



UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARACTERIZACIÓN Y CLASIFICACIÓN DE SEÑALES
ELÉCTRICAS CEREBRALES PARA APLICACIONES BCI
(INTERFAZ CEREBRO COMPUTADOR)**

**Tesis de Grado Previa a la Obtención del
Título de Ingeniero en Electrónica Y
Telecomunicaciones**

Autor:

Milton Andrés León Bustamante

Director de Tesis:

Ing. Luis Enrique Chuquimarca Jiménez, M. I.

Loja - Ecuador

2017

CERTIFICACIÓN


Ing. Luis Enrique Chuquimarca Jiménez, M.I.

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en “**CARACTERIZACIÓN Y CLASIFICACIÓN DE SEÑALES ELÉCTRICAS CEREBRALES PARA APLICACIONES BCI (INTERFAZ CEREBRO COMPUTADOR)**”, previo a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por el señor egresado: **Milton Andrés León Bustamante**, mismo que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 31 de agosto de 2017



Ing. Luis Enrique Chuquimarca Jiménez, M.I.

DIRECTOR DE TESIS

AUTORÍA

Yo, **MILTON ANDRÉS LEÓN BUSTAMANTE**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Firma: 

Cédula: 1105657645

Fecha: 2017-09-22

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **MILTON ANDRÉS LEÓN BUSTAMANTE**, declaro ser autor de la tesis titulada: **“CARACTERIZACIÓN Y CLASIFICACIÓN DE SEÑALES ELÉCTRICAS CEREBRALES PARA APLICACIONES BCI (INTERFAZ CEREBRO COMPUTADOR)”**, como requisito para optar al grado de: **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

- Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.
- La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veintidós días del mes de septiembre del dos mil diecisiete.

Firma: 

Autor: Milton Andrés León Bustamante

Cédula: 1105657645

Dirección: Loja (Cdla. Héroe del Cenepa).

Correo Electrónico: moltilion@live.com

Teléfono: 2-546068

Celular: 0985006957

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Luis Enrique Chuquimarca Jiménez, M.I.

Tribunal de Grado: Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

Ing. Andy Fabricio Vega León, Mg. Sc.

Ing. John Jossimar Tucker Yépez, Mg. Sc.

DEDICATORIA

Para Alida, Milton, César, Carlos y Juan Diego.

Los llevaré en el corazón siempre.

AGRADECIMIENTO

Quiero agradecer a la Universidad Nacional de Loja, a la Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables y a la Carrera de Ingeniería en Electrónica y Telecomunicaciones, a toda su planta docente, personal administrativo y trabajadores, por haberme permitido formar profesionalmente en sus prestigiosas aulas. Seguro de contar con todas las habilidades y conocimientos necesarios para abrirme paso dentro de mi área de laboral, espero retornar aunque sea un poco de valor a la sociedad.

Agradezco al Ing. Luis Chuquimarca por ser mi guía durante el desarrollo de este trabajo de fin de grado. Sus conocimientos, preparación y recomendaciones han sido fundamentales para que esta tesis llegue a buen puerto. Espero que varios estudiantes más puedan tener la fortuna de tenerlo como su director de tesis.

Extiendo mi gratitud al Ing. Diego Orellana, hombre de amplios conocimientos en su campo de especialización, por haberme introducido al fascinante mundo del procesamiento digital de señales y las interfaces cerebro-computador.

Mi eterna gratitud con el Ing. Andy Vega, que ha sido profesor, compañero y amigo durante todas las aventuras que hemos vivido en la universidad.

Tengo infinita gratitud hacia mi familia, a mis padres y hermanos, porque ellos me han dado tanto amor sin pedir nada a cambio. Así también agradezco a mis tíos y primos por siempre darme ánimos.

Jamás olvidare a mis amigos y compañeros de clase; con ellos hemos pasado por tanto y seguro lo seguiremos haciendo en el futuro.

“Cuando yo era pequeño mi madre me decía: Si te haces soldado llegarás a general, si te haces cura, llegarás a ser Papa. En cambio de todo eso decidí ser pintor y me convertí en Picasso”.

Pablo Ruiz Picasso

TABLA DE CONTENIDO

CERTIFICACIÓN	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
TABLA DE CONTENIDO	VIII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIII
1 TÍTULO	1
2 RESUMEN	2
2.1 RESUMEN	2
2.2 ABSTRACT	3
3 INTRODUCCIÓN	4
4 REVISIÓN DE LITERATURA	6
4.1 CAPÍTULO 1: CONCEPTOS BÁSICOS SOBRE LAS INTERFACES CEREBRO-COMPUTADOR (BCI)	6
4.1.1 Definición	6
4.1.2 Partes de una BCI	6
4.1.3 Técnicas de Adquisición de Señales Cerebrales para BCI	7
4.1.4 Tipos de BCIs	9
4.2 CAPÍTULO 2: PRINCIPALES SEÑALES CEREBRALES UTILIZADAS EN BCI	11
4.2.1 Ritmos Cerebrales	11
4.2.2 Señales Cerebrales más Usadas en Aplicaciones BCI	12
4.2.2.1 Potenciales Evocados Visuales	13
4.2.2.2 Potenciales Corticales Lentos	15

4.2.2.3 Potencial Evocado P300	16
4.2.2.4 Ritmos Sensoriomotores.....	18
4.3 CAPÍTULO 3: TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES USADAS EN BCIs	21
4.3.1 Pre-Procesamiento de la Señal	21
4.3.1.1 Filtros Digitales	21
4.3.2 Métodos de Extracción de Características.....	22
4.3.2.1 Common Spatial Patterns (CSP)	25
4.3.2.2 La Transformada Wavelet Continua	28
4.3.2.3 La Transformada Wavelet Discreta.....	30
4.3.2.4 Modelado Autorregresivo.....	33
4.3.3 Métodos de Clasificación	34
4.3.3.1 Redes Neuronales Artificiales	36
4.3.3.2 El Perceptrón	37
4.3.3.3 Separabilidad Lineal y el Problema XOR	39
4.3.3.4 El Perceptrón Multicapa	41
4.3.3.5 La Regla Delta Generalizada.....	44
5 MATERIALES Y MÉTODOS	48
5.1 DISEÑO E IMPLEMENTACIÓN DE UN DETECTOR DE ONDAS ALFA EN TIEMPO REAL.....	48
5.1.1 Adquisición de las Señales Cerebrales	48
5.1.2 Paradigma de Entrenamiento.....	49
5.1.3 Preparación de las Señales.....	52
5.1.4 Extracción de Características	54
5.1.5 Clasificación de Características.....	55
5.1.6 Implementación del Detector de Ondas Alfa	56
5.2 EXTRACCIÓN DE CARACTERÍSTICAS Y CLASIFICACIÓN DE RITMOS SENSORIOMOTORES	62
5.2.1 Señales Cerebrales a Analizar	63
5.2.1.1 Base de Datos IIIA de la Competencia BCI III.....	63
5.2.1.2 Base de Datos 2A de la Competencia BCI IV.....	64
5.2.2 Preparación de las Señales.....	67
5.2.3 Extracción de Características	70

5.2.4 Clasificación de Características.....	74
6 RESULTADOS	75
6.1 RESULTADOS DE LA CLASIFICACIÓN DE ONDAS ALFA.....	75
6.2 RESULTADOS DE LA CLASIFICACIÓN DE RITMOS SENSORIOMOTORES	76
6.2.1 Resultados del Método 1	76
6.2.2 Resultados del Método 2	78
6.2.3 Resultados del Método 3	79
7 DISCUSIÓN	81
8 CONCLUSIONES	83
9 RECOMENDACIONES.....	84
10 BIBLIOGRAFÍA	85
11 ANEXOS	88
11.1 GLOSARIO DE ACRÓNIMOS	88
11.2 CÓDIGO DE MATLAB PARA LA NORMALIZACIÓN DE LAS SEÑALES CEREBRALES	90
11.3 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS DE LOS RITMOS ALFA	91
11.4 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS SEGÚN EL MÉTODO 1 (PARÁMETROS WAVELET).....	93
11.5 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS SEGÚN EL MÉTODO 2 (WAVELET + AR).....	95
11.6 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS SEGÚN EL MÉTODO 3 (CSP + WAVELET + AR + VARIANZA)	98
11.7 CÓDIGO DE LA INTERFAZ GRÁFICA DE USUARIO (GUI) DEL DETECTOR DE ONDAS ALFA	101
11.8 CÓDIGO DE ARDUINO PARA EL SISTEMA DE ACTUADORES DEL DETECTOR DE ONDAS ALFA	107
11.9 FOTOGRAFÍAS DEL DESARROLLO DE LAS SESIONES DE ENTRENAMIENTO DE RITMOS ALFA	109

ÍNDICE DE FIGURAS

Figura 1.....	8
Figura 2.....	15
Figura 3.....	16
Figura 4.....	17
Figura 5.....	18
Figura 6.....	19
Figura 7.....	22
Figura 8.....	26
Figura 9.....	30
Figura 10.....	32
Figura 11.....	33
Figura 12.....	35
Figura 13.....	38
Figura 14.....	39
Figura 15.....	41
Figura 16.....	43
Figura 17.....	48
Figura 18.....	49
Figura 19.....	50
Figura 20.....	51
Figura 21.....	51
Figura 22.....	52
Figura 23.....	53
Figura 24.....	56
Figura 25.....	57
Figura 26.....	59
Figura 27.....	60
Figura 28.....	61
Figura 29.....	62
Figura 30.....	63
Figura 31.....	64
Figura 32.....	65
Figura 33.....	66
Figura 34.....	67
Figura 35.....	68

Figura 36.....	69
Figura 37.....	71
Figura 38.....	72
Figura 39.....	73
Figura 40.....	77
Figura 41.....	79
Figura 42.....	80
Figura 43.....	109
Figura 44.....	109

ÍNDICE DE TABLAS

Tabla 1..10

Tabla 2..13

Tabla 3..23

Tabla 4..35

Tabla 5..40

Tabla 6..54

Tabla 7..75

Tabla 8.76

Tabla 9..77

Tabla 10..78

Tabla 11..78

Tabla 12..79

Tabla 13..80

1 TÍTULO

CARACTERIZACIÓN Y CLASIFICACIÓN DE SEÑALES ELÉCTRICAS CEREBRALES PARA APLICACIONES BCI (INTERFAZ CEREBRO COMPUTADOR)

2 RESUMEN

2.1 RESUMEN

En el presente trabajo de fin de grado se exploraron varias técnicas de extracción de características y clasificación de patrones para el procesamiento de ondas alfa y ritmos sensoriomotores, con la finalidad de evaluar la posibilidad de aplicar estos métodos en sistemas BCI. En ambos casos las señales cerebrales fueron adquiridas mediante EEG.

El trabajo inició con una revisión del estado del arte sobre la tecnología de las BCI, para determinar cuáles son las técnicas más recomendadas para el análisis de las señales cerebrales escogidas.

Para la identificación de los ritmos alfa se emplearon parámetros estadísticos calculados a partir de la transformada wavelet como características para la clasificación y una red neuronal como clasificador; esto dio como resultado altos niveles en la detección de ritmos alfa. Las señales de prueba usadas se obtuvieron de una mujer adulta sana que dio su consentimiento previo para participar del experimento.

Con las técnicas empleadas anteriormente se desarrolló una BCI en tiempo real simple, que tiene por objetivo activar un conjunto de indicadores cuando detecta que el usuario ha sido capaz de generar ritmos alfa.

El análisis de los ritmos sensoriomotores se basó en la discriminación entre imaginación motora de mano izquierda y derecha, para lo cual se evaluaron tres métodos diferentes de extracción de características. Cada uno de estos métodos presentó diferentes niveles de precisión en la discriminación de cada clase, pero se obtuvo un método con un nivel de aciertos adecuado para su posible aplicación en BCI. Las características calculadas para cada método incluyen aquellas utilizadas en la detección de ritmos alfa, coeficientes AR y filtrado espacial mediante Common Spatial Patterns. La clasificación se realizó mediante una red neuronal.

Las señales usadas para el procesamiento de los ritmos sensoriomotores se obtuvieron a partir de las bases de datos IIIA y 2A, pertenecientes a las competencias BCI III y IV respectivamente. En total se analizaron las señales adquiridas de 12 individuos diferentes.

2.2 ABSTRACT

In this end-degree project, different feature extraction and pattern classification techniques were explored for the processing of alpha waves and sensorimotor rhythms in order to evaluate the possibility of using these methods in BCI applications. The brain signals used were acquired through EEG in both cases.

The work started with a state of the art review of BCI technology for identifying the most recommended techniques, used in the analysis of the chosen signals.

The features used for alpha rhythms identification were statistic parameters computed from the signal's wavelet transform and a neural network was used as classifier. This method achieved high levels of accuracy when detecting alpha activity. The test waves used were from a healthy adult woman who gave her consent prior to the study.

With these techniques a simple real-time BCI was developed. The goal of the system is to activate a serie of cues when the subject is capable of generating alpha waves.

The sensorimotor rhythms analysis was based in the discrimination between left hand motor imagery and right hand motor imagery. Three feature extraction methods were evaluated for this purpose. Each of these methods had diferent results but finally one was found with enough accuracy for its possible BCI implementation. The features extracted of each method included those used for the alpha waves detector, AR coefficients and spatial filtering based in Common Spatial Patterns. A neural network was used for classification.

The signals used for the sensorimotor rhythms processing were taken from the IIIA and 2A datasets of the BCI competitions III and IV, respectively. The signals of 12 test subjects were analized in total.

3 INTRODUCCIÓN

Una interfaz cerebro-computador (*brain computer interface* o BCI en inglés) es un mecanismo de interacción humano-máquina que permite traducir la actividad cerebral de una persona en acciones de control sobre máquinas y programas de computadora (Grimm, Allison, & Pfurtscheller, 2010). Las acciones realizadas a través de una BCI requieren un control consciente por parte del usuario, es decir que las BCI no generan acciones en base a la actividad inconsciente de una persona.

El funcionamiento de las BCI se basa en la medición de la actividad cerebral, para después procesarla y así determinar el comando que desea ejecutar el usuario (Grimm, Allison, & Pfurtscheller, 2010). Durante el procesamiento, se extraen ciertos parámetros que caracterizan a los patrones cerebrales usados en la BCI para luego traducirlos en las señales de control adecuadas. Por esta razón, para asegurar el correcto funcionamiento de una BCI es necesario la aplicación de técnicas que permitan determinar las intenciones del usuario con un alto grado de exactitud; si la etapa de procesamiento no puede distinguir los patrones generados por el usuario, no sería viable la implementación de una BCI.

Debido a que las interfaces cerebro-computador proveen un canal de comunicación directa entre el cerebro y exterior, estas se constituyen en medios de interacción con funcionalidades que sobrepasan aquellas de las interfaces humano-máquina que existen en la actualidad, por lo que pueden ser aprovechadas en importantes campos de aplicación tales como la restauración de la función motora de pacientes con severos impedimentos físicos. Tal es el caso de las personas que sufren esclerosis lateral amiotrófica (ALS), en donde el movimiento voluntario se va perdiendo de manera progresiva hasta desaparecer, sin embargo las funciones neuronales de estas personas se mantienen intactas, a pesar de que son incapaces de mover los músculos (Rao, 2013). A través de las BCI se podrían desarrollar sistemas para el manejo de delectadores con la mente con fines de comunicación para estas personas, o sillas de ruedas autónomas para transporte.

En la actualidad existen una gran diversidad de métodos para la extracción y clasificación de características de las señales cerebrales, cada una de las cuales presenta sus propias propiedades. La selección de los métodos que permitan obtener los mejores

resultados de clasificación en una aplicación BCI específica es uno de los principales problemas que se debe resolver al desarrollar uno de estos sistemas.

4 REVISIÓN DE LITERATURA

4.1 CAPÍTULO 1: CONCEPTOS BÁSICOS SOBRE LAS INTERFACES CEREBRO-COMPUTADOR (BCI)

4.1.1 Definición

Una interfaz cerebro-computador es un sistema de interacción humano-máquina capaz de traducir la actividad eléctrica cerebral de una persona en una interacción sobre un sistema físico o un programa de computadora.

El objetivo de una BCI no es el de generar una acción espiando la actividad cerebral de las personas, sino la de proveer al cerebro de un nuevo canal de salida que requiere de un control voluntario por parte del usuario (Vallabhaneni, Wang, & He, 2005). Esto quiere decir que las BCI generan una salida en base a la actividad neuronal consiente del usuario; no generan acciones de control en base a la actividad inconsciente de las personas.

4.1.2 Partes de una BCI

Un sistema BCI se puede dividir en seis etapas de funcionamiento. Estas etapas son (Henríquez, 2014):

- **Adquisición de la señal:** En esta etapa se registra la señal generada por el cerebro que dará lugar a la acción de control. Aquí la señal también es amplificada, filtrada y digitalizada para que puede ser utilizada por las siguientes etapas del sistema.
- **Pre-procesamiento de la señal:** En donde la señal se pasa a través de una serie de filtros digitales para eliminar ruido que pueda afectarla y prepararla para su procesamiento.
- **Etapa de extracción de características:** Aquí la señal es procesada para extraer de ella características permitirán diferenciarla entre distintas clases de ritmos cerebrales e identificar la acción que el usuario desea realizar.
- **Etapa de clasificación de características:** En donde las características extraídas en la etapa anterior son procesadas mediante algoritmos de clasificación para determinar a qué clase o tipo de actividad cerebral pertenecen. Generalmente las señales se clasifican en dos tipos de clases:

deseadas y no deseadas. Sin embargo, en algunas aplicaciones puede ser útil tener más de dos clases (Sarasa, 2015).

- **Control:** La etapa de control traduce las señales obtenidas a partir de la etapa de clasificación en una salida de control sobre un sistema físico o de software, dependiendo si la señal cerebral obtenida es una señal deseada o no deseada.

En la literatura también se puede encontrar las etapas de pre-procesamiento, extracción de características y clasificación en una única etapa conocida como procesamiento (Henríquez, 2014).

4.1.3 Técnicas de Adquisición de Señales Cerebrales para BCI

Las BCI pueden utilizar técnicas tanto de tipo invasivo como no invasivo para la adquisición de las señales cerebrales. A continuación se explica en qué consisten estos métodos (Henríquez, 2014).

- **BCI invasivas:** Son aquellas en las que se implantan electrodos directamente en la corteza cerebral del usuario, mediante una intervención quirúrgica, para adquirir las señales eléctricas generadas por neuronas individuales o grupos de estas. Estos sistemas tienen mayor precisión que las BCI no invasivas. La resolución temporal y espacial de estos es cerca de 100 veces mejor que la de los sistemas no invasivos (Sarasa, 2015). Sin embargo, las BCI invasivas presentan más peligros y riesgos hacia la salud de los usuarios. Un ejemplo de método invasivo es la electrocorticografía (ECoG).
- **BCI no invasivas:** En los métodos de adquisición no invasivos la actividad cerebral es registrada a través de diferentes tipos de señales que pueden ser eléctricas, magnéticas y metabólicas mediante sensores que no se implantan dentro del cuerpo humano. A pesar de que estas técnicas poseen una resolución y relación señal ruido menor a la de los métodos invasivos, muchas veces son suficientes para la mayoría de las aplicaciones. Algunos de los métodos de extracción no invasiva son:
 - Tomografía computarizada (CT).
 - Tomografía por emisión de fotón único (SPECT).
 - Imagen de resonancia magnética (MRI).

- Imagen por resonancia magnética funcional (fMRI).
- Magnetoencefalografía (MEG).
- Electroencefalografía (EEG).

De los métodos de adquisición invasivos y no invasivos que se ha mencionado, se expondrá las características de algunos de los más relevantes (Sarasa, 2015).

- **Electroencefalografía (EEG):** La excitación sináptica de las dendritas de las neuronas generan un flujo de corrientes eléctricas que pueden ser capturadas mediante el uso de electrodos colocados en el cuero cabelludo. Este proceso de adquisición se conoce como encefalografía. La señal registrada se mide como una diferencia de potencial entre la señal continua y un electrodo utilizado como punto de referencia (Sarasa, 2015), y su amplitud se encuentra dentro del rango de los microvoltios (μV). Los electrodos se colocan en el cráneo siguiendo el Sistema Internacional 10/20, de tal forma que se asegure el cubrimiento de toda el área del cráneo (Henríquez, 2014) (Sarasa, 2015). La figura 1 muestra la ubicación de los electrodos en el cuero cabelludo según este sistema.

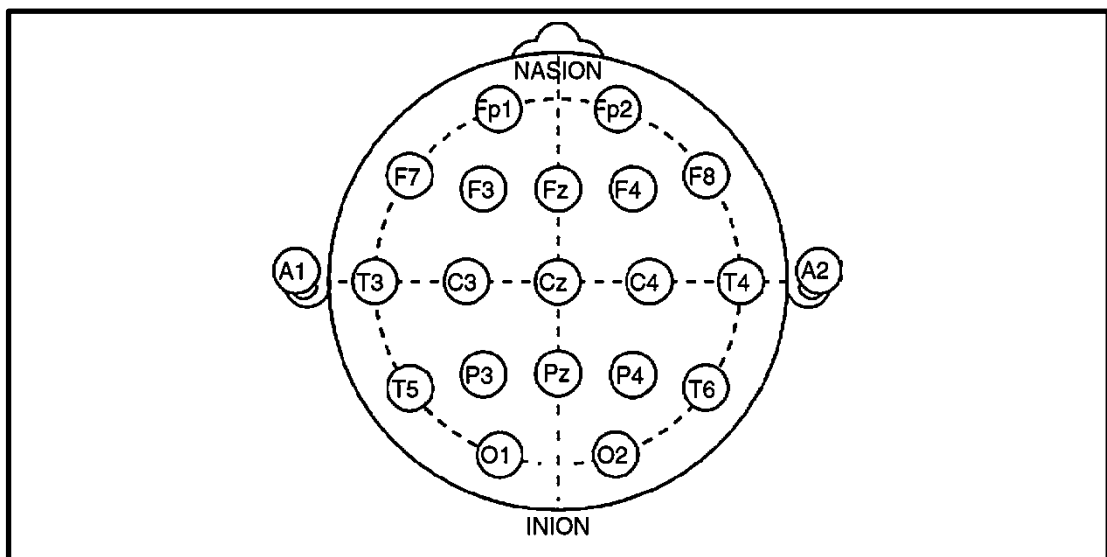


Figura 1. Ubicación de los electrodos según el Sistema Internacional 10/20 para EEG. Nomenclatura: C = central, P = parietal, T = temporal, F = frontal, Fp = frontal polar, O = occipital, A = mastoides.

Fuente: (Rao, 2013) .

- **Magnetoencefalografía (MEG):** En la magnetoencefalografía también se registra la actividad sináptica generada por el cerebro, pero a diferencia del EEG, la MEG registra los campos magnéticos dinámicos generados por la

actividad eléctrica cerebral. Las señales obtenidas a través de esta técnica son mucho más claras y menos afectadas por el ruido en comparación a las señales obtenidas a través de EEG, sin embargo necesitan de instrumentación mucho más sofisticada pues utilizan dispositivos superconductores de interferencia cuántica (SQUID) para registrar los campos magnéticos (Sarasa, 2015).

- **Electrocorticografía (ECoG):** En la electrocorticografía se utilizan electrodos subdurales implantados en la corteza cerebral del usuario para registrar la actividad eléctrica cerebral. Este proceso resulta en una señal con una mejor relación señal-ruido que las obtenidas a través de EEG y provee de una mejor resolución espacial como temporal. Sin embargo, esto viene asociado a los problemas de salud relativos al proceso de colocación de los electrodos, pues se necesita de una craneotomía para su implante, además la vida útil de los electrodos es limitada (Henríquez, 2014).

No se espera que las técnicas de adquisición basadas en ECoG, MEG y fMRI sean utilizadas para el desarrollo de BCI orientadas al uso cotidiano de personas debido a los riesgos que pueden llevar (ECoG) y al hardware altamente especializado que necesitan para su funcionamiento (Sarasa, 2015) (Nicolas-Alonso & Gomez-Gil, 2012). A pesar de la pobre SNR (Relación Señal-Ruido) de las señales EEG, esta técnica posee resoluciones temporal y espacial razonables, y utiliza hardware menos especializado y más barato; es por esta razón que para el presente trabajo de tesis se ha optado por el uso de una BCI basada en la adquisición de señales EEG.

4.1.4 Tipos de BCIs

Los sistemas BCI se pueden clasificar en base a la naturaleza de las señales de control que utilicen para su funcionamiento en dos tipos: sistemas BCI exógenos y sistemas BCI endógenos (Nicolas-Alonso & Gomez-Gil, 2012).

Los sistemas BCI exógenos utilizan señales neuronales que son generadas por medio de estímulos externos, tales como potenciales P300 o VEP. Los usuarios no necesitan entrenamiento para el control de estos sistemas puesto que las señales de control son moduladas en base a los estímulos externos. Estos sistemas también se

caracterizan por ofrecer velocidades de transmisión de información muy altas de hasta 60 bits/min (Nicolas-Alonso & Gomez-Gil, 2012).

Por otro lado, los sistemas BCI endógenos utilizan señales neuronales auto-reguladas por el usuario, tales como ritmos sensoriomotores o SCPs, como señales de control. Estos sistemas necesitan que el usuario pase por un proceso de enteramiento para que pueda aprender a modular de forma adecuada su actividad neuronal, de tal forma que pueda ser decodificada por el sistema BCI de forma exitosa (Nicolas-Alonso & Gomez-Gil, 2012). En la tabla 1 se enumeran las ventajas y desventajas que presentan los BCI endógenos y exógenos.

Tabla 1. Características de los sistemas BCI exógenos y endógenos.

Fuente (Nicolas-Alonso & Gomez-Gil, 2012).

TIPO DE SISTEMA BCI	SEÑALES CEREBRALES	VENTAJAS	DESVENTAJAS
Exógeno	<ul style="list-style-type: none"> • SSVEP • P300 	<ul style="list-style-type: none"> • Entrenamiento mínimo. • Configuración de las señales de control de forma rápida y fácil. • ITR alto (60 bits/min). • Pueden funcionar solamente con un canal EEG. 	<ul style="list-style-type: none"> • Atención permanente a estímulos externos. • Puede causar cansancio en algunos usuarios.
Endógeno	<ul style="list-style-type: none"> • Ritmos sensoriomotores (μ y β). • SCPs. 	<ul style="list-style-type: none"> • Funcionamiento independiente de cualquier estímulo externo. • Puede ser operado a libre voluntad del usuario. • Útil en usuarios con órganos sensoriales afectados. • Adecuado para aplicaciones para el control de cursores. 	<ul style="list-style-type: none"> • Entrenamiento durante largos periodos de tiempo. • No todos los usuarios pueden obtener un control adecuado. • Se necesitan varios canales de adquisición BCI para un desempeño adecuado. • ITR bajo (20 – 30 bits/min).

4.2 CAPÍTULO 2: PRINCIPALES SEÑALES CEREBRALES UTILIZADAS EN BCI

4.2.1 Ritmos Cerebrales

Gracias al monitoreo de la actividad eléctrica cerebral por medio de EEG se ha podido identificar varios tipos de ondas o ritmos cerebrales, cada uno de los cuales está vinculado a un estado de conciencia o actividad mental en específico. Estos ritmos se clasifican según la banda de frecuencia en la que operan, siendo estas la banda delta (δ), theta (θ), alfa (α), beta (β) y gamma (γ).

Las ondas delta (δ) aparecen por debajo de los 4 Hz (Nicolas-Alonso & Gomez-Gil, 2012). Aparecen en los niños de 1 año de edad y va descendiendo conforme el niño va creciendo. También se la puede encontrar en adultos en estado de sueño profundo, sin embargo su presencia exagerada en adultos despiertos se relaciona con enfermedades neurológicas. Estas ondas no son útiles para su aplicación en BCI (Henríquez, 2014).

Las ondas theta (θ) se encuentran en el rango de 4 a 7 Hz (Nicolas-Alonso & Gomez-Gil, 2012). Se las puede encontrar en niños jóvenes, niños viejos y en adultos en estados de meditación, somnolencia y sueño, sin embargo su presencia exagerada en adultos despiertos se considera anormal y se la vincula con enfermedades neurológicas, al igual que las ondas delta. Los ritmos theta aparecen en estados de concentración meditativa, y también se los ha relacionado con una amplia variedad de actividades cognitivas como cálculos mentales o durante la atención consiente.

Las ondas alfa se presentan con gran intensidad en las regiones occipitales de la cabeza y operan en el rango de 8 a 12 Hz (Nicolas-Alonso & Gomez-Gil, 2012). La amplitud de estas ondas aumenta cuando se cierra los ojos y en estados de relación, pero se suprime cuando se abren los ojos y existe esfuerzo mental. Estas ondas se relacionan con el procesamiento de información visual y la realización de esfuerzos mentales. Cuando se realiza un esfuerzo mental, la amplitud de estos ritmos se atenúa, por lo que se ha propuesto su uso como indicadores para medir el nivel de esfuerzo mental en un individuo (Nicolas-Alonso & Gomez-Gil, 2012). Los ritmos sensoriomotores MU (μ) aparecen en el mismo rango que la banda

alfa, sin embargo el origen de estas oscilaciones se vincula a estados mentales diferentes.

Las ondas beta (β) se definen en el rango que va de 12 a 30 Hz, y a menudo se las subdivide también en β_1 que va de 13,5 a 17 Hz, β_2 que va de 17,5 a 21 Hz y β_3 que va de 21,5 a 30 Hz (Henríquez, 2014) (Nicolas-Alonso & Gomez-Gil, 2012). Estos ritmos aparecen en las regiones frontales y centrales de la cabeza y se relacionan con actividades motoras. La amplitud de estas se atenúa en presencia de movimiento real o con imaginaria motora.

Las ondas gamma (γ) aparecen en la banda de 30 a 100 Hz (Nicolas-Alonso & Gomez-Gil, 2012). Han sido relacionadas con ciertas actividades motoras y percepciones. Algunos estudios han mostrado su relación con actividades motoras durante contracción muscular máxima, mientras que otros también han mostrado su vínculo durante la percepción de estímulos visuales y auditivos. Estas ondas no son usadas generalmente en BCI debido a que son muy sensibles a la presencia de artefactos.

4.2.2 Señales Cerebrales más Usadas en Aplicaciones BCI

Existen diferentes tipos de señales cerebrales que pueden ser usadas para el control de aplicaciones BCI. Cada señal de control varía en su naturaleza y representan un tipo de patrón en específico que tiene que ser clasificado.

Un problema en el desarrollo de las BCI es que las señales cerebrales son generadas por un conjunto de diferentes tareas cognitivas, por lo que identificar una única señal relacionada con una tarea cognitiva específica no es un trabajo simple. Sin embargo, en la actualidad se han estudiado varios fenómenos fisiológicos relacionados con ciertas señales cerebrales, a tal grado que las personas podrían modularlas a voluntad para su utilización en sistemas BCI (Nicolas-Alonso & Gomez-Gil, 2012). La tabla 2 muestra las principales señales cerebrales que se utilizan para el control de sistemas BCI modernos.

Tabla 2. Características de las señales de control más utilizadas en aplicaciones BCI.

Adaptada de: (Nicolas-Alonso & Gomez-Gil, 2012).

