Hossain, 2019)

### **2.6. Clasificación de señales de eeg de imágenes motoras con filtrado CSP a través de modelos de redes neuronales:**

En este artículo se realizó una propuesta la cual consistió en tres etapas principales: organización de las señales EEG, extracción de características y ejecución de algoritmos de clasificación. Las señales EEG utilizadas, representan cuatro acciones motoras: Movimientos de la mano izquierda, la mano derecha, la lengua y el pie; en el marco del paradigma de las imágenes motoras.

Estas señales de EEG se obtuvieron de una base de datos proporcionada por la Universidad Tecnológica de Graz. De estos datos, sólo se utilizaron las señales de EEG de dos sujetos sanos para el trabajo propuesto. La etapa de extracción de características se llevó a cabo mediante la aplicación de un algoritmo conocido como Patrón Espacial Común, además del método estadístico denominado Root Mean Square. Los algoritmos de clasificación utilizados fueron: K-Nearest Neighbors, Support Vector Machine, Multilayer Perceptron y Dendrite Morphological Neural Networks.

Estos algoritmos fueron evaluados con dos estudios. El primero tenía como objetivo evaluar el rendimiento en el reconocimiento entre dos clases de tareas de mano izquierda vs. mano derecha, mano izquierda vs. lengua, mano izquierda vs. pie, mano derecha vs. lengua. Mano izquierda vs. Pie, Mano derecha vs. Lengua, Mano derecha vs. Pie y Lengua contra el pie. El segundo estudio pretendía emplear los mismos algoritmos en el reconocimiento entre cuatro clases de tareas de imágenes motoras. Los resultados obtenidos fueron confiables ya que para el Sujeto 1 se obtuvo una precisión del  $93,9\% \pm 3,9\%$  y para el sujeto 2 una precisión del  $68,7\% \pm 7\%$ . (Virgilio-González, 2018)

---

### **2.7. Clasificación de la señal de EEG de imágenes motoras multiclase utilizando un enfoque de dispersión:**

En el presente trabajo se ha propuesto una técnica de clasificación basada en la representación dispersa para clasificar los datos del electroencefalograma MI de múltiples tareas.

El método propuesto calcula sólo la energía wavelet directamente a partir de los datos MI segmentados y construye un diccionario. La representación dispersa del diccionario se utiliza entonces para clasificar unos datos de prueba. El enfoque propuesto es más rápido, ya que trabaja con una sola característica y sin necesidad de ningún preprocesamiento. Además, con una longitud reducida de un periodo de imagen, el método propuesto proporciona una clasificación precisa en un tiempo de cálculo menor.

Se ha evaluado el rendimiento del enfoque propuesto y se ha comparado con otros clasificadores publicados en la literatura. Los resultados corroboran que el enfoque de sparsity propuesto funciona significativamente mejor que los clasificadores existentes.

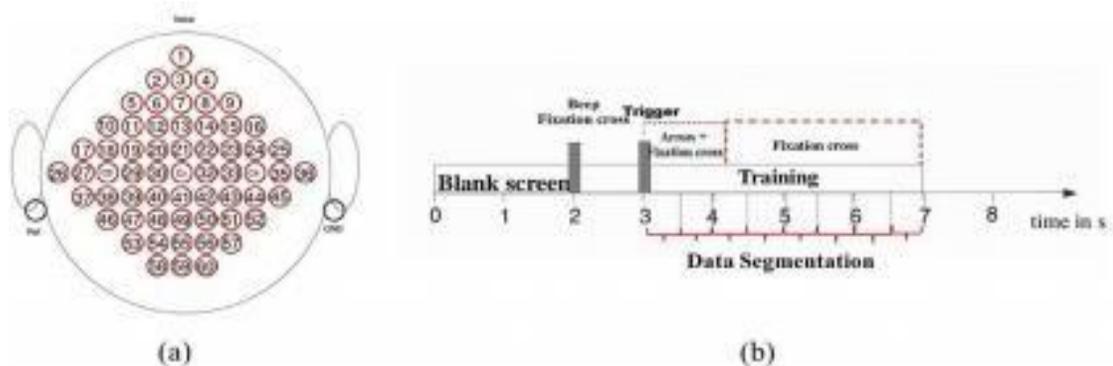
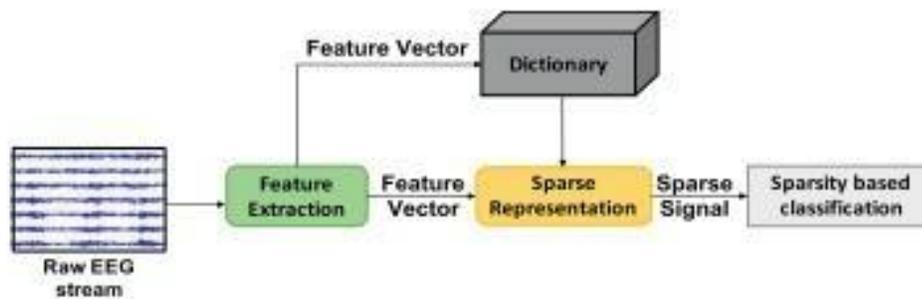


Fig. 1. (a) EEG electrode positions; (b) Training time of a trial.



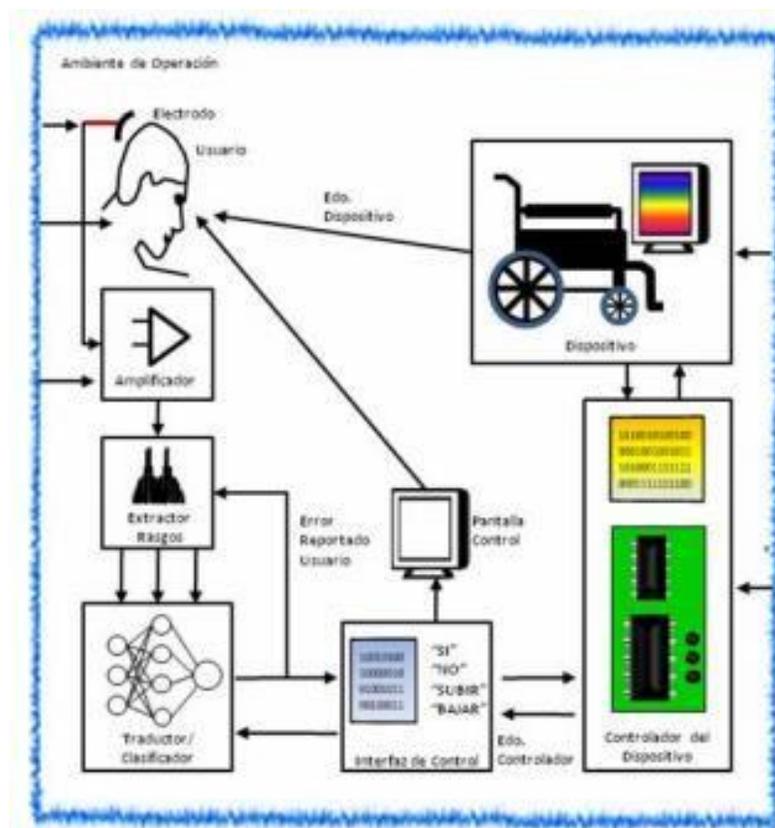
**Figura 3.** Diagrama de bloques de la metodología propuesta en la clasificación de señales EEG utilizando enfoque de dispersión.

**Adaptado de:** *Classification of multiclass motor imagery EEG signal using sparsity approach.* *Neurocomputing* por Sreeja, S. R., & Samanta, D. (2019). doi:10.1016/j.neucom.2019.08.037 (R. & D., 2019)

## 2.8. Interfaz cerebro computador modular basada en la interpretación del electroencefalograma (EEG) mediante RNA para el control de dispositivos electrónicos:

Este artículo describe el diseño e implementación de un prototipo de interfaz cerebro computador (ICC) modular con operación fuera de línea utilizado para simular la manipulación de un mando a distancia controlado únicamente por medio de la actividad cerebral. El sistema se basa en el análisis espectral de la actividad cerebral registrada durante la realización de tareas mentales sencillas. Se desarrollaron siete módulos independientes los cuales se implementaron en

forma de controles ActiveX. Cada control ActiveX realiza una etapa de procesamiento y cuando se combinan permiten establecer la interfaz entre el cerebro y la máquina. La clasificación de los patrones asociados a las tareas mentales fue realizada por una red neuronal del tipo perceptrón multicapa. El sistema implementado permite simular la operación de dispositivos electrodomésticos de uso común como receptores de radio o televisión por medio de la actividad cerebral. (Villegas, Lugo, & Pacheco, 2008)



**Figura 4.** Modelo de referencia para el diseño de interfaces cerebrales

**Adaptado de:** *Interfaz cerebro computador modular basada en la interpretación del electroencefalograma (EEG) mediante RNA para el control de dispositivos electrónicos* por Villegas, Angel; Lugo, Edgar; Pacheco, José R.; Villegas, Hyxia publicado en la Revista INGENIERÍA UC, vol. 15, núm. 2, agosto, 2008.

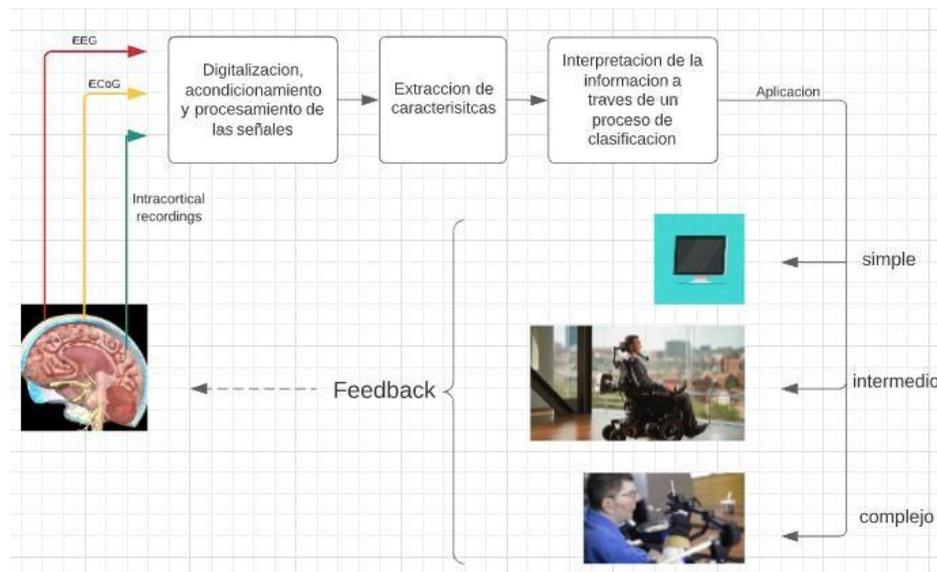
### **3. Marco Conceptual**

#### **3.1. Señales electroencefalográficas (EEG)**

Son el producto de la diferencia de potencial en la membrana celular de las neuronas. El registro de estos bioseñales es conocido como Electroencefalograma (EEG) y los ritmos de la actividad neuronal constituyen un lenguaje de comunicación propio de las neuronas (Marengo, 2021)

El EEG, es una prueba que se realiza con el fin de estudiar el funcionamiento del sistema nervioso central, la cual consiste esencialmente en registrar mediante electrodos especiales adheridos al cuero cabelludo las corrientes eléctricas que se forman en las neuronas cerebrales y que son la base del funcionamiento del sistema nervioso (Chávez Saenz, Et al., 2016)

El EEG mide las fluctuaciones de voltaje causadas por las corrientes iónicas entre las neuronas del cerebro y un período de tiempo, es decir, las aplicaciones de diagnóstico a menudo se denominan ondas cerebrales. Las señales provenientes del cerebro son muy complejas, ya que se trata de señales variantes según la localización de los electrodos y del usuario que utiliza la máquina, además dependen del nivel de concentración, el parpadeo involuntario e incluso el estado mental del usuario (Gomez Figueroa, 2016).



**Figura 5.** Diagrama de adquisición de señales cerebrales.

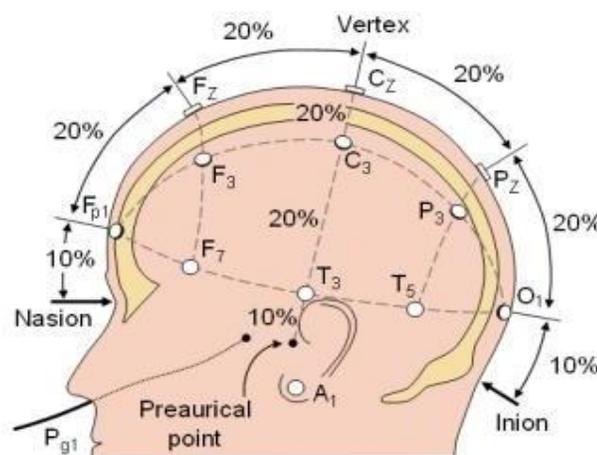
**Adaptado de:** Fuente propia.

### 3.1.1. Sistemas de ubicación de electrodos (10/20, 10/10 y 10/5)

El sistema de ubicación 10/20, es un estándar internacional estipulado por la Federación Internacional de Sociedades de Electroencefalografía (FISE) el cual describe en qué ubicación del cráneo deben ser adheridos los electrodos para medir un EEG (Homan, Herman, & Purdy, 1987)

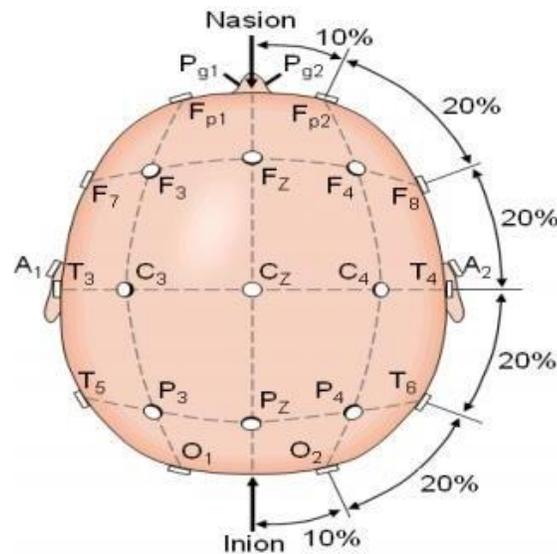
Este sistema se basa en la relación que existe entre la posición de los electrodos y la corteza cerebral, es decir que los electrodos vecinos se pueden ubicar a un 10% o 20% de la distancia total entre la zona frontal y posterior, o de derecha a izquierda del cráneo (Gomez Figueroa, 2016). Las etiquetas en la corteza cerebral hacen referencia al lóbulo o área del cerebro en las que se puede ubicar cada electrodo para registrar señales EEG (Shovon, Nazi, Dash, & Hossain, 2019).

- Áreas principales: Frontopolar (Fp), frontal (F), central (C), temporal (T), parietal (P) y occipital (O).
- La ubicación lateral está definida por números pares e impares, es decir, un número par hace referencia a la ubicación del electrodo en el hemisferio derecho y un número impar hace referencia a la ubicación del electrodo en el hemisferio izquierdo.
- Los electrodos ubicados en la línea vertical imaginaria (línea media) que divide el cráneo en dos mitades (derecha e izquierda) están etiquetados con la letra “z” (z=zero en inglés).



**Figura 6.** Vista de perfil sistema 10/20

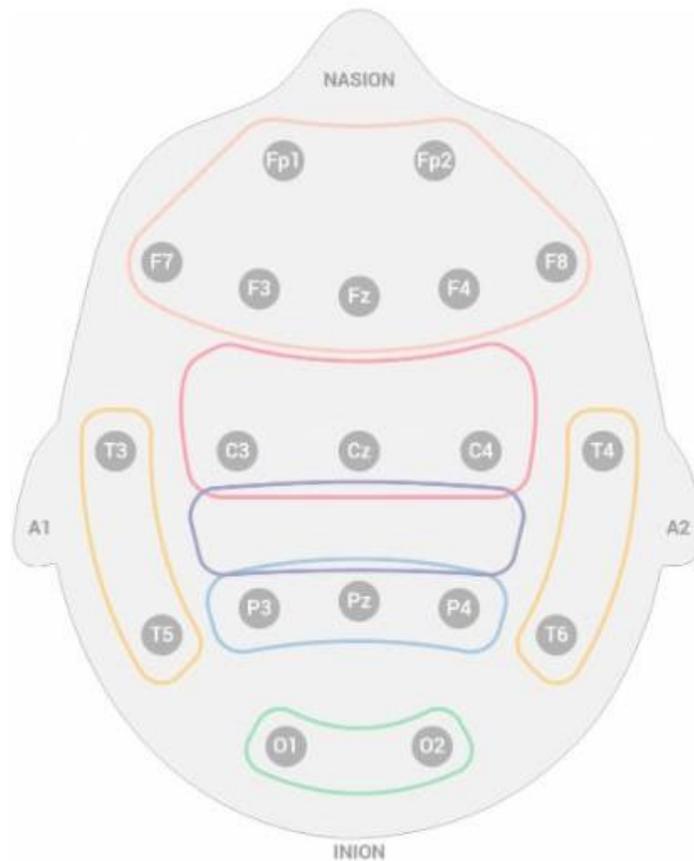
**Adaptado de:** Fisiología.facmed.unam.mx. (2021). Retrieved 7 March 2021, from <http://fisiologia.facmed.unam.mx/wp-content/uploads/2019/09/UTI-pr%C3%A1ctica-7-a.-Electroencefalograma.-AnexoManual.pdf>.



**Figura 7.** Vista superior sistema 10/20

**Adaptado de:** Fisiología.facmed.unam.mx. (2021). Retrieved 7 March 2021, from <http://fisiologia.facmed.unam.mx/wp-content/uploads/2019/09/UTI-pr%C3%A1ctica-7-a.-Electroencefalograma.-AnexoManual.pdf>.

Según la distribución de las distancias, el sistema 10/20 tiene un total de 21 electrodos respecto a las líneas centrales (central sagital y coronal), en el sistema 10/10 estas líneas se dividen en un 10% con un total de 81 electrodos y en el sistema 10/5 las líneas se dividen en un 5% con un total de 320 electrodos. El sistema estándar internacional permite asociar fácilmente el registro de la actividad cerebral de un electrodo con diferentes funciones según su ubicación craneal (figura 3) (Alda & Torreblanca, 2018)



**Figura 8.** *Relación entre funciones cerebrales y electrodos para EEG.*

**Adaptado de:** *Colocación de electrodos EEG en un Layout Fijo vs. Variable.* Bitbrain. Por Alda y Torreblanca, (2017). Recuperado el 5 de Mayo de 2021, de <https://www.bitbrain.com/es/blog/colocacion-electrodos-eeq>.

**Tabla 1.** *Relación de convenciones entre funciones cerebrales y electrodos para EEG*

Razonamiento, control del habla y del movimiento, emociones y resolución de problemas.

**Funciones sensoriomotoras.**

Somatosensoriales.
Atención, percepción y procesamiento de los estímulos relacionados con los sentidos.
Funciones visuales.
Memoria y audición.

