



POSGRADOS

MAESTRÍA EN _____ ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

DESARROLLO DE UN SISTEMA BCI BASADO EN
REDES NEURONALES Y MOVIMIENTOS DE LA
CABEZA PARA EL MANEJO DE UN ORDENADOR

AUTOR:

EDDY FABIAN CORRALES BASTIDAS

DIRECTOR:

CARMEN JOHANNA CELI SANCHEZ

QUITO - ECUADOR
2021

Autor:



Eddy Fabián Corrales Bastidas.

Ingeniero en Electrónica e Instrumentación.

Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Quito.

eddyco2307@gmail.com

Dirigido por:



Carmen Johanna Celi Sánchez.

Ingeniera en Sistemas Mención Robótica e Inteligencia
Artificial.

Magíster en Diseño Producción y Automatización
Industrial.

cceci@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2021 Universidad Politécnica Salesiana.

QUITO – ECUADOR – SUDAMÉRICA

EDDY FABIAN CORRALES BASTIDAS

***DESARROLLO DE UN SISTEMA BCI BASADO EN REDES
NEURONALES Y MOVIMIENTOS DE LA CABEZA PARA EL
MANEJO DE UN ORDENADOR***

Índice general

Índice de Figuras	6
Índice de Tablas	8
Abstract	9
1. Introducción	11
1.1. Descripción general del problema	12
1.2. Objetivos	12
1.2.1. Objetivo general	12
1.2.2. Objetivos específicos	12
1.3. Contribuciones	13
1.4. Organización del manuscrito	13
2. Marco Referencial	14
2.1. Estado del Arte	15
2.2. Pregunta de Investigación	19
2.3. Justificación	19
2.4. Propuesta de Solución	20
2.5. Alcance	21
2.6. Delimitación	21
2.6.1. Temporal	21
2.6.2. Espacial	21
2.6.3. Institucional	21
3. Marco Conceptual	22
3.1. Fundamentos Fisiológicos	23
3.1.1. El Cerebro Humano	23
3.1.2. Enfermedades Neurodegenerativas	23
3.1.3. Señales Electroencefalográficas (EEG)	24

3.2. Sistemas BCI	26
3.2.1. Medición de la actividad cerebral	27
3.2.2. Fases de los sistemas BCI	27
3.2.3. Estrategias de señales de control	29
3.3. Dispositivos y software comerciales de adquisición de señales EEG	30
3.3.1. Emotiv Insight	31
3.3.2. EmotivPRO	32
3.3.3. API Cortex	33
3.4. Redes neuronales artificiales	34
3.4.1. Elementos de una red neuronal	34
3.4.2. Mecanismos de aprendizaje	37
3.5. Entornos de programación y el algoritmo de Kalman	38
3.5.1. Python	39
3.5.2. Filtro de Kalman	39
4. Desarrollo del Sistema	41
4.1. Requerimientos previos	42
4.2. Realización del experimento	43
4.2.1. Objetivo del experimento	43
4.3. Etapas y metodología de diseño	44
4.4. Diseño del sistema BCI	46
4.4.1. Adquisición de señales	47
4.4.2. Extracción de características	48
4.4.3. Clasificación	56
4.5. Control del cursor del mouse	62
4.5.1. Diseño	63
4.6. API Cortex para el modo ON LINE	64
4.7. Interfaz gráfica de usuario	66
4.7.1. Librerías utilizadas	67
4.7.2. Funcionalidades de la interfaz	67
5. Pruebas y Resultados	69
5.1. Resultados en las etapas de desarrollo del sistema BCI	70
5.1.1. Etapa de adquisición de señales	70
5.1.2. Etapa de extracción de características	70
5.1.3. Etapa de clasificación	72
5.2. Pruebas del control del cursor del mouse	78
5.3. Funcionamiento del sistema	81
5.4. Pruebas sobre el sistema BCI	82

<i>ÍNDICE GENERAL</i>	5
5.4.1. Prueba test	83
5.4.2. Prueba sobre la interfaz de usuario	84
5.4.3. Prueba general sobre el sistema	85
5.5. Resultados generales sobre el sistema BCI	86
5.5.1. Discusión de resultados	90
Conclusiones y Recomendaciones	95
Referencias	98
A. Uso de la diadema Emotiv Insight	98
B. Entrenamientos virtuales	101
C. Manejo de archivos .csv	104