SEÑAL	FENÓMENO FISIOLÓGICO	NÚMERO DE ELECCIONES	NECESIDAD DE ENTRENAMIENTO	TAZA DE TRANSFERENCIA DE INFORMACIÓN (ITR)
Potenciales Visuales Evocados (VEP)	Modulaciones de señales cerebrales en la corteza visual.	Alto	No	60-100 bits/min
Potenciales Corticales Lentos (SCP)	Cambios de voltaje lentos en las señales cerebrales.	Bajo (2 o 4)	Si	5-12 bits/min
Potenciales Evocados P300	Picos de voltaje positivos debidos a estímulos poco frecuentes.	Alto	No	20-25 bits/min
Ritmos Sensoriomotores	Modulación de señales cerebrales sincronizadas con actividades motoras.	Bajo (2, 3, 4, 5)	Si	3-35 bits/min

4.2.2.1 Potenciales Evocados Visuales

Los Potenciales Evocados Visuales (*Visually Evoked Potentials* o *VEP* en inglés) son potenciales que aparecen en la corteza visual después de la recepción de un estímulo visual. Los VEPs se pueden clasificar según tres criterios: en base a la morfología del estímulo visual, según la frecuencia de la estimulación visual y por estimulación de campo. (Nicolas-Alonso & Gomez-Gil, 2012).

En base al primer tipo de clasificación, los VEPs pueden ser causados por estimulación flash o a través de patrones gráficos. Según el segundo criterio

las VEPs también se pueden subclasificar en VEPs transitorias (*Transient VEP* o *TVEP* en inglés) y VEPs de estado estable (*Steady State VEP* o *SSVEP* en inglés). Finalmente, de acuerdo al criterio de estimulación de campo, las VEPs se pueden clasificar en VEPs de campo entero, VEPs de medio campo VEPs de parte de campo.

Los SSVEPs son causados por la presentación repetida de un estímulo visual a una frecuencia mayor a 6 Hz y se manifiestan como señales sinusoidales que oscilan a una frecuencia fundamental igual a la frecuencia de presentación del estímulo.

Las BCIs basadas en SSVEPs generalmente muestran estímulos visuales parpadeantes, como teclas con números o letras, de tal manera que se induzca una VEP en el usuario al observar una de estas teclas. El usuario puede seleccionar una acción determinada observando cualquiera de las opciones disponibles. La figura 2 muestra el diseño de una BCI basada en SSVEPs utilizada para el marcado de números telefónicos. Cada número se muestra como una señal que parpadea a una frecuencia determinada. Los usuarios seleccionan el número que desean marcar observando cada uno de los números, lo que genera una SSVEP con una componente armónica a la misma frecuencia que el símbolo seleccionado. El sistema calcula la densidad de potencia de cada una de las frecuencias de las teclas y clasifica aquella que con más seguridad fue elegida (Henríquez, 2014).

Las BCI basadas en SSVEP tienen una gran variedad de ventajas lo que las hace de un interés especial para la investigación. De forma específica, son relativamente fáciles de detectar puesto que el espectro de potencia de las SSVEPs se concentra en una frecuencia igual al de la frecuencia de repetición del estímulo observado. Además tienen una buena SNR (Henríquez, 2014). Otra ventaja destacable es que por ser generadas a través de estímulos externos, los usuarios requieren poco entrenamiento para aprender a usar BCIs que utilicen este paradigma.

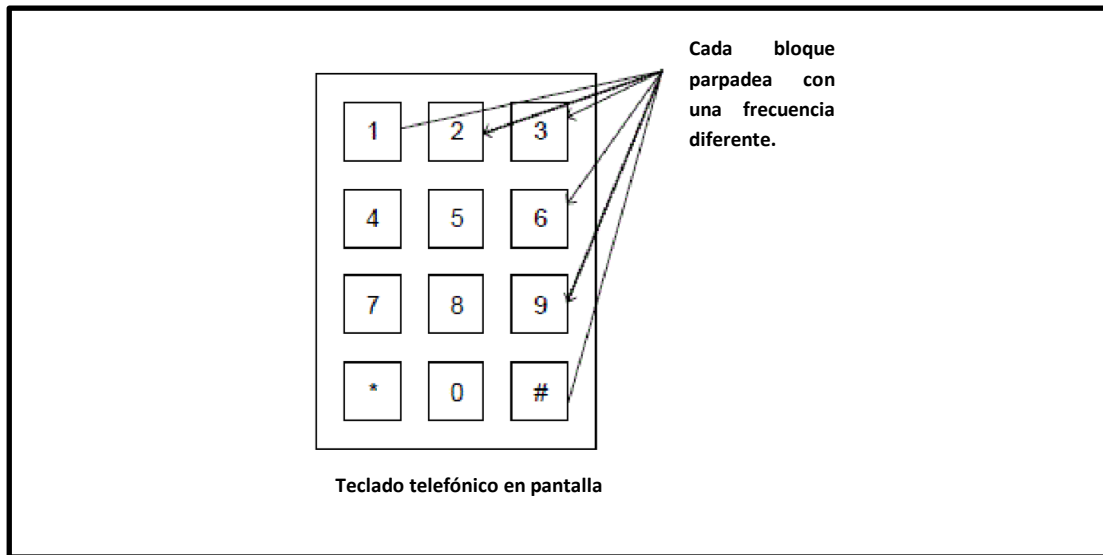


Figura 2. Estimulo visual utilizado en una BCI basada en SSVEPs para el marcado de números telefónicos.

Adaptado de: (Henríquez, 2014).

4.2.2.2 Potenciales Corticales Lentos

Los Potenciales Corticales Lentos (*Slow Cortical Potentials* o SCP en inglés) son cambios lentos de voltaje en un EEG que pueden ser apreciados en intervalos que van desde un segundo hasta varios segundos. Las componentes espectrales de estas señales se encuentran dentro de la banda de frecuencia de 0.1 Hz hasta 1 Hz (Henríquez, 2014). Los SCP son generados por cambios en el nivel de actividad cortical de una persona y se clasifican en dos tipos: SCP negativos y SCP positivos. Los SCP negativos son generados por el incremento de la actividad neuronal, mientras que los SCP positivos se relacionan con el decremento de actividad neuronal (Nicolas-Alonso & Gomez-Gil, 2012). En la figura 3 se muestran los dos tipos de SCP.

Se ha demostrado que las personas son capaces de modular a voluntad los SCP mediante entrenamiento a través de dispositivos de traducción de pensamientos (*thought-translation devices* o TTD en inglés) (Nicolas-Alonso & Gomez-Gil, 2012). Los TTD son dispositivos de comunicación cerebro-

computador utilizados para que un usuario aprenda a regular de forma voluntaria sus SCP, mediante el uso de estímulos visuales y auditivos.

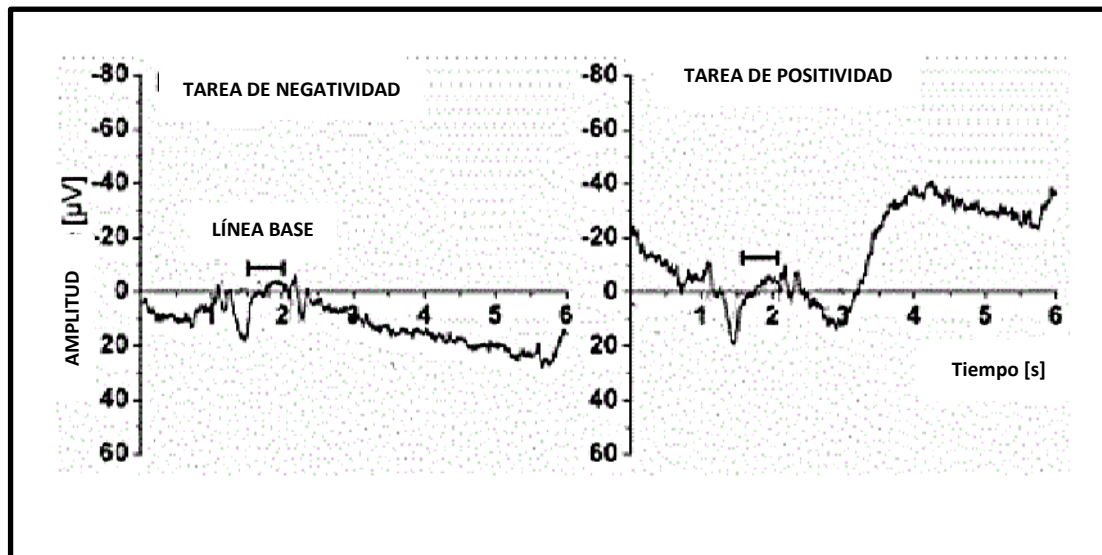


Figura 3. Señales EEG en donde se muestra SCP negativo y SCP positivo.

Adaptado de: (Henríquez, 2014).

Las BCI basadas en SCP necesitan que el usuario sea capaz de auto-modular de forma voluntaria sus SCP y se diseñan mediante un paradigma binario que consta de dos fases: una fase inicial de preparación y una fase activa. En la fase de preparación se mide un nivel de voltaje inicial durante un intervalo de 2 segundos y seguidamente, en la fase activa, el usuario realiza una tarea mental específica produciendo de forma voluntaria un aumento o disminución del nivel de voltaje medido durante un intervalo de 3 segundos (Henríquez, 2014).

La precisión en la clasificación de SCP auto-regulables ha logrado alcanzar porcentajes entre 70 y 80% (Nicolas-Alonso & Gomez-Gil, 2012). El punto débil de las SCP es que necesitan de largos periodos de entrenamiento para su funcionamiento que puede extenderse durante varios meses.

4.2.2.3 Potencial Evocado P300

El potencial evocado P300 es un pico de amplitud positiva que aparece en una EEG aproximadamente 300 ms después de que el usuario ha sido expuesto a un estímulo visual, auditivo o somatosensorial poco frecuente. La figura 3 muestra una señal P300 según se la define de forma habitual, con un incremento de voltaje 300 ms después de presentado un estímulo.

Sin embargo, es necesario mencionar que el potencial P300 no siempre se presenta tal y como es definido (Sarasa, 2015).

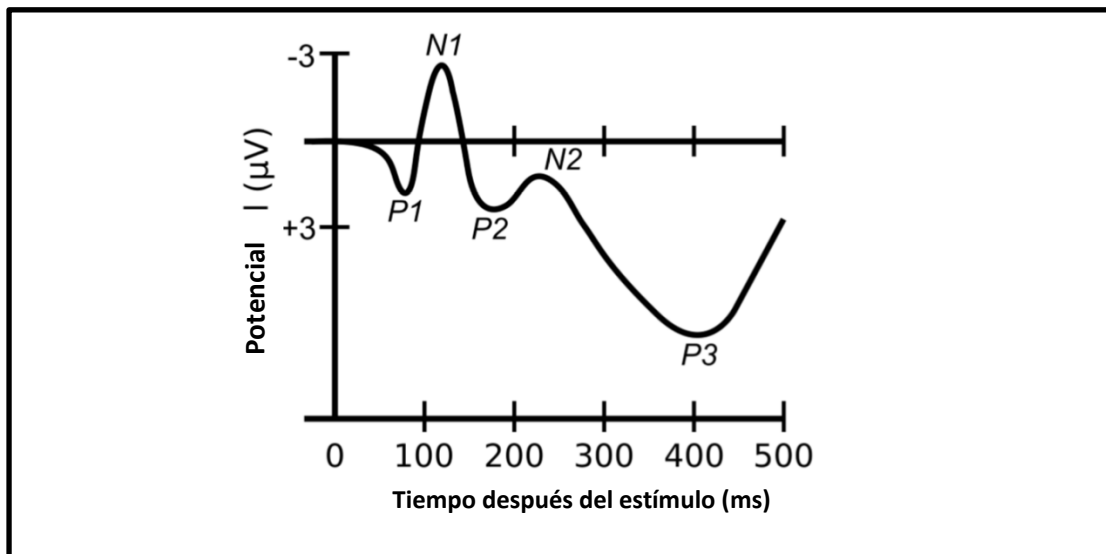


Figura 4. Señal EEG obtenida después de presentar al usuario un estímulo. A partir de 300 ms podemos observar un aumento en la amplitud en la onda.

Adaptado de: (Sarasa, 2015).

Implementaciones de sistemas BCI basadas en el potencial P300 utilizan, de forma habitual, una matriz que contiene números, letras o comandos (Nicolas-Alonso & Gomez-Gil, 2012). Se puede destacar el sistema BCI desarrollado por Donchin y Farwell en 1988, el cual es la primera BCI basada en este potencial (Henríquez, 2014). Este sistema consiste en una matriz de 6 filas y 6 columnas con números y letras con el objetivo de que el usuario sea capaz de escribir mensajes; en la figura 5 se presenta el diseño de la matriz utilizada.

Las filas y las columnas de la matriz se iluminan de forma aleatoria. Para que el usuario pueda seleccionar un símbolo se le pide que cuente el número de veces que éste se ilumina. Cada fila y columna se ilumina por lo menos una vez, por lo que la matriz en total se ilumina 12 veces. Esto que significa que cada letra se ilumina dos veces en total. Cuando se ilumina el símbolo objetivo el usuario debería mostrar una respuesta de amplitud considerable, por lo tanto, lo que se hace es promediar las respuestas que dieron cada uno

de los símbolos y aquel que presente el pico de mayor amplitud será la opción que el usuario buscaba seleccionar (Henríquez, 2014).

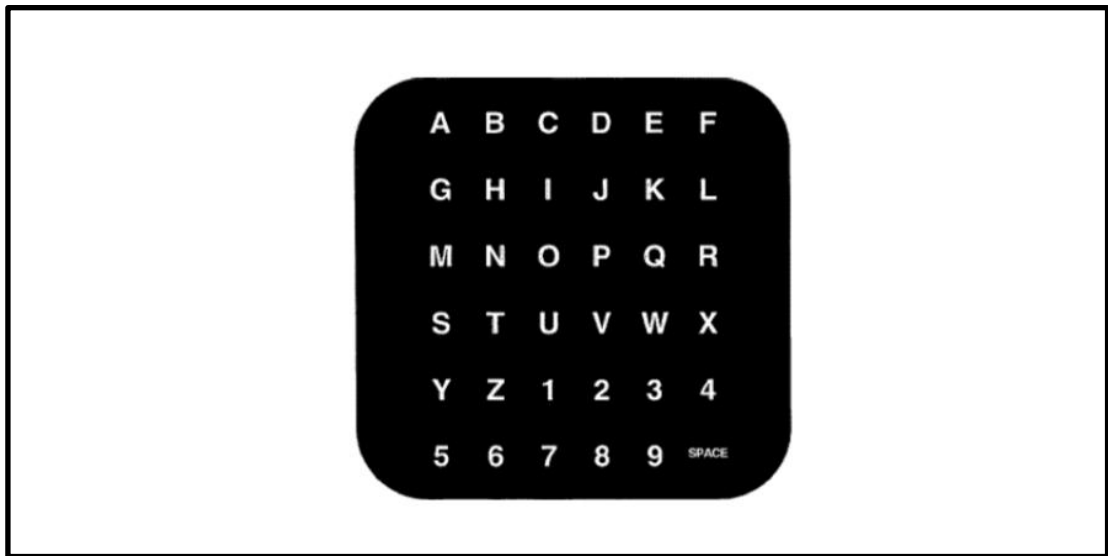


Figura 5. Matriz de letras que se iluminan aleatoriamente utilizada en una BCI basada en el potencial P300.

Fuente: (Henríquez, 2014).

Debido a que los sistemas BCI basados en el potencial P300 utilizan estímulos externos para que el usuario module las señales generadas, estos no necesitan de entrenamiento. Además, la aplicación de estos sistemas son de mucho interés para pacientes con severas deficiencia motoras, pues el control se basa en sus habilidades cognitivas y sensoriales (Henríquez, 2014). Sin embargo, estas BCI también adolecen de un tasa de transferencia de información (Information Transfer Rate o ITR en inglés) muy baja puesto que la clasificación basada solamente en un promediado de valores es muy simple, y la precisión de la detección de los potenciales P300 es muy lenta (Nicolas-Alonso & Gomez-Gil, 2012).

4.2.2.4 Ritmos Sensoriomotores

Los ritmos sensoriomotores son señales cerebrales en forma de oscilaciones, que se registran en la zona somatosensorial y motora de la corteza cerebral, cuyas componentes frecuenciales se concentran en las bandas μ (8 – 12 Hz) y β (13 – 28 Hz) respectivamente. La amplitud de estas señales varía cuando

el usuario realiza actividades mentales relacionadas con alguna actividad motora, sin embargo no se necesita movimiento real del sujeto para generar estos cambios. Esto es debido a que la modulación de amplitud de estos ritmos se puede realizar mediante imaginación motora.

La capacidad que tienen los usuarios para modular estos ritmos, hacen que puedan ser utilizados en sistemas BCI. En la figura 6 se muestra, tanto en el tiempo como en frecuencia, un ritmo sensoriomotor μ utilizado en una BCI para el movimiento de un cursor a través de una pantalla. El cursor se puede mover hacia un objetivo en la parte superior de la pantalla o hacia un objetivo en la parte inferior. La figura 6 muestra que la amplitud del ritmo μ es mayor cuando se encuentra en la parte superior de la pantalla, mientras que se vuelve menor cuando se encuentra en la parte inferior (Henríquez, 2014).

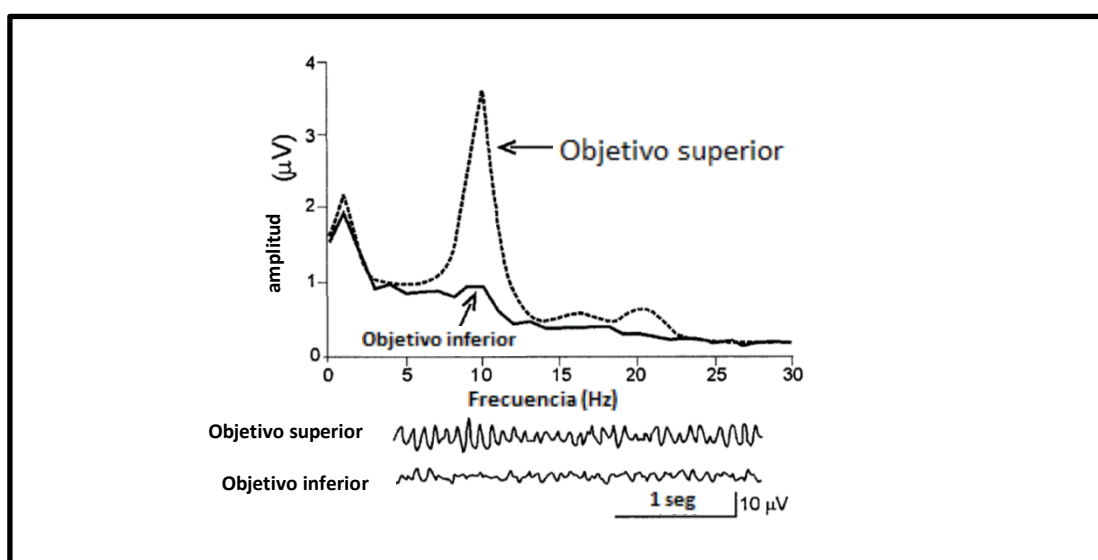


Figura 6. Panel superior: Espectro de frecuencia de un EEG donde se utiliza un ritmo sensoriomotor μ para el control de una BCI. Panel inferior: Registro en el tiempo de una señal EEG que muestra el uso de un ritmo sensoriomotor μ para el control de una BCI.

Adaptado de: (Henríquez, 2014).

Los usuarios que utilizan BCI basadas en ritmos sensoriomotores necesitan pasar a través de un proceso de entrenamiento para lograr modular de forma adecuada estas señales, de tal manera que puedan ser utilizadas por un sistema BCI. Los sistemas de entrenamiento generalmente utilizan retroalimentación con estímulos auditivos y visuales. Sin embargo, Hwang y

otros desarrollaron un sistema de entrenamiento que muestra la actividad cortical del usuario en tiempo real como estímulo de retroalimentación, que probó ser más eficiente que los métodos anteriormente mencionados (Nicolas-Alonso & Gomez-Gil, 2012).

Finalmente, cabe mencionar que los ritmos sensoriomotores han sido ampliamente estudiados en el campo de las BCIs, y algunos de los sistemas BCI más conocidos utilizan estos ritmos como señales de control. Los adelantos en la investigación de estos ritmos han demostrado que es posible predecir movimientos voluntarios de una persona antes de que los realice, en base a la modulación de los ritmos sensoriomotores que ésta presente (Nicolas-Alonso & Gomez-Gil, 2012).

4.3 CAPÍTULO 3: TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES USADAS EN BCIs

4.3.1 Pre-Procesamiento de la Señal

Las señales neurológicas aparecen generalmente como señales de baja intensidad que pueden estar enmascaradas bajo interferencias o ruido. Estas señales que no aportan información relevante para la BCI pueden tener varios orígenes, incluyendo otros procesos fisiológicos del cuerpo humano, los instrumentos utilizados para su adquisición y el medio ambiente en el que se realizan las pruebas (Henríquez, 2014).

La etapa de pre-procesamiento tiene la función de eliminar estas señales de interferencia para que se pueda identificar de una manera más simple la información de relevancia en las etapas de análisis posteriores. El pre-procesamiento puede ser realizado a través de varios tipos de filtros y algoritmos computacionales.

4.3.1.1 Filtros Digitales

Un filtro es un sistema lineal e invariante en el tiempo que tiene la capacidad de suprimir ciertas componentes de frecuencia de una señal pero deja pasar otras sin cambios (Oppenheim & Schafer, 2009).

Los filtros pueden clasificarse en (Pertence, 1991):

- **Filtros Pasa Bajo:** Que son aquellos que permiten el paso de todas las componentes de frecuencia que se encuentren por debajo de una frecuencia de corte f_c , mientras que suprime todas las frecuencias que se encuentren por encima de esta.
- **Filtros Pasa Alto:** Son aquellos que permiten el paso de las frecuencias que se encuentren por encima de cierta frecuencia de corte f_c , pero eliminan todas las componentes que se encuentren por debajo de esta.
- **Filtros Pasa Banda:** Estos filtros permiten el paso de las frecuencias que se encuentren ubicadas dentro de una banda delimitada por una frecuencia de corte inferior f_{c1} y una frecuencia de corte superior f_{c2} , mientras que las frecuencias afuera de la banda sin suprimirlas.

- **Filtros Rechaza Banda:** En estos filtros se suprimen todas las frecuencias que se encuentren dentro de una banda delimitada por una frecuencia de corte inferior y una frecuencia de corte superior, mientras que las componentes fuera de la banda no sufren ninguna modificación.

En la figura 7 se ilustra la respuesta en frecuencia ideal de cada uno de estos filtros. En la práctica, transiciones inmediatas como las mostradas en la figura harían que estos sistemas no sean causales, volviéndolos no realizables; para su implementación real, los filtros presentan pendientes de transición, y a veces ciertos rizados en las bandas de paso y de supresión, los cuales son determinados por ciertos parámetros de tolerancia.

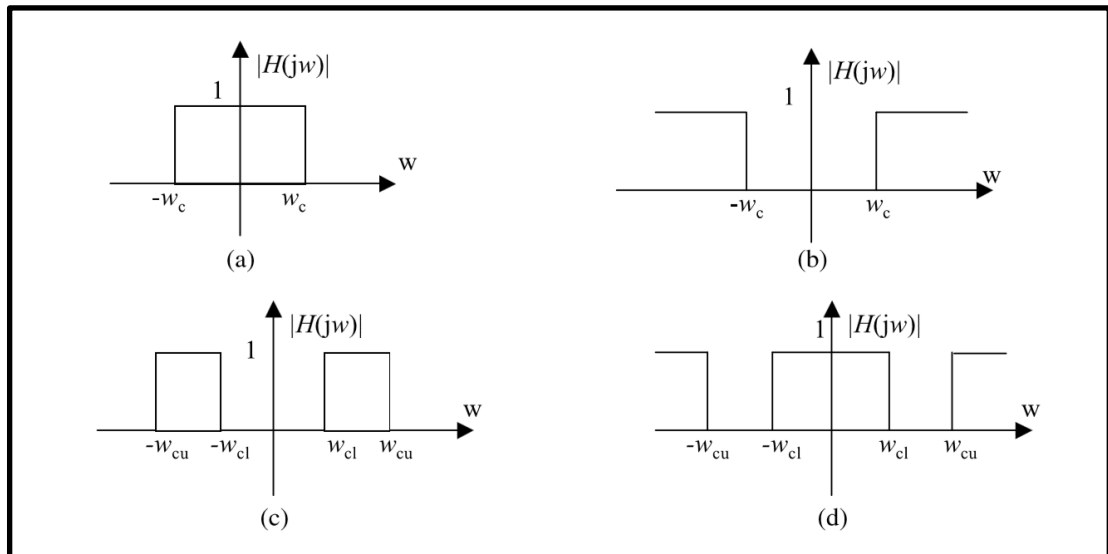


Figura 7. Respuesta en frecuencia ideal de diferentes tipos de filtros: a) Filtro paso bajo, b) Filtro paso alto, c) Filtro pasa banda, d) Filtro rechaza banda.

Adaptado de: (ElAli & Karim, 2008).

4.3.2 Métodos de Extracción de Características

Después de la adquisición de las señales cerebrales y su pre-procesamiento para eliminar ruido, estas señales pasan por una serie de técnicas de procesamiento digital para determinar la actividad mental que las generó, con la finalidad de identificar la acción de control que el usuario busca ejecutar.

“Las BCI se pueden considerar como un sistema de reconocimiento de patrones que clasifica cada patrón en una clase de acuerdo a sus características”. (Nicolas-Alonso & Gomez-Gil, 2012).

El reconocimiento de la señal cerebral se realiza en dos fases: la fase de extracción de características de la señal y la fase de clasificación de características. En la primera etapa se extraen una serie de parámetros de las señales cerebrales que permitirán identificar las intenciones del usuario, mientras que en la segunda etapa se aplican algoritmos de traducción sobre estos parámetros para reconocer la señal y ejecutar la acción de control correspondiente (Schalk & Mellinger, 2010).

Las señales cerebrales son de naturaleza no estacionaria, lo que significa que no presentan un comportamiento periódico a la largo del tiempo. Además, estas se generan en ambientes altamente ruidosos y a partir de diferentes fuentes, por lo que la señal de interés asociada a una actividad cognitiva específica podría estar enmascarada por otras actividades cognitivas y otros bio-potenciales, tales como actividad eléctrica cardíaca y muscular. Por esta razón, la utilización de técnicas de procesamiento digital tradicionales para la extracción de características, como el uso de filtros para extraer las bandas de frecuencia de interés, dan lugar a resultados de clasificación pobres.

Para la extracción de características de las señales cerebrales se utilizan varias técnicas de procesamiento moderno, cada una con diferentes propiedades. En la tabla 3 se presenta una lista de las diferentes técnicas de caracterización de señales cognitivas que se utilizan en la implementación de interfaces cerebro-computador.

Tabla 3. Lista de técnicas de extracción de características utilizadas en interfaces cerebro-computador.

Fuente: (Nicolas-Alonso & Gomez-Gil, 2012).

	MÉTODO	PROPIEDADES
Técnicas de Extracción de Reducción de Dimensión	Análisis de Componentes Principales (PCA)	<ul style="list-style-type: none"> - Transformación lineal. - Conjunto de observaciones posiblemente correlacionadas se transforma en un conjunto de variables sin correlación. - Representación óptima de los datos en términos del

		<p>error cuadrático medio mínimo.</p> <ul style="list-style-type: none"> - Nada garantiza que siempre se obtendrá una clasificación adecuada. - Es un método valioso para la reducción de ruido y dimensión. PCA necesita que los artefactos no tengan correlación con la señal EEG.
	Análisis de Componentes Independientes (ICA)	<ul style="list-style-type: none"> - Divide un conjunto de señales mezcladas en sus fuentes. - Se asume la independencia estadística mutua entre las fuentes. - Poderosa y robusta herramienta para la remoción de artefactos. Se necesita que los artefactos sean independientes de la señal EEG. - Puede corromper el espectro de frecuencia.
Técnicas de Extracción Espaciales	Patrones Espaciales Comunes (CSP)	<ul style="list-style-type: none"> - Filtro espacial diseñado para problemas de dos clases. Existe una extensión multi-clase. - Buenos resultados para BCIs síncronas. Menos efectiva para BCIs asíncronas. - Su desempeño se ve afectado por la resolución espacial. - Existen versiones mejoradas de CSP: WCSP, CCSP, CSSSP.
Técnicas de Extracción Temporales-Frecuenciales	Componentes Autorregresivos (AR)	<ul style="list-style-type: none"> - Modelo del espectro. - Alta resolución en la frecuencia para segmentos cortos de tiempo. - No es adecuado para señales no estacionarios. - Existe una versión adaptativa de AR: MVAAR.
	Filtro Adaptado (MF)	<ul style="list-style-type: none"> - Determina un patrón específico en base a su correspondencia con señales predeterminadas conocidas o modelos. - Adecuada para la detección de formas de onda con características temporales consistentes.
	Transformada Wavelet Continua (CWT)	<ul style="list-style-type: none"> - Provee tanto información temporal como frecuencial. - Adecuado para señales no estacionarias.
	Transformada Wavelet Discreta (DWT)	<ul style="list-style-type: none"> - Provee tanto información temporal como frecuencial. - Adecuado para señales no estacionarias. - Reduce la redundancia y complejidad de la CWT.

Como se puede apreciar en la tabla, cada método de extracción de características presenta diferentes ventajas y desventajas, y la selección de uno de estos métodos depende en gran medida del tipo de tarea que se busca realizar. Por ejemplo, los métodos de PCA e ICA se usan principalmente para la depuración de artefactos en las señales EEG (Nicolas-Alonso & Gomez-Gil, 2012). En el caso que nos compete, los métodos que se emplean más a menudo para la clasificación de ritmos sensoriomotores son aquellos basados en el análisis del dominio de la frecuencia, tales como la transformada wavelet, y los métodos basados en coeficientes autorregresivos (Schalk & Mellinger, 2010).

4.3.2.1 Common Spatial Patterns (CSP)

Las señales EEG en bruto obtenidas del cuero cabelludo tienen una resolución espacial muy pobre debido a las características conductivas del tejido del cráneo. Esta situación es problemática a la hora de analizar señales de débil intensidad, tales como los ritmos sensoriomotores, mientras otras fuentes en la cabeza producen señales más fuertes que operan en la misma banda de frecuencia como la actividad α , o si aparecen artefactos oculares y musculares (Blankertz, Tomioka, Lemm, Kawanabe, & Müller, 2008).

Para mejorar la intensidad de la señal de interés, algunos métodos tratan de calibrar el sistema a las características específicas de los usuarios; inspirados por esta alternativa, se ha demostrado que las técnicas basadas en datos para desarrollar filtros espaciales específicos a cada usuario proveen buenos resultados (Blankertz, Tomioka, Lemm, Kawanabe, & Müller, 2008).

Para entender cómo es que el filtrado espacial ayuda a la discriminación entre estados mentales, refiérase a la figura 8. Aquí se muestran los espectros de frecuencia de imaginación motora de mano derecha e izquierda tomados de una misma zona de la cabeza. Los espectros representan los mismos datos EEG, pero se utilizaron diferentes filtros espaciales para calcularlos. El espectro generado a partir de los datos en bruto es similar para los dos tipos de imaginación, mostrando un mismo pico alrededor de 9 Hz, siendo muy difícil discriminar entre clases. Los filtros bipolar y CAR (Referencia Promedio Común o Common Average Reference) muestran pequeñas diferencias entre los espectros de cada clase, pero es gracias a los filtros de Laplace y CSP

(Patrones Espaciales Comunes o Common Spatial Patterns) que se puede observar una clara diferencia entre las clases, ya que revelan la presencia de un segundo pico alrededor de 12 Hz en el espectro de imaginación de mano derecha.