### 3.2. Ondas o Ritmos Cerebrales

El cerebro humano genera impulsos eléctricos (potenciales de acción) que pasan a través de las neuronas. Los ritmos cerebrales son oscilaciones en la actividad cerebral, las cuales se registran en las áreas somatosensoriales y motoras, estos han sido estudiados y clasificados de acuerdo con sus rangos de frecuencia. Se han creado seis clases para definir y delimitar estos ritmos y se nombran usando las letras griegas ( $\alpha$ ,  $\beta$ ,  $\delta$ ,  $\theta$ ,  $\gamma$ , y  $\mu$ ). (Nicolas-Alonso & L. Gomez-Gil, 2012)

Todas las ondas cerebrales poseen características de amplitud y frecuencia, las cuales se miden en ciclos por segundo y en microvoltios. Se ha comprobado que hay una relación existente entre los distintos estados mentales y las características. (Chávez Saenz, Et al., 2016)

Tipo de onda	Voltaje	Frecuencia	Estado mental
Delta	10-50 $\mu$ V	0.5 – 4 Hz	La amplitud de las señales delta es relativamente grande comparada con las demás y se presentan durante el sueño profundo, debido a su baja frecuencia es muy fácil confundirla con ruido causado por el movimiento de los músculos del

			cuello o la quijada.
Theta	10-100 $\mu$ V	4 – 7.5 Hz	Solo es posible detectar pequeñas cantidades de este ritmo en hombres adultos. Este ritmo se presenta principalmente en niños, adolescentes, y adultos somnolientos durante etapas de meditación o sueño ligero.
Alpha	100-150 $\mu$ V	8 – 13 Hz	La amplitud de estas señales crece cuando se cierran los ojos y el cuerpo se relaja, pero disminuye cuando los ojos se abren o se hace esfuerzo mental. Este ritmo está ligado a el procesamiento visual en el lóbulo occipital y puede estar relacionado a las funciones de memoria..
Betta	150-200 $\mu$ V	14 – 26 Hz	Se presenta principalmente en el área central y frontal del cerebro. El ritmo beta se asocia principalmente con el movimiento o la intención de éste.
Gamma	+200 $\mu$ V	+ de 30 Hz	Existe evidencia de que la actividad gamma está relacionada con los estímulos visuales y auditivos. Debido a que las señales electromiográficas (EMG) y electrooculográficas (EOG) son muy parecidas, es común evitarlas al implementar interfaces cerebro-computadora.

### 3.3. Sistemas BCI

Una interfaz cerebro-computadora (brain-computer interfaces BCI) es un medio de comunicación cerebral a través de la actividad no muscular u hormonal con el mundo exterior (Clerc, et al., 2013) & (Springer International Publishing, 2015). Para la implementación de las BCI es necesario que los usuarios utilicen su actividad cerebral de manera explícita en lugar de las reacciones neuromusculares para comunicarse con una computadora, esta es capaz de interpretar las señales EEG adquiridas por medio de unos sensores colocados en el cuero cabelludo o directamente en el tejido cerebral para poder descifrar el mensaje de comunicación y/o control hacia un entorno exterior.

El amplio campo de aplicación de las BCI va desde asistir a personas con discapacidad motriz (Moreno, Et al., 2016), (Leeb, Et al., 2015) & (Leeb, Et al., 2007), controlar dispositivos a través de una vía cerebral (Chang, et al., 2010) & (Göhring, et al., & Rojas, 2013) hasta aplicaciones en la industria de los videojuegos (Krauledat, y otros, 2009) & (Bonnet, Lotte, & Lécuyer., 2013).

Según el tipo de mensaje que se interprete de la actividad cerebral, las BCI se pueden clasificar en mecanismos neurológicos conocidos como fuentes electrofisiológicas o procesos para generar las señales de control. Las fuentes más utilizadas en el desarrollo de las BCI son: la imaginación motora, los potenciales corticales lentos, los p300 y los potenciales evocados. Existen 2 paradigmas de control para llevar a cabo una comunicación con estas fuentes: discreto o continuo.

- Discreto: El usuario es libre de elegir dos o más opciones (por ejemplo, elige una tecla específica de un teclado virtual visualizado en la pantalla de una computadora).
- Continuo: El usuario es capaz de controlar un número de variables cinemáticas (por ejemplo, controlar un cursor en el monitor).

**3.3.1. Cuadro comparativo de cascos de adquisición**

**Tabla 3.** Comparación entre cascos de adquisición

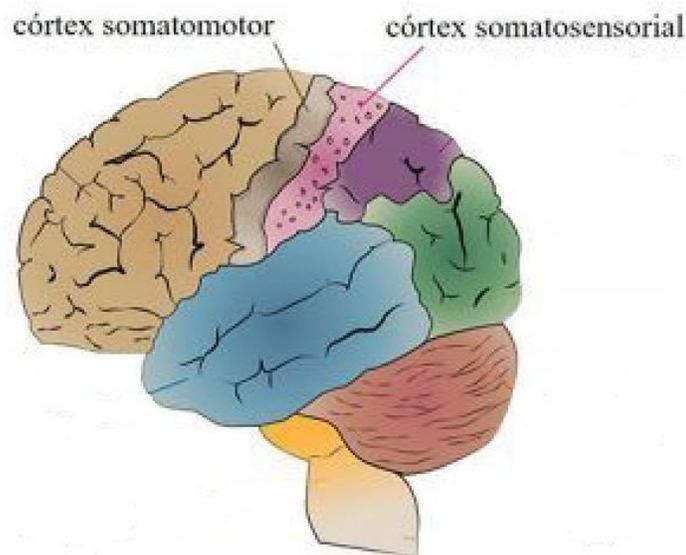
<b>BCI 2000</b>		<b>EPOC FLEX</b>		<b>EPOC X</b>		<b>EPOC +</b>	
							
<b>Sensores</b>	64	<b>Sensores</b>	32	<b>Sensores</b>	14	<b>Sensores</b>	14

<b>Ubicación de los sensores</b>	F3, F4, C3, C4, Pz, P3, P4 + Orejas vinculadas	<b>Ubicación de los sensores</b>	Configurable en cualquier ubicación 10-20	<b>Ubicación de los sensores</b>	AF3, AF4, F3, F4, FC5, FC6, F7, F8, T7, T8, P7, P8, O1, O2	<b>Ubicación de los sensores</b>	AF3, AF4, F3, F4, FC5, FC6, F7, F8, T7, T8, P7, P8, O1, O2
<b>Resolución</b>	14 bits o 16 bits por canal de respuesta 0.317 $\mu$ V/muestra	<b>Resolución</b>	14 bits por canal límite de velocidad de respuesta 65 $\mu$ V / muestra	<b>Resolución</b>	14 bits o 16 bits por canal	<b>Resolución</b>	14 bits o 16 bits por canal
<b>Frecuencia de Muestreo</b>	0,003-150 Hz	<b>Frecuencia de Muestreo</b>	0,16 - 43 Hz	<b>Frecuencia de Muestreo</b>	0,16 - 43 Hz	<b>Frecuencia de Muestreo</b>	0,16 - 43 Hz

**Adaptado de:** *Package: NEUROSPEC AG* por AG, N. (2018). DSI-7 BCI 2000 Research Neurosciences. Neurospec.com. Retrieved 16 April 2021, from <https://www.neurospec.com/Products/Details/1071/dsi-7-bci-2000-package>.

### 3.4. Imaginación Motora

Las actividades cerebrales relacionadas al movimiento corporal se les llama ritmos sensoriomotores (SMR), imaginar mover un miembro, contraer un músculo o simplemente tener intención de movimiento en cualquier parte del cuerpo cambia la actividad cerebral en la corteza. La disminución en una frecuencia específica de la actividad cerebral se les llama desincronización relacionada a eventos (ERD) mientras que el aumento es llamado sincronización relacionada a eventos (ERS). Los patrones ERD/ERS producidos durante la imaginación de movimiento se presentan mayormente en las bandas de frecuencia muy beta de las señales EEG (Grimann, et al., 2010)



**Figura 9.** Áreas de actividad motora en el encéfalo.

**Adaptado de:** ¿Por qué se aprende mejor en movimiento? La neurociencia tiene la respuesta. Educación – articuloseducativos.es. (2018). Retrieved 5 May 2021, from <http://articuloseducativos.es/educacion/2018/10/12/por-que-se-aprende-mejor-en-movimiento-la-neurociencia-tiene-la-respuesta/>

### **3.5. Transformada Wavelet (WT)**

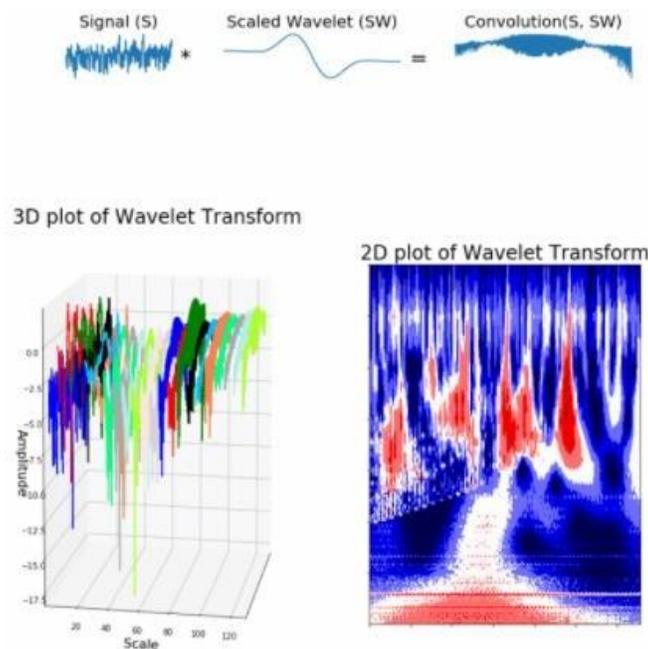
La Transformada Wavelet es utilizada para analizar señales con un espectro de frecuencia dinámico, esta herramienta matemática tiene una alta resolución en el dominio de la frecuencia y en el del tiempo.

Con la WT no sólo se puede interpretar qué frecuencias están presentes en una señal sino también en qué momento se han producido, y es gracias al trabajo realizado en diferentes escalas, es decir, primero se observa la señal con una escala (ventana grande) para analizar las características grandes y luego con escalas más pequeñas para analizar las características más pequeñas.

- Para valores de frecuencia pequeños y alta resolución en el dominio de la frecuencia, la wavelet tiene baja resolución en el dominio del tiempo.
- Para valores de frecuencia grandes y baja resolución en el dominio de la frecuencia, la wavelet tiene alta resolución en el dominio del tiempo.

#### **3.5.1 Funcionamiento de la Transformada Wavelet.**

La Transformada Wavelet utiliza una onda pequeña (ondícula) para obtener información de tiempo y de frecuencia, ya que la wavelet está localizada en el tiempo se puede multiplicar la señal original con la ondícula en diferentes ubicaciones del tiempo, es decir, se comienza con el origen de la señal y lentamente se mueve la ondícula hacia el fin de la señal, este proceso también es conocido como convolución. Después de haber hecho esto para la ondícula original (madre), se puede escalar de manera que se haga más grande y repetir el proceso



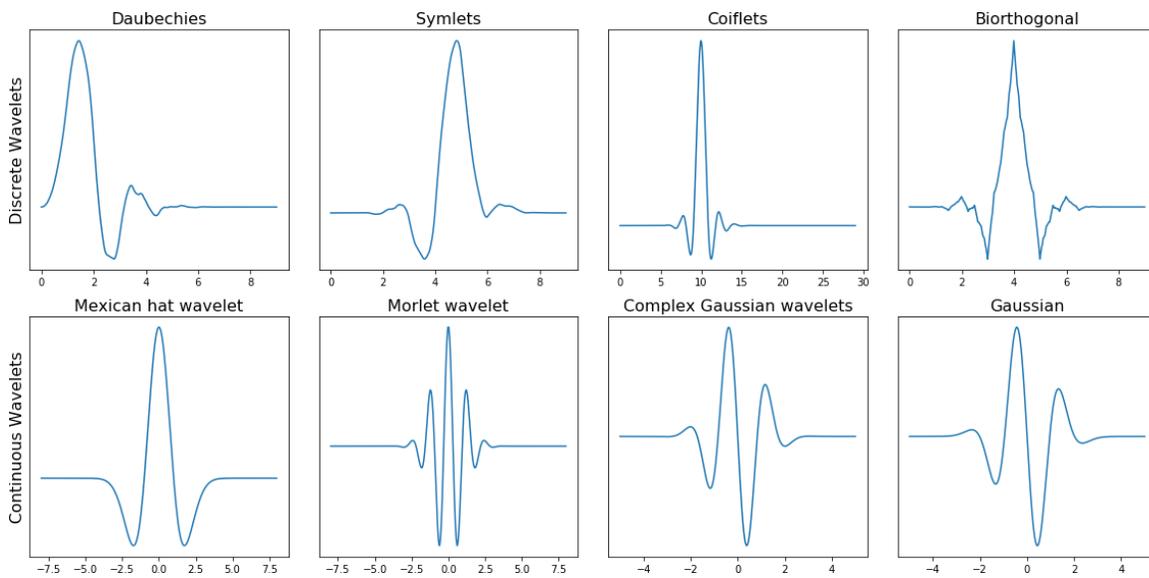
**Figura 10.** Funcionamiento de la Transformada Wavelet.

**Adaptado de:** *A guide for using the Wavelet Transform in Machine Learning.* Taspinar, A. (2018). Retrieved 30 April 2021, from <https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>.

En la figura anterior, la transformada Wavelet de una señal unidimensional tendrá dos dimensiones, la salida bidimensional de la Wavelet es la representación a escala de tiempo de la señal en forma de un escalograma.

### 3.5.2 Familias de las wavelets

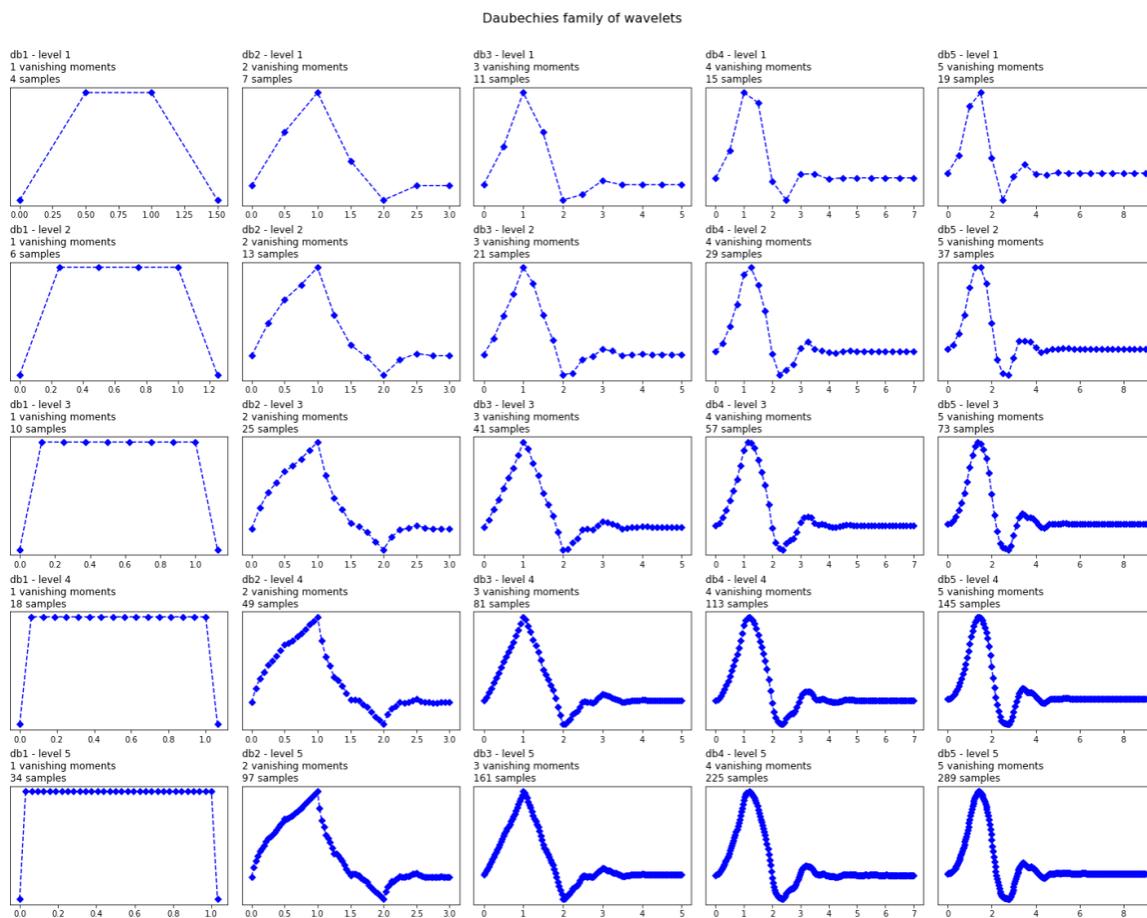
La WT contiene unas familias de ondículas que se diferencian entre sí por lo compacta y suave que se ve cada ondícula, es decir que una familia de ondículas específica se puede adaptar mejor a las características que se buscan en una señal.



**Figura 11.** Familias de las ondículas wavelet.

**Adaptado de:** *A guide for using the Wavelet Transform in Machine Learning.* Taspinar, A. (2018). Retrieved 30 April 2021, from <https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>.

### 3.5.2.1 Familia de ondículas Daubechies (db)



**Figura 12.** Familia de ondículas Daubechies en diferentes niveles de descomposición

**Adaptado de:** *A guide for using the Wavelet Transform in Machine Learning.* Taspinar, A. (2018). Retrieved 30 April 2021, from <https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>.

En la figura anterior se observa la familia de wavelets 'Daubechies' (db) compuesta por 5 columnas que representan el orden de cada db enumerado de la siguiente manera: (db1, db2, db3, db4 y db5). El número de orden de cada db indica la relación entre el orden de aproximación y la suavidad de la ondícula, a medida que aumenta el número de orden también aumenta el grado del polinomio de la ondícula y se vuelve más suave.

### **3.5.3 Transformada wavelet continua (CWT)**

Esta transformada constituye a una técnica relativamente nueva, la cual se ha propuesto como una herramienta de análisis sobre el comportamiento local de una señal.

Esta transformada es una representación tiempo-frecuencia, es decir, que para valores pequeños la CWT obtiene información de la señal localizada en el dominio del tiempo y, en el caso contrario la CWT obtiene información de la señal localizada en el dominio de la frecuencia. Cuando se presenta un cambio de valor también cambia el ancho de banda y la duración de la wavelet, pero su forma se mantiene igual. Por lo mencionado anteriormente se encuentra la diferencia principal entre la CWT y la transformada corta de Fourier (STFT), ya que la primera ocupa ventanas de corta duración para altas frecuencias y de larga duración para bajas frecuencias, en cambio la STFT ocupa una sola ventana con la misma duración tanto para altas como para bajas frecuencias (Hernández González C. , 2017).

Cabe anotar que esta transformada se utiliza comúnmente para generar un escalograma. (Schalk, McFarland, T., N., & J.R., 2004) & (Riveras Sanabria, 2012)

### **3.5.4 Transformada Wavelet Discreta (DWT)**

Esta transformada permite descomponer la señal en coeficientes de aproximación y de detalle para obtener una mejor resolución temporal respecto a otros algoritmos utilizados para el procesamiento de señales.

Los coeficientes de detalle facilitan la detección de la singularidad de interés ocurrida en la señal, esto se debe a que la wavelet madre posee  $n$  momentos de desvanecimientos. Esta propiedad de la DWT permite que se pueda utilizar como un operador diferencial multiescala de orden  $n$ , siendo muy útil en aplicaciones donde las señales presentan discontinuidades.