Índice de Figuras

3.1. Lóbulos de la corteza cerebral	23
3.2. Sistema Internacional 10-20. Posición de los electrodos	26
3.3. Medición de la actividad cerebral. (Método no invasivo)	27
3.4. Fases de los sistemas BCI	28
3.5. Kit Emotiv Insight	31
3.6. Estructura de una red neuronal	35
3.7. Componentes de una red neuronal	36
4.1. Secuencia del experimento	43
4.2. Etapas de diseño a cumplirse del sistema BCI	44
4.3. Diagrama general del algoritmo a desarrollarse	45
4.4. Proceso de adquisición de señales	47
4.5. Participante durante la toma de señales	47
4.6. Separación de los datos de los cinco canales EEG	48
4.7. Gráfica resultante del data set de guiños derechos	49
4.8. Determinación de eventos sobre un tramo de la señal	50
4.9. Columnas de Evento y Clase para el proceso de etiquetado	51
4.10. Ventanas de tiempo sobre un tramo de la señal	52
4.11. Transición del evento 0 al evento 1 en el proceso de etiquetado	53
4.12. Gráfica del data set de guiños derechos normalizado	54
4.13. Features obtenidos para el data set de guiños derechos	56
4.14. Arquitectura general de una red neuronal MLP	57
4.15. Lectura del archivo FeatureNet mediante la librería Pandas	58
4.16. Parámetros de configuración de la red MLPClassifier	58
4.17. Validación cruzada estratificada	59
4.18. Entrenamiento de la red neuronal	60
4.19. Validación del modelo mediante métricas de clasificación	61
4.20. Comunicación websocket entre la API Cortex y Python	65
4.21. Streams utilizados para el modo ON LINE	66

4.22. Pantalla principal de la interfaz gráfica de usuario	67
4.23. Teclado BCI	68
4.24. Sonidos BCI	68
5.1. Esquema general de la etapa de clasificación	73
5.2. Validación cruzada durante las iteraciones de entrenamiento	74
5.3. Scores de precisión durante el proceso de entrenamiento	75
5.4. Función de pérdida vs Iteraciones	76
5.5. Matriz de confusión resultante para el set de prueba	77
5.6. Reporte de clasificación obtenido del set de prueba	77
5.7. Valores medidos por los tres acelerómetros de la diadema	78
5.8. Filtro de Kalman unidimensional aplicado a datos de aceleración	79
5.9. Variación de la posición del cursor ante movimientos de la cabeza	80
5.10. Modos de funcionamiento del sistema BCI	81
5.11. Prueba test	83
5.12. Prueba sobre la interfaz de usuario	84
5.13. Prueba general sobre el sistema	85
5.14. Resultados generales de los cinco participantes	89
A.1. Colocación de la diadema Emotiv Insight	99
A.2. Calidad de conexión de los electrodos	100
B.1. Cubo virtual para entrenamientos mediante EmotivBCI	102
B.2. Entrenamientos realizados sobre el cubo virtual	103
B.3. Diagrama del espacio cerebral de los entrenamientos realizados	103
C.1. Ejemplo de formato de archivos .csv	105

Índice de Tablas

3.1. Nomenclatura Sistema Internacional 10-20	25
3.2. Dispositivos comerciales de adquisición de señales EEG	30
3.3. Características técnicas de la diadema Emotiv Insight	32
3.4. Funciones de activación de las redes neuronales	37
4.1. Identificadores de las clases a predecir	51
5.1. Consideraciones tomadas durante la adquisición de señales	70
5.2. Eventos identificados en el data set de guiños derechos	71
5.3. Eventos identificados en el data set de guiños izquierdos	71
5.4. Información en el data set de Features de guiños derechos	71
5.5. Información en el data set de Features de guiños izquierdos	72
5.6. Información recopilada del archivo FeatureNet	72
5.7. Composición de la red neuronal MLP empleada	73
5.8. Distribución de las muestras por cada clase para ambos sets	74
5.9. Características físicas de los participantes de prueba	82
5.10. Resultados de la prueba test	86
5.11. Resultados sobre el control del cursor del mouse	87
5.12. Resultados de la prueba sobre la interfaz de usuario	88
5.13. Resultados de la prueba del sistema	89
5.14. Resultados generales del sistema desarrollado	90
5.15. Resultados obtenidos por otros autores	90
A.1. Representación del nivel de contacto de los electrodos	100
C.1. Etiquetas y descripción de los archivos .csv	106

Resumen

El presente trabajo ha surgido con la finalidad de brindar un aporte a los sistemas BCI, a través del campo de la Inteligencia Artificial, es así, que se ha desarrollado una interfaz cerebro computadora (BCI), destinada al manejo de un ordenador y orientada principalmente a personas que poseen movilidad reducida en sus extremidades superiores.

Dado que un sistema BCI se caracteriza por hacer uso de señales cerebrales del usuario a través de relacionar ciertos patrones cerebrales, se estableció como patrones o estrategias de señales de control el uso de guiños de ojos. En virtud de esto, se seleccionó la diadema Emotiv Insight como dispositivo de captura de señales EEG.