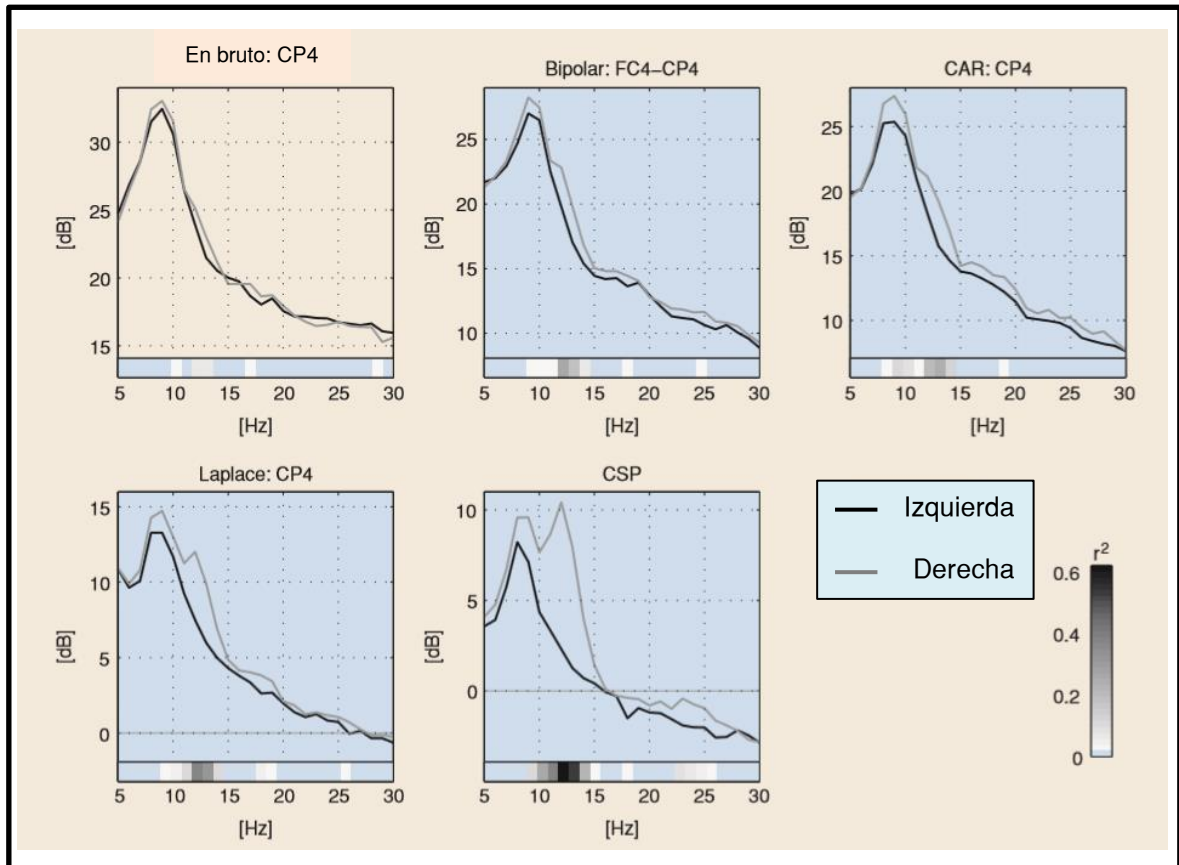


Figura 8. Espectro de frecuencia de imaginación motora de mano izquierda y derecha. Cada gráfico se calculó con la misma señal EEG pero se usó diferentes tipos de filtros espaciales.

Adaptado de: (Blankertz, Tomioka, Lemm, Kawanabe, & Müller, 2008).

CSP es un método matemático de extracción de características ampliamente aplicado en la clasificación de datos que pertenecen a problemas de dos clases (Nicolas-Alonso & Gomez-Gil, 2012). En el campo de las BCI, CSP permite proyectar las señales EEG en un subespacio donde las diferencias entre las clases se maximizan. Para lograr esto, se desarrollan filtros espaciales basados en CSP que permiten obtener señales de varianza óptima; es decir que el resultado de aplicar un filtro espacial CSP será que la diferencia en la varianza de las señales de dos clases se maximizará (Olías, 2016).

Para el cálculo de un filtro espacial CSP se puede aplicar el siguiente algoritmo:

Sea $X \in \mathbb{R}^{C \times T}$ un segmento de señal EEG que pertenece a una grabación de una clase de imaginaria de movimiento, de donde C es el número de canales y T el número de muestras adquiridas. Se utilizará X^i y X^d para denotar grabaciones EEG que pertenecen a imaginaria de clase izquierda y derecha respectivamente. CSP utiliza una matriz $W \in \mathbb{R}^{C \times C}$ que proyecta una señal X de su espacio de sensores originales hacia la señal $X_{CSP} \in \mathbb{R}^{C \times T}$, tal que

$$X_{CSP} = W^T X \quad (4.1)$$

de donde W^T es la transpuesta de W . Se denomina filtro espacial o filtro a cada vector columna $w_j \in \mathbb{R}^C$ ($j = 1, 2, 3, \dots, C$) que pertenece a W . Además, vale la pena mencionar que cada vector columna $a_j \in \mathbb{R}^C$ ($j = 1, 2, 3, \dots, C$) de una matriz $A = (W^{-1})^T$ se conoce como patrón espacial o patrón.

Para calcular W se parte de las matrices de covarianza espacial Σ^I y Σ^D pertenecientes a cada clase. Estas se calculan a partir de

$$\Sigma^I = \frac{X^i (X^i)^T}{\text{trace}[X^i (X^i)^T]} \quad (4.2) \quad \text{y},$$

$$\Sigma^D = \frac{X^d (X^d)^T}{\text{trace}[X^d (X^d)^T]} \quad (4.3)$$

De donde la función $\text{trace}(X)$ representa la operación de sumar todos los elementos en la diagonal de X . Seguidamente, W se puede obtener mediante la diagonalización simultánea de las dos matrices de covarianza, lo que matemáticamente se expresa como

$$\begin{aligned} W^T \Sigma^i W &= \Lambda^i, \\ W^T \Sigma^d W &= \Lambda^d, \end{aligned} \quad (4.4)$$

donde W debe cumplir que $\Lambda^I + \Lambda^D = I$, siendo I la matriz identidad. Esto conlleva a la resolución del problema de autovectores generalizado

$$W^T \Sigma^i W = \Lambda \wedge W^T (\Sigma^i + \Sigma^d) W = I \quad (4.5)$$

Por lo tanto, al resolver el problema planteado en (4.5) se encontrará una W tal que podrá satisfacer (4.4), siendo esta matriz el filtro espacial buscado. Una curiosidad muy encontrada en la literatura es que (4.5) puede resolverse fácilmente en Matlab con una sola línea de código: $[W, \Lambda] = \text{eig}(\Sigma^i, \Sigma^i + \Sigma^d)$ (Blankertz, Tomioka, Lemm, Kawanabe, & Müller, 2008).

4.3.2.2 La Transformada Wavelet Continua

La transformada de Fourier (*Fourier Transform* o FT en inglés) es una poderosa herramienta para el análisis de señales debido a que a través de ella es posible extraer información relevante de una señal en términos de la frecuencia. Esto es posible gracias a que la FT puede representar una señal $f(t)$ como una combinación lineal de senos y cosenos. Las frecuencias de mayor magnitud son las que aportan más para la constitución de $f(t)$, mientras que las de menor magnitud pueden asumirse despreciables, pudiéndolas eliminar sin que se aprecie alguna degradación en la señal. Es así que gracias a la representación frecuencial de una señal, se hace posible la visualización de características que pueden no ser aparentes en el dominio temporal.

Sin embargo, la FT no está exenta de limitaciones. Debido a que utiliza senos y cosenos como funciones base, los cuales ocupan una extensión temporal infinita, la FT tiene un desempeño pobre para la representación de señales no estacionarias, transitorias o que varían en el tiempo (Rao, 2013). Si se aplicara la FT a alguna señal perteneciente a alguno de los tipos mencionados anteriormente, se obtendría una función en donde se podría observar todas las componentes frecuenciales que son parte de la señal, pero no se tendría ninguna información acerca de cuándo se habrían originado estas componentes dentro de la señal. En otras palabras, la FT asume que las componentes frecuenciales en una señal son constantes en el tiempo, lo que puede llevar a análisis erróneos al representar señales no estacionarias o variantes en el tiempo.

La Transformada de Fourier de Tiempo Corto (*Short Time Fourier Transform* o STFT en inglés) se propuso como método de análisis para sobrellevar las limitaciones de la FT. Esta técnica define una pequeña ventana de tiempo dentro de la cual se realiza la FT de una parte de la señal a analizar, para

después, trasladar la ventana en el tiempo para realizar la FT de la siguiente parte de la función y así sucesivamente hasta analizar la totalidad de la señal. De esta manera la STFT permite el análisis de una señal en el dominio del tiempo y la frecuencia de forma simultánea. A pesar de esto, la STFT también tiene serias limitaciones en la representación de una señal, debido a que la elección del tamaño de la ventana es un compromiso entre resolución en el tiempo y la frecuencia; las ventanas de tamaños pequeños proveen de una gran resolución temporal a costa de una resolución frecuencial pobre, y de la misma manera, los tamaños de ventana grandes dan lugar a excelentes resoluciones frecuenciales pero con poca resolución temporal (Nicolas-Alonso & Gomez-Gil, 2012) (Rao, 2013). Esto es debido a que en el análisis y procesamiento de señales, la concentración de energía en el tiempo y en la frecuencia de una señal está gobernada por el principio de incertidumbre de Heisemberg, por lo que no se puede tener una resolución en el tiempo y la frecuencia de forma arbitrariamente exacta y simultánea. Es debido a estas limitaciones que se desarrolló la transformada Wavelet (*Wavelet Transform* o WT en inglés), en reemplazo a la STFT.

Una wavelet, u ondícula como muchas veces se traduce el término al español, es una función oscilante de duración finita, lo que significa que su energía se encuentra localizada en el tiempo, por lo que estas funciones se constituyen en una gran herramienta para el análisis de señales no estacionarias (Burrus, Gopinath, & Haitao, 1998) en contraste a la FT. La WT representa una señal como una combinación de copias escaladas y trasladadas en el tiempo de una wavelet madre $\psi(t)$, haciendo que la WT pueda sobrellevar las limitaciones de la STFT, porque el análisis de la señal se hace en el tiempo y la frecuencia a varias resoluciones, ya que la wavelet actúa como una especie de ventana temporal modulada. Al cambiar la escala, la cual determina el tamaño de la wavelet, se pueden obtener diferentes niveles de resolución en la frecuencia. La traslación de la wavelet sobre la señal a analizar permite obtener información temporal de cuando aparecen las componentes frecuenciales observadas en la escala. En la figura 9 se puede observar diferentes traslaciones en el tiempo y escalas de una wavelet madre ψ_{D4} .

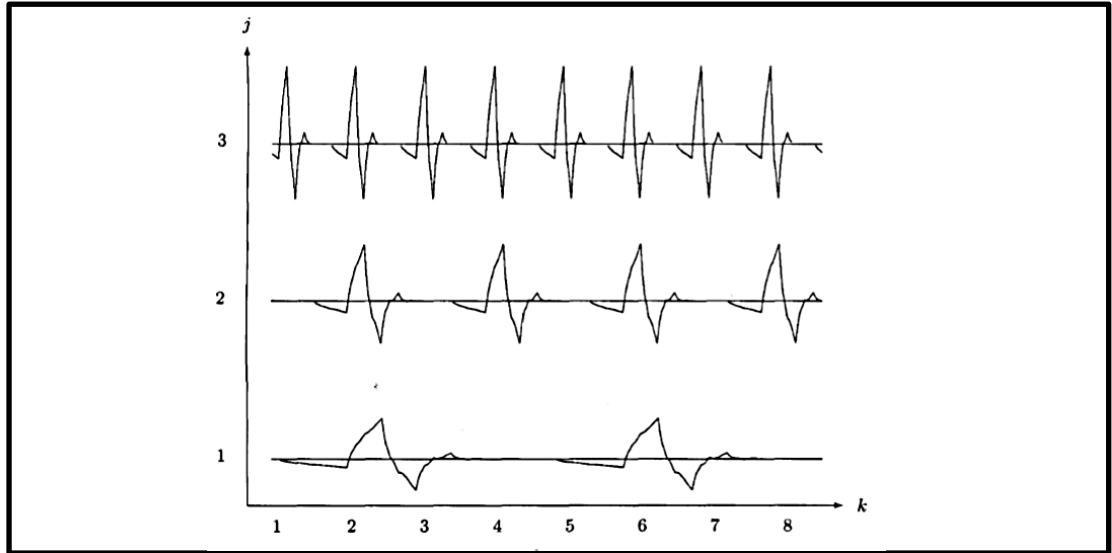


Figura 9. Traslación y escalado de una wavelet ψ_{D4} .

Adaptado de: (Burrus, Gopinath, & Haitao, 1998).

La transformada wavelet continua (CWT) de una señal $x(t)$ se define como

$$w(s, \tau) = \int_{-\infty}^{\infty} x(t) \psi_{s,\tau}^*(t) dt \quad (4.6)$$

donde $w(s, \tau)$ es el coeficiente wavelet que corresponde a la función wavelet $\psi_{s,\tau}(t)$. El símbolo de asterisco “*” representa al complejo conjugado de $\psi_{s,\tau}(t)$. La wavelet $\psi_{s,\tau}(t)$ es una versión escalada por un factor de s y con un desplazamiento en el tiempo τ de la wavelet madre $\psi(t)$. Matemáticamente, $\psi_{s,\tau}(t)$ se define como

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (4.7)$$

Existen varios tipos de wavelets madre, pero todas deben satisfacer la siguiente condición (Nicolas-Alonso & Gomez-Gil, 2012)

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (4.8)$$

4.3.2.3 La Transformada Wavelet Discreta

La CWT da lugar a una gran cantidad de redundancia debido a que involucra analizar una señal en un gran rango de frecuencias mediante varios escalamientos y traslaciones de la wavelet madre. Por esta razón, la transformada wavelet discreta (*Discrete Wavelet Transform* o DWT en inglés)

se desarrolló para reducir la redundancia de la CWT (Nicolas-Alonso & Gomez-Gil, 2012).

La diferencia principal de la DWT con la CWT es que la primera escala y traslada a la wavelet madre en una serie de valores discretos solamente, a diferencia de la CWT que permite traslaciones y escalas con valores continuos (Nicolas-Alonso & Gomez-Gil, 2012). Los valores que puede adoptar la escala en la DWT están dados por

$$s = 2^j: j = 1, 2, 3, 4 \dots \quad (4.9)$$

Mientras que los valores de traslación están dados por los valores de

$$\tau = 2^j m: m = 1, 2, 3, 4, \dots \quad (4.10)$$

Aplicar la DWT sobre una señal es equivalente a pasar dicha señal por un banco de filtros paso-alto y paso-bajo. Primeramente, la señal $x[n]$ a analizar se usa como entrada para dos filtros FIR (filtro de respuesta finita al impulso), uno pasa-bajo y el otro pasa-alto, que dejan pasar la mitad de las frecuencias inferiores y superiores del ancho de banda de $x[n]$ respectivamente. Este proceso se muestra en la figura 10. A este par de filtros se los conoce comúnmente como los filtros de análisis (Weeks, 2007). A continuación de la salida de cada filtro se encuentra un bloque diezmador para hacer más eficiente la transformada (Weeks, 2007). La salida del diezmado del filtro paso-bajo se conoce como el nivel de aproximación mientras que la salida del diezmado del filtro paso-alto se denomina como el nivel de detalle. La salida del nivel de aproximación se puede alimentar a otro par de filtros de análisis, y así sucesivamente, con lo que se pueden obtener tantos niveles de descomposición como lo permita la longitud de $x[n]$. Cada nivel de descomposición se conoce como *octava*. La salida de los niveles de detalle y aproximación de cada etapa son los coeficientes de la transformada wavelet.

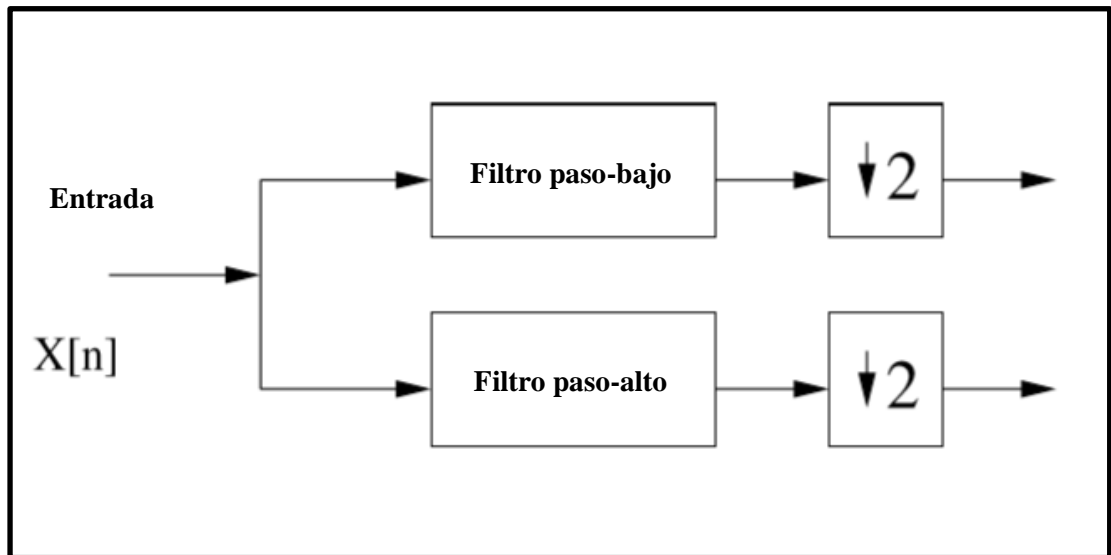


Figura 10. Filtros de análisis de la DWT.

Adaptado de: (Weeks, 2007).

Gracias a que la salida de cada octava tiene un tamaño igual a la mitad de la salida de la etapa precedente, la representación wavelet tiene un tamaño casi igual al de la señal original (Weeks, 2007), eliminando así la redundancia y la gran cantidad de coeficientes que se obtienen a través de la CWT.

El nivel de aproximación de cada octava se puede eliminar debido a que estos pueden ser calculados de nuevo al aplicar la transformada wavelet inversa (Weeks, 2007). El nivel de aproximación del último nivel de descomposición es el único que se mantiene. Los coeficientes de salida de cada octava se ordenan colocando primero los coeficientes del nivel de aproximación de la octava más alta y a continuación los niveles de detalle de cada octava en orden descendente, tal y como se puede observar en la figura 11.

Gracias a que la DWT se puede implementar de esta manera, una señal $x[n]$ puede ser analizada en sub-bandas de frecuencia progresivamente más pequeñas, con diferentes niveles de resolución (YouTube, 2016).

El primer problema que se debe solucionar cuando se trabaja con modelos AR es el de seleccionar el orden adecuado para el modelo, pues si se elige un modelo con un orden muy bajo, el espectro de la señal analizada tiende a suavizarse, mientras que si el orden es muy alto, pueden aparecer picos aleatorios en el espectro. Para el presente trabajo, se usó un modelo de orden 6 para el análisis de las señales EEG, en base a la recomendación de (Xu & Song, 2008).

Existen varias técnicas para el cálculo de los coeficientes AR, tales como el método Yule-Walker, Burg, de covarianza y algoritmos de avance-retroceso (Nicolas-Alonso & Gomez-Gil, 2012). Una vez más el trabajo de (Xu & Song, 2008) se tomó como referencia, por lo que se decidió aplicar el método de Burg para la estimación de los coeficientes AR.

4.3.3 Métodos de Clasificación

En la fase de clasificación se determina la actividad mental realizada por el usuario en base a las características extraídas en la etapa anterior. Para realizar esto se pueden utilizar algoritmos de regresión así como también algoritmos de clasificación, los cuales se subdividen a su vez en algoritmos de clasificación lineales y no lineales.

Los algoritmos de regresión tienen como objetivo emplear las características obtenidas de las señales EEG como variables independientes para predecir las intenciones de los usuarios. Por su parte, los algoritmos clasificadores usan las características obtenidas como variables independientes para definir límites entre las diferentes clases (Nicolas-Alonso & Gomez-Gil, 2012). En la figura 12 se muestra la diferencia entre los algoritmos de regresión y clasificación.

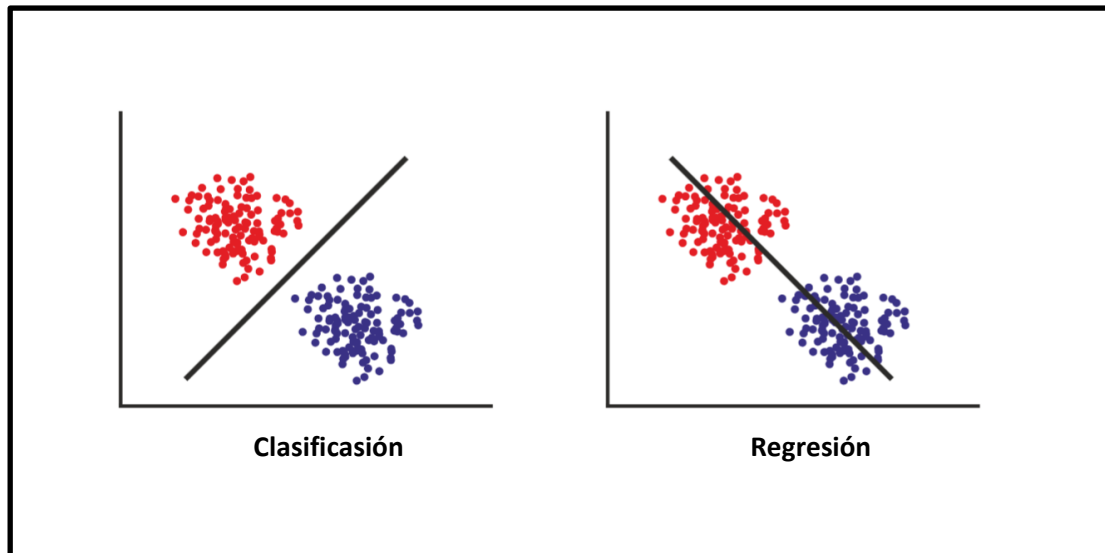


Figura 12. Diferencias entre los algoritmos de regresión y clasificación usados para la clasificación de las características de señales EEG.

Adaptado de: (Nicolas-Alonso & Gomez-Gil, 2012).

La tabla 4 muestra una lista con algunos de los métodos de clasificación más utilizados en BCIs.

Tabla 4. Listado de técnicas de clasificación de características.

Fuente (Nicolas-Alonso & Gomez-Gil, 2012).

	MÉTODO	PROPIEDADES
Modelo Generativo	Análisis Bayesiano	<ul style="list-style-type: none"> - Asigna el vector de propiedades a la clase etiquetada a la que tiene más probabilidad de pertenecer. - Produce límites de decisión no lineales. - No es muy popular en los sistemas BCI.
	Análisis de Discriminación Lineal (LDA)	<ul style="list-style-type: none"> - Clasificador simple con precisión aceptable. - Bajos requerimientos computacionales. - Falla en presencia de valores atípicos o fuerte ruido. Requiere regularización. - Contempla la clasificación de dos clases. Existen versiones multi-clase. - Existen versiones LDA mejoradas: BLDA, FLDA.
Lineal	Máquina de	<ul style="list-style-type: none"> - Modalidades lineales y no lineales (gausiana).

No Lineal	Soporte de Vectores (SVM)	<ul style="list-style-type: none"> - Método binario o multi-clase. - Maximiza la distancia entre las muestras de entrenamiento más cercanas y los hiperplanos. - Falla en presencia de valores atípicos o fuerte ruido. Requiere regularización. - Clasificación rápida.
	Clasificador de K-Vecinos más Cercanos (K-NNC)	<ul style="list-style-type: none"> - Utiliza distancias métricas entre la característica de prueba y sus vecinos. - Multi-clase. - Es eficiente con vectores de características de dimensiones bajas. Es muy sensible a las dimensiones de los vectores de características.
	Red Neuronal Artificial (ANN)	<ul style="list-style-type: none"> - Clasificador muy flexible. - Multi-clase. - Múltiples arquitecturas (PNN, Fuzzy ARTMAP ANN, FIRNN, PeGNC).

De la tabla anterior se puede concluir que al igual que los métodos de extracción de características, existen diversos tipos de clasificadores, cada uno con sus propiedades específicas. A la hora elegir un tipo de clasificador para una aplicación BCI, la tendencia es elegir algoritmos simples por sobre alternativas más complejas. Sin embargo, si los métodos más complejos presentan mejores resultados que los procesos simples, estos últimos tendrán que ser reemplazados (Nicolas-Alonso & Gomez-Gil, 2012).

4.3.3.1 Redes Neuronales Artificiales

Las redes neuronales artificiales (Artificial Neural Networks o ANN por sus siglas en inglés) son un modelo computacional que busca emular el sistema nervioso de los animales para la resolución de problemas. De esta manera, las redes neuronales son capaces de resolver con facilidad problemas a los cuales la computación convencional no les puede dar una solución adecuada (Forslund, 2003). Poseen una gran versatilidad, lo que les ha permitido ser aplicadas exitosamente en una gran cantidad de campos entre los que se cuentan la biología, la medicina, la industria, la ingeniería de control, y muchos más.

En términos computacionales, el cerebro (humano o animal) se puede considerar como un procesador en paralelo, no lineal, capaz de organizarse a sí mismo para adaptarse a su ambiente (Forslund, 2003). Es por esta razón que las dos principales ventajas que las redes neuronales poseen, gracias a las características heredadas de sus pares biológicos, es su capacidad de resolver problemas complejos, no lineales, no estacionarios y estocásticos, y su capacidad de auto-organización lo que permite que su aplicación se pueda generalizar a una gran cantidad de problemas.

Las AAN están compuestas por un número determinado de células neuronales que se interconectan entre sí, donde cada conexión posee un peso asociado. La salida de una neurona está determinada por los valores de cada una de sus entradas ponderados con sus respectivos pesos. Gracias a esto las redes neuronales aprenden a generar la salida deseada ajustando los pesos de sus interconexiones. El proceso por el cual las redes neuronales ajustan el peso de sus interconexiones para realizar una tarea determinada se denomina “aprendizaje”. “Se ha demostrado teóricamente que una red neuronal artificial puede aproximar cualquier mapeo continuo con precisión arbitraria” (Suzuki, 2013).

4.3.3.2 El Perceptrón

El perceptrón es uno de los primeros modelos neuronales que se desarrollaron. Fue propuesto por F. Rosenblatt en 1958, en base a los trabajos sobre células neuronales realizados por McCulloch y Pitts en años previos. La idea detrás del perceptrón era la de realizar tareas de clasificación, mediante la determinación de las superficies que separan las clases que se busca discriminar. La figura 13 ilustra este concepto.

El perceptrón es una estructura monocapa, compuesta por un número determinado de entradas y una o más salidas. También existe un parámetro de umbral θ que se utiliza como valor de comparación para producir la salida, y existen tantos como salidas tenga la red (Viñuela & Galván, 2004).

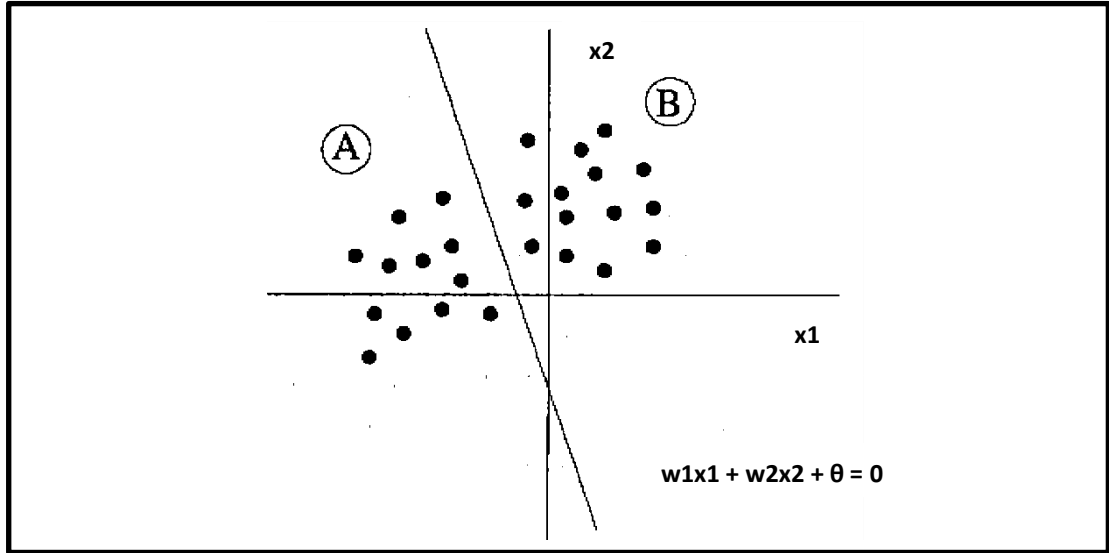


Figura 13. Diferenciación de dos clases realizada por un perceptrón. x_1 y x_2 son las entradas del modelo, w_1 y w_2 son los pesos de las entradas y θ es el valor de umbral.

Adaptado de: (Viñuela & Galván, 2004).

Como primer paso para generar una salida, la red hace la suma ponderada por los pesos w_n de todas las entradas i_n . Esto se puede expresar matemáticamente como (Viñuela & Galván, 2004)

$$y' = \sum_{i=1}^n x_i w_i \quad (4.12)$$

La salida es una función de la sumatoria ponderada de las entradas y el nivel de activación θ . Esta función es de tipo escalón que permite diferenciar entre dos clases, dependiendo si la sumatoria es mayor o menor a θ . Formalmente, esto se expresa como (Viñuela & Galván, 2004)

$$y = f(y', \theta) \quad (4.13)$$

Donde y es la salida del perceptrón y la función f se denomina función de activación.

Si se pasa θ al otro lado de la ecuación (4.13), se puede escribir

$$y = f(\sum_{i=1}^n \omega_i x_i + \theta) \quad (4.14)$$

Esta expresión nos permite tener un mejor entendimiento del umbral θ , que puede ser visto como el peso de una entrada ficticia a la cual se le aplica un

valor constante de -1 (Viñuela & Galván, 2004). El funcionamiento del perceptrón se muestra en la figura 14.

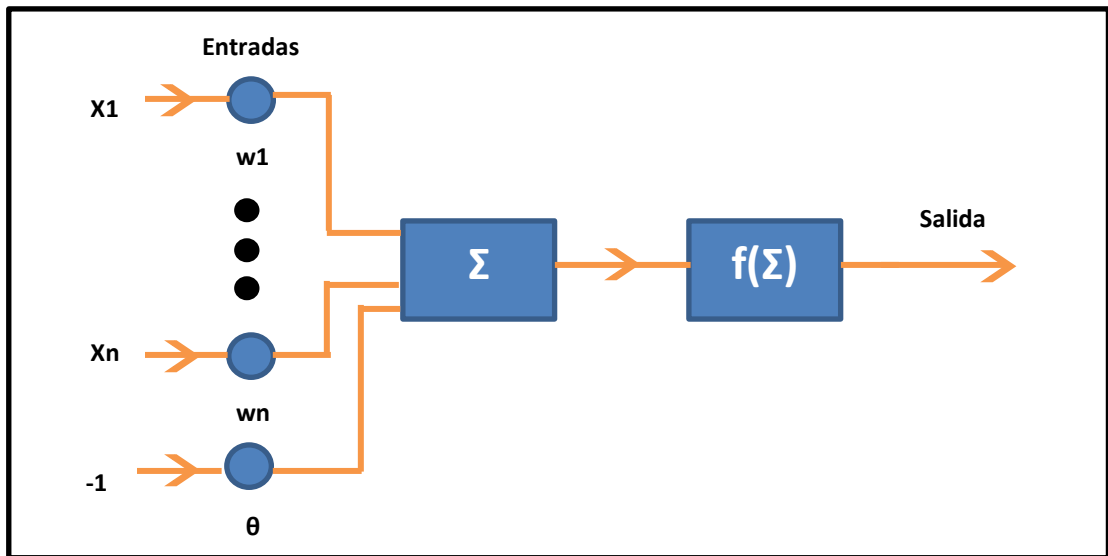


Figura 14. Funcionamiento del perceptrón.