### **3.6. Transformada de Fourier (FFT)**

Permite el análisis de una señal periódica en el dominio tiempo dado su contenido en frecuencia, debido a que los coeficientes de Fourier representan la contribución de las funciones seno y coseno según cada frecuencia. Bajo ciertas condiciones, es posible, después de haber realizado el estudio en el dominio frecuencia, retornar al dominio tiempo aplicando la anti transformada o la transformada inversa de Fourier. (E., Valdés-Santiago, et al, 2018)

La FT y su anti transformada tienen propiedades de simetría que resultan muy útiles para el análisis de señales, para conocer la evolución de la señal en cualquier instante del tiempo, es necesario encontrar el valor de la señal transformada en una determinada frecuencia, es decir que la FT permite la localización en frecuencia, pero no en tiempo. (R. Castro & M. Castro, 2021)

### **3.7. Identificación de Patrones**

En algún momento en la etapa de procesamiento de una señal es fundamental realizar un reconocimiento de patrones. Un patrón es una descripción estructural o cuantitativa de un objeto o de alguna otra entidad de interés en una señal y está formado por uno o más descriptores.

La observación de estas tareas particulares (imaginarse el movimiento de la mano) crea en los sistemas BCI un especial impacto específico en las señales cerebrales (los ERS/ERD) medidos en una localización en especial. Aun cuando dichos fenómenos cognitivos fueron extensamente estudiados y a pesar que para la aplicación BCI se considere una sola señal fisiológica viable (como los ritmos mu). (Muñoz Cardona, 2014)

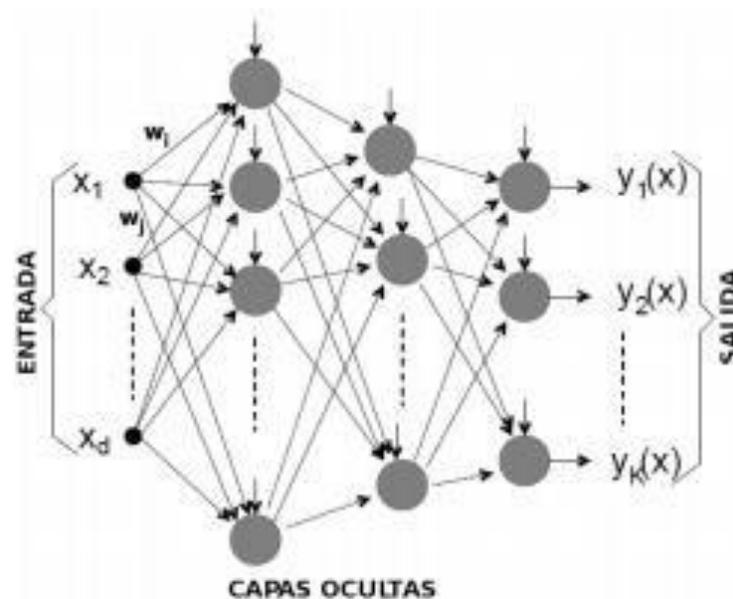
Esta aleatoriedad entre las señales es uno de los aspectos que más limitan el procesamiento de las señales de EEG de las interfaces, dichos cambios en la señal EEG registrados tienen la posibilidad de ser provocados por diferencias en medio de las

sesiones de prueba, falta de la calibración o pequeñas diferencias en medio de las posiciones de los electros entre sesiones. (W, C, KR, & M., 2012)

Las técnicas de preprocesamiento para dividir las señales cerebrales que son de interés, como los patrones cerebrales asociados con la estrategia de imaginación motora implementada; sin embargo, hay algunos otros componentes remanentes como el ruido, que no son de interés dado que las señales de los canales EEG tienden a estar altamente correlacionados. (Clerc, R.Wolpaw, & Winter, 2013)

### **3.8. Red neuronal artificial tipo feedforward (RNAf)**

Una red neuronal feed forward, posee una configuración de dos capas y un número suficiente de neuronas, se puede aproximar cualquier función continua a un grado de precisión arbitrario [51]. Una RNAf se caracteriza por ser un conjunto de neuronas que reciben información multivariable, la procesan y dan una respuesta que puede ser multivariable también. En la arquitectura feedforward la topología del arreglo de neuronas y sus interconexiones hace fluir la información de forma unidireccional para que nunca pueda pasar más de una vez a través de una neurona antes de generarse la respuesta de salida. (Patiño, et al., 2016)



**Figura 13.** Estructura típica de una red neuronal artificial tipo feedforward (RNAf).

**Adaptado de:** Red neuronal feedforward como estimador de patrones de corrientes en el interior del puerto de manzanillo sujeto a la acción de tsunamis. Por Vásquez López, J. (2014). Tomado de secretaría de comunicaciones y transportes instituto mexicano del transporte, No. 406, 17-22. Retrieved 10 May 2021, <https://www.imt.mx/archivos/Publicaciones/PublicacionTecnica/pt406.pdf>.

### 3.9. Métodos de Aprendizaje

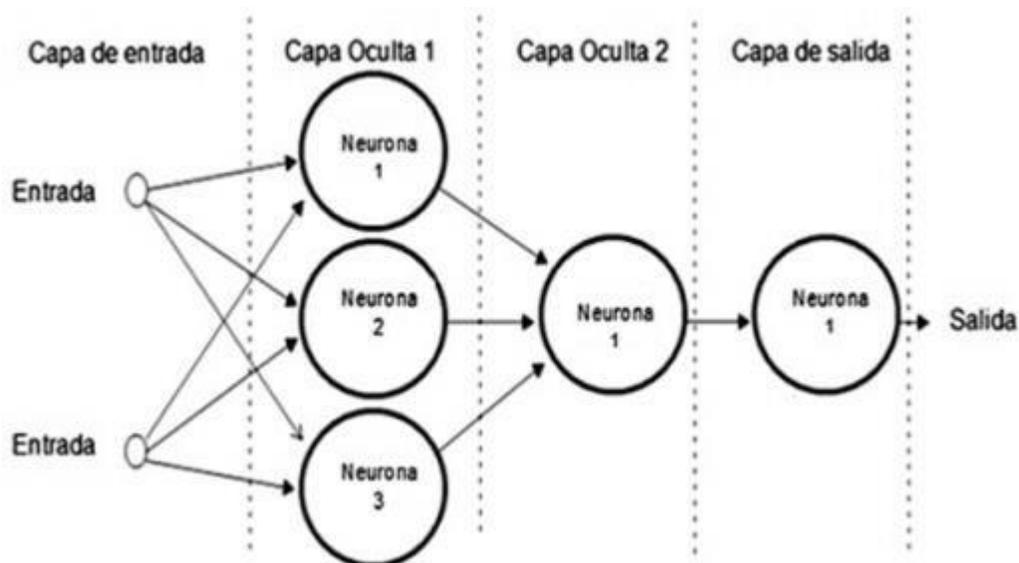
Para el aprendizaje supervisado (RNAf) se entrega a la red el grupo de datos (entradas y salidas), donde la entrada representa las características más sobresalientes que determinan el valor de la salida que la RNA debería exponer. Esta clase de aprendizaje debería ser apto de generalizar, revelando valores de salida para datos de ingreso que no fueron analizados antes.

Para el aprendizaje supervisado (RNAf) se otorga a la red el conjunto de datos (entrada-salida), donde la entrada representa las propiedades destacadas que

determinan el valor de la salida que la RNA debe presentar. Este tipo de aprendizaje debe ser apto de generalizar, revelando valores de salida para datos de entrada que no fueron analizados previamente. El algoritmo de aprendizaje hace que el ajuste de los pesos de la red sea capaz de clasificar sin solicitar un punto de inicio. (Delgado, et al., 2019)

Para que esta RNA profundice en el aprendizaje de la actividad requerida hace falta realizar la variación de los pesos por medio del proceso de entrenamiento, donde cada neurona de la RNA es responsable de adaptar los pesos a los datos de ingreso que recibe de las neuronas que la preceden y ya que las salidas adquiridas para los datos de entrenamiento tienen que ser lo más similares a las salidas pretendidas, hace falta establecer un valor que es dependiente de estas salidas nombrado función de error.

Para disminuir el error generado por la red, es necesario evaluar la diferencia entre el valor deseado y el proporcionado por la RNA con respecto a los datos de entrada. Los factores que influyen para disminuir la función del error son: la estructura de la red que lleva consigo la disminución del error, la función de error elegida, los datos de entrenamiento y las salidas deseadas. (Delgado, et al., 2019)



**Figura 14.** Esquema de una red neuronal feed-forward o MLP

**Adaptado de:** *Análisis de electroencefalograma usando redes neuronales artificiales* por Delgado, K., Ledesma, S., & Rostro, H. (2019).

En la figura anterior se observa la comunicación que las neuronas establecen y como la información debería fluir de las entradas hacia las salidas, evitando así generar ciclos de realimentación; Por esta razón este tipo de redes se llaman Redes de Propagación hacia Delante, feed-forward o MLP: Multi Layer Perceptron.

### 3.10. Matriz de Confusión

La Matriz de Confusión es una de las métricas más intuitivas y sencillas que se utiliza para encontrar la precisión y exactitud del modelo. Se utiliza en problemas de clasificación donde la salida puede ser de dos o más tipos de clases y nos brinda información sobre las medidas de sensibilidad y especificidad de la red neuronal.

Esta matriz, es una herramienta fundamental a la hora de evaluar el desempeño de un algoritmo de clasificación, ya que dará una mejor idea de cómo se está clasificando dicho algoritmo, a partir de un conteo de los aciertos y errores de cada una de las clases en la clasificación. Así se puede comprobar si el algoritmo está clasificando mal las clases y en qué medida. (EcuRed, 2019)

<b>Prediction</b>	Positive	True positives	False positives
	Negative	False negatives	True negatives

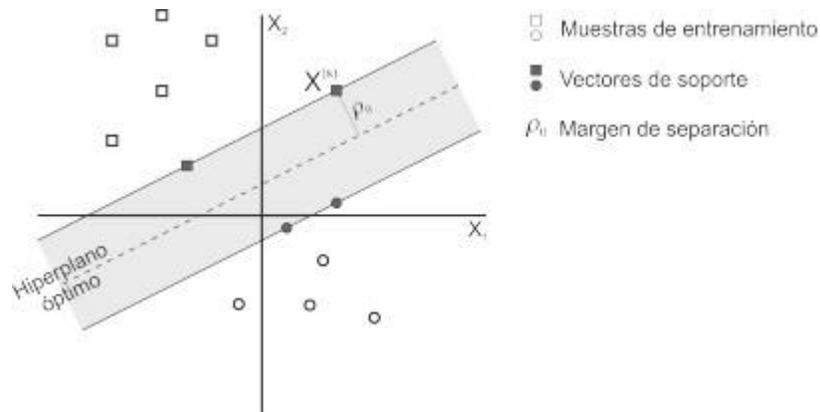
**Figura 15.** Matriz de confusión para un clasificador en dos clases.

**Adaptado de:** *Disposición de la matriz de confusión. InteractiveChaos por Burrueco, D. (2015).* Retrieved 13 April 2021, from <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/disposicion-de-la-matriz-de-confusion>.

En la imagen de arriba, puede ver las filas llamadas "positivas", estas filas se han clasificado como "verdaderas" y las de las filas "negativas" se clasifican como "negativas". En otras palabras, los verdaderos positivos se refieren a aquellos elementos que en realidad son afirmaciones y han sido clasificados como estos elementos. Y los falsos negativos son los elementos que han sido identificados como negativos no siéndolo. (Burrueco, 2015)

### 3.11. Máquina de soporte Vectorial

Una máquina de soporte vectorial SVM (Support Vector Machines) tienen sus orígenes en los trabajos sobre la teoría de aprendizaje, y fueron introducidas en los 90 por Vapnik y sus asistentes (Boser, Guyon, & Vapnik, 1992). Estos se crearon originalmente para resolver problemas de clasificación binaria y ahora se utilizan para resolver problemas como regresión y agrupamiento. SVM es una especie de máquina de aprendizaje binaria, que puede construir un hiperplano lineal a partir de la plataforma de entrenamiento, de modo que el margen de intervalo entre los vectores de características que pertenecen a dos clases diferentes es grande, lo cual es una ventaja para la solución de sobreajuste (Haykin, 2009)



**Figura 16.** Hiperplano óptimo obtenido por SVM para dos clases

**Adaptado de:** *Neural networks and learning machines*. Simon S Haykin., 2009 volume 3. Pearson Upper Saddle River, NJ, USA.

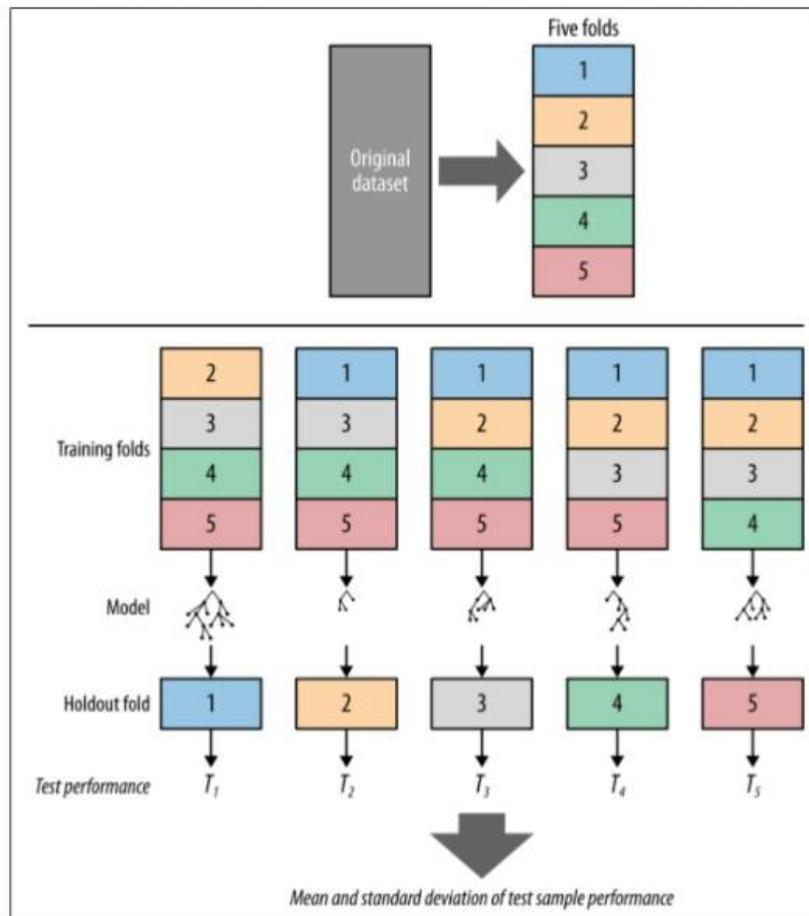
### 3.12. Validación Cruzada

El método de validación cruzada de K- Fold Cross-Validation, es un proceso iterativo que incluye la división aleatoria de datos en  $k$  grupos de aproximadamente el mismo tamaño,  $k-1$  grupos para entrenar el modelo y uno de los grupos para la validación. En cada iteración, este proceso se repite  $k$  veces utilizando diferentes grupos como verificación. Este proceso genera  $k$  estimaciones de error y utiliza su valor promedio como estimación final. (Rodrigo, 2021)

En esta iteración, los otros pliegues  $k-1$  se combinan para formar los datos de entrenamiento. Por lo tanto, en cada iteración tenemos  $(k-1) / k$  de los datos utilizados para el entrenamiento y  $1/k$  utilizado para la evaluación. Cada iteración produce un modelo, y por lo tanto una estimación del rendimiento de la generalización, por ejemplo, una estimación de la precisión. (Morales & Escalante, 2015)

Una vez finalizada la validación cruzada, todos los ejemplos se han utilizado sólo una vez para evaluar, pero  $k-1$  veces para entrenar. En este punto tenemos

estimaciones de rendimiento de todos los pliegues y podemos calcular la media y la desviación estándar de la precisión del modelo. (Morales & Escalante, 2015)



**Figura 17.** Validación cruzada usando iteraciones.

**Adaptado de:** *Machine Learning con Python – Sobreajuste* por Briega, R. (2021).

Relopezbriega.github.io. Retrieved 17 May 2021, from <https://relopezbriega.github.io/blog/2016/05/29/machine-learning-con-python-sobreajuste/>.

## 4. Materiales y Métodos

Con el propósito de definir los procesos que se llevarán a cabo en el desarrollo de la presente investigación y cumplir con los objetivos planteados, se propone a continuación la siguiente guía metodológica CDIO. Es de suma importancia resaltar que para el total cumplimiento de cada uno de los objetivos se realizará una investigación observacional, compuesta de conocimientos previos de ingeniería, normatividad y bibliografía como fuente de retroalimentación de cada uno de los procesos realizados.

Para el desarrollo de cada etapa de este proyecto, nos basamos en la guía metodológica CDIO, por consiguiente se realizó una breve revisión del estado del arte de las técnicas de caracterización y clasificación de señales EEG bajo el paradigma de imaginación motora, posteriormente se utilizaron registros electroencefalográficos de PhysioNet basados en imaginación motora para la creación de un subconjunto de datos de 5 sujetos tomando las siguientes rutinas experimentales: Imaginar abrir y cerrar el puño derecho o izquierdo y un periodo de descanso, dichos datos fueron procesados en una estrategia computacional implementada en el software python y finalmente interpretados según la intención de movimiento de cada sujeto.

### 4.1. Base de Datos.

La base de datos de EEG utilizada para este trabajo fue adquirida a través de un banco de señales electrofisiológicas llamado PhysioNet con acceso gratuito y disponible en el siguiente sitio web:

<https://www.physionet.org/physiobank/database/eegmmidb/>. Este es un conjunto de datos que contiene más 1500 registros de EEG, obtenidos de 109 sujetos saludables.

Esta base de datos tiene la opción de ser exportada en los siguientes formatos: EDF+, .MAT, .CSV, .TXT O .ZIP. , en este proyecto se exporta en .CSV, ya que en la elaboración de la estrategia computacional en python se trabaja con la librería PANDAS para realizar la lectura de los datos, cada formato mencionado contiene 64 señales EEG tomadas cada una a 160 Hz.

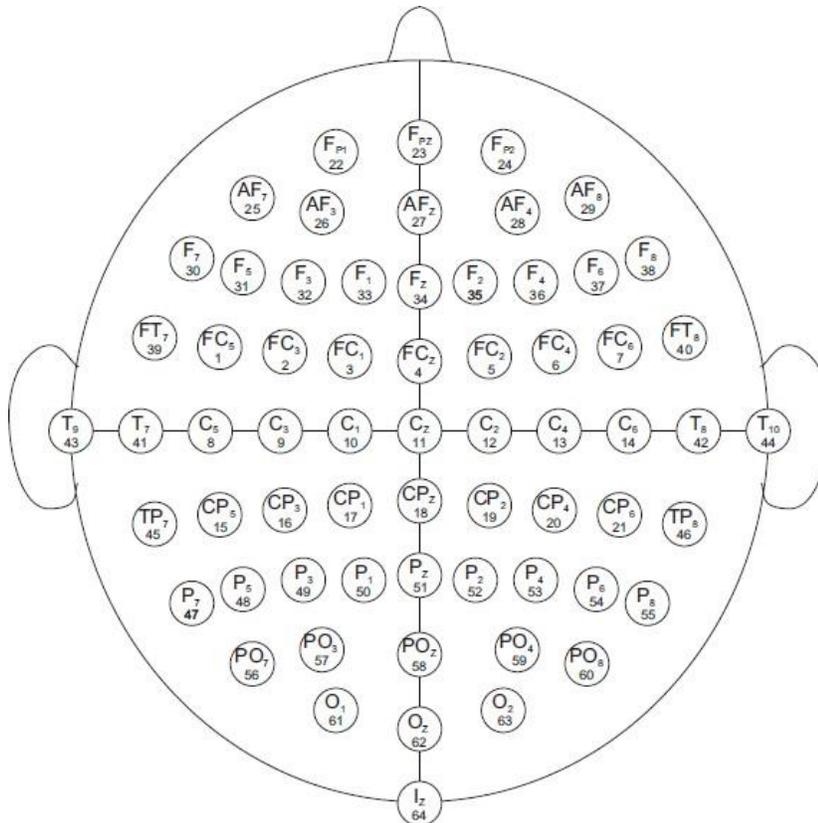
Los sujetos realizan diferentes tareas de imaginación motora registradas por medio del sistema BCI2000, cada sujeto realizó 14 pruebas experimentales conformadas de la siguiente manera: dos carreras iniciales de un minuto (una con los ojos abiertos, otra con los ojos cerrados) y tres carreras de dos minutos de cada una de las cuatro tareas siguientes:

1. Aparece un objetivo en el lado izquierdo o derecho de la pantalla. El sujeto abre y cierra el puño correspondiente hasta que el objetivo desaparece. Entonces el sujeto se relaja.
2. Aparece un objetivo en el lado izquierdo o derecho de la pantalla. El sujeto se imagina abriendo y cerrando el puño correspondiente hasta que el objetivo desaparece. Entonces el sujeto se relaja.
3. Aparece un objetivo en la parte superior o inferior de la pantalla. El sujeto abre y cierra ambos puños (si el objetivo está arriba) hasta que el objetivo desaparece. Entonces el sujeto se relaja.
4. Aparece un objetivo en la parte superior o inferior de la pantalla. El sujeto se imagina abriendo y cerrando ambos puños (si el objetivo está arriba) hasta que el objetivo desaparece. Entonces el sujeto se relaja.

Para almacenar, extraer y usar esta base de datos, los autores generan un archivo de anotaciones compatible con PhysioBank correspondientes a las siguientes anotaciones (T0, T1 o T2):

- T0 corresponde al descanso
- T1 corresponde al inicio del movimiento (real o imaginario) de
  - el puño izquierdo (en los recorridos 3, 4, 7, 8, 11 y 12)
- T2 corresponde al inicio del movimiento (real o imaginario) de
  - el puño derecho (en los recorridos 3, 4, 7, 8, 11 y 12)

Los recorridos experimentales de interés para este proyecto son los siguientes: Recorrido tres y cuatro, que corresponden a abrir y cerrar el puño izquierdo o derecho e imaginar abrir y cerrar puño izquierdo o derecho respectivamente.



**Figura 18.** Ubicación de los 64 electrodos.

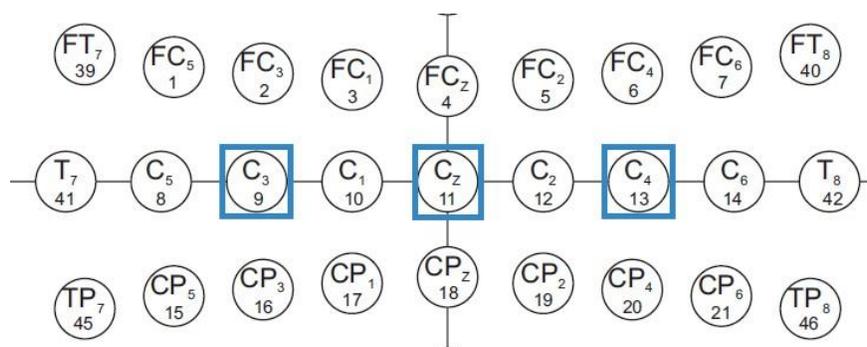
**Adaptado de:** *A General-Purpose Brain-Computer Interface (BCI) System*. IEEE por Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R., 2004. Transactions on Biomedical Engineering

En la figura anterior se evidencia la ubicación de los 64 electrodos basada en el sistema internacional 10/10 para captar las señales EEG producidas por la aplicación de la técnica de imaginación motora.

## 4.2. Selección de Canales

Según lo indagado en la literatura (Sleight, Jason, Pillai, & Mohan., 2009) Se describe que la actividad cerebral registrada en los canales EEG contienen información redundante por lo cual se realizó un experimento con diferentes selecciones de canales de EEG para utilizarlos como entrada a un algoritmo de clasificación, utilizando sólo los canales C3, C4 y Cz de los 64 disponibles basados en el formato internacional 10-20.

A su vez (Deecke, Weinberg, & Brickett., 2009)] argumentan que al utilizar los 64 canales, el algoritmo de clasificación debería ser capaz de separar los canales relevantes (aquellos que contienen información importante para discriminar entre movimientos motores ejecutados e imaginados) de los canales irrelevantes (aquellos que no contienen información importante para discriminar entre movimientos motores ejecutados e imaginados). Por consiguiente en los estudios neurológicos anteriores se demuestra que la actividad neuronal relacionada con los movimientos motores ejecutados e imaginados se encuentra casi exclusivamente en los canales C3, C4 y Cz de los registros del EEG. (Alomari, 2014)



**Figura 19.** Selección de electrodos C3, C4 y Cz, los cuales registran el movimiento de las manos y los pies.

**Adaptado de:** Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., Wolpaw, J.R. BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. IEEE Transactions on Biomedical Engineering 51(6):1034-1043, 2004.

### **4.3. Diseño y Desarrollo**

El presente trabajo de investigación propone el desarrollo de un clasificador basado en redes neuronales interfaz cerebro-computador (BCI) no invasivo, apoyado en la imaginación del movimiento para la extracción de características y la clasificación de las señales EEG; debido a que las registros de la actividad cerebral presentan ruido y además son señales no estacionarias, se debe realizar un proceso de filtrado para eliminar frecuencias innecesarias, por esta razón a las EEG de los canales seleccionados (C3, C4 y Cz) se les realizó un filtro pasa banda con frecuencias de corte entre [0.5 - 50 Hz], este proceso se realiza durante un tiempo de 4.1 segundos y posteriormente el sujeto pasa a un estado de relajación durante 4.2 segundos, esto quiere decir que cada 2 minutos de señal EEG incluye 15 eventos de actividad cerebral separados por un intervalo de descanso. Cabe anotar que el conjunto de datos tiene una frecuencia de muestreo de 160 Hz, es decir que cada evento contiene 656 muestras de la señal original.

#### **4.3.1. Módulo Computacional**

En la etapa de adquisición de las señales EEG, los registros se tomaron de la plataforma gratuita PhysioNet en conjunto con EEG Motor Movement/Imagery Dataset (eegmmidb), siendo seleccionados para el desarrollo de este proyecto 5 sujetos de manera aleatoria separando 3 tareas específicas correspondientes a: descanso (T0), movimiento real o imaginario del puño izquierdo (T1) y el movimiento real o imaginario del puño derecho (T2). Al obtener los registros de los 5 sujetos se formó una matriz de 13.120 filas por 3 columnas por cada canal seleccionado (C3, C4, CZ) con la finalidad de extraer características y propiedades de los componentes de la señal.

El propósito de las técnicas matemáticas (FFT y DWT) es obtener información de frecuencia y descomponer la señal EEG en un subconjunto de coeficientes de resolución múltiple, sus características de expansión y traslación permiten la determinación precisa de todos los componentes producidos por cada posición de la señal, las diferentes funciones de escala y desplazamiento contienen información

sobre las ondas alfa y beta, siendo estas de nuestro interés; apoyados en los siguientes estimadores de amplitud se obtienen características de cada uno de estos detalles.

- **MEDIA CUADRÁTICA (RMS):**

$$RMS = \sqrt{\frac{1}{N} * \sum_{n=1}^N Di^2(n)}$$

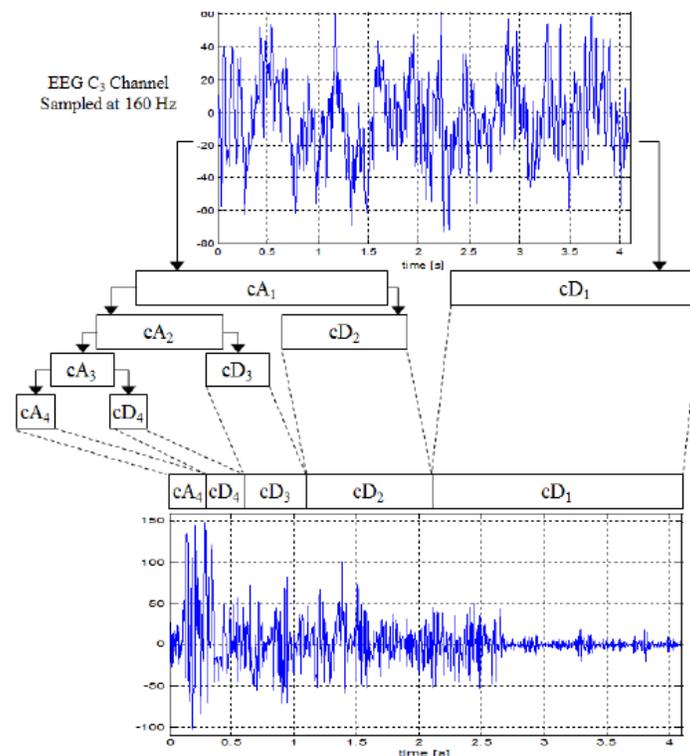
- **VALOR MEDIO ABSOLUTO (MAV):**

$$MAV = \frac{1}{N} \sum_{n=1}^N |Di(n)|$$

Como resultado, la wavelet escalada más fina y larga, permite ver toda la información de la señal, mientras que las pequeñas escalas muestran detalles por acercamiento en las componentes de la señal. Existen muchas wavelets madre agrupadas según su utilidad, las Daubechies son las más apropiadas para aplicarse en análisis de señales discretas (Alomari, 2014).

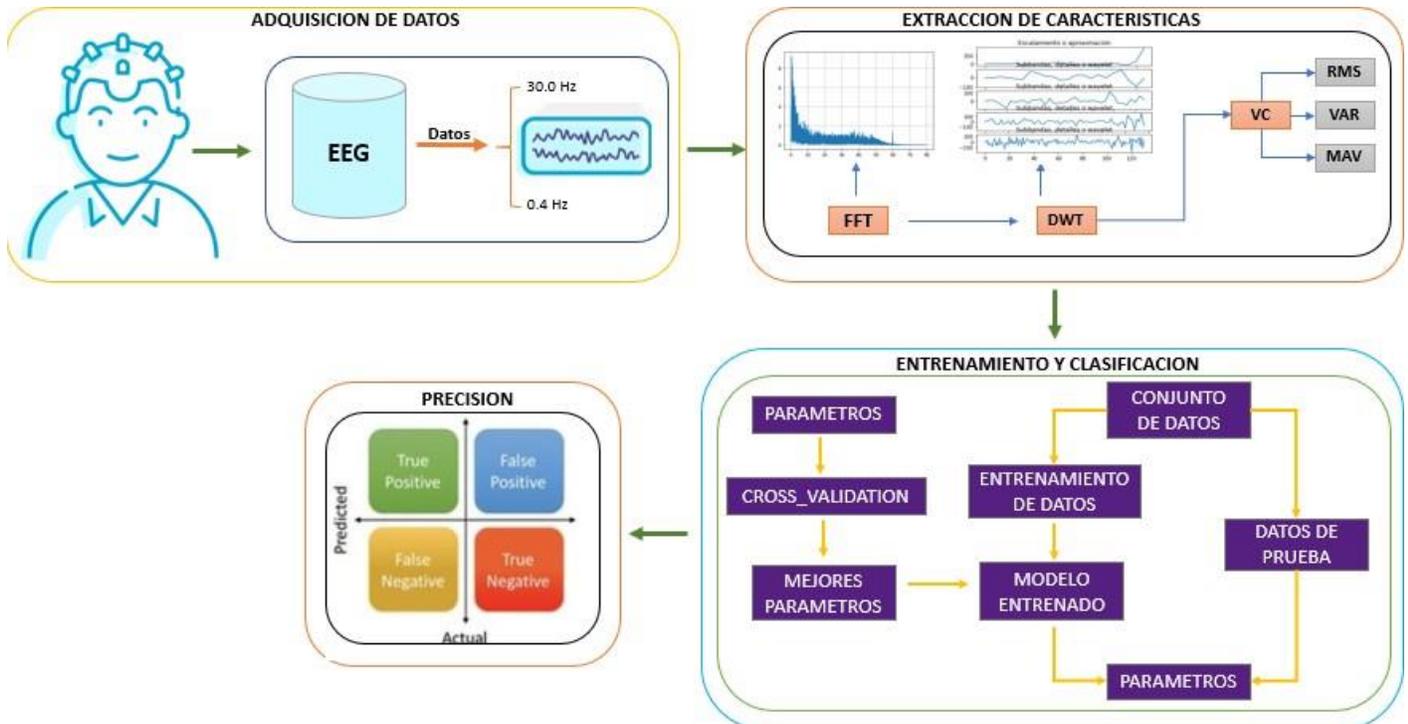
La familia de wavelets presenta un rendimiento alto al momento de realizar la clasificación puesto que las propiedades de la wavelet de una ventana escalable permite localizar los componentes de la señal. Estas propiedades de dilatación y traslación permiten extraer todos los componentes para cada posición creando diferentes escalas y funciones desplazadas (en el dominio del tiempo) de una señal. Como resultado, las wavelet más finas y de gran escala, permiten obtener toda la información de la señal (la imagen grande), mientras que las escalas pequeñas muestran los detalles de la señal haciendo zoom en los componentes de la señal.

El objetivo principal de la DWT es descomponer la señal de EEG registrada en subconjuntos de coeficientes de resolución múltiple: un subconjunto de coeficientes detallados ( $cD_n$ ) y un subconjunto de coeficientes de aproximación ( $cA_n$ ) en el nivel  $n$ . Así, en el primer nivel de descomposición obtenemos  $cD_1$  y  $cA_1$ , luego la primera aproximación  $cA_1$  puede transformarse en  $cD_2$  y  $cA_2$  en el segundo nivel, y así sucesivamente. Para nuestros experimentos, el nivel de descomposición se estableció para generar detalles de cuatro niveles.



**Figura 21:** Descomposición multiresolución de la señal EEG del canal C3

### 4.3.2. Descripción del algoritmo



**Figura 22.** Diagrama de bloques de la metodología implementada

**Adaptado de:** Fuente Propia

Para el procesamiento de las EEG se diseñó una estrategia algorítmica en la plataforma python la cual consta de 4 etapas, para este diseño fue necesario entender el funcionamiento y extraer de la base de datos de PhysioNet, a continuación, se realiza una breve descripción de cada una de las etapas por las que se compone el algoritmo.

#### 4.3.2.1 Adquisición de Datos (Concebir)

Se escogen aleatoriamente los pacientes a estudiar en cada uno de los canales seleccionados (C3, C4 y Cz), la base de datos fue construida en función de las tareas que los pacientes realizaron T0: descanso, T1: movimiento real o imaginario del puño izquierdo y T2: movimiento real o imaginario del puño derecho.

Posteriormente en en la plataforma se importan las librerías necesarias para el desarrollo e implementación del algoritmo, estas librerías permiten leer datos, visualizar

los datos, implementar filtros, extraer datos característicos, clasificar y visualizar la precisión de la clasificación de los datos.

Para el perfecto funcionamiento del algoritmo en python se usaron las siguientes librerías:

```
%matplotlib inline
import numpy as np
import pandas as pd
import pywt
import seaborn as sns
```

- **NUMPY:** Permite el análisis de grandes volúmenes de datos.
- **PANDAS:** Permite el manejo y análisis de estructuras de datos.
- **MATPLOTLIB:** Permite el análisis y la representación gráfica de datos. (Alberca, 2021)
- **PYWT:** Permite la transformación de ondas. (Wasilewski, 2021)
- **SEABORN:** Permite la visualización gráfica de datos. (Rodríguez, 2021)

```
import matplotlib.pyplot as plt
import matplotlib.gridspec as GridSpec
from scipy.fftpack import fft
import openpyxl

from google.colab import files
files.upload()
```

- **GRIDSPEC:** Permite el diseño de cuadrículas dentro de una figura.
- **SCIPY.FFTPACK:** Modulo que permite el cálculo de la transformada de fourier. (Scipy, 2021)
- **OPENPYXL:** Permite la lectura, creación y modificación de ficheros de Excel. (Python, 2021)
- **FROM:** Es usado para subir un archivo desde el disco local de tu pc al entorno interactivo de google colab.

```
from scipy.signal import butter, lfilter
```

- **SciPy:** Proporciona al usuario comandos y clases de alto nivel para manipular y visualizar datos. El subpaquete `scipy.signal` es que permite el procesamiento de la señal en este caso con el diseño de filtro analógico y digital Butterworth. (Brownlee, 2021)

```
import tensorflow
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers.core import Dense
```

- **TENSORFLOW:** Es una biblioteca básica que se puede usar para crear modelos de aprendizaje profundo directamente o mediante el uso de bibliotecas contenedoras que simplifican el proceso. (Brownlee, 2021)
- **SKLEARN.MODEL\_SELECTION (TRAIN\_TEST\_SPLIT):** Permite dividir la base de datos en 2 grupos (datos de entrenamiento y datos de prueba) destinados al entrenamiento y validación del modelo. (IC. Inc, 2021)
- **KERAS:** Es una API de alto nivel de TensorFlow la cual permite accesibilidad para resolver problemas de aprendizaje automático, con un enfoque en el aprendizaje profundo moderno. El módulo `sequential` es adecuado para una pila simple de capas, en la cual cada capa tiene exactamente un tensor de entrada y un tensor de salida. (Team, K., 2021)
- **KERAS.LAYERS.SCORE:** Las capas son los componentes básicos de las redes neuronales en Keras. Una capa consta de una función de cálculo de tensor de entrada y tensor de salida y algún estado, que se mantiene en las variables de TensorFlow (los pesos de la capa).

```
from sklearn import metrics
from sklearn import svm
```

- **SKLEARN:** Es una biblioteca de aprendizaje automático de código abierto que admite el aprendizaje supervisado y no supervisado. También proporciona varias herramientas para el ajuste de modelos, preprocesamiento de datos, selección y evaluación de modelos y muchas otras utilidades.
- **METRICS:** Implementa funciones que evalúan el error de predicción para propósitos específicos. (Metrics and scoring, 2021)
- **SVM:** Permite la creación de la máquina de soporte vectorial.

```
from sklearn.metrics import confusion_matrix
```

- **SKLEARN.METRICS:** Implementa funciones que evalúan el error de predicción para propósitos específicos. [<https://scikit-learn.org/stable/modules/classes.html?highlight=metrics#sklearn-metrics-metrics>]
- **CONFUSION\_MATRIX:** Calcula la matriz de confusión para evaluar la precisión de una clasificación. (Sklearn Metrics, 2021)

```
from sklearn.metrics import precision_score
```