Fuente: El autor.

4.3.3.3 Separabilidad Lineal y el Problema XOR

En 1969, Minsky y Papert publicaron un libro en donde ponían de manifiesto una de las más grandes limitaciones que poseen los perceptrones monocapa, pues estos no pueden discriminar entre clases que no sean linealmente separables. Recordando que el objetivo del perceptrón era el de constituirse en un sistema capaz de resolver problemas de clasificación de forma automática, no es de sorprenderse que durante esa época la comunidad científica se sintiera decepcionada de las redes neuronales, haciendo que casi la totalidad de los estudios en esta área desaparecieran. El interés en las redes neuronales no se recobraría hasta 20 años después, cuando Rumerhat y McClelland propusieron una solución para sobrellevar esta limitante (Forslund, 2003) (Graupe, 2007).

El problema de la separabilidad lineal se puede entender fácilmente utilizando el álgebra lineal. Consideremos un perceptrón con N entradas que se pueden escribir como un vector columna de dimensión $N+1$, $x = (-1, x_1, x_2, \dots, x_n)$ y los pesos de cada entrada se pueden expresar como un vector columna $w = (\theta, w_1, w_2, \dots, w_n)$, también de dimensión $N+1$. Entonces la ecuación (4.14) se

puede escribir como el producto vectorial $x^T w$. Con esta terminología, queda claro que la clasificación del vector de entrada x depende de si el producto $x^T w$ es positivo o negativo. El límite que separa a las clases que se busca diferencias está dado por la ecuación lineal

$$x^T w = 0 \quad (4.15)$$

que define un hiperplano de dimensión N . Esto quiere decir que un espacio compuesto por dos clases será dividido por una recta, un espacio de tres dimensiones será separado por un plano y así sucesivamente (Forslund, 2003).

Refiérase a la figura 15, en ella se puede ver que la operación booleana XOR no puede ser resuelta por un perceptrón, debido a que sus clases no se pueden separar usando una sola recta. Esto significa que sus clases no son linealmente separables. Resolver la operación XOR implica resolver el conjunto de ecuaciones lineales

$$\begin{cases} 0 \cdot w_1 + 0 \cdot w_2 < \theta \\ 0 \cdot w_1 + 1 \cdot w_2 > \theta \\ 1 \cdot w_1 + 0 \cdot w_2 > \theta \\ 1 \cdot w_1 + 1 \cdot w_2 < \theta \end{cases} \quad (4.16)$$

el cual no tiene solución (Forslund, 2003). Obviamente existen muchos otros problemas que no se pueden resolver usando un hiperplano de dimensión N , sino que requieren de mapeos mucho más complejos. La tabla 5 muestra la tabla de verdad de la operación XOR, en donde se utiliza -1 en lugar de usar 0 como dígito binario.

Tabla 5. Tabla de verdad de la operación XOR.

Fuente: El autor.

x_1	x_2	Y
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1

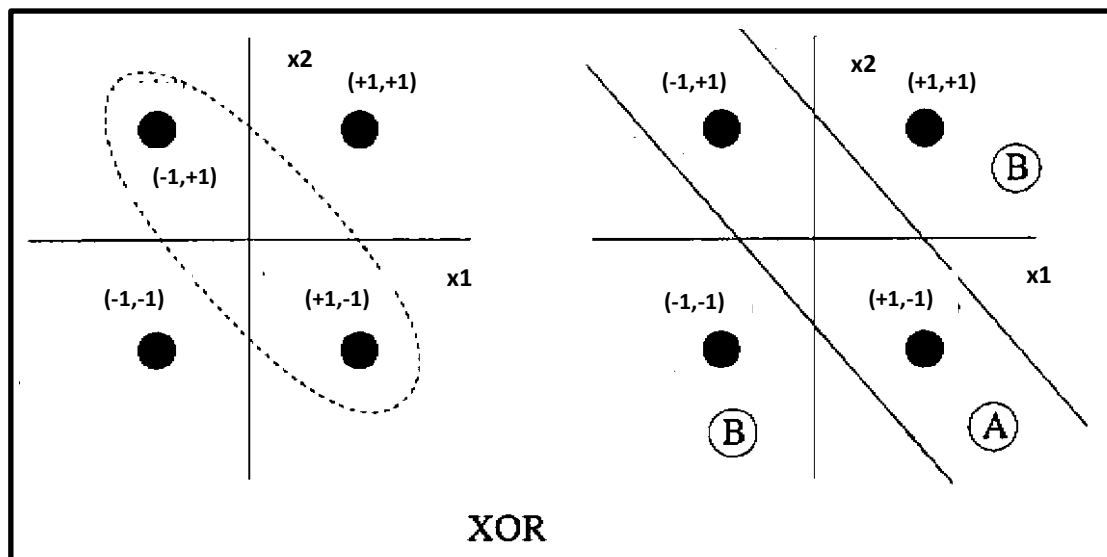


Figura 15. Entradas de la operación XOR representadas en el plano bidimensional y regiones de frontera que separan dichas salidas. Las clases a las que pertenecen las entradas no se pueden separar utilizando una única recta.

Adaptado de: (Viñuela & Galván, 2004).

4.3.3.4 El Perceptrón Multicapa

Cuando Papert y Minsky demostraron la incapacidad del perceptrón para resolver problemas que no sean linealmente separables, también mostraron que mediante la unión de varios perceptrones en cascada era posible resolver algunos problemas no lineales, tales como la operación XOR; sin embargo no presentaron una regla de aprendizaje que se pudiera utilizar para ajustar los pesos de este modelo, ya que el algoritmo de aprendizaje del perceptrón monocapa no se puede aplicar en este caso. Como se mencionó anteriormente, sería 20 años después, en 1989, cuando Rumerlhat, Hinton y Williams presentaron un método en el que la diferencia entre la salida generada por la red y la salida deseada se retropropaga hacia las capas anteriores para ajustar sus pesos. Este método de aprendizaje se conoce como la regla delta generalizada o la regla de retropropagación generalizada (Viñuela & Galván, 2004).

El desarrollo del perceptrón multicapa supuso un avance enorme en el campo de las redes neuronales debido a su gran número de propiedades y ventajas.

Como sistema de clasificación, el perceptrón multicapa no sufre del problema de separabilidad lineal, además se ha demostrado que una red de tres capas puede resolver cualquier problema de clasificación si el número de nodos que la conforman es lo suficientemente alto (Forslund, 2003). Otra cualidad de esta red es que tiene la capacidad de aproximar cualquier función continua compacta en el espacio R^n , con por lo menos una capa oculta de neuronas. Gracias a esto se pueden utilizar redes neuronales para generar cualquier tipo de funciones sean estas lineales o no lineales (Viñuela & Galván, 2004).

Estas redes han sido aplicadas con éxito para la resolución de problemas en una gran variedad de áreas diferentes, como reconocimiento del habla [Cohen et al., 1993], reconocimiento de caracteres ópticos [Sackinger et al., 1992], reconocimiento de caracteres escritos [Guyon, 1991], control de procesos [Werbos, 1989], modelización de sistemas dinámicos [Narendra and Parthasarathy, 1990], conducción de vehículos [Pormeleau, 1992], diagnósticos médicos [Baxt, 1992], predicción de series temporales [Weigend et al., 1990], etc (Viñuela & Galván, 2004, pág. 46).

Tal como su nombre lo indica, el perceptrón multicapa se divide en un número determinado de capas, cada una de las cuales contiene un conjunto de neuronas. La red contiene tres tipos de capas diferentes: la capa de entrada, las capas ocultas y la capa de salida. Esta arquitectura se muestra en la figura 16. La capa de entrada tiene la función de transmitir los valores de entrada hacia la siguiente capa sin realizar ningún cambio en ellos. Por su parte, la capa de salida comunica la salida de la red hacia el exterior y las capas ocultas realizan operaciones no lineales en los patrones de entrada para generar las salidas apropiadas.

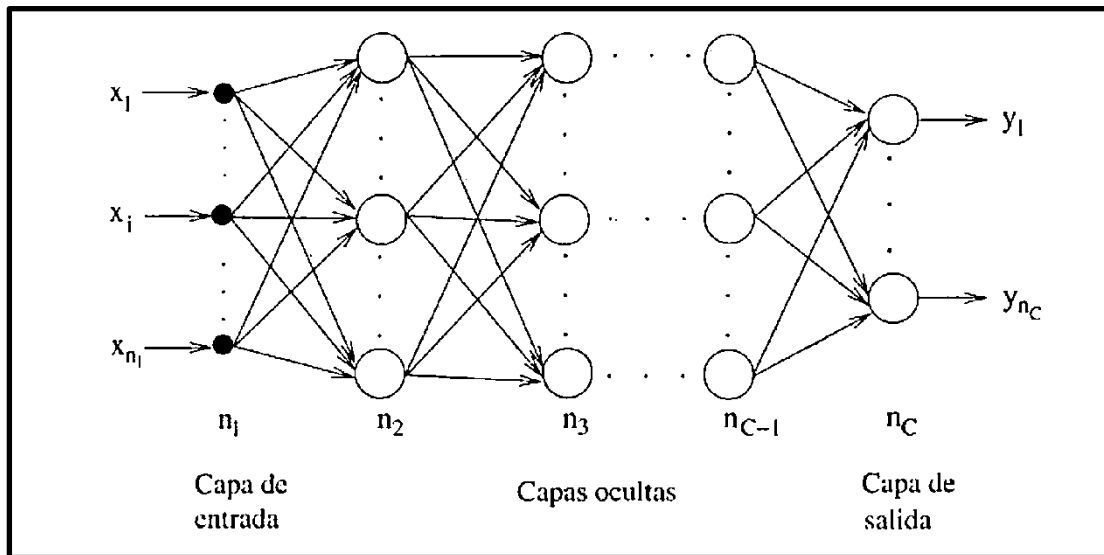


Figura 16. Arquitectura del perceptrón multicapa.

Adaptado de: (Viñuela & Galván, 2004).

El perceptrón multicapa funciona de la siguiente manera (Viñuela & Galván, 2004):

- Las señales de entrada son propagadas hacia las siguientes capas por las neuronas de la capa de entrada, sin realizar ningún cambio en ellas.
- Las capas ocultas procesan la información recibida por las neuronas de entrada para generar la salida deseada. Las neuronas de esta capa pueden ser vistas como perceptrones simples, debido a que la salida que generan es función de la sumatoria ponderada por los pesos de sus entradas más un valor de umbral; la función de activación utilizada es común para todas las neuronas en las capas ocultas. De esta manera, las neuronas en la primera capa oculta generan una salida en función de sus entradas y la propagan hacia la siguiente capa oculta, y así sucesivamente hasta llegar a la capa de salida.
- Las neuronas de la capa de salida funcionan de la misma manera que las de las capas ocultas, generando una salida que es función de la sumatoria ponderada de la suma de las entradas más un valor de umbral. Estas salidas son el resultado definitivo de la red. Generalmente, la función de activación utilizada en las neuronas de la capa oculta y la capa de salida es la misma, sin embargo dependiendo

de la aplicación, se puede utilizar una función de activación diferente para las neuronas de salida.

4.3.3.5 La Regla Delta Generalizada

El mecanismo a través del cual una red neuronal ajusta los pesos de sus interconexiones y los umbrales de sus neuronas para obtener una salida determinada se conoce como regla o algoritmo de aprendizaje. El algoritmo de aprendizaje utilizado en el perceptrón multicapa es de tipo *supervisado*, lo que significa que los parámetros se ajustan para que la salida de la red sea lo más parecida posible a una salida provista por un supervisor; esto quiere decir que para cada patrón o clase de entrada de la red debe disponer de un patrón de salida deseada (Viñuela & Galván, 2004).

En una red perfectamente entrenada, la salida que genera para un patrón de entrada dado es igual a la salida deseada para ese patrón. Las redes no entrenadas, es decir que no hayan sido sometidas a un algoritmo de aprendizaje, darán lugar a una salida que difiere con la deseada en menor o mayor grado. Por lo tanto, es conveniente definir una función de error que nos permita estimar la diferencia que existe entre las salidas de una red y las salidas deseadas. Generalmente esta función se define como (Viñuela & Galván, 2004)

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (4.17)$$

de donde N es el número de patrones que existen y e(n) es el error cometido por la red para el patrón n, definido como la distancia euclidiana

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} [s_i(n) - y_i(n)]^2 \quad (4.18)$$

Donde $Y(n) = [y_1(n), \dots, y_{n_c}(n)]$ y $S(n) = [s_1(n), \dots, s_{n_c}(n)]$ son el vector de salida de la red y el vector de salida deseado para el patrón n, respectivamente, y n_c es el número de neuronas en la capa C.

Al entrenar una red, lo que se busca es que $E \rightarrow 0$, por lo tanto el proceso de aprendizaje de una red no es más que un problema de optimización multidimensional, no lineal, en el que se busca encontrar un mínimo en la función de error (Forslund, 2003) (Viñuela & Galván, 2004). La mayoría de estos métodos se basan en iniciar en un punto arbitrario de la función de error y

a través de un proceso iterativo, dirigirse hacia un mínimo siguiendo una dirección determinada. En el perceptrón multicapa, la técnica de optimización más utilizada es el método de descenso del gradiente, en donde la dirección que se sigue es la dirección negativa del gradiente de E (Viñuela & Galván, 2004).

El método de descenso de gradiente más utilizado es el método del gradiente estocástico, en el cual los parámetros w de la red, tanto pesos como umbrales, se modifican utilizando la siguiente ley de aprendizaje (Viñuela & Galván, 2004):

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w} \quad (4.19)$$

de donde α es la razón de aprendizaje de la red, que influye en la magnitud del desplazamiento del error hacia el mínimo buscado.

En el perceptrón multicapa, el método del descenso de gradiente se puede aplicar de forma exitosa como método de aprendizaje a través de la regla delta generalizada, también conocida como algoritmo de retropropagación, llamado así porque determina el valor de E a la salida de una red, y luego lo propaga hacia las capas anteriores con el fin de ajustar sus parámetros para minimizar E.

A continuación se describe la regla delta generalizada tal y como es presentada por Viñuela y Galván:

Considérese un perceptrón multicapa que contenga C capas, $C-2$ capas ocultas y n_c neuronas en la capa c , donde $c = 1, 2, 3, \dots, C$. Además, sea $W^c = (w_{ij}^c)$ la matriz de pesos asociada a las conexiones que van de la capa c a la capa $c+1$ para $c = 1, 2, \dots, C-1$, donde w_{ij}^c es el peso de la conexión que va desde la neurona i de la capa c hacia la neurona j de la capa $c+1$; y sea $U^c = (u_i^c)$ el vector de umbrales para cada neurona de la capa c donde $c = 2, \dots, C$ y a_i^c es la activación de la neurona i en la capa c . Sea, también, f la función de activación que utilizan todas las neuronas de la red; la regla delta generalizada define la manera en que se modificarán los parámetros de la red en dos partes (Viñuela & Galván, 2004):

- Primero, se definen las ecuaciones que dictan la manera en que se deben modificar los parámetros de la capa oculta C-1 hacia la capa de salida. La expresión que determina como se modifican los pesos en las interconexiones entre estas capas es

$$w_{ji}^{C-1}(n) = w_{ji}^{C-1}(n-1) + \alpha \delta_i^C(n) a_j^{C-1}(n) \quad (4.20)$$

para $j = 1, 2, \dots, n_{C-1}$ e $i = 1, 2, \dots, n_C$.

Los umbrales de la capa de salida se ajustan según

$$u_i^C(n) = u_i^C(n-1) + \alpha \delta_i^C(n) \quad (4.21)$$

para $i = 1, 2, \dots, n_C$; y en donde para (4.20) y (4.21)

$$\delta_i^C(n) = -[s_i(n) - y_i(n)] f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \quad (4.22)$$

- En segundo lugar se definen las expresiones que regirán los ajustes que se deben hacer para los parámetros en el resto de las capas ocultas; es decir, las ecuaciones que modificarán los pesos de la capa c a la capa $c+1$ y los umbrales en las neuronas de la capa $c+1$ para $c = 1, 2, \dots, C-2$. Es así que los pesos se modifican por

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_j^{c+1}(n) a_k^c(n) \quad (4.23)$$

donde $a_k^c(n)$ es la activación de la neurona k en la capa c y w_{kj}^c es el peso de la interconexión que va desde la neurona k en la capa c hacia la neurona j en la capa $c+1$, para $k = 1, 2, \dots, n_c$, $j = 1, 2, \dots, n_{c+1}$ y $c = 1, 2, \dots, C-2$.

Los umbrales se ajustan según

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n) \quad (4.24)$$

para $j = 1, 2, \dots, n_{c+1}$ y $c = 1, 2, \dots, C-2$; de donde para (4.23) y (4.24)

$$\delta_j^{c+1}(n) = f' \left(\sum_{k=1}^{n_c} w_{kj}^c a_k^c + u_j^c \right) \sum_{i=1}^{n_{c+1}} \delta_i^{c+2}(n) w_{ji}^c \quad (4.25)$$

Ahora que se ha definido la regla delta generalizada, es posible aplicarla para el entrenamiento de un perceptrón multicapa. El proceso presentado a continuación también fue tomado de Viñuela y Galván (Viñuela & Galván,

2004). Sea $\{(X(n), S(n)), n = 1, 2, \dots, N\}$ la representación del problema que se desea resolver, donde $X = [x_1(n), \dots, x_{nc}(n)]$ es el conjunto de clases o patrones de entrada de la red, $S(n) = [s_1(n), \dots, s_{nc}(n)]$ son las salidas deseadas para dichas entradas y N es el número total de clases disponibles para el entrenamiento; los pasos de conforman el algoritmo de aprendizaje del perceptrón multicapa son:

1. Se inicializan los pesos y umbrales de la red en valores arbitrarios cercanos a cero.
2. Se selecciona una clase n del conjunto de entrenamiento $(X(n), S(n))$ como entrada para la red, de tal forma esta genere una respuesta $Y(n)$ a su salida.
3. Se calcula la diferencia existente entre la salida $Y(n)$ y la salida deseada $S(n)$ mediante la ecuación (4.18).
4. Se aplica la regla delta generalizada para ajustar los parámetros de la red, siguiendo los siguientes pasos:
 - 4.1 Se calcula el parámetro δ de las neuronas en la capa de salida usando (4.22).
 - 4.2 Se calculan los valores δ de todas las neuronas en las capas ocultas usando (4.25), empezando desde la última capa oculta y retropropagando los valores δ hacia las neuronas de la capa de entrada.
 - 4.3 Con los nuevos valores δ , se modifican los pesos y umbrales de la capa de salida mediante las expresiones (4.20) y (4.21) y los parámetros del resto de capas con (3.23) y (4.24).
5. Se realizan los pasos 2, 3 y 4 para el resto de clases del conjunto de entrenamiento, con lo que se completa una repetición o ciclo de aprendizaje.
6. Se evalúa el error de entrenamiento E , dado por (4.17).
7. Se repiten los pasos 2, 3, 4, 5 y 6 hasta que se alcance un mínimo en la función E .

5 MATERIALES Y MÉTODOS

5.1 DISEÑO E IMPLEMENTACIÓN DE UN DETECTOR DE ONDAS ALFA EN TIEMPO REAL

En el presente trabajo de titulación se desarrolló un detector de ondas alfa en tiempo real basado en Matlab. Esta aplicación BCI tiene por objetivo activar una alarma cada vez que un usuario sea capaz de generar ondas alfa. El usuario con el que se desarrolló este sistema fue una mujer sana de 52 años que accedió previamente a ser parte del estudio.

5.1.1 Adquisición de las Señales Cerebrales

5.1.1.1 Amplificador EEG

La señal EEG se adquiere por medio de un amplificador OpenBCI de 32 bits, el cual está constituido principalmente por un conversor analógico-digital ADS1299 de bajo ruido para mediciones de bio-potenciales, y un microcontrolador PIC32MX250F128B compatible con la IDE de Arduino. El amplificador cuenta con 8 canales de adquisición en donde la señal se muestrea a una frecuencia de 250 Hz y se codifica utilizando 24 bits de resolución. Los datos digitalizados se envían hacia una PC a través de un módulo Bluetooth, RFduino. En la figura 17 se muestra el OpenBCI.

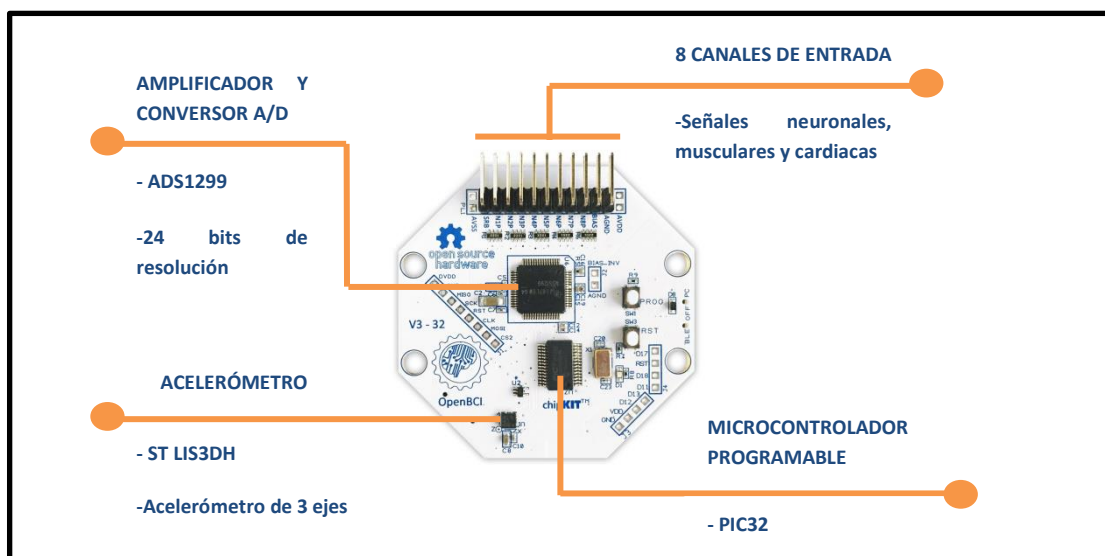


Figura 17. Amplificador de bio-potenciales OpenBCI de 32 bits.

Adaptado de: (OpenBCI Shop, 2016).

OpenBCI es un dispositivo de hardware y software libre, por lo que existe extensa documentación sobre su funcionamiento generada tanto por sus desarrolladores, como por sus usuarios. Los esquemas PCB y el firmware del microcontrolador se pueden encontrar en la página web oficial de OpenBCI para que cualquiera pueda hacer uso de ellos.

5.1.1.2 Ubicación de los Electrodoes para Obtención de EEG

Se usaron dos electrodos ubicados en la zona occipital del cráneo según el sistema 10/20, para adquirir la actividad neurológica que manejará al sistema BCI. El lóbulo de la oreja izquierda se usó como referencia y el lóbulo de la derecha como tierra. La figura 18 muestra la configuración utilizada.

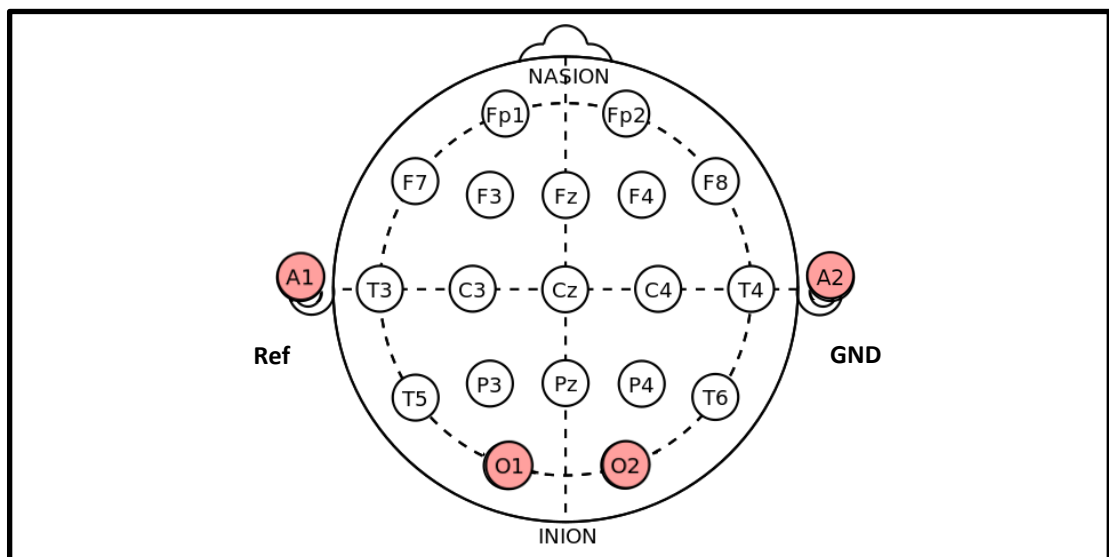


Figura 18. Ubicación de los electrodos usados para adquirir las señales del sistema BCI detector de ondas alfa.

Fuente: El autor.

La selección del área occipital para grabar los ritmos alfa se hizo en base a la literatura revisada, pues es sabido que estos ritmos se presentan con gran intensidad en este lugar (Henríquez, 2014) (Nicolas-Alonso & Gomez-Gil, 2012).

5.1.2 Paradigma de Entrenamiento

Para que el sistema propuesto sea capaz de identificar correctamente los ritmos alfa generados por un usuario, primero debe aprender correctamente cuáles son las características que definen unívocamente a este tipo de señal. Por esta razón se

diseñó un procedimiento de entrenamiento en el que se registra la actividad EEG del usuario, tanto aquella que representa actividad alfa como aquella que no, para luego ser usadas como ejemplos de aprendizaje por el sistema.

En el paradigma propuesto, el sujeto se sienta frente a una pantalla blanca que contiene una cruz en su centro. Al inicio de la prueba se presenta un indicador acústico que le informa al usuario del inicio del procedimiento y que debe mirar fijamente a la cruz en el centro de la pantalla. En el instante $t = 10s$ se genera un segundo estímulo acústico que le señala al sujeto que debe cerrar los ojos hasta la finalización de la prueba. En $t = 25s$, se presenta un último estímulo sonoro que indica el final de la prueba. La figura 19 representa gráficamente el paradigma de entrenamiento.

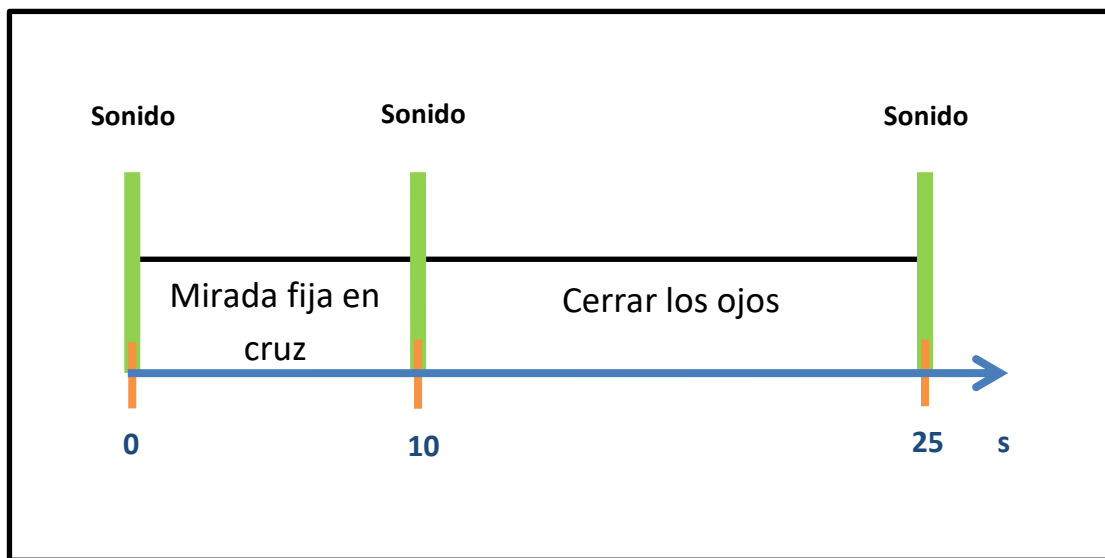


Figura 19. Descripción del paradigma de entrenamiento utilizado para la adquisición de ondas alfa.

Fuente: El autor.

Las ondas alfa se presentan con gran intensidad cuando se cierra los ojos (Henríquez, 2014) (Nicolas-Alonso & Gomez-Gil, 2012). Mientras el sujeto mantiene fija la mirada en la cruz se obtienen muestras que servirán como ejemplos de señales EEG sin presencia de ritmos alfa; por otro lado, cuando el sujeto cierra los ojos se obtienen muestras de ritmos alfa.

Estas pruebas se hicieron durante cinco días, haciendo cinco pruebas por día, dando lugar a un total de 25 pruebas. La figura 20 muestra la pantalla blanca

usada en estas sesiones, mientras que en la figura 21 se puede observar la realización de una de estas.



Figura 20. Pantalla blanca usada durante las sesiones de entrenamiento para la adquisición de ondas alfa.

Fuente: El autor.

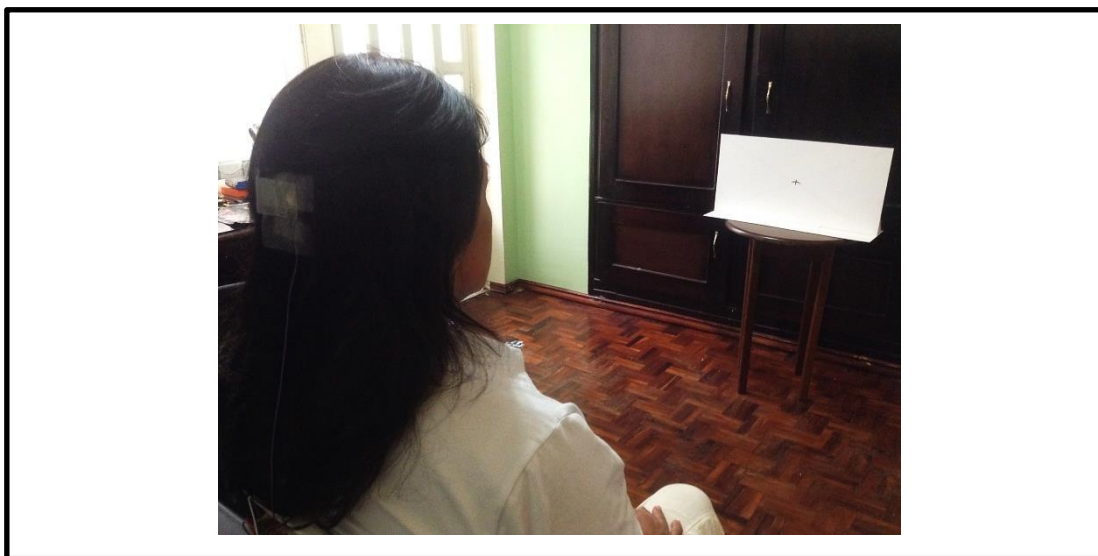


Figura 21. Desarrollo de una sesión de entrenamiento de ondas alfa.