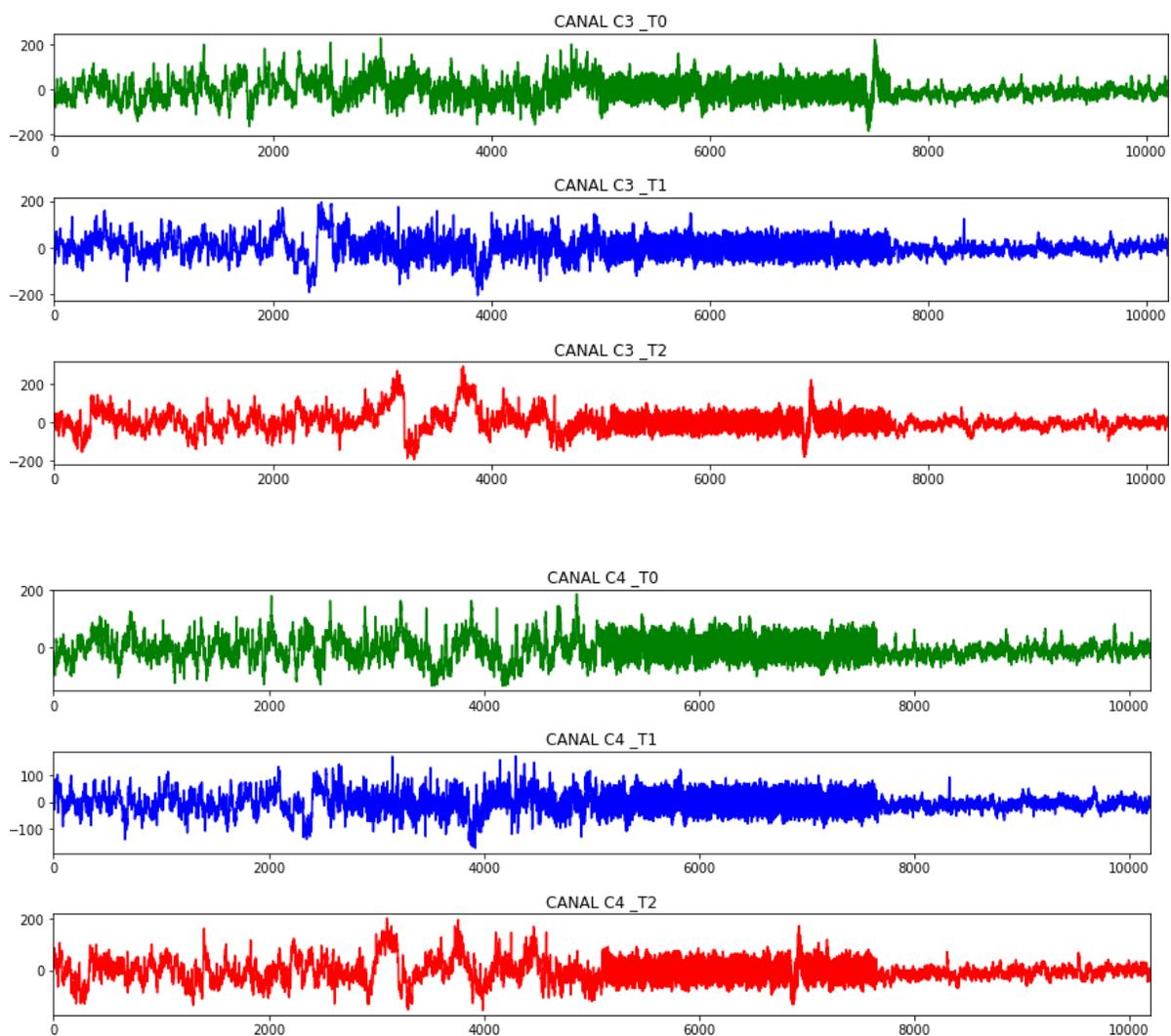
- **PRECISION\_SCORE:** La precisión es la razón donde está el número de verdaderos positivos y el número de falsos positivos. La precisión es intuitivamente la capacidad del clasificador de no etiquetar como positiva una muestra que es negativa. El mejor valor es 1 y el peor valor es 0. (Sklearn Metrics, 2021)

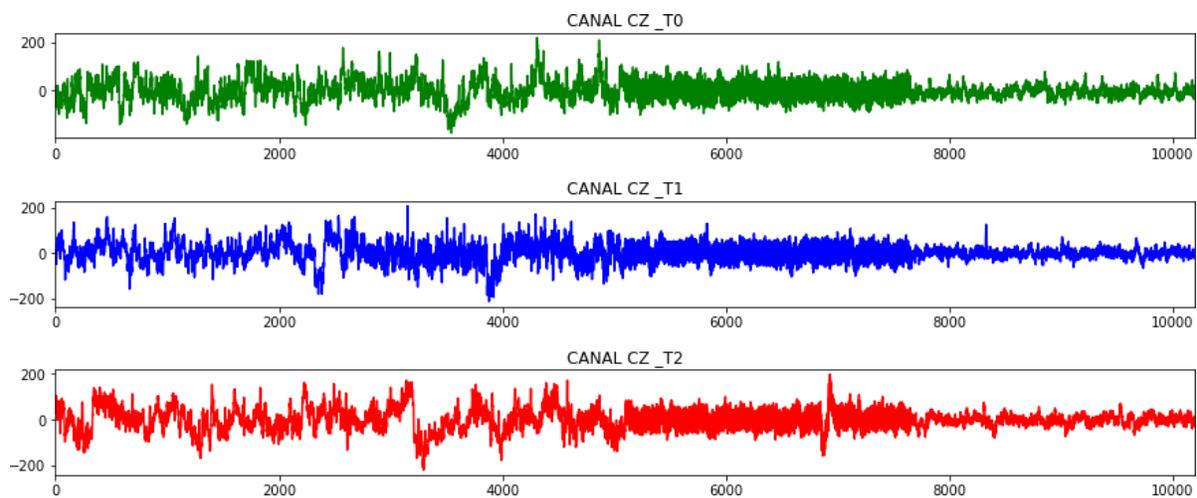
#### 4.3.2.2. Extracción de características (diseñar)

Para extraer datos característicos se hizo uso de 2 herramientas matemáticas (FFT y DWT) con el fin de filtrar las señales originales y deshacer datos que no eran de

nuestro interés, en esta etapa fue necesario implementar un ventaneo de las EEG de cada canal seleccionando las bandas de frecuencia relacionadas a los ritmos cerebrales involucrados en la imaginación motora (alfa y beta), con el fin de obtener coeficientes de aproximación y de detalle para extraer variables probabilísticas (valor RMS y valor MAV) las cuales permiten implementar técnicas de clasificación.

#### 4.3.2.2.1. Señal EEG pura



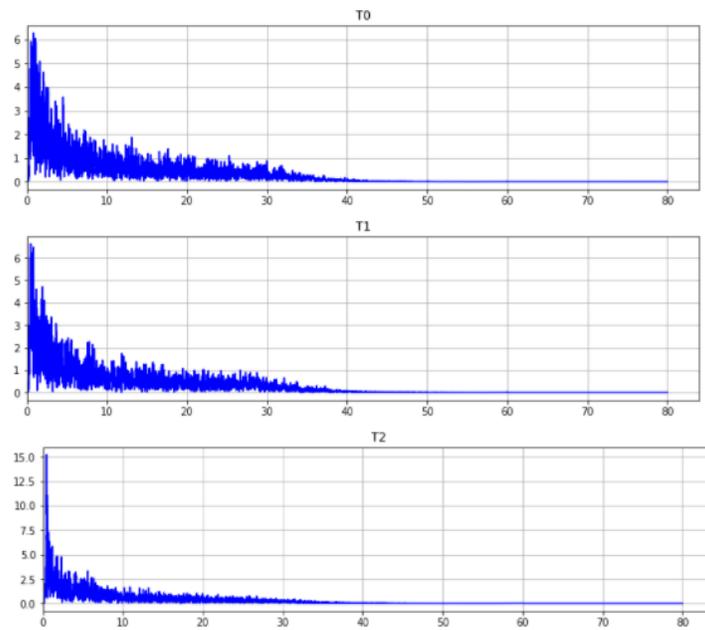


**Figura 233.** Gráficas de señales EEG puras para cada canal (C3,C4 y CZ).

**Adaptado de:** Fuente propia.

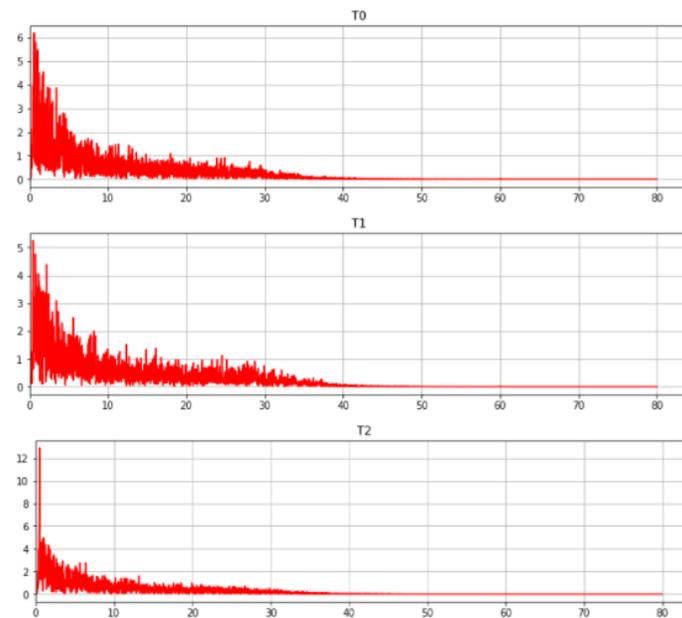
#### *4.3.2.2. Transformada de Fourier para cada canal*

Seguido a lo anterior, se hace la FFT con un filtro pasabanda para cada canal con la intención de borrar frecuencias no deseadas a los ritmos cerebrales relacionados con la imaginación motora. Esta transformada nos posibilita pasar una señal al dominio de frecuencia para de esta forma obtener información que no es notable en el dominio temporal. Ejemplificando, es más simple saber sobre qué ancho de banda se centra la energía de una señal analizándola en el dominio de la frecuencia.



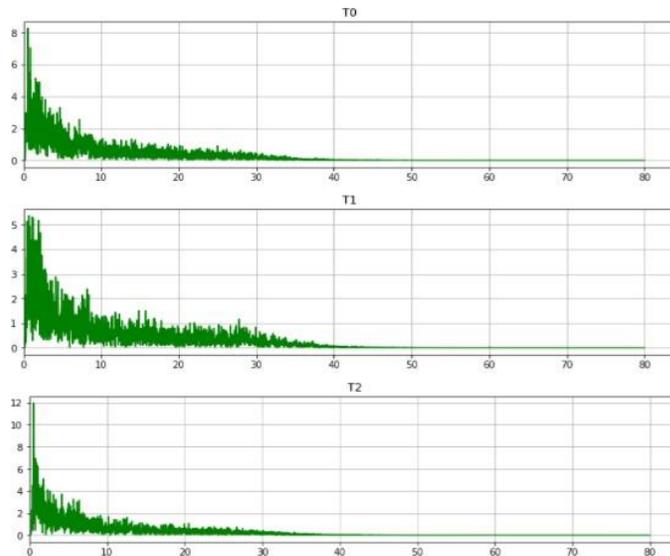
**Figura 24.** Transformada de Fourier para cada tarea del canal C3

**Adaptado de:** Fuente propia



**Figura 255.** Transformada de Fourier para cada tarea del canal C4

**Adaptado de:** Fuente Propia

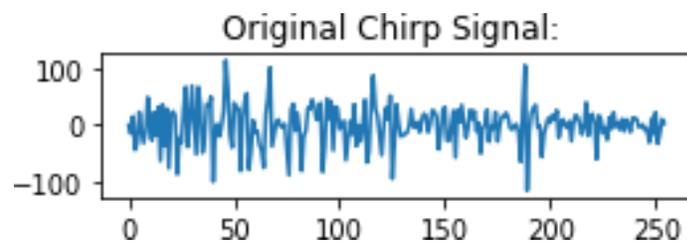


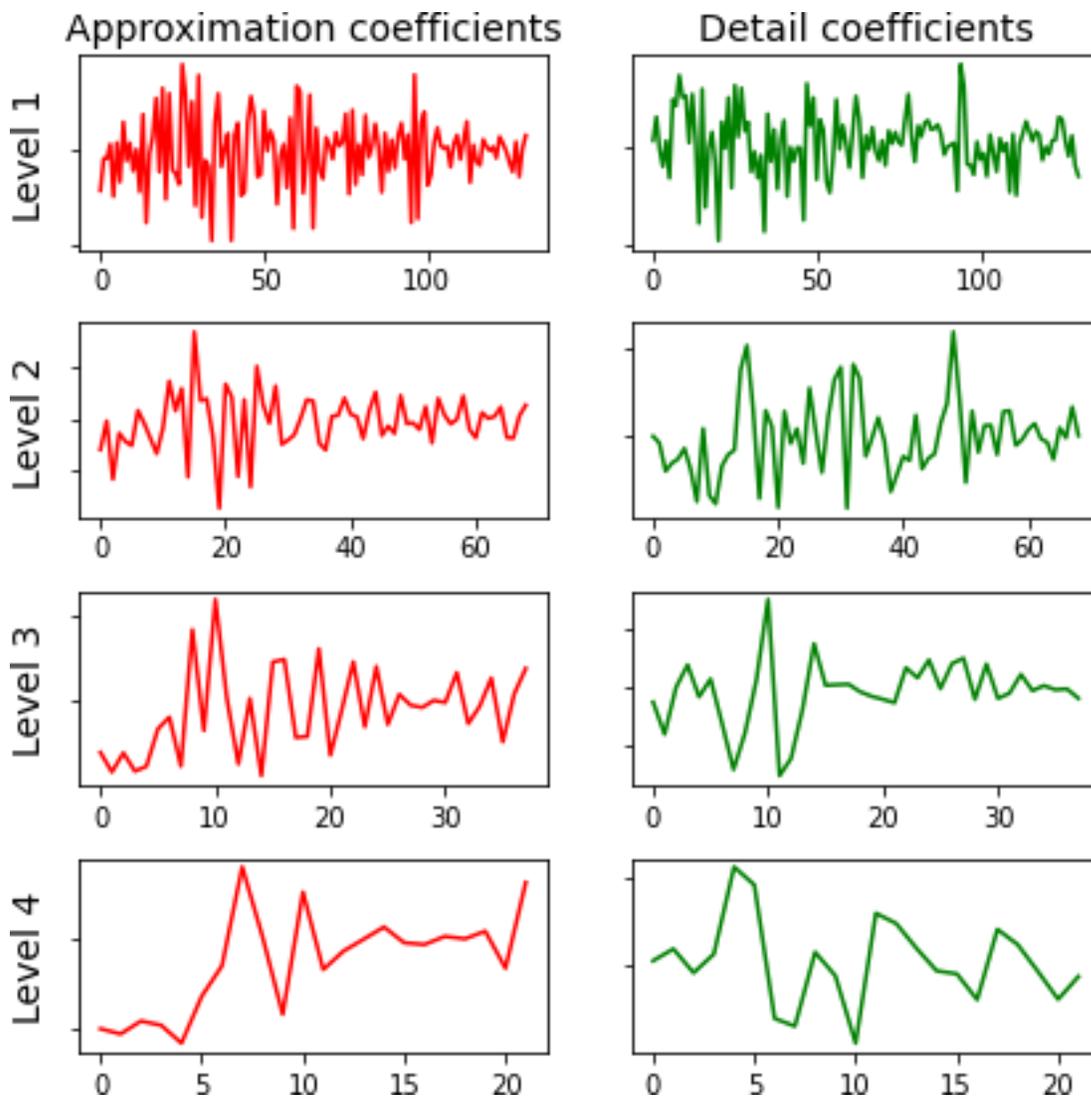
**Figura 26.** Transformada de Fourier para cada tarea del canal Cz

**Adaptado de:** Fuente Propia

#### 4.3.2.2.3. Transformada Wavelet

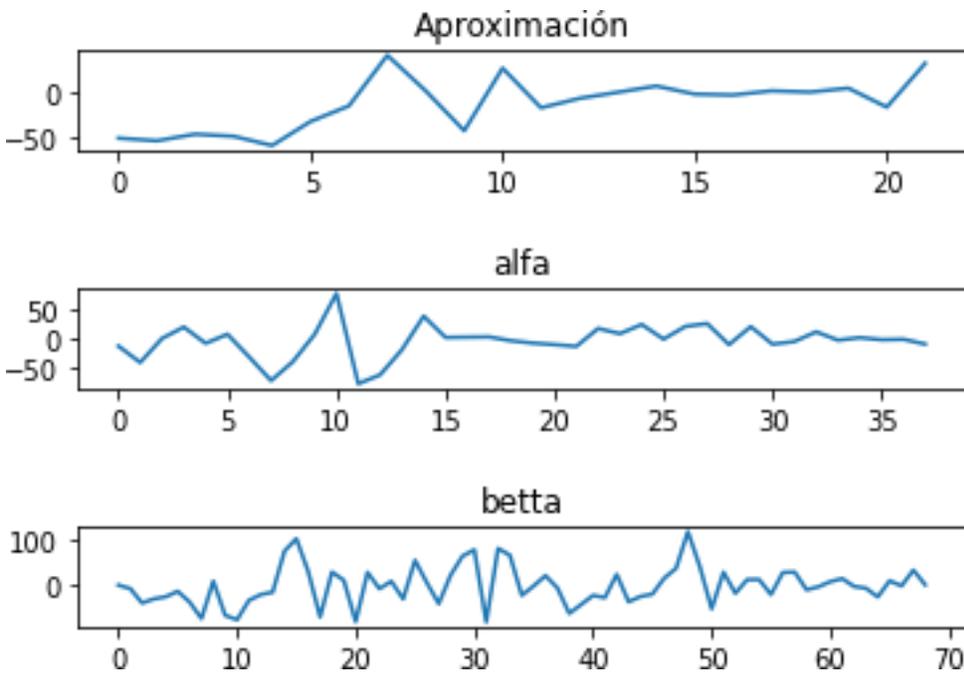
Más adelante visualizamos la DWT que nos permite obtener una representación, descomposición y recomposición de las señales que presenten cambios abruptos en sus elementos de tiempo-frecuencia de manera inmediata, por medio del estudio de multi-resolución con ventanas de longitud variable, adaptadas al cambio de frecuencia de la señal para cada canal con el objeto de hacer la división de los ritmos cerebrales de interés.





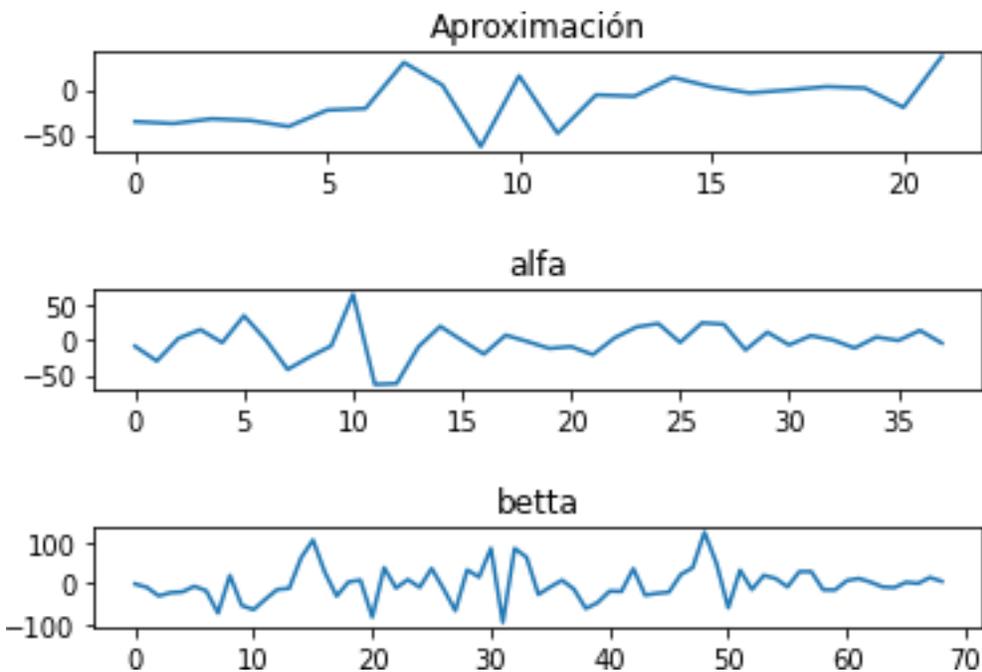
**Figura 27.** Transformada wavelet para cada tarea del canal C3

**Adaptado de:** Fuente propia



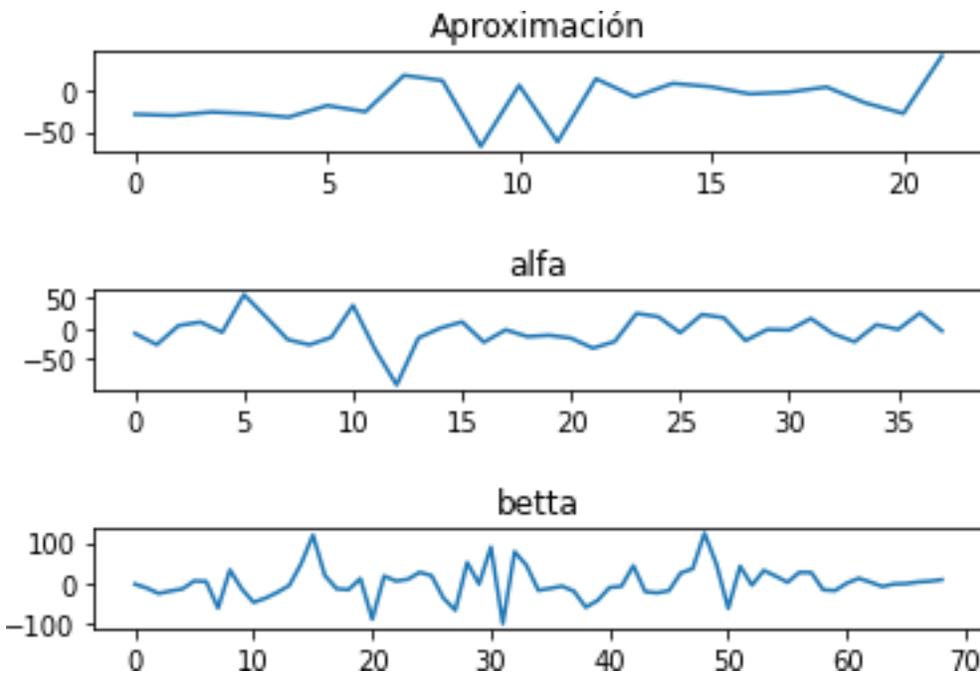
**Figura 28.** Separación de bandas de la tarea T0 aplicando DWT para el canal C3

**Adaptado de:** Fuente propia



**Figura 28.** Separación de bandas de la tarea T1 aplicando DWT para el canal C3

**Adaptado de:** Fuente propia



**Figura 290.** Separación de bandas de la tarea T2 aplicando DWT para el canal C3

**Adaptado de:** Fuente propia

#### 4.3.2.3. Estimadores de Amplitud:

Para finalizar esta etapa, encontramos los estimadores de amplitud utilizados para el entrenamiento y la clasificación de los datos estudiados.

```
def rms(data):  
    valor_rms = np.sqrt(np.mean(np.square(data)))  
    return valor_rms  
  
def mav(data):  
    valor_mav = np.mean(np.abs(data))  
    return valor_mav
```

## 1. ENTRENAMIENTO Y CLASIFICACIÓN (Implementar)

Se implementaron 2 técnicas de clasificación basadas en (SVM, Red neuronal) obteniendo por cada una de ellas un rango de precisión en el cual se puede interpretar cuál de las técnicas clasificó las EEG de una mejor manera.

### Clasificación con SVM

```
from sklearn import metrics
from sklearn import svm
clf = svm.SVC(kernel='linear') # Linear Kernel
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

### Modelo de clasificación con la Red Neuronal.

```
kf = KFold(5, shuffle=True, random_state=42) # Use for Kold classification
fold = 0
for train, test in kf.split(X):
    fold+=1
    print(f"Fold #{fold}")
    x_train = X[train]
    y_train = target[train]
    x_test = X[test]
    y_test = target[test]
    model = Sequential()
    model.add(Dense(30, input_dim=len(X[0]),...
    activation='tanh'))#tanh
    model.add(Dense(20, activation='tanh'))
    model.add(Dense(10, activation='tanh'))
    model.add(Dense(1, activation='relu'))
    model.compile(loss='mean_squared_error',...,
    optimizer='adam', metrics=['accuracy'])
    fit_history=model.fit(x_train,y_train,...,
    validation_data=(x_test,y_test),verbose=0,epochs=1000)
    pred = np.round(model.predict(x_test))
    score = metrics.accuracy_score(y_test, pred)
    print(f"Fold score (accuracy): {score}")
```

#### 4.3.2.4. Precisión (Operar)

En esta última etapa se realiza la matriz de confusión para las 2 técnicas de clasificación desarrolladas durante el algoritmo, con el fin de determinar cuál de éstas es más precisa que la otra y por ende sería la apropiada para clasificar las EEG puestas a prueba en esta estrategia computacional.

```
pred = np.round(model.predict(x_test))

from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)

array([[ 6,  2,  0],
       [ 0,  5,  0],
       [ 0,  1, 10]])

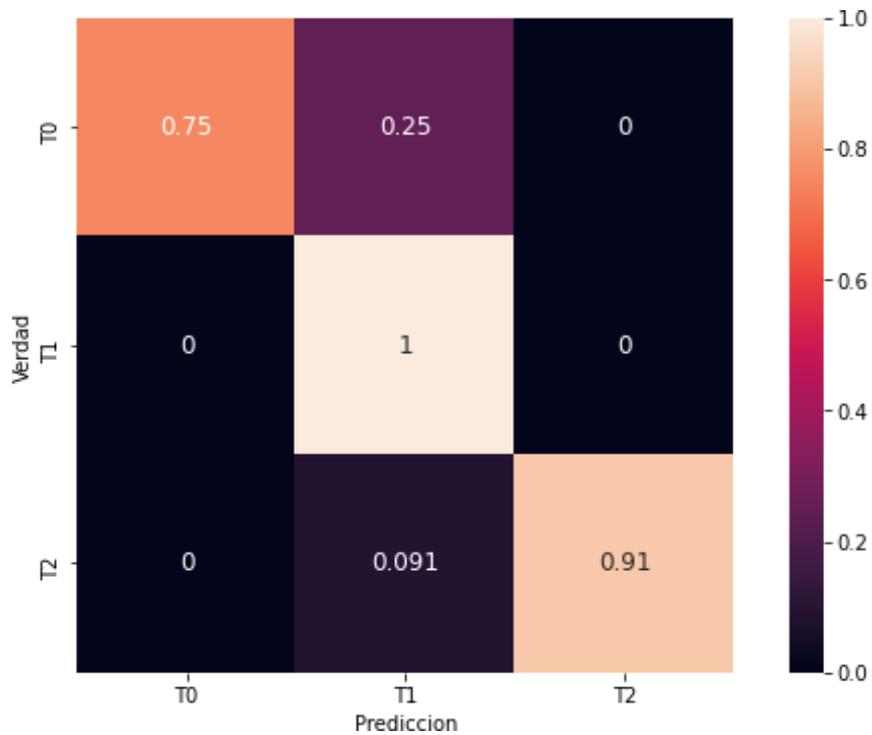
precision = precision_score(y_test, y_pred,...,
average='micro')
precision

0.875
```

Matriz de confusión del modelo (Red Neuronal).

```
cm = confusion_matrix(y_test, y_pred)
cm_norm = cm.astype('float')...,
/ cm.sum(axis=1)[:, np.newaxis]

df_cm = pd.DataFrame(cm_norm, ['T0', 'T1', 'T2'],...,
['T0', 'T1', 'T2'])
plt.figure(figsize = (10,6))
conf = sns.heatmap(df_cm, annot=True, square=True,...,
annot_kws={"size": 12})
conf.set_xlabel('Prediccion')
conf.set_ylabel('Verdad')
```



**Figura 31. Matriz de confusión del modelo (Red Neuronal)**

**Adaptado por:** Fuente propia.

Matriz de confusión de la SVM

```

y_predSVM = np.round(clf.predict(X_test))

from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_predSVM)

array([[ 4,  2,  0],
       [ 3, 12,  0],
       [ 0,  0,  9]])

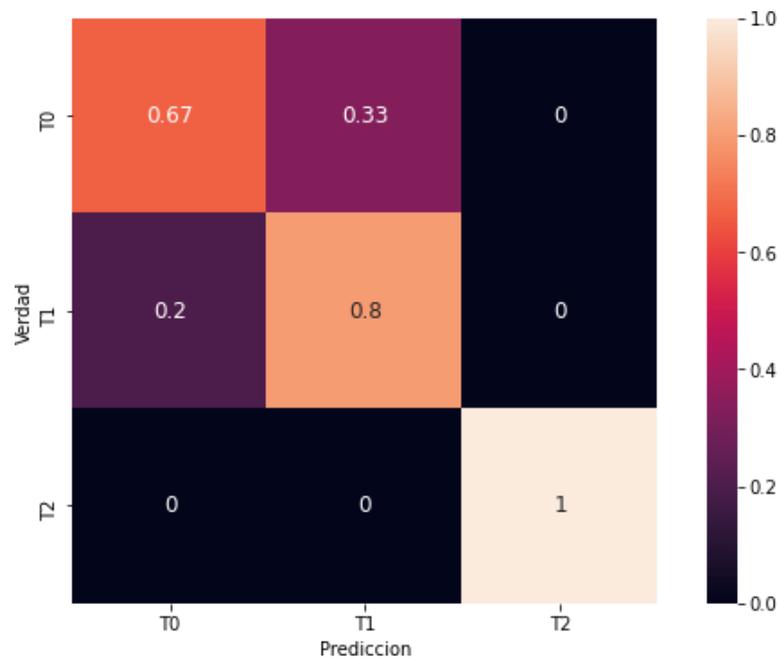
precisionSVM = precision_score(y_test, ...,
y_predSVM, average='micro')
precisionSVM

```

```
0.8333333333333334

cmSVM = confusion_matrix(y_test, y_predSVM)
cm_normSVM = cmSVM.astype('float') / cmSVM.sum(axis=1)[:, np.newaxis]

df_cmSVM = pd.DataFrame(cm_normSVM, ['T0', 'T1', 'T2'], ['T0', 'T1', 'T2'])
plt.figure(figsize = (10,6))
conf = sns.heatmap(df_cmSVM, annot=True, square=True, annot_kws={"size": 12})
conf.set_xlabel('Prediccion')
conf.set_ylabel('Verdad')
```



**Figura 30.** Matriz de confusión de la SVM.

**Adaptado por:** Fuente propia

## 5. Análisis y Resultados

En este capítulo se pretende explicar los resultados obtenidos en cada una de las etapas de clasificación, patrones de clasificación en Red Neuronal y SVM para comprobar que tan efectivo han sido las técnicas de clasificación. Es importante tener en cuenta que todos estos resultados son evaluados bajo las mismas condiciones en las señales EEG, es decir mismos sujetos, paradigmas y detalles técnicos de adquisición.

### 5.1 Resultados de la Matriz de confusión

El porcentaje de acierto en las técnicas de clasificación fue el mismo tanto para SVM como la Red Neuronal, es necesario aclarar que para la clasificación y el entrenamiento se utilizó el valor RMS de la onda alfa, ya que este nos permitió obtener una precisión mayor que la onda beta y el valor MAV.

En la diagonal de cada matriz encontramos los siguientes resultados:

<b>Tabla 4.</b> <i>Tabla de resultados de clasificadores SVM y Red Neuronal</i>						
<b>Modelo de clasificación</b>	<b>T0 aciertos</b>	<b>T0 fallos</b>	<b>T1 aciertos</b>	<b>T1 fallos</b>	<b>T2 aciertos</b>	<b>T2 fallos</b>
<b>SVM</b>	0.83	0.17	0.91	0.091	0.92	0.077
<b>Red Neuronal</b>	0.83	0.17	0.91	0.091	0.92	0.077

La tabla No. 4 nos muestra los porcentajes de acierto y fallo de los clasificadores en cada una de las tareas estudiadas, se puede analizar que tanto para SVM y la Red Neuronal la probabilidad de acierto para cada tarea fue la siguiente: T0 = 83%, T1 = 91% y T2 = 92%, estos resultados nos arrojan una precisión para ambos clasificadores de un 90%.

Es necesario aclarar que cada vez que el algoritmo es ejecutado genera un nuevo vector de pesos y demás métricas, lo que nos lleva a obtener nuevos resultados (mayores o menores) en las clasificaciones y precisiones de cada técnica.

Ahora bien, se desea presentar al lector la tabla de resultados de la matriz de confusión también en la siguiente medida.

<b>Tabla 5.</b> Tabla de resultados de matriz de confusión					
<b>CLASIFICADORES</b>	<b>MTOS</b>	<b>ACIERTOS</b>	<b>FALLOS</b>	<b>TOTAL</b>	<b>TOTAL, MTOS</b>
<b>SVM</b>	T0	9	1	10	<b>30</b>
	T1	11	1	12	
	T2	8	0	8	
<b>RNAf</b>	T0	7	1	8	<b>24</b>
	T1	5	0	5	
	T2	9	2	11	

Estos resultados permiten interpretar la precisión de cada uno de los clasificadores utilizados, en este caso se debe realizar la sumatoria de los aciertos de cada modelo y dividirlo por el total de movimientos y multiplicarlo por cien para obtener la precisión del modelo, así como se observa a continuación:

#### **PORCENTAJE DE FALLAS**

SVM:

$$((1 + 1 + 0) \div 30) * 100 = 6.66\%$$

RNAf:

$$((1 + 0 + 2) \div 24) * 100 = 12.5\%$$

## PORCENTAJE DE ACIERTOS

SVM:

$$((8 + 9 + 11) \div 30) * 100 = 93.3\%$$

RNAf:

$$((5 + 7 + 9) \div 24) * 100 = 87.5\%$$

En la tabla se observa que los clasificadores tuvieron una precisión mayor al 87% de los datos y el porcentaje de fallas fue menor al 13%, lo cual permite interpretar que los modelos de clasificación utilizados en este proyecto logran clasificar de manera adecuada las tres intenciones de movimiento estudiadas (T0, T1 y T2)

## 5.2 Resultados de Clasificadores

### Máquina de soporte vectorial

```
from sklearn import metrics
from sklearn import svm
clf = svm.SVC(kernel='linear') # Linear Kernel
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

print("Accuracy:", metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.8333333333333334
```

```
# evaluamos el modelo
scores = model.evaluate(X_test, y_test)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_test_function.<locals>.test_function at 0
1/1 [=====] - 0s 110ms/step - loss: 0.0392 - accuracy: 0.7391

accuracy: 73.91%
```

**Figura 31.** Clasificación de la Red Neuronal.

**Adaptado por:** Fuente propia.

En este análisis de resultados se evidencia un rango de clasificación del [73-90%], esto nos lleva a interpretar que el clasificador basado en SVM presenta una mayor precisión, estos clasificadores fueron validados en la matriz de confusión.

### 5.3 Resultados de Precisión

Se logra evidenciar que la clasificación de la SVM y la Red Neuronal fue del 0.9, es decir una precisión del 90% en la clasificación de las tareas realizadas por los sujetos (T0, T1 y T2), con esto podemos analizar que en esta ejecución del algoritmo las técnicas de clasificación son adecuadas para la identificación de la intención de movimiento de los miembros superiores.

#### Precisión de la matriz de confusión de la SVM.

```
y_predSVM = np.round(clf.predict(X_test))

confusion_matrix(y_test, y_predSVM)
array([[ 4,  2,  0],
       [ 3, 12,  0],
       [ 0,  0,  9]])

precisionSVM = precision_score(y_test, ...,
y_predSVM, average='micro')
```

```
precisionSVM  
  
0.8333333333333334
```

### **Precisión de la matriz de confusión de la RNAf.**

```
import numpy as np  
y_pred = np.round(model.predict(x_test))  
  
confusion_matrix(y_test, y_pred)  
array([[ 6,  2,  0],  
       [ 0,  5,  0],  
       [ 0,  1, 10]])  
  
precision = precision_score(y_test, y_pred,...,  
average='micro')  
precision  
  
0.875
```

## 5. Conclusiones y recomendaciones

### 5.1. Conclusiones

En este capítulo, principalmente podemos concluir que la extracción de características utilizando la FFT y la DWT fueron las apropiadas para la utilización de las técnicas de clasificación, ya que los filtros de la señal y el ventaneo de la misma fue conveniente para seleccionar los ritmos cerebrales (alfa y beta).

La transformada wavelet evidenció ser una de las maneras más rápidas de filtrar las bandas de frecuencia (alfa y beta) a diferencia del proceso de filtrado tradicional el cual debe realizarse de manera separada aislando una por una las frecuencias de interés.

Para la etapa de clasificación se usaron máquinas de soporte vectorial probando 2 tipos de kernels (RBF y Lineal), siendo el kernel lineal el más efectivo debido a que separa la señal en sus diferentes tareas. Esta técnica de clasificación dio como resultado una precisión correspondiente al 90%, esto permite interpretar que las tareas fueron clasificadas de una manera adecuada obteniendo solo un 10% de datos clasificados de manera incorrecta demostrando la eficacia del método.

Uno de los inconvenientes fue la implementación de la red neuronal como clasificador ya que al ejecutar el algoritmo su precisión variaba, es por eso que se optó por modificar el número de capas, las funciones de activación y el número de neuronas por capa con el fin de obtener una clasificación con un alto grado de precisión.

Por otra parte en este trabajo se propone la RNAf para clasificar las señales de imaginación motora de EEG al cual se le realizó la clasificación de las tres tareas mentales (T0, T1 Y T2), mostrando resultados más prometedores en relación con lo encontrado en el estado del arte; Esta clasificación establece un enfoque novedoso para la investigación de la interfaz cerebro-ordenador.

Y finalmente se concluye que la validación de los dos clasificadores en la matriz de confusión da como resultado una precisión del 0.9, interpretando este resultado una buena técnica de clasificación.

Desde la óptica de funcionalidad y producción científica, lo cierto es que el aporte de la presente obra representa una profundización sobre las señales EEG encaminadas a mejorar los prototipos ya identificados en cuanto a la necesidad de producción en masa de manos robóticas motoras, las cuales importan a un número considerable de la población en condición de discapacidad que podrá tener acceso a ellas para mejorar su calidad de vida.

Los avances a la luz de la tecnología y con el uso de herramientas como los softwares analizados suponen un verdadero beneficio interdisciplinar entre la ingeniería de sistemas, la ingeniería biomédica y las demás áreas que contribuyen a la robótica, comoquiera que resulta imperioso que cada una de las áreas del conocimiento que intervienen en su producción representen la viabilidad de adquisición para los usuarios en términos de seguridad, garantía y funcionalidad accesible.

Por ello, desde el código implementado, hasta el tratamiento y el estudio de resultados de los análisis de laboratorio planteados, son un medio idóneo y necesario para el último fin altruista de este tipo de facilidades en pro de los seres humanos.

No es un secreto que todo instrumento puesto a disposición de la comunidad debe garantizar este tipo de contenidos mínimos en aras de ofrecer un verdadero beneficio y una herramienta confiable que realmente facilite las condiciones de utilización de cada mecanismo

## **5.2. Recomendaciones**

Se recomienda utilizar una red neuronal convolucional y comparar los resultados obtenidos en esta investigación para así tener una mayor certeza de cuál de los clasificadores es más preciso en la identificación del movimiento de la mano derecha o izquierda bajo el paradigma de imaginación motora.

Para futuras investigaciones en esta área, se propone la implementación de un procedimiento diferente para la extracción de características o el uso de clasificadores con más poder computacional para poder controlar sillas de ruedas, prótesis, y hasta combinar esta línea con la realidad virtual o la realidad aumentada.

## A. Anexo: CODIGO IMPLEMENTADO