Fuente: El autor.

Refiérase a la figura 22; aquí se puede observar la señal EEG adquirida por el electrodo O2 en una de las pruebas de entrenamiento. La presencia de los ritmos alfa se puede notar a simple vista a partir de los 10s de la prueba, instante en el que el usuario cierra los ojos.

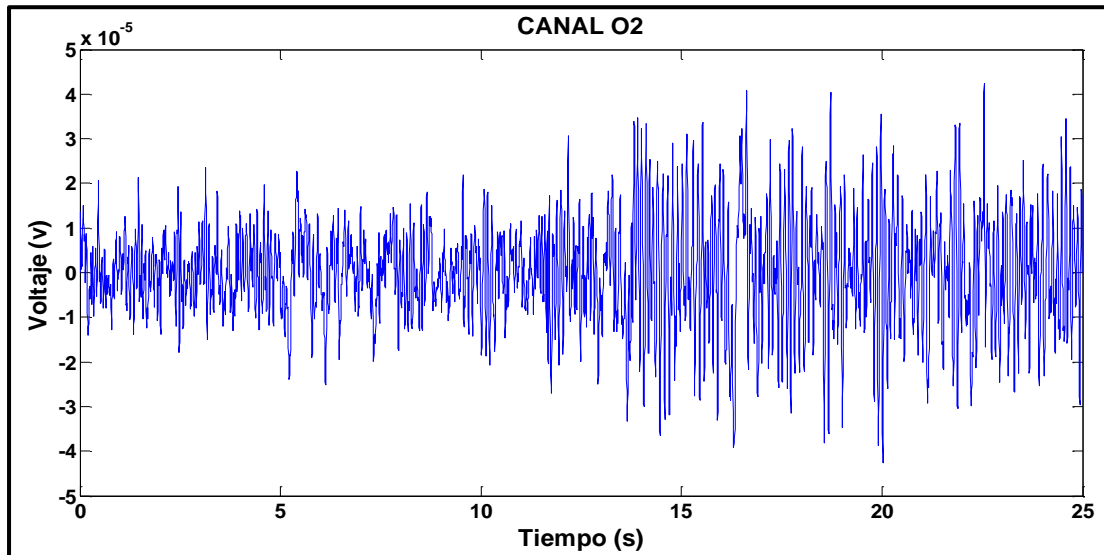


Figura 22. Señal EEG adquirida por el canal O2 durante una sesión de entrenamiento.

Fuente: El autor.

5.1.3 Preparación de las Señales

Las muestras obtenidas de las sesiones pasan por un proceso de preparación y pre-procesado para que el entrenamiento del sistema de detección de ondas alfa se pueda hacer con mayor facilidad.

5.1.3.1 Separación de Datos Útiles

En primer lugar se separa la información útil de cada prueba. El criterio de separación usado es que de cada prueba, se toma toda la parte de la grabación que corresponde a la fijación de la mirada en la cruz, mientras que de la parte que corresponde a los ojos cerrados se toma la señal que va desde 3 segundos después de que se presentó el segundo estímulo acústico hasta el final de la prueba. La figura 23 ilustra esta separación para una mejor comprensión. Este criterio de separación de datos asegura que para el entrenamiento del sistema de detección se usarán solo datos con información relevante, dejando de la ambigüedad que las señales pudieran tener en la transición existente entre el periodo de la mirada fija y el cierre de los ojos.

La señal resultante del proceso de normalización es adimensional y su amplitud se limita al intervalo que va de 0 a 1. Para la normalización se utilizó la fórmula

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (5.1)$$

de donde $X = \{x_1, x_2, \dots, x_n\}$ es un canal EEG sin normalizar, $Z = \{z_1, z_2, \dots, z_n\}$ es el canal normalizado, x_i es el i -ésimo valor del canal sin normalizar, z_i es el i -ésimo valor del canal normalizado, $\min(x)$ es el valor mínimo de x y $\max(x)$ es el valor máximo de x . En el anexo 11.2 se muestra la implementación de esta fórmula como una función de Matlab.

5.1.4 Extracción de Características

La extracción de características se realiza a través de la transformada wavelet discreta. Para esto se divide la señal de cada canal en 4 niveles de descomposición tal y como se muestra en la tabla 6. La wavelet madre utilizada es la *Daubechies* de orden 10, debido su gran desempeño en (Xu & Song, 2008). En la tabla 6 se puede observar que el rango de frecuencia del nivel de detalle D4 concuerda con la frecuencia de los ritmos alfa, por lo que únicamente los coeficientes wavelet de este nivel serán utilizados para el proceso de caracterización.

Tabla 6. Rangos de frecuencia para los diferentes niveles de descomposición obtenidos en la DWT.

Fuente: El autor.

NIVEL DE DESCOMPOSICIÓN	SUBCONJUNTO DE COEFICIENTES	RANGO DE FRECUENCIAS
1	D1	62,5 – 125 Hz
2	D2	31,25 – 62,5 Hz
3	D3	15,625 – 31,25 Hz
4	D4	7,813 – 15,625 Hz
4	A4	0 – 7,813 Hz

A partir de los coeficientes seleccionados se calcula:

- El valor absoluto medio de los coeficientes.

- La desviación estándar de los coeficientes.
- La potencia promedio de los coeficientes.

Estos parámetros representan la distribución de frecuencia y la cantidad de cambios en la distribución de frecuencia del nivel de descomposición (Xu & Song, 2008), y serán los utilizados como entradas para el sistema de clasificación. Por lo tanto, se calculan 3 características para el nivel D4 de cada uno de los 2 canales EEG, obteniéndose un total de 6 características wavelet.

5.1.5 Clasificación de Características

Las redes neuronales artificiales se han usado en varios estudios de clasificación de patrones EEG con excelentes resultados (Alomari, Awada, Samaha, & Alkamha, 2014). En la presente tesis se utilizó una red neuronal de dos capas como método de clasificación de características.

La primera capa oculta de la red neuronal se compone de 20 neuronas que utilizan la función tangente sigmoideal como función de activación. Esta función está definida por la ecuación (Dorofki, Elshafie, Jaafar, Karim, & Mastura, 2012)

$$Tansig(n) = \frac{2}{1+e^{-2n}} - 1 \quad (5.2)$$

La capa de salida se constituye de dos neuronas, representando las dos posibles clases de salida, y utilizan la función softmax como función de activación. La función softmax está dada por (Sarle, 2014)

$$p_i = \frac{e^{q_i}}{\sum_{j=1}^c q_j} \quad (5.3)$$

de donde q_i es un vector tal que $i = 1, 2, 3, \dots, c$ y p_i es la salida de la función softmax.

El entrenamiento de la red se realizó a través del algoritmo de retropropagación de gradiente conjugada escalada. Para el proceso de aprendizaje se usó el 80% de ventanas reservadas para este fin; de este mismo grupo de entrenamiento, el 90% de ventanas se usó para ajustar los pesos de la red y el 10% se utilizó para medir la capacidad de generalización de la misma.

Cuando el entrenamiento de la red finaliza, se le presenta el 20 % de las ventanas restantes para evaluar los resultados obtenidos. Para conseguir una red con la mejor capacidad de generalización, la red fue entrenada 10 veces y cada uno de estos estados fue comprobado con el grupo de evaluación. El estado que presentó los mejores resultados fue utilizado en la implementación del detector de ondas alfa.

La siguiente figura muestra la estructura de la red neuronal.

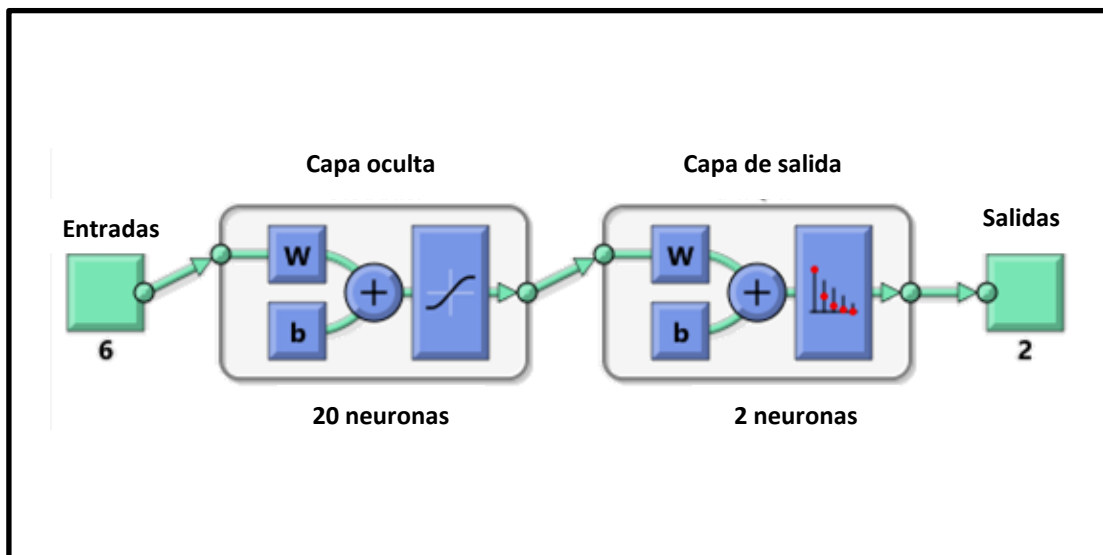


Figura 24. Red neuronal para la clasificación de ritmos alfa.

Fuente: El autor.

Cabe destacar que durante el entrenamiento Matlab normaliza por defecto las entradas de la red para que caigan dentro del rango de -1 a 1. En el desarrollo del presente trabajo, esta característica se mantuvo intacta.

5.1.6 Implementación del Detector de Ondas Alfa

El objetivo del detector de ondas alfa es que el usuario sea capaz de activar un conjunto de actuadores cuando este logre generar ondas alfa. Para su implementación se empleó la red neuronal entrenada anteriormente como sistema de clasificación. La señal EEG adquirida por el amplificador es analizada mediante un algoritmo implementado en Matlab y en base a la clase a la que pertenece la señal cerebral se envía un comando a través de Bluetooth hacia un sistema de actuadores controlado por un microcontrolador. El funcionamiento de este sistema BCI se ilustra en la figura 25.

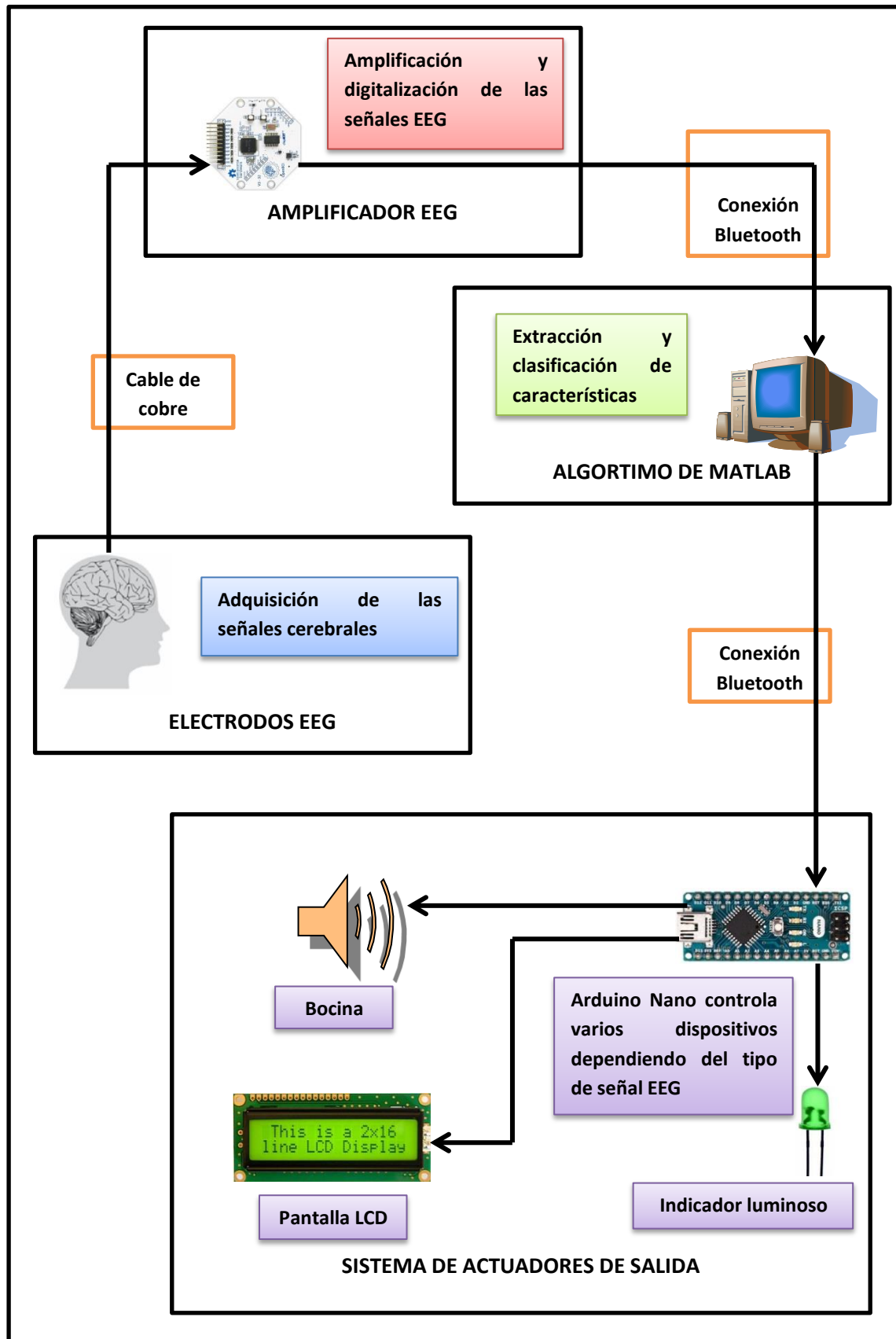


Figura 25. Descripción del funcionamiento de la BCI implementada.

Fuente: El autor.

El algoritmo implementado para la clasificación y extracción de características de las ondas alfa se describe a continuación:

1. Se adquiere 1 segundo de señal EEG a partir de los electrodos O1 y O2 del sistema internacional 10/20.
2. La ventana de tiempo adquirida se filtra a través de un filtro pasabanda para eliminar posibles señales de interferencia tales como el ruido de sinusoidal de 60 Hz generado por las líneas eléctricas.
3. A continuación se normaliza la amplitud de la señal cerebral a través de la fórmula 4.1.
4. Seguidamente se inicia con la extracción de características de la ventana de tiempo. Para esto, la señal de cada canal se descompone en 4 niveles de detalle wavelet usando la DWT.
5. Luego se calcula el valor absoluto medio, la desviación estándar y la potencia promedio de los coeficientes pertenecientes al nivel de detalle 4.
6. Las características wavelet se combinan en un único vector que se utilizará como entrada para la red neuronal. El vector estará compuesto por 6 elementos; 3 características wavelet obtenidas por canal.
7. El vector de características se ingresa a la red neuronal, la cual determinará si la señal captada son ondas alfa o si son otro tipo de actividad EEG.
8. Se presenta la salida de la red neuronal para que el sistema actuador pueda realizar la acción pertinente dependiendo del resultado de la clasificación.

Los pasos de este algoritmo se pueden observar en forma de diagrama de flujo en la figura 26.

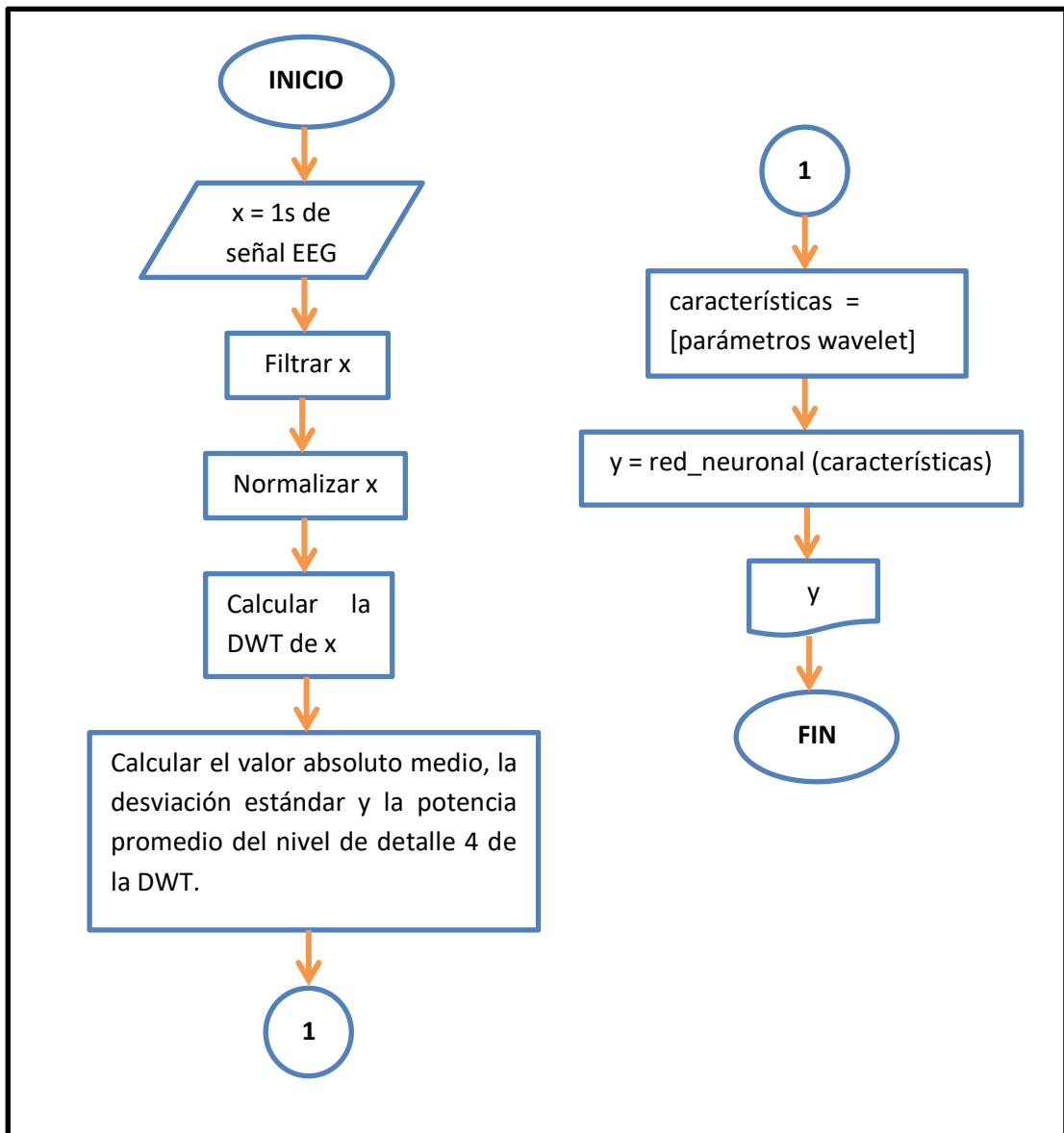


Figura 26. Algoritmo para la extracción de características y clasificación de ondas alfa.

Fuente: El autor.

Un sistema actuador, controlado por un microcontrolador, maneja una bocina, una pantalla LCD y un indicador luminoso para señalar la presencia de ondas alfa. El microcontrolador se comunica con la computadora mediante una conexión Bluetooth. El algoritmo bajo el cual funciona este sistema se describe a continuación:

1. Se lee un comando enviado por el programa de extracción de características y clasificación del puerto de comunicación Bluetooth.
2. Si este comando indica que se han detectado ondas alfa:
 - 2.1. Encender el indicador acústico.

- 2.2. Encender el indicador luminoso.
- 2.3. Presentar en la pantalla LCD “Se han detectado ondas alfa”.
3. De lo contrario, si el comando no indica la presencia de ondas alfa:
 - 3.1. Apagar el indicador acústico.
 - 3.2. Apagar el indicador luminoso.
 - 3.3. Presentar en la pantalla LCD “Escaneando señales cerebrales”.

El diagrama de flujo de este programa se muestra a continuación.

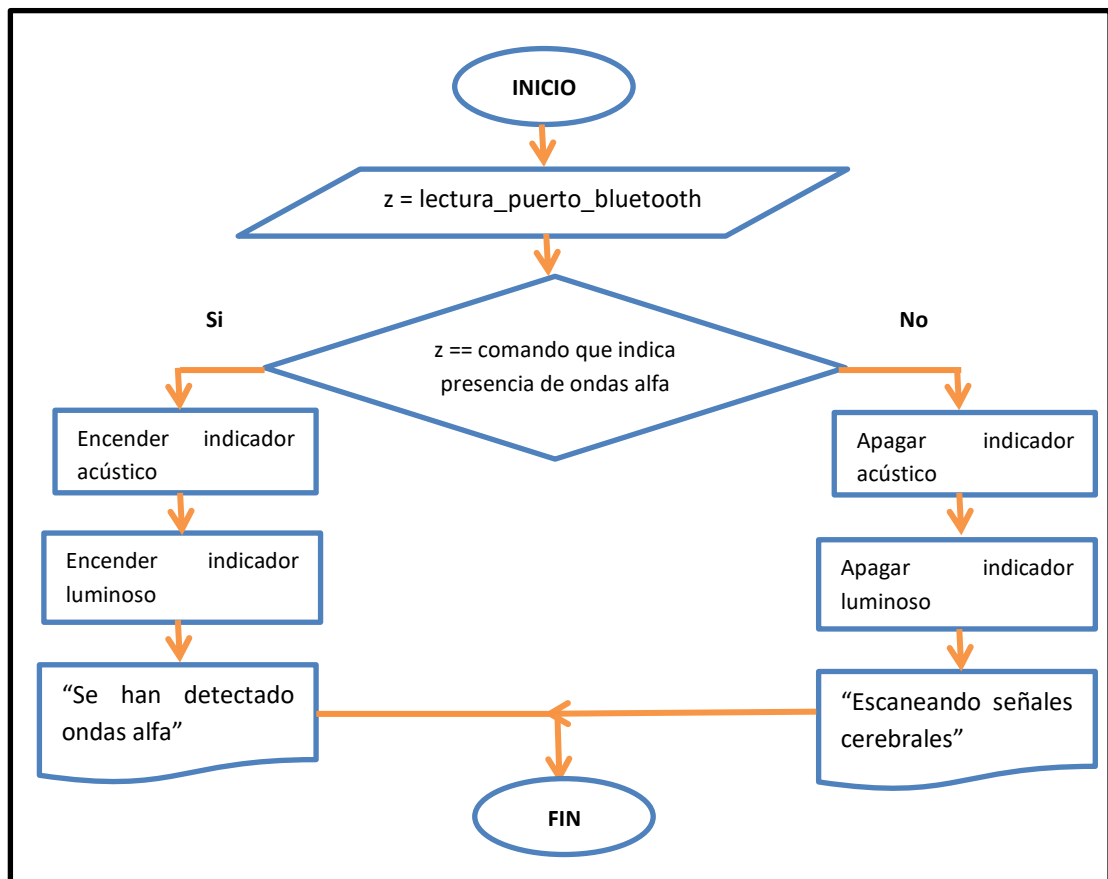


Figura 27. Diagrama de flujo del programa ejecutado por el sistema actuador.

Fuente: El autor.

Para una mejor facilidad de uso, el algoritmo de extracción de características y clasificación se implementó en forma de interfaz gráfica de usuario (GUI o *graphic user interface* en inglés), la cual se controla a través de cuatro botones:

- Un botón para inicializar la conexión serial con el amplificador EEG.
- Un botón para inicializar la conexión Bluetooth con el sistema actuador.
- Además existe un botón para probar el estado de la conexión Bluetooth.

- Y un botón de inicio para inicializar la BCI.

Además, la GUI también grafica en tiempo real las señales EEG que se obtienen del usuario. El diseño de la GUI se puede observar en la figura 28.

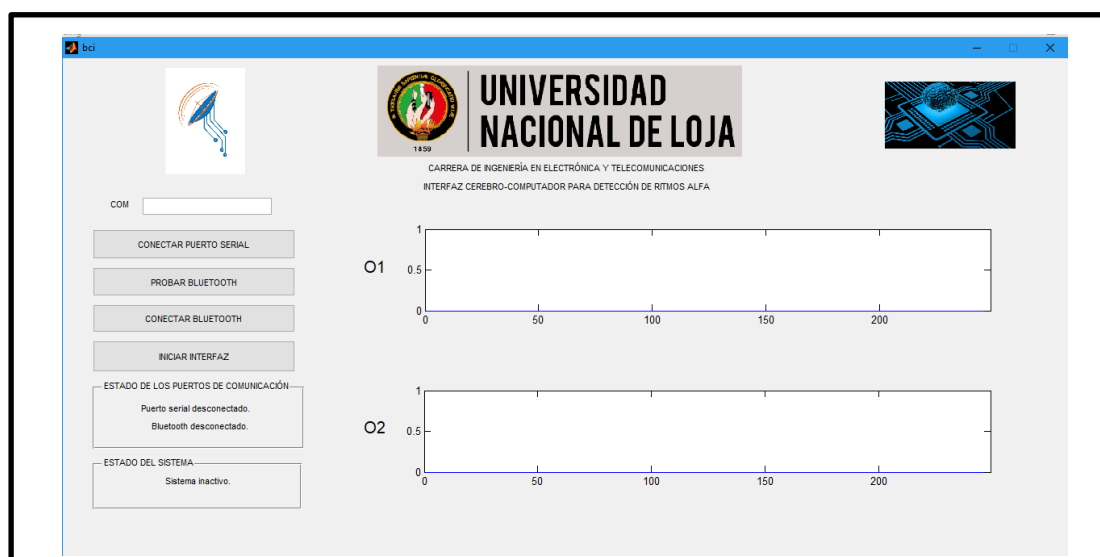


Figura 28. GUI del detector de ondas alfa en tiempo real.

Fuente: El autor.

El sistema actuador se implementó mediante una placa Arduino Nano con microcontrolador ATmega328. Cuando recibe una letra “o” en formato ASCII, interpreta que se han detectado ritmos alfa y realiza las acciones descritas en el algoritmo anterior, mientras que si recibe una “s” entiende que la señal EEG no representa actividad alfa y realiza las acciones debidas. La GUI envía estos caracteres al sistema actuador una vez que determina la clase a la que pertenece una ventana temporal.

En la figura 29 se ilustra un ejemplo del funcionamiento del sistema actuador.



a)



b)

Figura 29. a) Mientras no se detectan ondas alfa el sistema actuador indica que se continúa escaneando la actividad mental del usuario. b) Al detectar ondas alfa suena una alarma, se enciende un indicador luminoso y se informa de esto mediante una pantalla LCD.

Fuente: El autor.

5.2 EXTRACCIÓN DE CARACTERÍSTICAS Y CLASIFICACIÓN DE RITMOS SENSORIOMOTORES

Dentro de la presente tesis también se realizó el estudio de diferentes métodos de extracción de características para la clasificación de ritmos sensoriomotores, con la finalidad de identificar una metodología que presente un nivel de aciertos aceptable para su posible aplicación en sistemas BCI. En este trabajo se buscó una metodología

que permita discriminar entre imaginación motora de mano izquierda y de mano derecha.

5.2.1 Señales Cerebrales a Analizar

Para el análisis de las metodologías propuestas se utilizaron dos bases de datos de grabaciones EEG proporcionados por la *Graz University of Technology* para las competencias BCI III y IV, respectivamente. El objetivo de estas competencias es la de validar técnicas de procesamiento de señales y clasificación para BCI (sitio web de BCI Competition III, 2017) (sitio web de BCI Competition IV, 2017).

5.2.1.1 Base de Datos IIIA de la Competencia BCI III

Esta base de datos contiene grabaciones BCI de tres personas, a las cuales se les pidió realizar cuatro tipos de imaginación motora: movimiento de mano izquierda, mano derecha, pies y lengua. Existen 60 grabaciones para cada clase (Schlögl, 2017).

En cada grabación se registraron 60 canales EEG según las posiciones mostradas en la figura 30. Se utilizó el mastoides derecho como referencia para las señales y el mastoides izquierdo como tierra. Las señales cerebrales se digitalizaron con una frecuencia de muestreo de 250 Hz y fueron filtradas con un filtro pasabanda de 1 a 50 Hz (Schlögl, 2017).

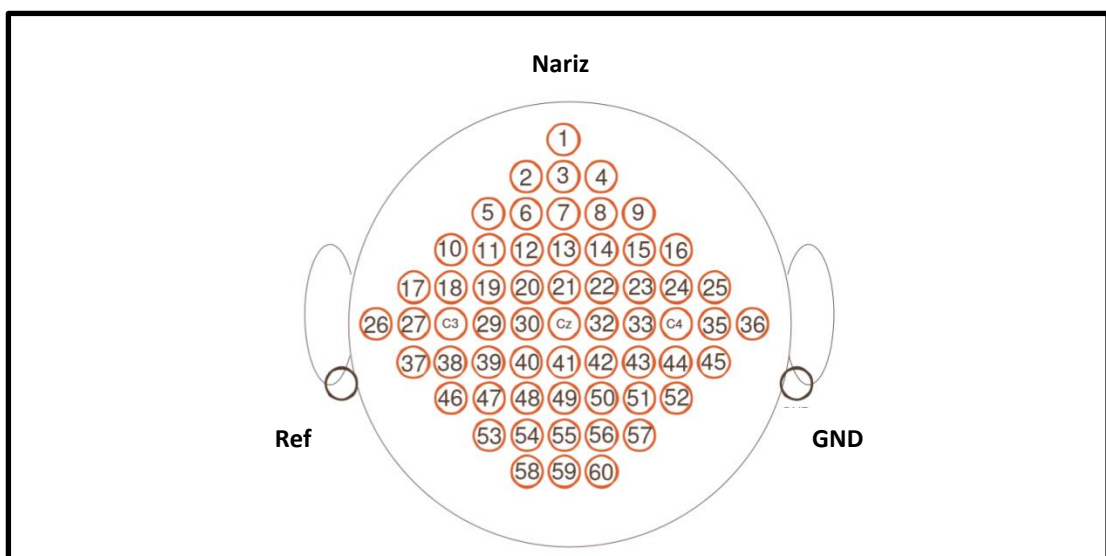


Figura 30. Posición de los electrodos EEG usados para registrar la base de datos IIIA.

Adaptado de: (Schlögl, 2017).

Refiérase a la figura 31; el paradigma utilizado para la obtención de las señales EEG se describe a continuación: El sujeto se sienta frente a una pantalla en una silla cómoda con descanso para brazos. En cada prueba se le pide al usuario realizar imaginación motora de mano derecha, izquierda, pies o lengua dependiendo de una señal visual. En el inicio de la grabación, la pantalla se muestra oscura durante los dos primeros segundos; en el instante $t = 2s$ aparece un estímulo acústico que indica el inicio de la prueba y se muestra una cruz en la pantalla. Entonces, a partir de $t = 3s$ se muestra una flecha apuntando hacia la izquierda, derecha, arriba o abajo y en ese instante el sujeto debe imaginar mover su mano izquierda, derecha, pies o lengua respectivamente. La flecha desaparece después de un segundo pero el sujeto debe mantener su actividad mental hasta que la cruz desaparezca en $t = 7s$ (Schlögl, 2017).

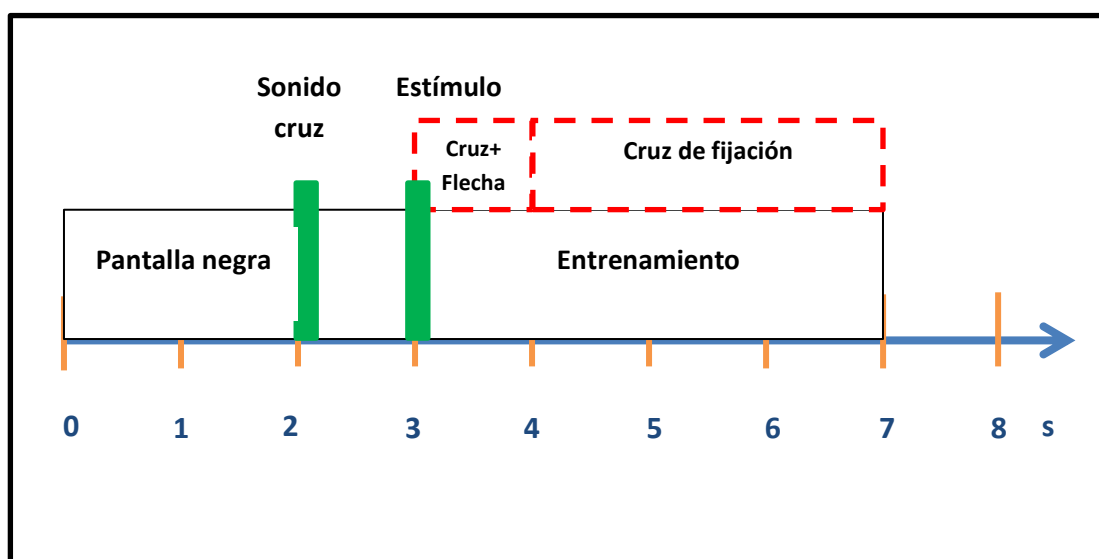


Figura 31. Descripción del paradigma utilizado para la adquisición de las señales EEG de la base de datos IIIA.

Adaptado de: (Schlögl, 2017).

5.2.1.2 Base de Datos 2A de la Competencia BCI IV

Esta base de datos está conformada por las grabaciones EEG de 9 personas y contiene la información de 4 tareas de imaginación motora distintas; estas son imaginación de movimiento de mano izquierda, de mano derecha, pies y lengua.

Dos sesiones en días diferentes se registraron para cada sujeto. Cada sesión se compone de 6 pruebas separadas por pausas breves. Una prueba consiste de 48 grabaciones (12 por cada una de las 4 clases posibles), de donde se obtiene un total de 288 grabaciones por sesión (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017, pág. 1).

La estructura de una sesión se puede visualizar en la figura 32; al iniciar cada sesión se realiza una grabación para estimar el efecto de los artefactos oculares en las señales EEG a través de EOG (electrooculografía); esta grabación se divide en tres etapas (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017):

- La primera se realiza con los ojos abiertos y tiene una duración de dos minutos.
- La segunda se realiza con los ojos cerrados y tiene una duración de un minuto.
- En la tercera se hace una grabación donde se mueven los ojos y tiene una duración de un minuto.

Después de esta grabación inician las pruebas del sujeto.

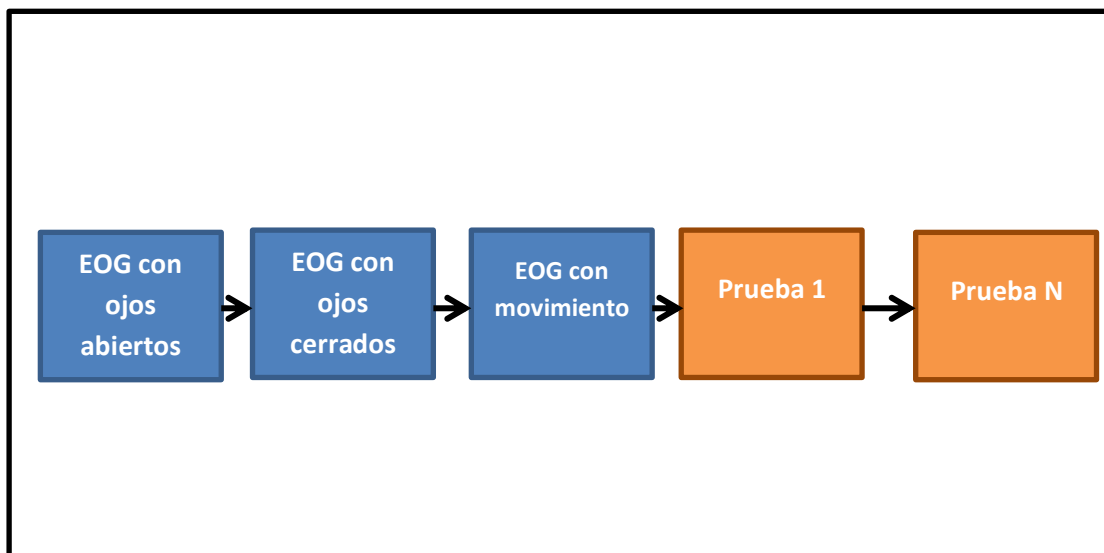


Figura 32. Estructura de una sesión.

Adaptado de: (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017).

Para la adquisición de la información motora, se sienta al sujeto en una silla cómoda con descanso para brazos en frente de una pantalla. Al iniciar la grabación se muestra una cruz sobre el fondo negro, acompañado por un corto indicador acústico. En el instante $t = 2s$ aparece una flecha apuntando hacia la izquierda, derecha, arriba o abajo en correspondencia con las imaginерías motoras de mano izquierda, derecha, pies y lengua; la duración de este estímulo visual es de 1.25s. Al usuario se le pide mantener la tarea mental hasta que la cruz desaparezca de la pantalla en $t = 6s$. Para finalizar existe un pequeño periodo de descanso donde la pantalla se pone oscura de nuevo (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017). La estructura de estas grabaciones se muestra en la figura 33.

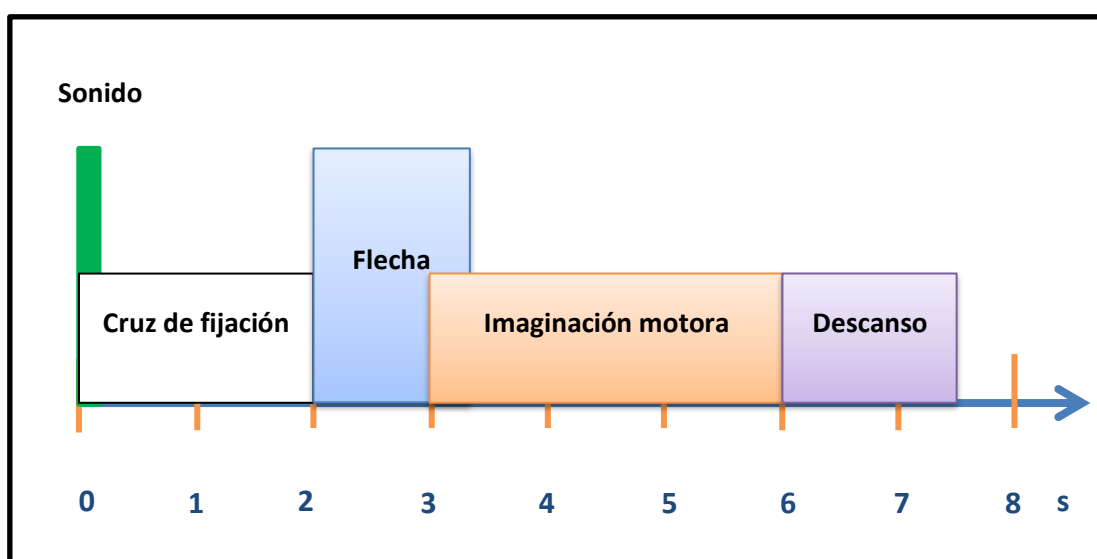


Figura 33. Estructura de una grabación.

Adaptado de: (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017).

Para la adquisición de las señales EEG se utilizaron 22 electrodos colocados en el cuero cabelludo utilizando el estándar internacional 10-20. El mastoide izquierdo se utilizó como punto de referencia y el mastoide derecho como tierra. Las señales se digitalizaron con una frecuencia de muestreo de 250 Hz y se filtraron con un filtro pasabanda con un ancho de banda de 0.5 a 100 Hz. Adicional a los canales EEG, también se registraron tres canales EOG con la misma frecuencia de muestreo y procesados con el mismo filtro, mediante tres

electrodos colocados en el rostro (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017). El montaje de los electrodos EEG y EOG se muestra en la figura 34.

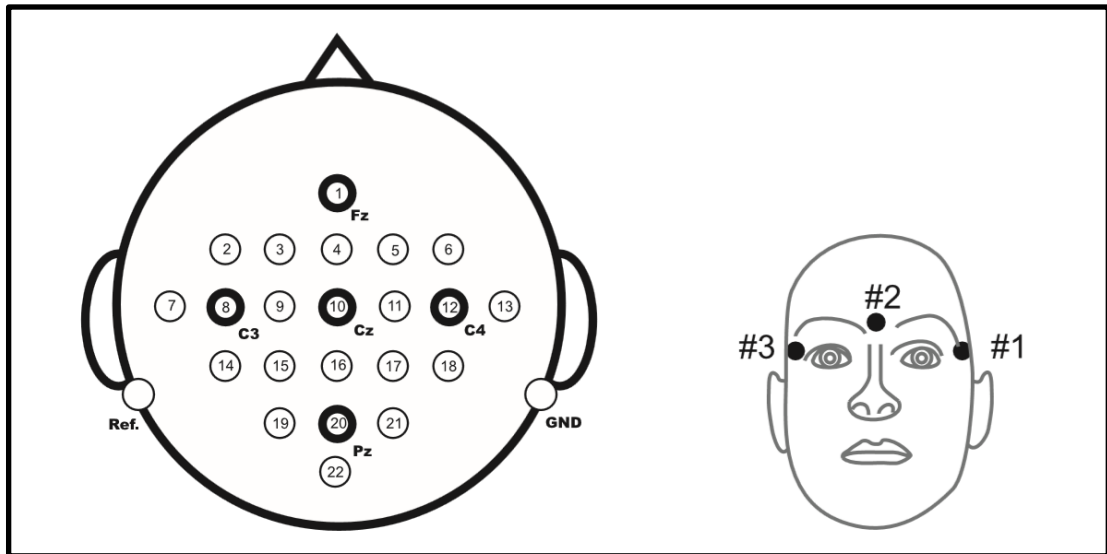


Figura 34. Izquierda: Posición de los electrodos para el registro de EEG. Derecha: Posición de electrodos para el registro de EOG.

Adaptado de: (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2017).

5.2.2 Preparación de las Señales

Tanto la base de datos IIIA, como la base de datos 2A fueron recopiladas a través de varios canales utilizando múltiples electrodos colocados en distintas regiones del cuero cabelludo; sin embargo se ha demostrado que la información obtenida a través de múltiples canales EEG es redundante, y en el caso específico de los ritmos sensoriomotores, la información que se correlaciona más fuertemente con la actividad motora de los puños se concentra principalmente en los electrodos C3, Cz y C4 del sistema internacional 10/20 (Alomari, Awada, Samaha, & Alkamha, 2014). Es por esta razón que para el presente trabajo se optó por trabajar únicamente con los canales EEG: C3, Cz, y C4. La figura 35 muestra la posición de los electrodos seleccionados según el estándar 10/20.

La reducción en el número de canales a analizar también conlleva ahorro en el uso de recursos computacionales, además de una reducción en el tiempo de procesamiento de las señales.

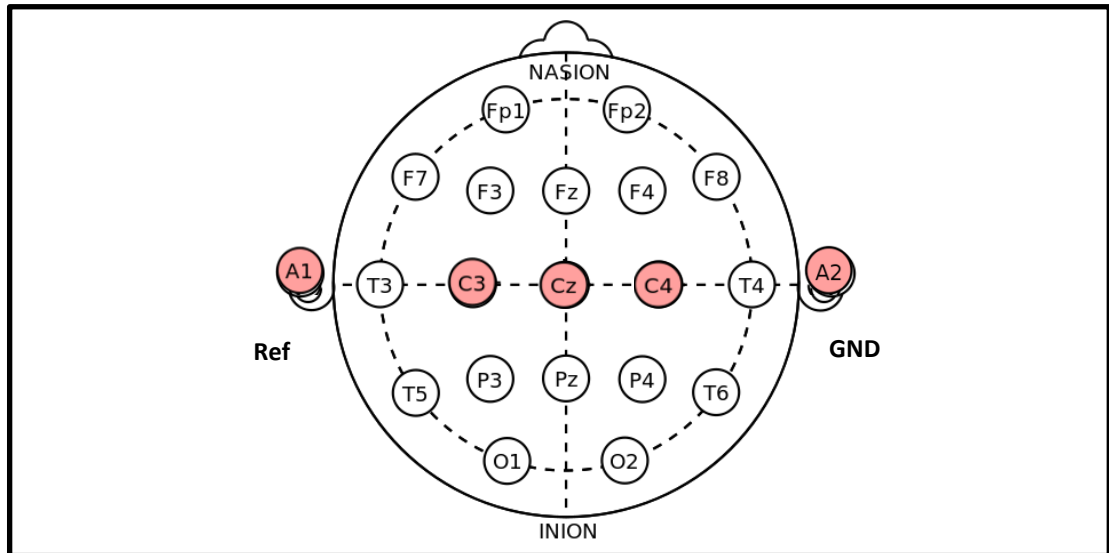


Figura 35. Electrodo seleccionados para la clasificación de imaginación motora.

Adaptado de: (Wikimedia Commons, 2015).

Las dos bases de datos utilizadas se basan en un paradigma de adquisición de señales EEG en el que el usuario debe realizar la tarea mental correspondiente después de la aparición de un estímulo visual. Para la aplicación y evaluación de las técnicas de extracción y clasificación de características utilizadas en el presente trabajo, primero se separaron las porciones de las grabaciones EEG que contienen información relevante sobre la imaginación motora del resto de la grabación. Para esto, de todas las grabaciones se separó la porción de la señal que va desde un segundo después de que se haya presentado el estímulo visual al usuario, hasta el momento en el que el usuario deja de realizar la actividad mental; esto asegura que el análisis de las señales se realice únicamente sobre los periodos que contienen la imaginación motora que se busca clasificar. En la figura 36 se muestra la porción de la señal que se selecciona para el análisis en cada base de datos.

Con los datos seleccionados se construyen dos matrices por usuario, de tal manera que una de las matrices contenga únicamente información correspondiente a imaginación motora de mano izquierda y la otra información de imaginación motora de mano derecha. Las filas de las matrices corresponden a cada muestra de las grabaciones EEG, mientras que las columnas corresponden a los electrodos seleccionados.

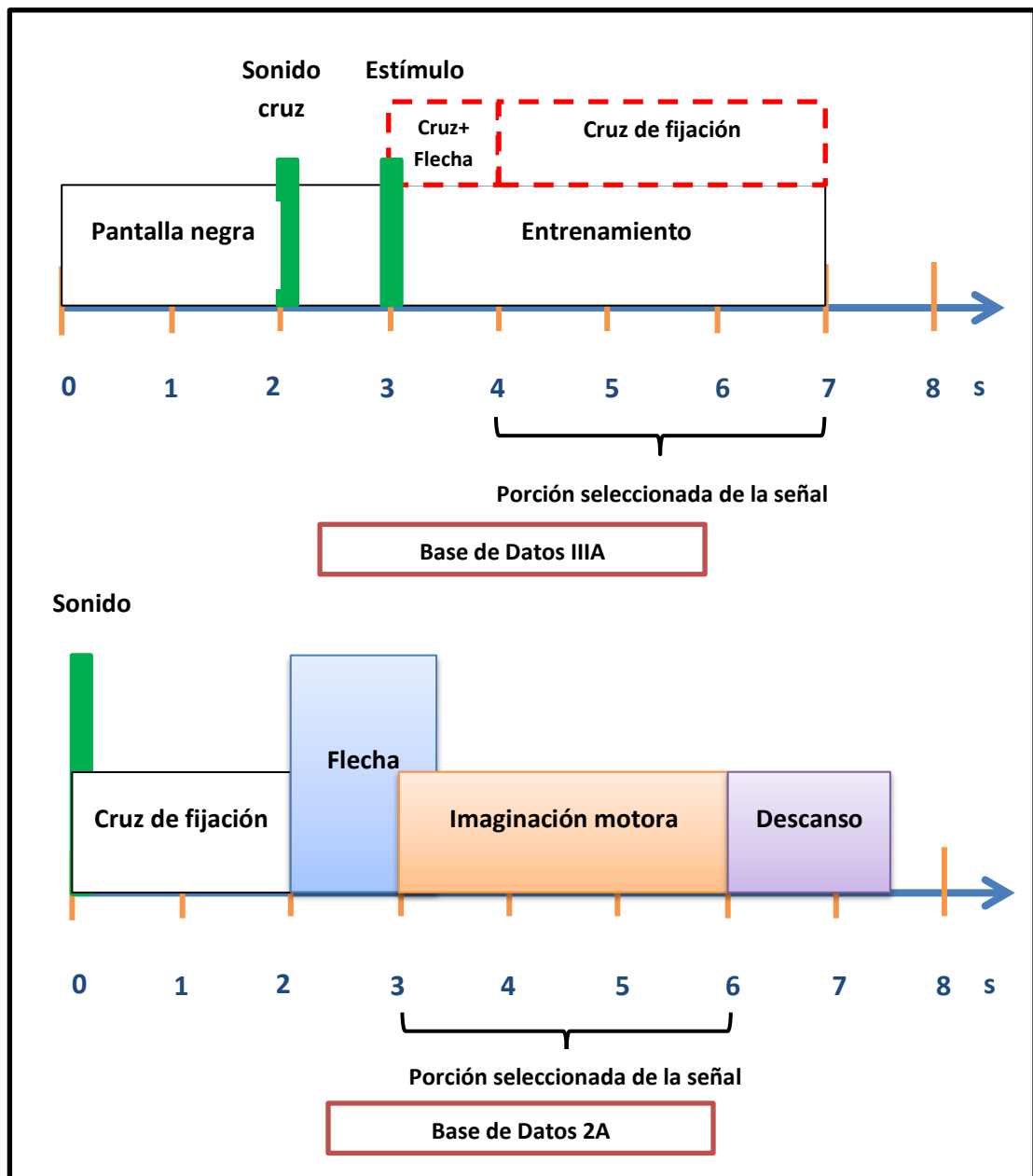


Figura 36. Periodos seleccionados de cada grabación EEG para su análisis.

Fuente: El autor.

Algunas de las grabaciones contienen artefactos oculares que corrompen la información neuronal, haciendo que sea imposible la clasificación de las partes afectadas; por esta razón todas las grabaciones que contengan artefactos oculares fueron descartadas de las matrices.

El proceso de extracción y clasificación de características se realizó sobre ventanas temporales con un tamaño de 250 muestras. Debido a que la frecuencia de muestreo en las dos bases de datos es de 250 Hz, las ventanas tienen un tamaño

equivalente a 1s de grabación. Las matrices de cada usuario se dividen en estas ventanas y el 80% de las ventanas de imaginación de cada mano se utilizan para el entrenamiento del sistema de caracterización y clasificación; el 20% de las ventanas restantes se usarán como datos de validación para el sistema propuesto.

5.2.3 Extracción de Características

5.2.3.1 Método 1

La primera metodología de extracción de características propuesta inicia normalizando la amplitud de las ventanas de entrenamiento mediante la fórmula 5.1.

A continuación se calcula la DWT de las señales usando 4 niveles de descomposición según como se indica en la tabla 6. En esta tabla se puede observar que los rangos de frecuencia de los niveles de detalle D3 y D4 concuerdan con el rango de frecuencia de los ritmos sensoriomotores β y μ respectivamente, por lo que únicamente los coeficientes wavelet de estos niveles serán utilizados para el proceso de caracterización.

Al igual que en el proceso de caracterización de ritmos alfa, con los coeficientes wavelet seleccionados se calcula:

- El valor absoluto medio de los coeficientes.
- La desviación estándar de los coeficientes.
- La potencia promedio de los coeficientes.

Estos parámetros serán las características utilizadas para la identificación de las clases de imaginación motora. Estas tres características se calculan para cada uno de los niveles D3 y D4 de cada canal, lo que da lugar un vector de características total de 18 elementos. En la figura 37 se ilustra este proceso de extracción de características para un canal EEG.

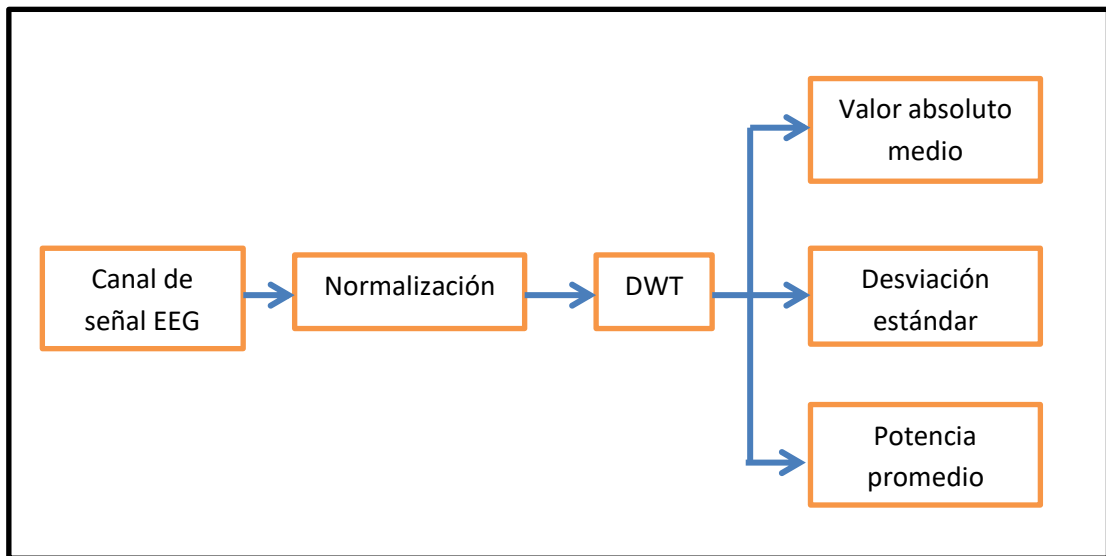


Figura 37. Descripción del primer método de extracción de características.

Fuente: El autor.

5.2.3.2 Método 2

Este inicia con la normalización de la ventana EEG mediante la ecuación 5.1. A continuación se extraen las características que son de dos tipos: características wavelet y coeficientes AR.

Los coeficientes wavelet se obtienen aplicando la DWT a los canales normalizados y calculando el valor absoluto medio, la desviación estándar y la potencia promedio de los coeficientes de los niveles de detalle D3 y D4. Este procedimiento da lugar a 18 características wavelet.

De cada canal EEG normalizado también se calculan los coeficientes AR que serán usados como características de clasificación. El modelo AR utilizado es de orden 6 y los coeficientes se estiman a través el método de Burg, tal y como es propuesto por Xu y Song (Xu & Song, 2008). Esto da lugar a 6 coeficientes por canal, obteniéndose un total de 18 coeficientes AR.

El vector de características final que se utilizará para la clasificación de los ritmos sensoriomotores tendrá una longitud de 36 elementos, fruto de la combinación de los parámetros wavelet y los coeficientes AR. La figura 37 describe en forma de diagrama de bloques este proceso de extracción de características para uno de los canales EEG.

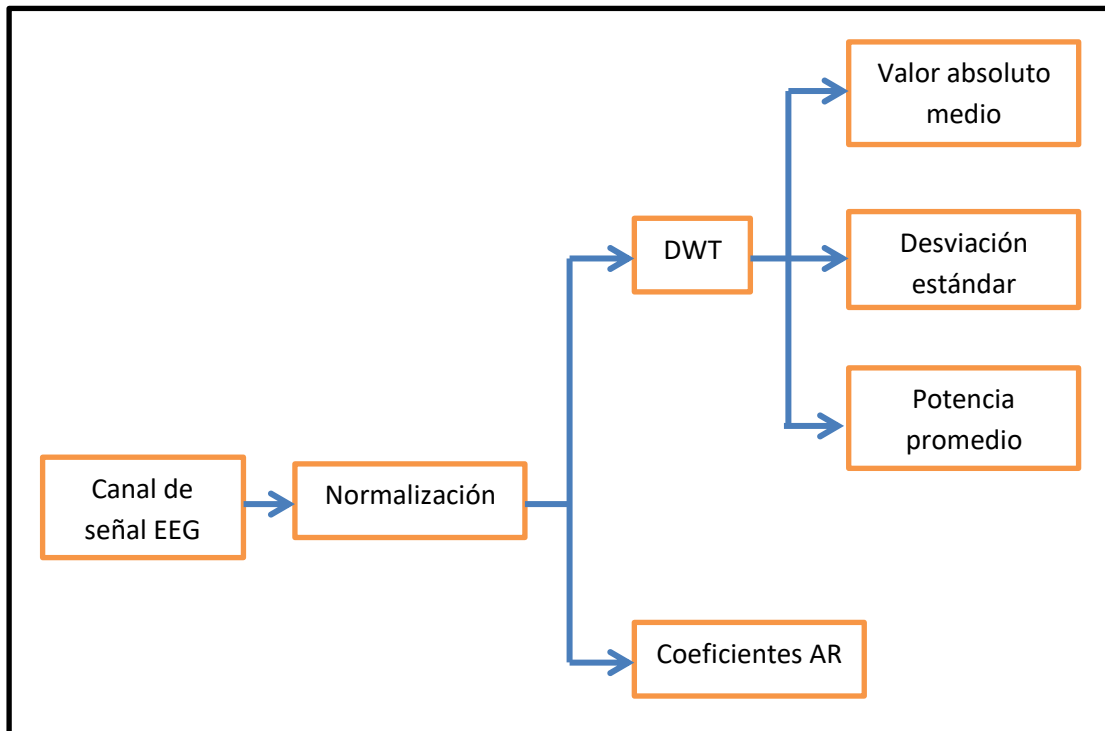


Figura 38. Descripción del segundo método de extracción de características.

Fuente: El autor.

5.2.3.3 Método 3

Primero se normaliza la ventana EEG a analizar mediante la fórmula 5.1. A continuación se filtra espacialmente la señal normalizada mediante CSP para mejorar la capacidad de discriminación entre las clases de imaginación motora .

El filtro espacial W se obtiene resolviendo el problema de autovectores generalizado propuesto en la ecuación (4.5) y se lo aplica a los datos normalizados mediante una simple multiplicación de matrices según (4.1).

A continuación se calculan las características que serán de tres tipos: características wavelet, coeficientes AR y varianza de la señal.

El procedimiento para calcular las características wavelet es el mismo que el descrito en los dos métodos anteriores, dando lugar un total de 18 características.

Para calcular los coeficientes AR de cada canal, se aplicó un modelo AR de orden 6 mediante el método de Burg; este proceso da lugar a 18 características.

La utilización de la varianza de las señales como características de clasificación se decidió porque la aplicación de CSP da como resultado la maximización de la diferencia entre la varianza de las clases (Olías, 2016). El proceso de extracción de características de varianza consiste en hacer pasar la señal normalizada por un banco de dos filtros pasabanda cuyos rangos de frecuencia van de 8 a 12 Hz y de 13 a 30 Hz respectivamente; estas bandas de paso se corresponden con los rangos de operación de los ritmos μ y β . Seguidamente se calcula la varianza de la salida de cada filtro, siendo estos valores los que se usarán como características para la clasificación. Como se calculan dos varianzas por canal se obtiene un total de 6 características.

Las características obtenidas se agrupan en un solo vector que tendrá una longitud total de 42 elementos. Una descripción en forma de diagrama de bloques de este proceso de extracción de características para un canal se muestra en la figura 39.

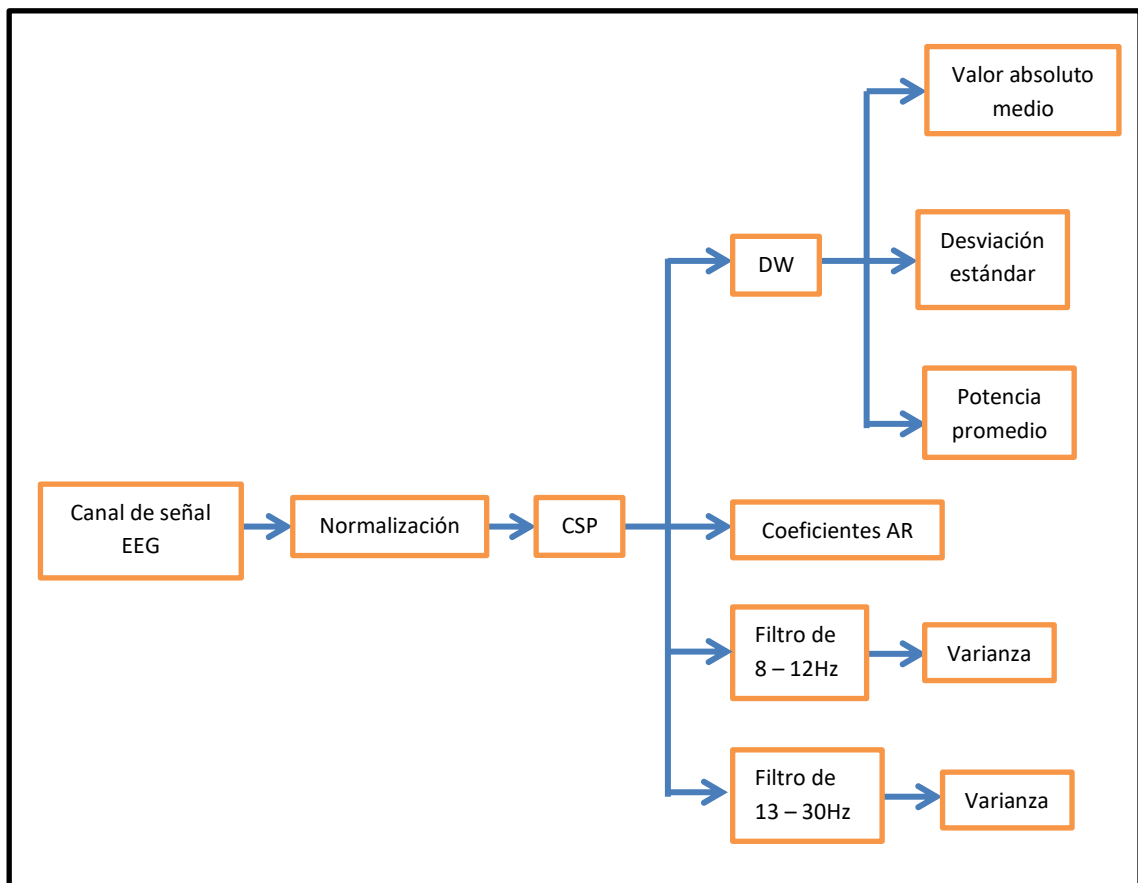


Figura 39. Descripción del tercer método de extracción de características.

Fuente: El autor.

5.2.4 Clasificación de Características

La clasificación de las características obtenidas con los métodos descritos anteriormente se realizó mediante una red neuronal de dos capas, la cual tiene la misma estructura que la usada para la clasificación de las ondas alfa, siendo la única diferencia el número de entradas. Esta red se puede observar en la figura 23.

Durante el entrenamiento de la red se usó el 80% de ventanas reservadas para cada sujeto como muestras de aprendizaje. De este mismo conjunto de aprendizaje, el 90% se usó para ajustar los pesos de la red, mientras que el 10% restante sirvió para medir la capacidad de generalización de la red durante el entrenamiento.