```
%matplotlib inline
import numpy as np
import pandas as pd
import pywt
import seaborn as sns
#import scipy.io as sio
import matplotlib.pyplot as plt
import matplotlib.gridspec as GridSpec
from scipy.fftpack import fft
import openpyxl
```

### Importar base de datos en formato .CSV

```
from google.colab import files
files.upload()

C3=pd.read_csv('C3_4sujetos.csv',header=0)
print(C3)
C4=pd.read_csv('C4_4sujetos.csv',header=0)
print(C4)
Cz=pd.read_csv('Cz_4sujetos.csv',header=0)
print(Cz)

C3 = 'C3_4sujetos.csv'
C4 = 'C4_4sujetos.csv'
Cz = 'Cz_4sujetos.csv'

C3=pd.read_csv(C3)
C4=pd.read_csv(C4)
Cz=pd.read_csv(Cz)

C3=np.array(C3)
C4=np.array(C4)
Cz=np.array(Cz)

C3.shape
C4.shape
Cz.shape
```

**Graficas de los datos raw para los canales C3, C4 y Cz**

```

#C3
plt.subplot(3,1,1),plt.plot(C3[:,0],color = 'green')
plt.subplot(3,1,2),plt.plot(C3[:,1],color = 'blue')
plt.subplot(3,1,3),plt.plot(C3[:,2],color = 'red')
#C4
plt.subplot(3,1,1),plt.plot(C4[:,0],color = 'green')
plt.subplot(3,1,2),plt.plot(C4[:,1],color = 'blue')
plt.subplot(3,1,3),plt.plot(C4[:,2],color = 'red')
#Cz
plt.subplot(3,1,1),plt.plot(Cz[:,0],color = 'green')
plt.subplot(3,1,2),plt.plot(Cz[:,1],color = 'blue')
plt.subplot(3,1,3),plt.plot(Cz[:,2],color = 'red')

```

### Espectro en frecuencia de los canales C3, C4 y Cz

```

#T0
yC3 = fft(C3[:,0])
yC4 = fft(C4[:,0])
yCz = fft(Cz[:,0])
L=len(C4[:,0])
T = 1 / 160
xf = np.linspace(0.0, 1.0/(2.0*T), L//2)

plt.subplot(3,1,1),plt.plot(xf, 2.0/L * np.abs(yC3[0:L//2]),color = 'green
')
plt.subplot(3,1,2),plt.plot(xf, 2.0/L * np.abs(yC4[0:L//2]),color = 'blue'
)
plt.subplot(3,1,3),plt.plot(xf, 2.0/L * np.abs(yCz[0:L//2]),color = 'red')
plt.grid()
plt.show()
#T1
yC3 = fft(C3[:,1])
yC4 = fft(C4[:,1])
yCz = fft(Cz[:,1])
L=len(C4[:,1])
T = 1 / 160
xf = np.linspace(0.0, 1.0/(2.0*T), L//2)

plt.subplot(3,1,1),plt.plot(xf, 2.0/L * np.abs(yC3[0:L//2]),color = 'green
')
plt.subplot(3,1,2),plt.plot(xf, 2.0/L * np.abs(yC4[0:L//2]),color = 'blue'
)
plt.subplot(3,1,3),plt.plot(xf, 2.0/L * np.abs(yCz[0:L//2]),color = 'red')

```

```

plt.grid()
plt.show()
#T2
yC3 = fft(C3[:,2])
yC4 = fft(C4[:,2])
yCz = fft(Cz[:,2])
L=len(C4[:,2])
T = 1 / 160
xf = np.linspace(0.0, 1.0/(2.0*T), L//2)

plt.subplot(3,1,1),plt.plot(xf, 2.0/L * np.abs(yC3[0:L//2]),color = 'green
')
plt.subplot(3,1,2),plt.plot(xf, 2.0/L * np.abs(yC4[0:L//2]),color = 'blue'
)
plt.subplot(3,1,3),plt.plot(xf, 2.0/L * np.abs(yCz[0:L//2]),color = 'red')
plt.grid()
plt.show()

```

### **Función para implementar un filtro pasa-bandas**

```

from scipy.signal import butter, lfilter

def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='bandpass')
    return b, a

def butter_bandpass_filter(data, lowcut, highcut, fs, order=5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    y = lfilter(b, a, data)
    return y

C3F=np.ndarray(shape=(3, 10200))
C4F=np.ndarray(shape=(3, 10200))
CZF=np.ndarray(shape=(3, 10200))
C3F.shape
C4F.shape
CZF.shape

```

### **Aplicar filtro pasa-banda en las bandas de 0.4Hz a 30 Hz**

```
fs = 160
lowcut = 0.4
highcut = 30
for i in range(0, 3):
    signalA=C3[:,i]
    signalB=C4[:,i]
    signalC=Cz[:,i]

    C3F[i,:] = butter_bandpass_filter(signalA, lowcut, highcut, fs,order=5
)
    C4F[i,:] = butter_bandpass_filter(signalB, lowcut, highcut, fs,order=5
)
    CZF[i,:] = butter_bandpass_filter(signalC, lowcut, highcut, fs,order=5
)

C3FT=np.transpose(C3F)
C4FT=np.transpose(C4F)
CZFT=np.transpose(CZF)
C3FT.shape

# FFT C3
#T0
yf1 = fft(C3FT[:,0])
L=len(C3FT[:,0])
T = 1 / 160
xf1 = np.linspace(0.0, 1.0/(2.0*T), L//2)

## T1
yf2 = fft(C3FT[:,1])
L=len(C3FT[:,1])
T = 1 / 160
xf2 = np.linspace(0.0, 1.0/(2.0*T), L//2)

## T2
yf3 = fft(C3FT[:,2])
L=len(C3FT[:,2])
T = 1 / 160
xf3 = np.linspace(0.0, 1.0/(2.0*T), L//2)

import matplotlib.pyplot as plt
fig=plt.figure(figsize=(12,6))
```

```
fig1, ax1 = plt.subplots(1, 1, figsize=(12,3));
ax1.plot(xf1, 2.0/L * np.abs(yf1[0:L//2]), color='blue'), plt.grid()
ax1.set_xlim(0)
ax1.set_title("T0")
fig2, ax2 = plt.subplots(1, 1, figsize=(12,3));
ax2.plot(xf2, 2.0/L * np.abs(yf2[0:L//2]), color='blue'), plt.grid()
ax2.set_xlim(0)
ax2.set_title("T1")
fig3, ax3 = plt.subplots(1, 1, figsize=(12,3));
ax3.plot(xf3, 2.0/L * np.abs(yf3[0:L//2]), color='blue'), plt.grid()
ax3.set_xlim(0)
ax3.set_title("T2")
plt.show()
# FFT C4
#T0
yf1 = fft(C4FT[:,0])
L=len(C4FT[:,0])
T = 1 / 160
xf1 = np.linspace(0.0, 1.0/(2.0*T), L//2)

## T1
yf2 = fft(C4FT[:,1])
L=len(C4FT[:,1])
T = 1 / 160
xf2 = np.linspace(0.0, 1.0/(2.0*T), L//2)

## T2
yf3 = fft(C4FT[:,2])
L=len(C4FT[:,2])
T = 1 / 160
xf3 = np.linspace(0.0, 1.0/(2.0*T), L//2)

import matplotlib.pyplot as plt
fig=plt.figure(figsize=(12,6))

fig1, ax1 = plt.subplots(1, 1, figsize=(12,3));
ax1.plot(xf1, 2.0/L * np.abs(yf1[0:L//2]), color='red'), plt.grid()
ax1.set_xlim(0)
ax1.set_title("T0")
fig2, ax2 = plt.subplots(1, 1, figsize=(12,3));
ax2.plot(xf2, 2.0/L * np.abs(yf2[0:L//2]), color='red'), plt.grid()
ax2.set_xlim(0)
```

```
ax2.set_title("T1")
fig3, ax3 = plt.subplots(1, 1, figsize=(12,3));
ax3.plot(xf3, 2.0/L * np.abs(yf3[0:L//2]), color='red'), plt.grid()
ax3.set_xlim(0)
ax3.set_title("T2")
plt.show()
# FFT Cz
#T0
yf1 = fft(CZFT[:,0])
L=len(CZFT[:,0])
T = 1 / 160
xf1 = np.linspace(0.0, 1.0/(2.0*T), L//2)

## T1
yf2 = fft(CZFT[:,1])
L=len(CZFT[:,1])
T = 1 / 160
xf2 = np.linspace(0.0, 1.0/(2.0*T), L//2)

## T2
yf3 = fft(CZFT[:,2])
L=len(CZFT[:,2])
T = 1 / 160
xf3 = np.linspace(0.0, 1.0/(2.0*T), L//2)

import matplotlib.pyplot as plt
fig=plt.figure(figsize=(12,6))

fig1, ax1 = plt.subplots(1, 1, figsize=(12,3));
ax1.plot(xf1, 2.0/L * np.abs(yf1[0:L//2]), color='green'), plt.grid()
ax1.set_xlim(0)
ax1.set_title("T0")
fig2, ax2 = plt.subplots(1, 1, figsize=(12,3));
ax2.plot(xf2, 2.0/L * np.abs(yf2[0:L//2]), color='green'), plt.grid()
ax2.set_xlim(0)
ax2.set_title("T1")
fig3, ax3 = plt.subplots(1, 1, figsize=(12,3));
ax3.plot(xf3, 2.0/L * np.abs(yf3[0:L//2]), color='green'), plt.grid()
ax3.set_xlim(0)
ax3.set_title("T2")
plt.show()
```

**Separación en ventanas de 255 datos**

```

C3F1 = np.transpose(np.array_split(C3FT[:,0], 255))#256-512-1024
C3F2 = np.transpose(np.array_split(C3FT[:,1], 255))
C3F3 = np.transpose(np.array_split(C3FT[:,2], 255))

C4F1 = np.transpose(np.array_split(C4FT[:,0], 255))#256-512-1024
C4F2 = np.transpose(np.array_split(C4FT[:,1], 255))
C4F3 = np.transpose(np.array_split(C4FT[:,2], 255))

CZF1 = np.transpose(np.array_split(CZFT[:,0], 255))#256-512-1024
CZF2 = np.transpose(np.array_split(CZFT[:,1], 255))
CZF3 = np.transpose(np.array_split(CZFT[:,2], 255))

A=np.concatenate((C3F1, C3F2, C3F3), axis=0)
B=np.concatenate((C4F1, C4F2, C4F3), axis=0)
C=np.concatenate((CZF1, CZF2, CZF3), axis=0)

data=np.concatenate((A,B,C), axis=0)
np.shape(data)

```

**Descomposicion de la señal utilizando DWT (cD1(40-80Hz), cD2(20-40Hz) ritmo Alfa, cD3(10-20Hz) ritmo Beta y cD4(5-10Hz) ritmo Theta)**

```

x = np.linspace(0, 1, num=255)
chirp_signal = A[0]

fig, ax = plt.subplots(figsize=(4,1))
ax.set_title("Original Chirp Signal: ")
ax.plot(chirp_signal)
plt.show()

data = chirp_signal
waveletname = 'db4'

fig, axarr = plt.subplots(nrows=4, ncols=2, figsize=(6,6))
for ii in range(4):
    (data, coeff_d) = pywt.dwt(data, waveletname)
    axarr[ii, 0].plot(data, 'r')
    axarr[ii, 1].plot(coeff_d, 'g')
    axarr[ii, 0].set_ylabel("Nivel {}".format(ii + 1), fontsize=14, rotation=90)
    axarr[ii, 0].set_yticklabels([])
    if ii == 0:

```

```

        axarr[ii, 0].set_title("Coeficientes de Aproximación", fontsize=14
)
        axarr[ii, 1].set_title("Coeficientes de Detalle", fontsize=14)
        axarr[ii, 1].set_yticklabels([])
plt.tight_layout()
plt.show()

cA,cD1,cD2,cD3,cD4=pywt.wavedec(A[0], 'db4', level=4)

plt.subplot(5,1,1),plt.plot(cA),plt.title('Aproximación')
#plt.subplot(5,1,2),plt.plot(cD1),plt.title('Subbandas, detalles o wavelet
')
plt.subplot(5,1,3),plt.plot(cD2),plt.title('alfa')
plt.subplot(5,1,5),plt.plot(cD3),plt.title('beta')
#plt.subplot(5,1,5),plt.plot(cD4),plt.title('Subbandas, detalles o wavelet
')
len(cA),len(cD1),len(cD2),len(cD3),len(cD4)
cA,cD1,cD2,cD3,cD4=pywt.wavedec(A[1], 'db4', level=4)

plt.subplot(5,1,1),plt.plot(cA),plt.title('Aproximación')
#plt.subplot(5,1,2),plt.plot(cD1),plt.title('Subbandas, detalles o wavelet
')
plt.subplot(5,1,3),plt.plot(cD2),plt.title('alfa')
plt.subplot(5,1,5),plt.plot(cD3),plt.title('beta')
#plt.subplot(5,1,5),plt.plot(cD4),plt.title('Subbandas, detalles o wavelet
')
len(cA),len(cD1),len(cD2),len(cD3),len(cD4)
cA,cD1,cD2,cD3,cD4=pywt.wavedec(A[2], 'db4', level=4)

plt.subplot(5,1,1),plt.plot(cA),plt.title('Aproximación')
#plt.subplot(5,1,2),plt.plot(cD1),plt.title('Subbandas, detalles o wavelet
')
plt.subplot(5,1,3),plt.plot(cD2),plt.title('alfa')
plt.subplot(5,1,5),plt.plot(cD3),plt.title('beta')
#plt.subplot(5,1,5),plt.plot(cD4),plt.title('Subbandas, detalles o wavelet
')
len(cA),len(cD1),len(cD2),len(cD3),len(cD4)

C3D2=np.ndarray(shape=(120, 38))
C3D3=np.ndarray(shape=(120, 69))
C4D2=np.ndarray(shape=(120, 38))
C4D3=np.ndarray(shape=(120, 69))

```

```
CZD2=np.ndarray(shape=(120, 38))
CZD3=np.ndarray(shape=(120,69))
RMS1= np.ndarray(120)
RMS2= np.ndarray(120)
RMS3= np.ndarray(120)
RMS4= np.ndarray(120)
RMS5= np.ndarray(120)
RMS6= np.ndarray(120)
MAV1= np.ndarray(120)
MAV2= np.ndarray(120)
MAV3= np.ndarray(120)
MAV4= np.ndarray(120)
MAV5= np.ndarray(120)
MAV6= np.ndarray(120)
np.shape(C3D2)
```

### Funciones para la extraccion de caracterísiticas temporales RMS y MAV

```
def rms(data):
    valor_rms = np.sqrt(np.mean(np.square(data)))
    return valor_rms

def mav(data):
    valor_mav = np.mean(np.abs(data))
    return valor_mav
```

### Extraccion de características en el dominio del tiempo usando DWT

```
for i in range(0, 120):
    sig1=A[i]
    sig2=B[i]
    sig3=C[i]
    cA,cD1,cD2,cD3,cD4=pywt.wavedec(sig1,'db4',level=4)
    cA1,cD11,cD12,cD13,cD14=pywt.wavedec(sig2,'db4',level=4)
    cA2,cD21,cD22,cD23,cD24=pywt.wavedec(sig3,'db4',level=4)
    #C3[i,:]=cA
    #C4[i,:]=cA1
    #Cz[i,:]=cA2
    C3D2[i,:]=cD2
    C3D3[i,:]=cD3
    C4D2[i,:]=cD12
    C4D3[i,:]=cD13
    CZD2[i,:]=cD22
    CZD3[i,:]=cD23
```

```

#Valor RMS
RMS1[i]=rms(cD2) #Ritmo Alfa canal C3
RMS2[i]=rms(cD3)
RMS3[i]=rms(cD12) #Ritmo Alfa canal C4
RMS4[i]=rms(cD13)
RMS5[i]=rms(cD22) #Ritmo Alfa canal Cz
RMS6[i]=rms(cD23)
#Valor MAV
MAV1[i]=mav(cD2) #Ritmo Alfa canal C3
MAV2[i]=mav(cD3)
MAV3[i]=mav(cD12) #Ritmo Alfa canal C4
MAV4[i]=mav(cD13)
MAV5[i]=mav(cD22) #Ritmo Alfa canal Cz
MAV6[i]=mav(cD23)

```

### Vector de características

```

X=np.transpose(np.block([[RMS1],[RMS3],[RMS5]]))
np.shape(X)

```

### Graficas de los valores RMS ó MAV

```

plt.subplot(5,1,1),plt.plot(RMS1)
plt.subplot(5,1,3),plt.plot(RMS3)
plt.subplot(5,1,5),plt.plot(RMS5)

```

### Vector de clases 0->T0, 1->T1, 2->T2

```

num_features=np.ones(40)
target=np.concatenate((0*num_features, 1*num_features, 2*num_features), axis=None)

```

```

np.shape(target)

```

```

import tensorflow
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers.core import Dense

```

### Vector de entrenamiento y de test para el clasificador SVM

```

X_train,X_test,y_train,y_test=train_test_split(X,target)
X_train.shape
X_test.shape

```

### ENTRENAMIENTO DE LA SVM

SVC implementa el enfoque “uno contra uno” para la clasificación multiclase. Si `n_clase` es el número de clases, entonces se construyen  $n\_clase * (n\_clase - 1) / 2$  clasificadores y cada uno entrena datos de dos clases

```
from sklearn import metrics
from sklearn import svm
clf = svm.SVC(kernel='linear') # Linear Kernel
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))

y_predSVM = np.round(clf.predict(X_test))
from sklearn.metrics import precision_score
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_predSVM)

precisionSVM = precision_score(y_test, y_predSVM, average='micro')
precisionSVM

cmSVM = confusion_matrix(y_test, y_predSVM)
cm_normSVM = cmSVM.astype('float') / cmSVM.sum(axis=1)[:, np.newaxis]
df_cmSVM = pd.DataFrame(cm_normSVM, ['T0', 'T1', 'T2'], ['T0', 'T1', 'T2'])
)
plt.figure(figsize = (10,6))
conf = sns.heatmap(df_cmSVM, annot=True, square=True, annot_kws={"size": 12})
conf.set_xlabel('Prediccion')
conf.set_ylabel('Verdad')
```

### ENTRENAMIENTO DE LA RED NEURONAL

**K-Fold Cross Validation:** El método K-Fold Cross-Validation consiste en dividir los datos de forma aleatoria en `k` grupos de aproximadamente el mismo tamaño, `k-1` grupos se emplean para entrenar el modelo y uno de los grupos se emplea como validación. Este proceso se repite `k` veces utilizando un grupo distinto como validación en cada iteración. El proceso genera `k` estimaciones del error cuyo promedio se emplea como estimación final

```
from sklearn import metrics
from sklearn.model_selection import StratifiedKFold
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation
from sklearn.model_selection import KFold
```

```
# Cross-Validate
kf = KFold(5, shuffle=True, random_state=42) # Use for KFold classification
n
fold = 0
for train, test in kf.split(X):
    fold+=1
    print(f"Fold #{fold}")
    x_train = X[train]
    y_train = target[train]
    x_test = X[test]
    y_test = target[test]
    model = Sequential()
    model.add(Dense(30, input_dim=len(X[0]), activation='tanh'))#tanh
    model.add(Dense(20, activation='tanh'))
    model.add(Dense(10, activation='tanh'))
    model.add(Dense(1, activation='relu'))
    model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
    fit_history=model.fit(x_train,y_train,validation_data=(x_test,y_test),verbose=0,epochs=1000)
    pred = np.round(model.predict(x_test))
    score = metrics.accuracy_score(y_test, pred)
    print(f"Fold score (accuracy): {score}")

def plot_learning_curves(history):
    pd.DataFrame(history.history).plot(figsize=(10, 5))
    plt.grid(True)
    plt.gca().set_ylim(0, 1)
    plt.xlabel('epocas', size = 16)
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    plt.savefig('Figure_epochs.png')

import numpy as np
y_pred = np.round(model.predict(x_test))
confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='micro')
precision
cm = confusion_matrix(y_test, y_pred)
cm_norm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
df_cm = pd.DataFrame(cm_norm, ['T0', 'T1', 'T2'], ['T0', 'T1', 'T2'])
plt.figure(figsize = (10,6))
```

```
conf = sns.heatmap(df_cm, annot=True, square=True, annot_kws={"size": 12})
conf.set_xlabel('Prediccion')
conf.set_ylabel('Verdad')
```

### **Extracción de pesos y bias para implementación en sistema embebido #**

```
Recuperamos bias y pesos de la 1 capa oculta
weights_HL, biases_HL = model.layers[0].get_weights()
# Recuperamos bias y pesos de la 2 capa de salida
weights_HL2, biases_HL2 = model.layers[1].get_weights()
# Recuperamos bias y pesos de la 3 capa de salida
weights_HL3, biases_HL3 = model.layers[2].get_weights()
# Recuperamos bias y pesos de la capa de salida
weights_OL, biases_OL = model.layers[3].get_weights()
weights_OL, biases_OL
```

## Bibliografía

- Abdeltawab, A., & Ahmad, A. (2020). Classification of Motor Imagery EEG Signals Using Machine Learning. *2020 IEEE 10th International Conference on System Engineering and Technology (ICSET)*. pp. 196-201.
- AG, N. (16 de Abril de 2018). Package: NEUROSPEC AG. *DSI-7 BCI 2000 Research Neurosciences. Neurospec.com. Retrieved 16 April 2021, .* From <https://www.neurospec.com/Products/Details/1071/dsi-7-bci-2000-package>.
- Alberca, A. (2021). La librería Numpy | Aprende con Alf. Aprende con Alf. Retrieved 18 May 2021, from <https://aprendeconalf.es/docencia/python/manual/numpy/>.
- Alda, A., & Torreblanca, N. (2018). Colocación de electrodos EEG en un Layout Fijo vs. Variable. Bitbrain. Retrieved 2 May 2021, from <https://www.bitbrain.com/es/blog/colocacion-electrodos-eeeg>.
- Alomari, M. H. (2014). EEG Mouse: A Machine Learning-Based Brain Computer Interface. (IJACSA) International Journal of Advanced Computer Science and Applications 5.4. pp. 193-198.
- Amiri, S., Fazel-Rezai, R., & Asadpour, V. (2013). A review of hybrid brain-computer interface systems," *Advances in Human-Computer Interaction*. Vol. I. pp. 1-8.
- Ang, K., Chua, K., Phua, K., Wang, C., Chin, Z., Kuah, C., . . . Guan, C. (April de 2014). A randomized controlled trial of EEG-based motor imagery brain-computer interface robotic rehabilitation for stroke. *Clinical EEG and Neuroscience: bol.* 46, pp. 310-320.
- Arboleda, Carolina, García, Eliana, Posada, Alejandro, . . . Róbinson. (04 de Abril de 2009). Diseño y construcción de un prototipo de interfaz cerebro-computador para facilitar la comunicación de personas con discapacidad motora. pp. 105-115.
- Blank, A., French, J. A., Pehlivan, A. U., & O'Malley, M. K. (June de 2015). Current trends in robot-assisted upper-limb stroke rehabilitation: Promoting patient engagement in therapy," *Current Physical Medicine and Rehabilitation Reports*". Vol 2, pp. 184-195.

- Bonnet, L., Lotte, F., & Lécuyer., A. (June de 2013). Two brains, one game: Design and evaluation of a multiuser bci video game based on motor imagery. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(2). pp. 185–198.
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory (COLT '92)*. Association for Computing Machinery, New York. pp. 144-152.
- Briega, R. (2021). Machine Learning con Python – Sobreajuste de Relopezbriega.github.io. Retrieved 17 May 2021, from <https://relopezbriega.github.io/blog/2016/05/29/machine-learning-con-python-sobreajuste/>.
- Brownlee, J. (2021). Introduction to the Python Deep Learning Library TensorFlow. Retrieved 18 May 2021, from <https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow/>.
- Burrucco, D. (2015). Disposición de la matriz de confusión. InteractiveChaos. Retrieved 13 April 2021, from <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/disposicion-de-la-matriz-de-confusion.>].
- Chang, H. C., Deng, H. T., Lee, P. L., Wu, C. H., & Shyu, K. K. (Aug de 2010). Real-time control of an ssvap-actuated remote-controlled car. *In Proceedings of SICE Annual Conference 2010*. pp.1884-1887.
- Chávez Saenz, V., Torres Ramírez, D., Herrera Ogaz, J., & Hernández Rodríguez, A. (2016). Adquisición y análisis de señales electroencefalográficas utilizando el dispositivo Emotiv EPOC+. *Revista De Tecnología E Innovación, (Vol. 3)*, 107-118. Retrieved 2 May 2021, from [https://ecorfan.org/bolivia/researchjournals/Tecnologia\\_e\\_innovacion/vol3num7/Revista\\_de\\_Tecnologia\\_e\\_Innovaci%C3%B3n\\_V3\\_N7.pdf#page=116](https://ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol3num7/Revista_de_Tecnologia_e_Innovaci%C3%B3n_V3_N7.pdf#page=116).
- Clerc, M., R.Wolpaw, J., & Winter, E. (2013). Brain Computer Interfaces, Principles and Practise. *BioMedical Engineering OnLine*. 12. 10.1186/1475-925X-12-22.
- DANE. (Junio de 2020). Estadísticas vitales (eevv) nacimientos y defunciones 1 trimestre 2020 PR Techport. *Departamento Administrativo Nacional de Estadística*. Bogotá, Colombia.

- Deecke, L., Weinberg, H., & Brickett., P. (2009). Magnetic fields of the human brain accompanying voluntary movement: Bereitschaftsmagnetfeld." *Experimental Brain Research* 48.1 (1982):. pp. 144-148.
- Delgado, K., Ledesma, S., & Rostro, H. (2019). Análisis de electroencefalograma usando redes neuronales artificiales. *Acta Universitaria* 29, e1672. doi. <http://doi.org/10.15174.au.2019.1672>.
- E., T.-C., Valdés-Santiago, D., León-Mecías, Á., & Ramírez-Beltrán, J. (2018). Detección y Localización de Ruptura Súbita mediante Transformada Wavelet Discreta y Correlación Cruzada. *Revista Iberoamericana De Automática E Informática Industrial*. pp. 211.
- EcuRed. (Septiembre de 2019). Consultado el 22:34, mayo 16, en [https://www.ecured.cu/index.php?title=Matrices\\_de\\_confusi%C3%B3n&oldid=3547576](https://www.ecured.cu/index.php?title=Matrices_de_confusi%C3%B3n&oldid=3547576).].
- Fernández-Gómez, E., & Sánchez-Cabeza, A. (2018). Imaginería motora: revisión sistemática de su efectividad en la rehabilitación de la extremidad superior tras un ictus. *Rev Neurol*. pp. 137-146.
- Göhring, D., Latotzky, D., Wang, M., & Rojas, R. (2013). Semi-autonomous car control using brain computer interfaces. *Intelligent Autonomous Systems*. pp. 393–408.
- Gomez Figueroa, L. (2016). Análisis de Señales EEG para detección de eventos oculares, musculares y cognitivos (Master en Automática y Robótica). Universidad Politécnica de Madrid.
- Graimann, B., Allison, B. Z., & Pfurtscheller, G. (2010). Brain-computer interfaces: Revolutionizing human-computer interaction. . *Springer Science & Business Media*.
- Guo, L., Su, K., & Liang., D. (2017). Classification of motor imagery EEG signals based on wavelet transform and sample entropy. *IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 905-910.
- Haykin, S. S. (2009). *Neural networks and learning machines, volume 3*. Pearson Upper Saddle River, NJ, USA:.

- Hernández González, C. (2017). Clasificación de señales EEG bajo paradigma de imaginación motora usando descomposición wavelet y máquinas de soporte vectorial (Maestro en Ciencias en el Área de Electrónica). Instituto Nacional de Astrofísica, Óptica y Electrónica.
- Hernández González, C. (2017). Clasificación de señales EEG bajo paradigma de imaginación motora usando descomposición wavelet y máquinas de soporte vectorial (Maestro en Ciencias en el Área de Electrónica). Instituto Nacional de Astrofísica, Óptica y Ele.
- Homan, R. W., Herman, J., & Purdy, P. (1987). Cerebral location of international 10-20 system electrode placement. *Electroencephalography and clinical neurophysiology* 66.4 pp. 376-382.
- IC. Inc. (2021). InteractiveChaos Retrieved 18 May 2021, from <https://interactivechaos.com/es/python/function/sklearnmodelselectiontraintestsp>.
- Kandel, R. (October de 2012). Principles of Neural Science. Fifth Edition (Principles of Neural Science (Kandel)). McGraw-Hill Education / Medical.
- Krauledat, M., Grzeska, K., Sagebaum, M., Blankertz, B., Vidaurre, C., Müller, K.-R., & Schröder, M. (2009). Playing pinball with non-invasive bci. In *Advances in neural information processing systems*. pp. 1641–1648.
- Leeb, R., Friedman, D., Müller-Putz, G. R., Scherer, R., Slater, M., & Pfurtscheller, G. (2007). Self-paced (asynchronous) bci control of a wheelchair in virtual environments: a case study with a tetraplegic. *Computational intelligence and neuroscience*.
- Leeb, R., Tonin, L., Rohm, M., Desideri, L., Carlson, T., & Millán, J. d. (Jun de 2015). Towards independence: A bci telepresence robot for people with severe motor disabilities. *Proceedings of the IEEE*, 103(6) pp. 969–982.
- Ma, T., Li, H., Deng, L., Yang, H., Lv, X., Li, P., . . . Xu, P. (Feb de 2017). The hybrid BCI system for movement control by combining motor imagery and moving onset visual evoked potential,” *Journal of Neural Engineering*. Vol. 14, p. 026015.
- Marengo, S. (2021). Afhic.com. Retrieved from [http://www.afhic.com/wp-content/uploads/2020/04/491\\_AFHIC\\_Seleccion-AFHIC.pdf](http://www.afhic.com/wp-content/uploads/2020/04/491_AFHIC_Seleccion-AFHIC.pdf).

- Metrics and scoring. (2021). Quantifying the quality of predictions — scikit-learn 0.24.2 documentation. (2021). Retrieved 18 May 2021, from [https://scikit-learn.org/stable/modules/model\\_evaluation.html#model-evaluation](https://scikit-learn.org/stable/modules/model_evaluation.html#model-evaluation).
- Morales, E., & Escalante, H. (2015). Evaluación, validación y sobre-ajuste. Presentation, Mexico.
- Moreno, I., Boya, C., Serracín, J. R., Quintero, J., & Antelis, J. (2016). Soporte para personas con movilidad reducida, basado en bci. In Memorias de Congresos UTP, volume 1, pages 5–7.
- Muller-Putz, G., Schwarz, A., Pereira, J., & Ofner, P. (2016). From classic motor imagery to complex movement intention decoding,” in “Progress in Brain Research. PP.39-70.
- Muñoz Cardona, J. (2014). Clasificación De Patrones De Imaginación Motora En Una Interfaz Cerebro Computador De Bajo Costo Usando Software Libre (Maestría En Ingeniería Eléctrica). Universidad Tecnológica De Pereira.
- N. Padfield; J. Zabalza; H. Zhao; V. Masero; J. Ren. (Marzo de 2019). EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges. Vol. 19,. P1423.
- Nicolas-Alonso, & L. Gomez-Gil, J. (2012). Brain Computer Interfaces, a Review. Sensors, 12(2), 1211–1279. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/s120201211>.
- Patiño, M., Castro, W., Peluffo, D., & Medina, C. (2016). Análisis De Señales EEG Para Detección De Intenciones Motoras Aplicadas A Sistemas BCI.
- Pfurtscheller. (2004). Brain-computer interfaces: State of the art and future prospects," in Proceedings of the 12th European Signal Processing Conference: EUROSIPCO. pp. 509-510.
- Quiroga, N., & Calla, E. (2019). Sistema de control de mano robótica replicadora de movimientos para manipular sustancias químicas. Acta Nova. 9 (1), pp. 82-114, Tomado de [http://www.scielo.org.bo/scielo.php?script=sci\\_arttext&pid=S1683-07892019000100005&lng=es&tlng=es](http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892019000100005&lng=es&tlng=es).
- R. Castro, L., & M. Castro, S. (2021). Wavelets y sus Aplicaciones. Lecture, - Bahía Blanca - Argentina.

- R., S. S., & D., S. (2019). Classification of multiclass motor imagery EEG signal using sparsity approach. *Neurocomputing*. doi:10.1016/j.neucom.2019.08.037.
- Rahman, M. A., Khanam, F., Hossain, K., Khurshed, M., & Ahmad, M. (2019). Clasificación de señales de EEG de imágenes motoras de cuatro clases utilizando PCA, Wavelet y red neuronal de dos etapas" *Revista internacional de aplicaciones y ciencias informáticas avanzadas (IJACSA)*. Recuperado de: <http://dx.doi.org/10.14569/IJACSA.2019.0100562>.
- Riveras Sanabria, F. (2012). Análisis con wavelets de alteraciones electrocardiográficas en pacientes chagsticos cronicos (Magíster en Matemáticas Aplicadas). Universidad EAFIT.
- Rodrigo, J. (2021). Validación de modelos predictivos (machine learning): Cross-validation, OneLeaveOut, Bootstraping. *Cienciadedatos.net*. Retrieved 12 May 2021, from [https://www.cienciadedatos.net/documentos/30\\_cross-validation\\_oneleaveout\\_bootstrap.](https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap.)]
- Rodríguez, D. (2021). Visualización de datos en Python con Seaborn - Analytics Lane. Retrieved 18 May 2021, from <https://www.analyticslane.com/2018/07/20/visualizacion-de-datos-con-seaborn/>.
- Schalk, G., McFarland, D., T., H., N., B., & J.R., W. (2004). BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering* 51(6):1034-1043.
- Scipy. (2021). Computación científica de alto nivel — Scipy lecture notes. Retrieved 18 May 2021, from <https://claudiovz.github.io/scipy-lecture-notes-ES/intro/scipy.html#transformada-rapida-de-fourier-scipy-fftpack>.
- Shovon, T., Nazi, Z., Dash, S., & Hossain, M. (2019). Classification of Motor Imagery EEG Signals with multi-input Convolutional Neural Network by augmenting STFT. *5th International Conference on Advances in Electrical Engineering (ICAEE)*. pp. 398-403.
- Sklearn Metrics. (2021). scikit-learn 0.24.2 documentation. Retrieved 18 May 2021, from [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html?highlight=confusion\\_matrix#sklearn.metrics.confusion\\_matrix](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html?highlight=confusion_matrix#sklearn.metrics.confusion_matrix).

- Sleight, Jason, Pillai, P., & Mohan., S. (2009). Classification of executed and imagined motor movement EEG signals. Ann Arbor: University of Michigan: 1-10.] .
- Springer International Publishing. (2015). Brain-Computer Interfaces. 74th ed., pp. 3-97.
- Taspinar, A. (2018). A guide for using the Wavelet Transform in Machine Learning. . Retrieved 30 April 2021, from <https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in.machine-learning/>.
- Team, K. (2021). Keras documentation: About Keras. Retrieved 18 May 2021, from <https://keras.io/about/>.
- Vásquez López, J. (2014). Red neuronal feedforward como estimador de patrones de corrientes en el interior del puerto de manzanillo sujeto a la acción de tsunamis. Tomado de secretaría de comunicaciones y transportes instituto mexicano del transporte,.
- Villegas, A., Lugo, E., & Pacheco, J. R. (Agosto de 2008). Villegas, Hyxia Interfaz cerebro computador modular basada en la interpretación del electroencefalograma (EEG) mediante RNA para el control de dispositivos electrónicos . *Revista INGENIERÍA UC*. Carabobo, Venezuela: Vol. 15, núm. 2, Universidad de Carabobo.
- Virgilio-González, C. (2018). Classification of Motor Imagery EEG Signals with CSP Filtering Through Neural Networks Models. 10.1007/978-3-030-04491-6\_10.
- W, S., C, V., KR, M., & M., K. (2012). Stationary common spatial patterns for brain-computer interfacing. *J Neural Eng.* Apr;9(2):026013. doi: 10.1088/1741-2560/9/2/026013. Epub 2012 Feb 20. PMID: 22350439.
- Wasilewski, F. (2021). PyWavelets - Wavelet Transforms in Python — PyWavelets Documentation. Retrieved 18 May 2021, from <https://pywavelets.readthedocs.io/en/latest/>.
- Yahya, N., & Musa, H. (2019). Classification of Motor Functions from Electroencephalogram (EEG) Signals Based on an Integrated Method Comprised of Common Spatial Pattern and Wavelet Transform Framework. 19(22):4878.
- Yahya, N., Musa, H., Ong, Z., & Elamvazuthi, I. (2019). Clasificación de las funciones motoras a partir de señales de electroencefalograma (EEG) basada en un método integrado compuesto por un patrón espacial común y un marco de

transformación de ondas. *Sensores (Basilea, Suiza)*. 19, (22), 4878 Tomado de:  
<https://doi.org/10.3390/s19224878>.

Zhang, J., Wang, B., Zhang, C., Xiao, Y., & Wang, M. Y. (March de 2019). An EEG/EMG/EOG-based multimodal human-machine interface to real-time control of a soft robot hand," *Frontiers in Neurorobotics*. Vol 13.