Con la finalidad de obtener una red con las mejores prestaciones se siguió el mismo procedimiento usado en la clasificación de ritmos alfa, en donde se realizaron 10 entrenamientos distintos con el 80% de las ventanas, cada uno de los cuales fue evaluado con el 20% de ventanas que conforman el grupo de evaluación. La red con los mejores resultados al clasificar el grupo de validación fue seleccionada. Este procedimiento se siguió para cada uno de los métodos de extracción de características.

6 RESULTADOS

6.1 RESULTADOS DE LA CLASIFICACIÓN DE ONDAS ALFA

Como se mencionó anteriormente, para el análisis de las señales EEG obtenidas mediante el paradigma de entrenamiento de ondas alfa, estas fueron divididas en ventanas de 125 muestras. El 80% de las ventanas se usó como ejemplos de aprendizaje, mientras que con el 20% restante se evaluó la red neuronal entrenada.

Al entrenar redes neuronales en Matlab, es necesario dividir en tres grupos los datos que se usan para el aprendizaje; el primer grupo se usa para ajustar los pesos de la red, mientras que los otros dos sirven para medir su capacidad de generalización. Es por esto que del 80% de ventanas usadas para el aprendizaje, el 90% de estas se usó para ajustar los pesos de la red mientras que el 10% restante se usó para medir la capacidad de generalización (5% en cada uno de los dos grupos). Después del entrenamiento se evaluó la red con el 20% de ventanas reservadas para esto.

Cada proceso de entrenamiento en Matlab puede llevar a resultados diferentes debido a variaciones en las condiciones iniciales de la red. Es por esta razón que con los mismos datos de aprendizaje se realizaron 10 procesos de entrenamiento diferentes, cada uno de los cuales fue analizado con el mismo grupo de evaluación; todo esto con la finalidad de obtener la red neuronal con la mejor capacidad de clasificación. El porcentaje de aciertos que cada estado de entrenamiento consigue al clasificar el grupo de evaluación se muestra a continuación.

Tabla 7. Resultados de la clasificación del grupo de evaluación obtenidos por cada estado de entrenamiento.

Fuente: El autor.

USUARIO	PORCENTAJE DE ACIERTOS DE CADA ESTADO DE ENTRENAMIENTO										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
Mujer	88,2%	88,2%	82,7%	88,2%	86,4%	88,2%	89,1%	86,4%	89,1%	88,2%	89,1%

La red con los mejores resultados se selecciona para su utilización en el detector de ondas alfa en tiempo real. En este caso, como el mejor resultado se repite dos veces se escoge cualquiera de estos indistintamente.

6.2 RESULTADOS DE LA CLASIFICACIÓN DE RITMOS SENSORIOMOTORES

6.2.1 Resultados del Método 1

El procedimiento para el análisis de las técnicas de extracción de características aplicadas a los ritmos sensoriomotores es similar al utilizado para las ondas alfa. Para cada usuario de las bases de datos se crean dos matrices que contienen la información de imaginación motora de mano izquierda y de mano derecha de la persona respectivamente. Las señales EEG de las dos matrices se dividieron en ventanas temporales de 125 muestras. El 80% de las ventanas de cada una de las dos matrices se combina en una sola matriz para ser utilizada como ejemplo de aprendizaje, mientras que el 20% restante de cada matriz se combina para ser usadas para evaluar la red neuronal entrenada. De la misma manera, se hicieron 10 entrenamientos distintos con los mismos datos para obtener una red neuronal con los mejores resultados de clasificación.

La tabla 8 muestra los resultados obtenidos mediante el primer método de extracción de características al clasificar el grupo IIIA y la tabla 9 los obtenidos para el grupo 2A.

Tabla 8. Resultados de clasificación de imaginación motora obtenidos por el primer método de extracción de características para el grupo IIIA.

Fuente: El autor

USUARIO	ACIERTOS OBTENIDOS PARA EL GRUPO IIIA POR EL PRIMER MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
SUJETO 1	60%	51,1%	64,4%	64,4%	64,4%	53,3%	46,7%	48,9%	57,8%	64,4%	64,4%
SUJETO 2	48,3%	55,2%	55,2%	51,7%	58,6%	58,6%	65,5%	58,6%	51,7%	41,4%	65,5%
SUJETO 3	48%	52%	52%	68%	44%	60%	52%	72%	64%	48%	72%

Tabla 9. Resultados de clasificación de imaginaria motora obtenidos por el primer método de extracción de características para el grupo 2A.

Fuente: El autor.

USUARIO	ACIERTOS OBTENIDOS PARA EL GRUPO 2A POR EL PRIMER MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
SUJETO 1	48,8%	59,5%	46,4%	50%	59,5%	50%	50%	53,6%	41,7%	45,2%	59,5%
SUJETO 2	34,9%	42,2%	36,1%	53%	44,6%	51,8%	51,8%	51,8%	41%	47%	53%
SUJETO 3	66,3%	60,2%	69,9%	62,7%	62,7%	63,9%	56,6%	67,5%	66,3%	65,1%	69,9%
SUJETO 4	50,6%	46,8%	50,6%	51,9%	53,2%	45,6%	50,6%	54,4%	46,8%	45,6%	54,4%
SUJETO 5	51,3%	42,3%	42,3%	37,2%	44,9%	53,8%	50%	38,5%	53,8%	47,4%	53,8%
SUJETO 6	52,2%	55,1%	49,3%	47,8%	52,2%	56,5%	49,3%	47,8%	49,3%	46,4%	56,5%
SUJETO 7	55,6%	50,6%	56,8%	50,6%	44,4%	46,9%	43,2%	42%	53,1%	49,4%	56,8%
SUJETO 8	62,5%	56,3%	58,8%	61,3%	62,5%	58,8%	57,5%	57,5%	56,3%	48,8%	62,5%
SUJETO 9	58,6%	58,6%	54,3%	57,1%	38,6%	61,4%	54,3%	54,3%	54,3%	50%	61,4%

El mejor resultado se toma para cada usuario y se asume como el estado de entrenamiento que posee la mejor capacidad de generalización.

Para tener una visión más general de la exactitud de la primera metodología de extracción de características, se calcula el promedio de los mejores resultados de clasificación. Este cálculo se muestra en la figura 40.

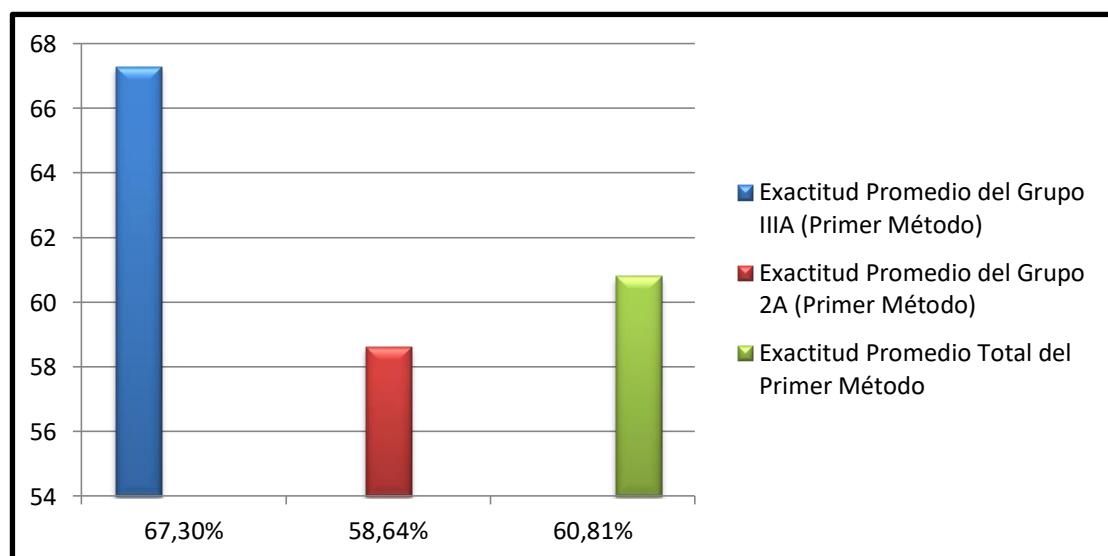


Figura 40. Exactitud de clasificación promedio del primer método de extracción de características.

Fuente: El autor.

6.2.2 Resultados del Método 2

La capacidad de discriminación del segundo método de extracción de características se evaluó utilizando el mismo procedimiento usado para el primer método. Las tablas 10 y 11 muestran los resultados obtenidos para el grupo IIIA y 2A respectivamente.

Tabla 10. Resultados de clasificación de imaginaria motora obtenidos por el segundo método de extracción de características para el grupo IIIA.

Fuente: El autor.

USUARIO	ACIERTOS OBTENIDOS PARA EL GRUPO IIIA POR EL SEGUNDO MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
SUJETO 1	51,1%	71,1%	57,8%	64,4%	64,4%	53,3%	51,1%	46,7%	62,2%	66,7%	71,1%
SUJETO 2	51,7%	48,3%	62,1%	51,7%	48,3%	51,7%	58,6%	51,7%	58,6%	55,2%	62,1%
SUJETO 3	48%	36%	68%	60%	56%	36%	56%	64%	52%	60%	68%

Tabla 11. Resultados de clasificación de imaginaria motora obtenidos por el segundo método de extracción de características para el grupo 2A.

Fuente: El autor.

USUARIO	ACIERTOS OBTENIDOS PARA EL GRUPO 2A POR EL SEGUNDO MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
SUJETO 1	53,6%	46,4%	56%	44%	50%	47,6%	41,7%	47,6%	51,2%	53,6%	56%
SUJETO 2	42,2%	56,6%	48,2%	48,2%	51,8%	49,4%	50,6%	45,8%	44,6%	49,4%	56,6%
SUJETO 3	68,7%	67,5%	56,6%	60,2%	65,1%	65,1%	62,7%	66,3%	69,9%	63,9%	69,9%
SUJETO 4	48,1%	51,9%	60,8%	45,6%	49,4%	54,4%	50,6%	51,9%	45,6%	40,5%	60,8%
SUJETO 5	51,3%	44,9%	46,2%	44,9%	59%	41%	51,3%	53,8%	47,4%	46,2%	59%
SUJETO 6	52,2%	44,9%	47,8%	52,2%	44,9%	52,2%	47,8%	56,5%	43,5%	52,2%	56,5%
SUJETO 7	49,4%	49,4%	56,8%	38,3%	54,3%	50,6%	56,8%	48,1%	42%	56,8%	56,8%
SUJETO 8	48,8%	56,3%	53,8%	51,2%	52,5%	48,8%	52,5%	47,5%	57,5%	51,2%	57,5%
SUJETO 9	62,9%	67,1%	54,3%	57,1%	61,4%	55,7%	54,3%	67,1%	60%	61,4%	67,1%

Se calculó el promedio de los mejores resultados de clasificación para tener una visión más general de la capacidad de discriminación del segundo método de extracción de características. Este resultado se muestra a continuación.

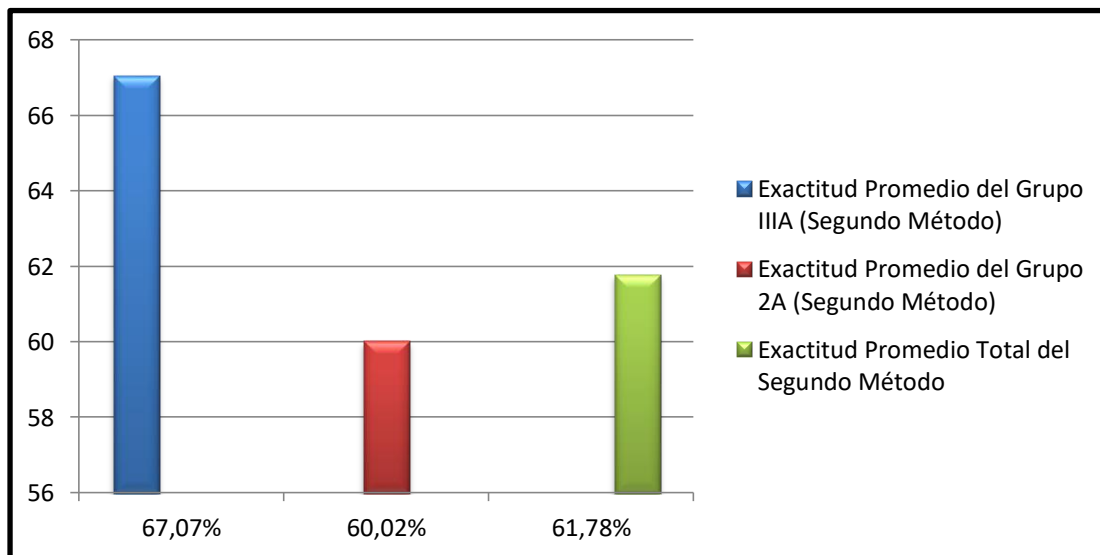


Figura 41. Exactitud de clasificación promedio del segundo método de extracción de características.

Fuente: El autor.

6.2.3 Resultados del Método 3

Para analizar los resultados del tercer método de extracción de características se procedió de la misma manera que en la evaluación de los métodos anteriores.

Las tablas 12 y 13 muestran los resultados obtenidos para el grupo IIIA y 2A respectivamente.

Tabla 12. Resultados de clasificación de imaginera motor obtenidos por el tercer método de extracción de características para el grupo IIIA.

Fuente: El autor.

USUARIO	ACIERTOS OBTENIDOS PARA EL GRUPO IIIA POR EL TERCER MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
SUJETO 1	48,9%	51,1%	51,1%	48,9%	53,3%	55,6%	46,7%	46,7%	57,8%	53,3%	57,8%
SUJETO 2	65,5%	31%	48,3%	55,2%	44,8%	51,7%	51,7%	72,4%	55,2%	51,7%	72,4%
SUJETO 3	68%	44%	52%	48%	76%	64%	72%	56%	64%	72%	76%

Tabla 13. Resultados de clasificación de imaginiería motora obtenidos por el tercer método de extracción de características para el grupo 2A.

Fuente: El autor.

USUARIO	ACIERTOS OBTENIDOS PARA EL GRUPO 2A POR EL TERCER MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS										MEJOR RESULTADO
	1	2	3	4	5	6	7	8	9	10	
SUJETO 1	51,2%	52,4%	58,3%	35,7%	58,3%	53,6%	52,4%	53,6%	53,6%	52,4%	58,3%
SUJETO 2	51,8%	51,8%	49,4%	49,4%	60,2%	47%	51,8%	47%	51,8%	43,4%	60,2%
SUJETO 3	73,5%	72,3%	69,9%	74,7%	75,9%	74,7%	71,1%	69,9%	73,5%	75,9%	75,9%
SUJETO 4	50,6%	55,7%	44,3%	49,4%	45,6%	54,4%	49,4%	54,4%	58,2%	49,4%	58,2%
SUJETO 5	48,7%	47,4%	44,9%	38,5%	60,3%	37,2%	44,9%	57,7%	46,2%	47,4%	60,3%
SUJETO 6	46,4%	47,8%	40,6%	52,2%	46,4%	50,7%	50,7%	58%	46,4%	47,8%	58%
SUJETO 7	53,1%	61,7%	58%	54,3%	60,5%	51,9%	55,6%	51,9%	46,9%	54,3%	61,7%
SUJETO 8	55%	56,3%	66,3%	60%	61,3%	55%	61,3%	47,5%	48,8%	57,5%	66,3%
SUJETO 9	60%	64,3%	70%	58,6%	62,9%	65,7%	61,4%	44,3%	60%	67,1%	70%

La exactitud promedio de este método de extracción de características se muestra a continuación.

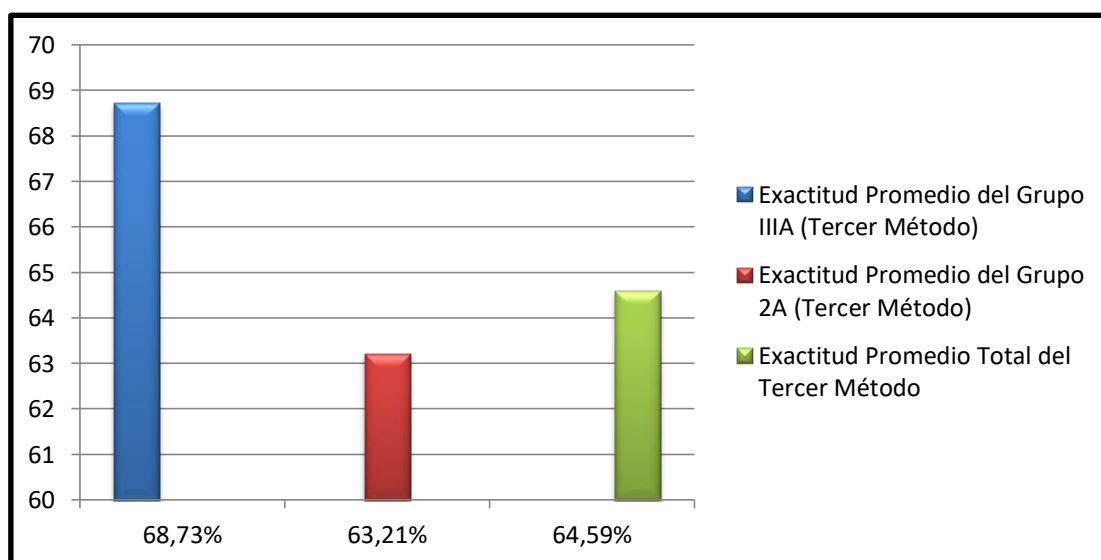


Figura 42. Exactitud de clasificación promedio del tercer método de extracción de características.

Fuente: El autor.

7 DISCUSIÓN

La metodología de extracción y clasificación de características basada en transformada wavelet y redes neuronales que se aplicó para la detección de ondas alfa, muestra un alto porcentaje de aciertos en la discriminación entre actividad cerebral sin ondas alfa y ritmos alfa. El mejor resultado tiene un nivel de aciertos del 89,1%. Gracias a esta elevada eficiencia en la clasificación se pudo implementar una aplicación BCI simple, de fácil manejo para el usuario.

La detección de ondas alfa ha sido utilizada exitosamente en BCI para el control de movimiento de un móvil a control remoto (Audette, 2017), para controlar la aceleración de un helicóptero de juguete (Vigué, 2017) y para controlar un juego de vídeo (Moyes & Jiang, 2018). Además los ritmos alfa podrían ser usados para medir el esfuerzo mental realizado por un usuario (Nicolas-Alonso & Gomez-Gil, 2012) y se han hecho trabajos en donde su análisis, junto con el de las bandas delta, theta, beta y gamma se aplica para medir el nivel de somnolencia de un usuario (Shen, Li, & Shi, 2017). Por esta razón se propone como posible trabajo futuro una aplicación para detectar los niveles de sueño de los transportistas mientras realizan viajes, como medida de prevención de accidentes.

El tiempo de respuesta de la BCI implementada es de por lo menos 1s, debido a que la ventana de análisis que se usa para extraer los comandos del usuario tiene esta duración. Para un trabajo inicial, esta latencia no representa ningún problema, pero pueden presentar inconvenientes para BCI destinadas al uso diario y consumo masivo ya que no permitirían un uso fluido y rápido del dispositivo. Por esto se hace necesario reducir la ventana de análisis planteada, sin embargo debido al éxito que se ha obtenido en los proyectos revisados en la literatura y en la presente tesis, se piensa que esto no conllevaría ningún problema en absoluto.

Para el análisis de los ritmos sensoriomotores se examinó una variedad de métodos de extracción de características, cada uno con diferentes niveles de precisión en la discriminación de imaginaria motora de mano izquierda y derecha. El primer método aplicado es exactamente el mismo que el empleado para la detección de ondas alfa y se basa en la captura de las principales características frecuenciales de la señal mediante la DWT; esto se hizo así debido a que para la caracterización de los ritmos sensoriomotores el dominio de la frecuencia es uno de los más usados (Schalk &

Mellinger, 2010). Sin embargo, la precisión del sistema resultó mucho menor a la lograda en la clasificación de ondas alfa.

En el segundo método se usaron coeficientes AR junto con los parámetros wavelet debido al éxito que estos dos tipos de características han tenido en trabajos anteriores (Xu & Song, 2008), obteniéndose solamente una leve mejoría en la clasificación.

El tercer método utiliza filtrado CSP para maximizar las diferencias entre clases y las características que se calculan son los coeficientes AR, parámetros wavelet y varianza de las señales EEG, obteniéndose una mejoría visible en relación con los métodos anteriores. El nivel de aciertos es del 64,59% en promedio para cada usuario, lo que es aceptable para su posible aplicación en BCI asíncronas con salida binaria; es decir, BCIs que sean capaces de realizar dos acciones, ya sea apagar o encender una luz, responder a preguntas de “sí” y “no” o controlar el movimiento de un cursor o un móvil de una dimensión (arriba y abajo, derecha e izquierda). Cada una de las dos acciones es representada por la imaginación motora de mano derecha e izquierda.

En la clasificación de ritmos sensoriomotores la cantidad de aciertos varía en gran medida para cada uno de los sujetos evaluados; para el sujeto con el que se obtuvieron los mejores resultados con el tercer método, la exactitud es del 76% mientras que para el sujeto con el menor desempeño es del 57,8%. Posiblemente esta gran varianza en la clasificación se deba a que el desempeño de una BCI también es dependiente de la capacidad del usuario para generar los patrones cerebrales que la comandarán. Un proceso de entrenamiento que permita a los usuarios aprender a controlar su actividad neurológica de mejor manera puede dar lugar a una mejora en los resultados de los algoritmos.

8 CONCLUSIONES

- El filtrado espacial CSP, la potencia promedio, desviación estándar y valor absoluto medio de los coeficientes wavelet dentro de las bandas μ y β , los coeficientes AR de un modelo de orden 6 y la varianza de las señales EEG son parámetros que han demostrado una eficiencia aceptable para discriminar imaginaria motora de manos izquierda y derecha. Esto hace que puedan ser usados para la caracterización de ritmos sensoriomotores en aplicaciones BCI.
- Las redes neuronales son clasificadores universales que han sido utilizados en varias aplicaciones de identificación de patrones. En este caso, son capaces de traducir las características calculadas a partir de los ritmos sensoriomotores con niveles de discriminación aceptables. Por esta razón se concluye que las redes neuronales pueden ser usadas como técnicas de clasificación en aplicaciones BCI.
- La aplicación de los parámetros wavelet calculados y el uso de redes neuronales han permitido la detección de ondas alfa de una mujer adulta con altos niveles de exactitud; esto viabilizó el diseño y la implementación de una aplicación BCI que activa un sistema de indicadores cada vez que el usuario es capaz de generar actividad alfa.
- EEG es uno de los métodos más populares en el ámbito de las BCI para la adquisición de señales cerebrales. Este método utiliza varios electrodos colocados en el cuero cabelludo para la obtención de la actividad neurológica. Sin embargo, a través del presente trabajo se puede comprobar que no es necesario utilizar todos los canales de grabación EEG para la clasificación tanto de ondas alfa como de ritmos sensoriomotores, sino que es suficiente utilizar aquellos electrodos que se encuentren sobre las zonas del cráneo responsables de la generación de cada tipo de actividad cerebral.

9 RECOMENDACIONES

- La normalización de cada ventana EEG es un proceso necesario, debido a que la amplitud de la señal EEG depende de varios factores, como la impedancia de los electrodos, y puede variar a través de diferentes sesiones de entrenamiento y diferentes usuarios. Mediante la normalización se tiene un marco de referencia común para la comparación de las señales entre varios sujetos.
- Las señales EEG pueden ser corrompidas por señales de interferencia generadas por el movimiento de los ojos y parpadeos. Estas señales, conocidas como artefactos oculares, pueden enmascarar las débiles señales EEG y producir resultados erróneos en el proceso de clasificación. Para evitar estos inconvenientes se recomienda eliminar las muestras que se encuentren contaminadas por artefactos antes del proceso de entrenamiento de la red neuronal, o utilizar algoritmos para la eliminación de dichos artefactos, que son capaces de recuperar la información distorsionada por ellos.

10 BIBLIOGRAFÍA

- [1] (6 de junio de 2017). Obtenido de sitio web de BCI Competition III: <http://www.bbc.de/competition/iii/>
- [2] (6 de junio de 2017). Obtenido de sitio web de BCI Competition IV: <http://www.bbc.de/competition/iv/>
- [3] Ahmadi, A., Dehzangi, O., & Jafary, R. (2012). Brain-computer interface signal processing algorithms: A computational cost vs. accuracy analysis for wearable computers. *Ninth International Conference on Wearable and Implantable Body Sensor Networks* (págs. 40-45). Londres: IEEE.
- [4] Alomari, M., Awada, E., Samaha, A., & Alkamha, K. (2014). Wavelet-Based Feature Extraction for the Analysis of EEG Signals Associated with Imagined Fists and Feet Movements. *Computer and Information Science*, 17-27.
- [5] Audette, W. (24 de 8 de 2017). *EEG Hacker*. Obtenido de EEG Hacker: <http://eeghacker.blogspot.com/2015/03/brain-controlled-shark-attack.html>
- [6] Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Müller, K.-R. (2008). Optimizing Spatial Filters for Robust EEG Single-Trial Analysis. *IEEE SIGNAL PROCESSING MAGAZINE*, 41-56.
- [7] Brunner, C., Leeb, R., Müller-Putz, G. R., Schlögl, A., & Pfurtscheller, G. (6 de Junio de 2017). Obtenido de sitio web de BCI Competition IV: http://www.bbc.de/competition/iv/desc_2a.pdf
- [8] Burrus, C., Gopinath, R., & Haitao, G. (1998). *Introduction to wavelets and wavelet transforms: a primer*. Houston: Prentice-Hall.
- [9] Dorofki, M., Elshafie, A. H., Jaafar, O., Karim, O. A., & Mastura, S. (2012). Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data. *2012 International Conference on Environment, Energy and Biotechnology* (págs. 39-44). Kuala Lumpur: IACSIT Press.
- [10] ElAli, T., & Karim, M. (2008). *Continuous Signals and Systems with Matlab*. Boca Raton: CRC Press.
- [11] Forslund, P. (2003). *A Neural Network Based Brain-Computer Interface for Classification of Movement Related EEG*. Linköping: Linköping University.
- [12] Graimann, B., Allison, B., & Pfurtscheller, G. (2010). Brain-Computer Interfaces: A Gentle Introduction. En *Brain-Computer Interfaces* (págs. 1-27). Springer.
- [13] Graupe, D. (2007). *PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS*. Singapur: World Scientific.
- [14] Henríquez, C. (2014). *Estudio de Técnicas de análisis y clasificación de señales EEG en el contexto de Sistemas BCI (Brain Computer Interface)*. Madrid: Universidad Autónoma de Madrid.

- [15] Joyce, C., Gorodnitsky, I., & Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, 313-325.
- [16] Moyes, C., & Jiang, M. (24 de 8 de 2018). *BCI using EEG*. Obtenido de BCI using EEG: https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2012/cwm55/cwm55_mj294/#
- [17] Mukul, M., & Matsuno, F. (2010). Relative spectral power (RSP) and temporal RSP as features for movement imagery EEG classification with linear discriminant analysis. *SICE Annual Conference 2010* (págs. 439-448). Taiwan: IEEE.
- [18] Nicolas-Alonso, L., & Gomez-Gil, J. (2012). Brain Computer Interfaces, a Review. *Sensors*, 1211-1279.
- [19] Olías, J. (2016). *Estudio del método Common Spatial Patterns y sus variantes en interfaces cerebro-ordenador*. Sevilla: Universidad de Sevilla.
- [20] *OpenBCI Shop*. (12 de Diciembre de 2016). Obtenido de OpenBCI Shop: <http://shop.openbci.com/collections/frontpage/products/openbci-32-bit-board-kit?variant=784651699>
- [21] Oppenheim, A., & Schaffer, R. (2009). *Tratamiento de Señales en Tiempo Discreto*. Madrid: Prentice Hall.
- [22] Pertence, A. (1991). *Amplificadores operacionales y filtros activos. Teoría, proyectos y aplicaciones prácticas*. España: McGraw-Hill.
- [23] Rao, R. (2013). *Brain-computer interfacing: an introduction*. New York: Cambridge University Press.
- [24] Sarasa, G. (2015). *DEVELOPMENT OF A TOOLBOX TO HANDLE ONLINE EEG SIGNALS ACROSS THE OPENBCI PLATFORM*. Madrid: Universidad Autónoma de Madrid.
- [25] Sarle, W. (27 de Marzo de 2014). *faqs.org*. Obtenido de sitio web de faqs.org: <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-12.html>
- [26] Schalk, G., & Mellinger, J. (2010). *A Practical Guide to Brain-Computer Interfacing with BCI2000*. Londres: Springer.
- [27] Schlögl, A. (6 de Junio de 2017). Obtenido de sitio web de BCI Competition III: http://www.bbci.de/competition/iii/desc_IIIa.pdf
- [28] Shen, J., Li, B., & Shi, X. (2017). Real-Time Detection of Human Drowsiness via a Portable Brain-Computer Interface. *Open Journal of Applied Sciences*, 98-113.
- [29] Suzuki, K. (2013). *Artificial Neural Networks – Architectures and Applications*. Rijeka: InTech.

- [30] Vallabhaneni, A., Wang, T., & He, B. (2005). Brain—computer interface. En B. He, *Neural Engineering* (págs. 85-121). Springer US.
- [31] Vigué, I. (24 de 08 de 2017). *IRENE'S BLOG/PORTFOLIO*. Obtenido de IRENE'S BLOG/PORTFOLIO: <https://irenevigueux.wordpress.com/2016/07/14/a-toy-helicopter-throttle-controlled-by-alpha-waves/>
- [32] Viñuela, P., & Galván, I. (2004). *Redes de Neuronas Artificiales. Un Enfoque Práctico*. Madrid: Prentice Hall.
- [33] Weeks, M. (2007). *Digital signal processing using matlab and wavelets*. Hingham: Infinty Science Press.
- [34] *Wikimedia Commons*. (25 de Agosto de 2015). Obtenido de Wikimedia Commons: https://commons.wikimedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg
- [35] Xu, B.-G., & Song, A.-G. (2008). Pattern recognition of motor imagery EEG using wavelet transform. *Journal of Biomedical Science and Engineering*, 64-67.
- [36] *YouTube*. (18 de Agosto de 2016). Recuperado el 20 de Noviembre de 2016, de MATLAB: <https://www.youtube.com/watch?v=F7Lg-nFYooU>

11 ANEXOS

11.1 GLOSARIO DE ACRÓNIMOS

A

- ALS
Amyotrophic Lateral Sclerosis (Esclerosis Lateral Amiotrófica), 4
- ANN
Artificial Neural Network (Red Neuronal Artificial), 36
- AR
AutoRegressive Components (Componentes Autorregresivos), 24, 33, 34, 58, 71, 83
- ASCII
American Standard Code for Information Interchange (Código Estándar Estadounidense para el Intercambio de Información), 60, 61

B

- BCI
Brain-Computer Interface (Interfaz Cerebro-Computador), XIII, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 26, 35, 36, 48, 49, 56, 57, 58, 61, 62, 63, 64, 81, 82, 83

C

- CAR
Common Average Reference (Referencia Promedio Común), 25
- CSP
Common Spatial Patterns (Patrones Espaciales Comunes), 24, 25, 26, 27, 72, 73, 82, 83
- CT
Computed Tomography (Tomografía Computarizada), 7
- CWT
Continuous Wavelet Transform (Transformada Wavelet Continua), 24, 30, 31, 32

D

- DWT
Discrete Wavelet Transform (Transformada Wavelet Discreta), 24, 30, 31, 32, 33, 54, 58

E

- ECoG
Electrocorticography (Electrocorticografía), 7, 9
- EEG

- Electroencephalography (Electroencefalografía), 2, 3, 8, 9, 10, 11, 15, 16, 17, 19, 23, 24, 25, 26, 27, 34, 35, 48, 49, 50, 51, 52, 53, 54, 55, 56, 58, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 75, 82, 83, 84

- EOG
Electrooculography (Electrooculografía), 65, 66, 67

F

- FIR
Finite Impulse Response (Respuesta Finita al Impulso), 31
- FIRNN
Finite Impulse Response Neural Network (Red Neuronal de Respuesta Finita al Impulso), 36
- fMRI
Functional Magnetic Resonance Imaging (Imagen por Resonancia Magnética Funcional), 8, 9
- FT
Fourier Transform (Transformada de Fourier), 28, 29

G

- GUI
Graphic User Interface (Interfaz Gráfica de Usuario), 60, 61

I

- ICA
Independent Component Analysis (Análisis de Componentes Independientes), 24, 25
- ITR
Information Transfer Rate (Taza de Transferencia de Información), 10, 18

K

- K-NNC
K-Nearest Neighbor Classifier (Clasificador de K Vecinos más Cercanos), 36

L

- LDA
Linear Discriminant Analysis (Análisis de Discriminación Lineal), 35

M

- MEG
 - Magnetoencephalography (Magnetoencefalografía), 8, 9
- MF
 - Matched Filter (Filtro Adaptado), 24
- MRI
 - Magnetic Resonance Imaging (Imagen de Resonancia Magnética), 7

P

- PCA
 - Principal Component Analysis (Análisis de Componentes Principales), 23, 25
- PeGNC
 - Probability estimating Guarded Neural Classifiers, 36
- PNN
 - Probabilistic Neural Networks (Red Neuronal Probabilística), 36

S

- SCP
 - Slow Cortical Potentials (Potenciales Corticales Lentos), 13, 15, 16
- SNR
 - Signal-to-Noise Ratio (Relación Señal-Ruido), 9

SPECT

- Single-Photon Emission Computed Tomography (Tomografía por Emisión de Fotón Único), 7

SSVEP

- Steady State VEP (VEP de Estado Estable), 10, 14

STFT

- Short Time Fourier Transform (Transformada de Fourier de Tiempo Corto), 28, 29

SVM

- Support Vector Machine (Máquina de Soporte de Vectores), 35

T

TTD

- Thought-Translation Device (Dispositivo de Traducción de Pensamiento), 15

TVEP

- Transient VEP (VEP Transitoria), 14

V

VEP

- Visual Evoked Potentials (Potenciales Visuales Evocados), 9, 13, 14

W

WT

- Wavelet Transform (Transformada Wavelet), 29

11.2 CÓDIGO DE MATLAB PARA LA NORMALIZACIÓN DE LAS SEÑALES CEREBRALES

```
function [ norm_data ] = min_max( data )
%UNIVERSIDAD NACIONAL DE LOJA
%Carrera de Ingeniería en Electrónica y Telecomunicaciones
%Esta función realiza la normalización min-max de una matriz de
%datos temporales, cuyas filas representan cada punto temporal y las
%columnas cada canal de adquisición.
%data es una matriz de señales sin normalizar y norm_data es una
matriz con
%las señales ya normalizadas.
%
%Autor: Milton León
%2017

mins = min(data); %Calculamos el mínimo de cada vector en matriz
maxs = max(data); %Calculamos el máximo de cada vector en matriz

[rows,columns] = size(data);

norm_data = zeros(rows, columns); %Definimos matriz de salida

for x = 1:columns %Bucle para calcular normalización de cada canal
    for y = 1:rows %Bucle para normalizar cada punto temporal
        norm = (data(y,x) - mins(x))/(maxs(x) - mins(x)); %Se
normaliza el punto
        norm_data(y,x) = norm; %Se almacena el punto en la matriz de
salida
    end
end
end
```

11.3 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS DE LOS RITMOS ALFA

```
function [CAR] = características_alfa(x)

%UNIVERSIDAD NACIONAL DE LOJA
%CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES
% [car] = características_alfa(x) extrae las características wavelet
% de una matriz x que contiene datos EEG.
%Las características wavelet se ubican en el vector columna car.
%La variable x es una matriz de 250 x 2, en donde las filas
representan
%cada punto temporal de una grabación EEG y las columnas representan
los
%canales de adquisición O1 y O2 respectivamente.
%
%Autor: Milton León
%2017

%Definimos constantes a utilizar en el programa
%n = 125; %Longitud de las señales en cada canal

%-----Pre-procesamiento de la Señal-----
---
x = filtro_eeg(x);
x = x(3:250-2,:); %Eliminamos efectos de borde
x = min_max(x);

%-----Transformada Wavelet de las Señales-----
[w_c3,l1] = wavedec(x(:,1), 4, 'db10'); %T.W del canal 1
[w_c4,l2] = wavedec(x(:,2), 4, 'db10'); %T.W del canal 2

%Obtención de los niveles de detalle cD4
%Para canal 1
%Se obtienen los coeficientes de detalle 4 mediante el vector l1
d4_c3 = w_c3(l1(1)+1:l1(1)+l1(2));

%Para canal 2
d4_c4 = w_c4(l2(1)+1:l2(1)+l2(2));

%-----Cálculo de las Características de los Coeficientes-----
-----

%Cálculo de la desviación estándar-----
-----
dev1 = std(d4_c3);
dev2 = std(d4_c4);

%Cálculo de la potencia promedio de los coeficientes-----
-----
p1 = bandpower(d4_c3);
p2 = bandpower(d4_c4);

%Cálculo del Valor Absoluto Medio-----
%Calculamos el MAV para cada nivel de detalle
%Para el canal 1
%Nivel de detalle D4
MAV1 = mean(abs(d4_c3));
```



```
%Para el canal 2
%Nivel de detalle D4
MAV2 = mean(abs(d4_c4));

CAR = [dev1; MAV1; p1; dev2; MAV2; p2];
```

11.4 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS SEGÚN EL MÉTODO 1 (PARÁMETROS WAVELET)

```
function [CAR] = características(x)

%UNIVERSIDAD NACIONAL DE LOJA
%CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES
% [car] = características(x) extrae las características wavelet de
% una matriz x que contiene imaginaria EEG de mano derecha e
% izquierda.
%Las características wavelet se ubican en el vector columna car.
%La variable x es una matriz de 250 x 3, en donde las filas
%representan
%cada punto temporal de una grabación EEG y las columnas representan
%los
%canales de adquisición C4, Cz y C3 respectivamente.
%
%Autor: Milton León
%2017

%Definimos constantes a utilizar en el programa
%n = 250; %Longitud de las señales en cada canal

%-----Pre-procesamiento de la Señal-----
---
%x = filtro_eeg(x);
%x = x(3:125-2,:); %Eliminamos efectos de borde
x = min_max(x); %Normalizamos la señal EEG

%-----Transformada Wavelet de las Señales-----
[w_c4,11] = wavedec(x(:,1), 4, 'db10'); %T.W del canal 1
[w_cz,12] = wavedec(x(:,2), 4, 'db10'); %T.W del canal 2
[w_c3,13] = wavedec(x(:,3), 4, 'db10'); %T.W del canal 3

%Obtención de los niveles de detalle cD3 y cD4
%Para canal 1
%Se obtienen los coeficientes del detalle 3 mediante el vector l1
d3_c4 = w_c4(l1(1)+l1(2)+1:l1(1)+l1(2)+l1(3));
%Se obtienen los coeficientes del detalle 4 mediante el vector l1
d4_c4 = w_c4(l1(1)+1:l1(1)+l1(2));

%Para canal 2
d3_cz = w_cz(l2(1)+l2(2)+1:l2(1)+l2(2)+l2(3));
d4_cz = w_cz(l2(1)+1:l2(1)+l2(2));

%Para canal 3
d3_c3 = w_c3(l3(1)+l3(2)+1:l3(1)+l3(2)+l3(3));
d4_c3 = w_c3(l3(1)+1:l3(1)+l3(2));

%-----Cálculo de las Características de los Coeficientes-----
-----

%Cálculo de la desviación estándar-----
-----
%Para canal 1
```

```

%Nivel de detalle cD3
dev1 = std(d3_c4);
%Nivel de detalle cD4
dev2 = std(d4_c4);

%Para canal 2
dev3 = std(d3_cz);
dev4 = std(d4_cz);

%Para canal 3
dev5 = std(d3_c3);
dev6 = std(d4_c3);

%Cálculo de la potencia promedio de los coeficientes-----
-----
%Para canal 1
%Nivel de detalle cD3
p1 = bandpower(d3_c4);
%Nivel de detalle cD4
p2 = bandpower(d4_c4);

%Para canal 2
p3 = bandpower(d3_cz);
p4 = bandpower(d4_cz);

%Para canal 3
p5 = bandpower(d3_c3);
p6 = bandpower(d4_c3);

%Cálculo del Valor Absoluto Medio-----
%Calculamos el MAV para cada nivel de detalle
%Para el canal 1
%Nivel de detalle D3
MAV1 = mean(abs(d3_c4));
%Nivel de detalle D4
MAV2 = mean(abs(d4_c4));

%Para el canal 2
MAV3 = mean(abs(d3_cz));
MAV4 = mean(abs(d4_cz));

%Para el canal 3
MAV5 = mean(abs(d3_c3));
MAV6 = mean(abs(d4_c3));

%Construimos el vector de características-----
-----
CAR = [dev1; MAV1; p1; dev2; MAV2; p2; dev3; MAV3; p3; dev4; MAV4;
p4;...
dev5; MAV5; p5; dev6; MAV6; p6];

end

```

11.5 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS SEGÚN EL MÉTODO 2 (WAVELET + AR)

```
function [CAR] = características2(x)

%UNIVERSIDAD NACIONAL DE LOJA
%CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES
% [car] = características2(x) extrae las características wavelet y AR
de
% una matriz x que contiene imaginaria EEG de mano derecha e
izquierda.
%Las características AR y wavelet se ubican en el vector columna car.
%La variable x es una matriz de 250 x 3, en donde las filas
representan
%cada punto temporal de una grabación EEG y las columnas representan
los
%canales de adquisición C4, Cz y C3 respectivamente.
%
%Autor: Milton León
%2017

%Definimos constantes a utilizar en el programa
%n = 250; %Longitud de las señales en cada canal

%-----Pre-procesamiento de la Señal-----
---
%x = filtro_eeg(x);
%x = x(3:125-2,:); %Eliminamos efectos de borde
x = min_max(x); %Normalizamos la señal EEG

%-----Transformada Wavelet de las Señales-----
[w_c4,l1] = wavedec(x(:,1), 4, 'db10'); %T.W del canal 1
[w_cz,l2] = wavedec(x(:,2), 4, 'db10'); %T.W del canal 2
[w_c3,l3] = wavedec(x(:,3), 4, 'db10'); %T.W del canal 3

%Obtención de los niveles de detalle cd3 y cd4
%Para canal 1
%Se obtienen los coeficientes del detalle 3 mediante el vector l1
d3_c4 = w_c4(l1(1)+l1(2)+1:l1(1)+l1(2)+l1(3));
%Se obtienen los coeficientes del detalle 4 mediante el vector l1
d4_c4 = w_c4(l1(1)+1:l1(1)+l1(2));

%Para canal 2
d3_cz = w_cz(l2(1)+l2(2)+1:l2(1)+l2(2)+l2(3));
d4_cz = w_cz(l2(1)+1:l2(1)+l2(2));

%Para canal 3
d3_c3 = w_c3(l3(1)+l3(2)+1:l3(1)+l3(2)+l3(3));
d4_c3 = w_c3(l3(1)+1:l3(1)+l3(2));

%-----Cálculo de las Características de los Coeficientes-----
-----

%Cálculo de la desviación estándar-----
-----
%Para canal 1
%Nivel de detalle cd3
```

```

dev1 = std(d3_c4);
%Nivel de detalle cD4
dev2 = std(d4_c4);

%Para canal 2
dev3 = std(d3_cz);
dev4 = std(d4_cz);

%Para canal 3
dev5 = std(d3_c3);
dev6 = std(d4_c3);

%Cálculo de la potencia promedio de los coeficientes-----
-----
%Para canal 1
%Nivel de detalle cD3
p1 = bandpower(d3_c4);
%Nivel de detalle cD4
p2 = bandpower(d4_c4);

%Para canal 2
p3 = bandpower(d3_cz);
p4 = bandpower(d4_cz);

%Para canal 3
p5 = bandpower(d3_c3);
p6 = bandpower(d4_c3);

%Cálculo del Valor Absoluto Medio-----
%Calculamos el MAV para cada nivel de detalle
%Para el canal 1
%Nivel de detalle D3
MAV1 = mean(abs(d3_c4));
%Nivel de detalle D4
MAV2 = mean(abs(d4_c4));

%Para el canal 2
MAV3 = mean(abs(d3_cz));
MAV4 = mean(abs(d4_cz));

%Para el canal 3
MAV5 = mean(abs(d3_c3));
MAV6 = mean(abs(d4_c3));

%-----Cálculo de los Coeficientes de Autorregresión-----
-----
%Se calculan los coeficientes del modelo de autorregresión de orden
%6 para cada canal

m1 = ar(x(:,1),6,'burg','Ts',0.004); %Calculo de modelo AR por método
de Burg
m2 = ar(x(:,2),6,'burg','Ts',0.004);
m3 = ar(x(:,3),6,'burg','Ts',0.004);

ar1 = m1.a; %Extracción de los coeficientes del modelo AR para canal 1
ar1 = ar1'; %Obtenemos la transpuestas de los coeficientes AR para
%ubicación en vector de características
ar2 = m2.a;
ar2 = ar2';

```

```

ar3 = m3.a;
ar3 = ar3';

%Construimos el vector de características-----
-----

CAR = [dev1; MAV1; p1; dev2; MAV2; p2; ar1(2:7); dev3; MAV3; p3; dev4;
MAV4; p4; ar2(2:7); ...
      dev5; MAV5; p5; dev6; MAV6; p6; ar3(2:7)];

end

```

11.6 CÓDIGO DE MATLAB PARA LA EXTRACCIÓN DE CARACTERÍSTICAS SEGÚN EL MÉTODO 3 (CSP + WAVELET + AR + VARIANZA)

```
function [CAR] = caracteristicas3(x)

%UNIVERSIDAD NACIONAL DE LOJA
%CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES
% [car] = caracteristicas3(x) extrae las características wavelet, AR y
de varianza
%de una matriz x que contiene imaginería EEG de mano derecha e
izquierda.
%Las características se ubican en el vector columna car.
%La variable x es una matriz de 250 x 3, en donde las filas
representan
%cada punto temporal de una grabación EEG y las columnas representan
los
%canales de adquisición C4, Cz y C3 respectivamente.
%
%Autor: Milton León
%2017

%Definimos constantes a utilizar en el programa
%n = 250; %Longitud de las señales en cada canal

%-----Pre-procesamiento de la Señal-----
---
%x = filtro_eeg(x);
%x = x(3:125-2,:); %Eliminamos efectos de borde
x = min_max(x); %Normalizamos la señal EEG

%-----Transformada Wavelet de las Señales-----
[w_c4,11] = wavedec(x(:,1), 4, 'db10'); %T.W del canal 1
[w_cz,12] = wavedec(x(:,2), 4, 'db10'); %T.W del canal 2
[w_c3,13] = wavedec(x(:,3), 4, 'db10'); %T.W del canal 3

%Obtención de los niveles de detalle cD3 y cD4
%Para canal 1
%Se obtienen los coeficientes del detalle 3 mediante el vector l1
d3_c4 = w_c4(11(1)+11(2)+1:11(1)+11(2)+11(3));
%Se obtienen los coeficientes del detalle 4 mediante el vector l1
d4_c4 = w_c4(11(1)+1:11(1)+11(2));

%Para canal 2
d3_cz = w_cz(12(1)+12(2)+1:12(1)+12(2)+12(3));
d4_cz = w_cz(12(1)+1:12(1)+12(2));

%Para canal 3
d3_c3 = w_c3(13(1)+13(2)+1:13(1)+13(2)+13(3));
d4_c3 = w_c3(13(1)+1:13(1)+13(2));

%-----Cálculo de las Características de los Coeficientes-----
-----

%Cálculo de la desviación estándar-----
-----
%Para canal 1
```

```

%Nivel de detalle cD3
dev1 = std(d3_c4);
%Nivel de detalle cD4
dev2 = std(d4_c4);

%Para canal 2
dev3 = std(d3_cz);
dev4 = std(d4_cz);

%Para canal 3
dev5 = std(d3_c3);
dev6 = std(d4_c3);

%Cálculo de la potencia promedio de los coeficientes-----
-----
%Para canal 1
%Nivel de detalle cD3
p1 = bandpower(d3_c4);
%Nivel de detalle cD4
p2 = bandpower(d4_c4);

%Para canal 2
p3 = bandpower(d3_cz);
p4 = bandpower(d4_cz);

%Para canal 3
p5 = bandpower(d3_c3);
p6 = bandpower(d4_c3);

%Cálculo del Valor Absoluto Medio-----
%Calculamos el MAV para cada nivel de detalle
%Para el canal 1
%Nivel de detalle D3
MAV1 = mean(abs(d3_c4));
%Nivel de detalle D4
MAV2 = mean(abs(d4_c4));

%Para el canal 2
MAV3 = mean(abs(d3_cz));
MAV4 = mean(abs(d4_cz));

%Para el canal 3
MAV5 = mean(abs(d3_c3));
MAV6 = mean(abs(d4_c3));

%-----Cálculo de los Coeficientes de Autorregresión-----
-----
%Se calculan los coeficientes del modelo de autorregresión de orden
%6 para cada canal

m1 = ar(x(:,1),6,'burg','Ts',0.004); %Calculo de modelo AR por método
de Burg
m2 = ar(x(:,2),6,'burg','Ts',0.004);
m3 = ar(x(:,3),6,'burg','Ts',0.004);

ar1 = m1.a; %Extracción de los coeficientes del modelo AR para canal 1
ar1 = ar1'; %Obtenemos la transpuestas de los coeficientes AR para

```



```

%ubicación en vector de características
ar2 = m2.a;
ar2 = ar2';
ar3 = m3.a;
ar3 = ar3';

%Cálculo de la varianza de la amplitud de la señal de cada banda
alfa = filtro_alfa(x);
beta = filtro_beta(x);
var_alfa1 = var(alfa(:,1));
var_alfa2 = var(alfa(:,2));
var_alfa3 = var(alfa(:,3));
var_beta1 = var(beta(:,1));
var_beta2 = var(beta(:,2));
var_beta3 = var(beta(:,3));

%Construimos el vector de características-----
-----

CAR = [dev1; MAV1; p1; dev2; MAV2; p2; ar1(2:7); var_alfa1; var_beta1;
dev3; MAV3; p3; dev4;...
      MAV4; p4; ar2(2:7); var_alfa2; var_beta2; dev5; MAV5; p5; dev6;
MAV6; p6; ar3(2:7); var_alfa3; var_beta3];

end

```

11.7 CÓDIGO DE LA INTERFAZ GRÁFICA DE USUARIO (GUI) DEL DETECTOR DE ONDAS ALFA

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

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

% Last Modified by GUIDE v2.5 01-Jul-2017 16:42:21

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

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

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

```

% varargin    command line arguments to bci (see VARARGIN)

%-----
--
%Configuramos las imagenes de la GUI
%Para el logotipo de la universidad
axes(handles.logo_unl);
background = imread('logo_unl_horizontal-01.png');
axis off;
imshow(background);
%Para el logotipo de la carrera
axes(handles.logo_cieyt);
background = imread('cieyt.jpg');
axis off;
imshow(background);
%Para el logotipo de la BCI
axes(handles.logo_bci);
background = imread('pic_bci.png');
axis off;
imshow(background);

%Configuramos ejes de la GUI
%Para el eje O1
axes(handles.axes_o1);
line1 = plot(zeros(1,246));
axis([0 249 0 1]);
handles.line1 = line1;
%Para el eje O2
axes(handles.axes_o2);
line2 = plot(zeros(1,246));
axis([0 249 0 1]);
handles.line2 = line2;

%-----
-

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

% Update handles structure
guidata(hObject, handles);

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

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

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

function edit1_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%          str2double(get(hObject,'String')) returns contents of edit1
%          as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

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

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

% Hint: get(hObject,'Value') returns toggle state of boton_serial
estado_serial = get(hObject,'Value');
if estado_serial == 1
    a = get(handles.edit1,'String'); %Se obtiene el num. del puerto
    serial
        b = 'COM';
        port = strcat(b,a); %Definimos el puerto a utilizar
        %Creamos objeto serial para placa OBCI
        puerto = serial(port,'BaudRate',115200,'Parity','none','StopBits',
1, 'InputBufferSize', 2^18);
        fopen(puerto); %Inicializa el puerto
        flushinput(puerto); %Borramos datos que pueden encontrarse en el
buffer del puerto
        set(handles.texto_puerto,'String','Conectando puerto serial...');
        %Reiniciamos placa OpenBCI
        fwrite(puerto,'v','uchar');
        pause(3); %Esperamos que el comando se vuelva efectivo
        flushinput(puerto);
        set(handles.texto_puerto,'String','Puerto serial conectado. ');
        set(handles.boton_serial,'String','DESCONECTAR PUERTO SERIAL');
        handles.puerto = puerto; %Guardamos el objeto puerto para poder
usarlo en otras funciones
        guidata(hObject,handles);
else
    puerto = handles.puerto;
    %Terminamos el streaming de datos de la placa OBCI
    fwrite(puerto,'s','uchar');
    set(handles.texto_puerto,'String','Desconectando puerto
serial...');
    pause(1); %Esperamos a que el comando se vuelva efectivo

```

```

    %Cerramos y eliminamos el puerto serial de la placa OBCI
    flushinput(puerto); %Eliminamos los datos en el buffer de entrada
    fclose(puerto);
    delete(puerto);
    set(handles.texto_puerto,'String','Puerto serial desconectado.');
```

end

```

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

% Hint: get(hObject,'Value') returns toggle state of boton_blue
estado_blue = get(hObject,'Value');
if estado_blue == 1
    %Creamos objeto bluetooth
    arduino = Bluetooth('CHURON',1);
    set(handles.texto_blue,'String','Conectando Bluetooth...');
    pause(3); %Esperamos que comando de creación sea efectivo
    fopen(arduino);
    flushinput(arduino); %Borramos datos que pueden encontrarse en el
buffer
    set(handles.texto_blue,'String','Bluetooth conectado.');
```

set(handles.boton_blue,'String','DESCONECTAR BLUETOOTH');

handles.arduino = arduino; %Guardamos el objeto arduino para que pueda ser usado por otras funciones

guidata(hObject,handles);

```

else
    arduino = handles.arduino;
    %Cerramos y eliminamos la comunicación con la placa Arduino
    flushinput(arduino); %Eliminamos los datos en el buffer de entrada
    fclose(arduino);
    delete(arduino);
    clear arduino;
    set(handles.texto_blue,'String','Bluetooth desconectado.');
```

set(handles.boton_blue,'String','CONECTAR BLUETOOTH');

end

```

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

% Hint: get(hObject,'Value') returns toggle state of boton_bci
estado_interfaz = get(hObject,'Value');
```

if estado_interfaz == 1

```

    set(handles.boton_bci,'String','DETENER INTERFAZ');
```

set(handles.texto_sistema,'String','Adquiriendo datos');

puerto = handles.puerto;

arduino = handles.arduino;

line1 = handles.line1;

line2 = handles.line2;

```

    %Definimos variables que controlarán la lectura de datos continua
    muestras = 250; %Número de muestras que se deben obtener en cada
ventana de tiempo
```

```

eeg = zeros(muestras, 2); %Matriz que contendrá los datos de cada
ventana
packet_index = 0; %Indice que indica el número de la siguiente
trama a leer

%Iniciamos streaming de datos
fwrite(puerto, 'b', 'uchar');

%Iniciamos la lectura de datos continua del amplificador EEG
%-----
-----

while 1 %Bucle para obtener datos EEG de forma continua

    %Esperamos a que existan bytes disponibles
    bytes_disponibles = get(puerto, 'BytesAvailable');
    while bytes_disponibles < 33
        bytes_disponibles = get(puerto, 'BytesAvailable');
    end

    packet_counter = 0; %índice que indica el número de paquetes
que se han leído

    while packet_counter < muestras

        lectura = fread(puerto, 33);
        %Comprobamos si los 33 bytes son iguales a la
estructura de una trama
        if (lectura(1) == 160) && (lectura(2) == packet_index)
&& (lectura(33) == 192)
            [canales, ~] = unpackOBCI_alfa(lectura, 1); %Si los
son, desempaquetamos datos
            packet_counter = packet_counter + 1; %Actualizamos
el contador de tramas
            eeg(packet_counter, :) = canales; %Ubicamos datos
en matriz de salida
            packet_index = packet_index + 1; %Actualizamos el
contador del número de la siguiente trama
            if packet_index > 255 %Si el índice de trama
es mayor a 255 se pone a cero
                packet_index = 0;
            end
        else
            %Si no tienen la estructura de una trama se debe
sincronizar la
            %lectura
            [canales_sync, packet_index, packet_counter] =
sincronizar_alfa(puerto, packet_counter);
            eeg(packet_counter-1:packet_counter, :) =
canales_sync;
        end
    end

    %Escalamos los datos obtenidos en unidades de voltaje real
eeg = (eeg.*((4.5/24)/((2^23)-1)));

    %Graficamos los datos EEG en los ejes correspondientes
eeg = filtro_eeg(eeg);
eeg = eeg(3:250-2, :);
eeg = min_max(eeg);

```

```

        %Para el eje O1
        set(line1, 'ydata', eeg(:,2));
        %Para el eje O2
        set(line2, 'ydata', eeg(:,1));
        drawnow; %Dibujar

        %Bloque comentado que contiene la caracterización y
clasificación
        %de las señales EEG. Descomentar cuando se necesite finalizar
GUI
        %-----
        -----
        %Calculamos el vector de características de los datos EEG
        carac = caracteristicas_alfa(eeg);
        %Clasificamos la señal EEG en base a sus características
        categoria = red_alfa2(carac);
        %Realizamos una acción en base a la clasificación
        if categoria(1,1) >= categoria(2,1)
            fwrite(arduino, 'o', 'uchar');
            fwrite(arduino, 00001010, 'uchar');
        else
            fwrite(arduino, 's', 'uchar');
            fwrite(arduino, 00001010, 'uchar');
        end
        %-----
        -----

        %pause(0.001); %Introducimos una pausa para evaluar estado de
boton
        if get(hObject, 'Value') == 0
            break;
        end

        end

    else
        set(handles.boton_bci, 'String', 'INICIAR INTERFAZ');
        set(handles.texto_sistema, 'String', 'Sistema inactivo. ');
        %Terminamos el streaming de datos de la placa OBCI
        puerto = handles.puerto;
        fwrite(puerto, 's', 'uchar');
        pause(0.5);
        flushinput(puerto);
    end

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

    %Rutina para probar el estado de la conexión Bluetooth
    arduino = handles.arduino;
    fwrite(arduino, 'o', 'uchar');
    fwrite(arduino, 00001010, 'uchar');
    pause(1);
    fwrite(arduino, 's', 'uchar');
    fwrite(arduino, 00001010, 'uchar');

```

11.8 CÓDIGO DE ARDUINO PARA EL SISTEMA DE ACTUADORES DEL DETECTOR DE ONDAS ALFA

/* UNIVERSIDAD NACIONAL DE LOJA
Carrera de Ingeniería en Electrónica y Telecomunicaciones

Código para la detección de ritmos alfa.

Se enciende un indicador luminoso, un buzzer y se muestra con una pantalla LCD que se han detectado ondas alfa.

Este programa fue implementado en un Arduino Nano con procesador ATmega328.

Autor: Milton León
2017
*/

```
#include <LiquidCrystal.h>

//Creamos variable para control de LCD
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

char x;

void setup() {

    //Definimos los pines que actuaran como salidas para indicar presencia
    //de ondas alfa
    pinMode(3,OUTPUT); //LED
    pinMode(4,OUTPUT); //Buzzer

    //Definimos el estado inicial de todos los pines
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);

    //Inicializamos comunicación serial
    Serial.begin(9600);
    //Esperamos a que se inicializa el puerto serial
    while (!Serial);

    //Inicializamos el control del display LCD
    lcd.begin(20,4);

}

void loop() {
    //Se espera hasta que existan datos que leer en el puerto serial
    if (Serial.available () > 0 )

    {x = Serial.read(); //Leemos el puerto serial

    //Encender indicadores si se detecta orden
    if (x == 'o') {
        digitalWrite(3,HIGH);
        digitalWrite(4,HIGH);
        lcd.clear();
        lcd.setCursor(4,2);
        lcd.print("ONDAS ALFA");
        lcd.setCursor(4,3);
    }
    }
```



```

        lcd.print("DETECTADAS");
    }

    //Apagamos indicadores si se detecta orden
    else if (x == 's') {
        digitalWrite(3,LOW);
        digitalWrite(4,LOW);
        lcd.clear();
        lcd.setCursor(4,2);
        lcd.print("ESCANEANDO");
        lcd.setCursor(4,3);
        lcd.print("ONDAS EEG");
    }

}
}

```

11.9 FOTOGRAFÍAS DEL DESARROLLO DE LAS SESIONES DE ENTRENAMIENTO DE RITMOS ALFA

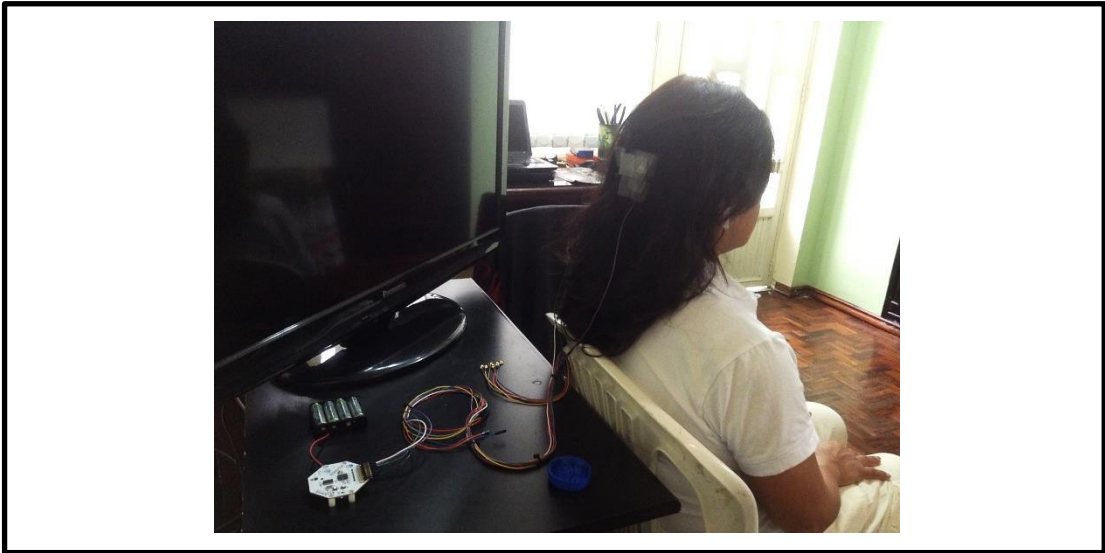


Figura 43. Vista del amplificador EEG y la colocación de electrodos para una sesión de entrenamiento.

Fuente: El autor.



Figura 44. Electrodo de referencia colocado en el lóbulo de la oreja izquierda.

Fuente: El autor.