El tratamiento de las señales obtenidas se las realizó en la etapa de extracción de características, en la que se consideró aspectos de series temporales y ventanas de tiempo para el proceso de etiquetado de las señales y la determinación de eventos. Por su parte, para el proceso de clasificación de las señales se utilizó una red neuronal MLP, como algoritmo de clasificación del data set de features, alcanzando el clasificador durante el proceso de entrenamiento una precisión de clasificación del 91.53 %. Esto permitió generar un modelo de clasificación, a fin de clasificar en tiempo real las señales captadas por la diadema sobre tres posibles acciones de los eventos del mouse: click izquierdo, click derecho y none.

Para el control del cursor del mouse a través de movimientos de la cabeza del usuario, se utilizó los acelerómetros de la diadema y el Filtro de Kalman como herramienta para minimizar ruidos y estimar la posición real del cursor del mouse.

Finalmente, se realizaron pruebas para validar el sistema donde se incluyó personas sanas y personas con algún tipo de limitación física en sus extremidades superiores, alcanzando el sistema una precisión del 88.06 %.

Palabras clave: Sistemas BCI, Inteligencia Artificial, red neuronal MLP, señales EEG, movimientos de cabeza.

Abstract

The present work has arisen with the purpose of providing a contribution to the BCI systems, through the field of Artificial Intelligence; thus, a brain-computer interface (BCI) has been developed, aimed at the management of a computer and mainly oriented to people who have reduced mobility in their upper limbs.

Since a BCI system is characterized by making use of the user's brain signals by relating certain brain patterns, the use of eye winks was established as patterns or control signal strategies. By virtue of this, the Emotiv Insight headset was selected as the EEG signal capture device.

The treatment of the signals obtained was carried out in the feature extraction stage, in which aspects of time series and time windows were considered for the process of signal labeling and event determination. For the signal classification process, a MLP neural network was used as the algorithm for classifying the data set of features, with the classifier achieving a classification accuracy of 91.53 % during the training process. This allowed to generate a classification model, in order to classify in real time the signals captured by the headset on three possible actions of the mouse events: left click, right click and none.

To control the mouse cursor through the user's head movements, the headband accelerometers and the Kalman Filter were used as a tool to minimize noise and estimate the real position of the mouse cursor.

Finally, tests were carried out to validate the system, including healthy people and people with some type of physical limitation in their upper limbs, achieving an accuracy of 88.06 %.

Keywords: BCI systems, Artificial Intelligence, MLP neural network, EEG signals, head movements.

Capítulo 1

Introducción

Desde la antigüedad el ser humano ha constituido un pilar fundamental en el avance de la ciencia, gracias a sus capacidades mentales que le permiten razonar, crear, imaginar, etc., permitiendo tener un mundo con comodidades, por medio de la tecnología. Sin embargo, sus actividades cotidianas, su estilo de vida, su genética, etc., pueden llevarlo a desencadenar a futuro enfermedades de tipo neurodegenerativas, ocasionando la pérdida de movilidad en sus extremidades; por lo que se ha visto en la necesidad de utilizar sistemas que le permitan comunicarse con el mundo que lo rodea.

Con el paso de los años, para cubrir esta necesidad se ha empleado los sistemas BCI (Brain Computer Interface, por sus siglas en inglés), que se basan en el registro de la actividad cerebral para proporcionar a los usuarios nuevas formas de controlar dispositivos externos [Alazrai et al., 2019].

De acuerdo a [Ramadan and Vasilakos, 2017], investigaciones en el campo de los sistemas BCI han incrementado significativamente en los últimos 10 años, siendo Estados Unidos líder en números de investigaciones y empresas que trabajan en este tipo de sistemas, a su vez, cerca de 50.000 dispositivos de asistencia han sido producidos para ayudar a personas con discapacidades.

Al mismo tiempo, el desarrollo de las distintas ramas de la Ingeniería Electrónica, han permitido integrar a los sistemas BCI algoritmos sofisticados que permitan clasificar tareas mentales y de esa manera crear aplicaciones innovadoras que faciliten la forma de vida de personas con algún tipo de discapacidad. Por esta razón, este trabajo se centra en aportar al desarrollo de los sistemas BCI hacia el ámbito de la Inteligencia Artificial, con el propósito de manejar un ordenador a base de la actividad cerebral de la persona.

1.1. Descripción general del problema

Con el paso del tiempo, el ser humano se ha vuelto más propenso a sufrir algún tipo de enfermedad, por lo tanto, en la actualidad se realizan investigaciones científicas que permiten que la medicina y la tecnología vayan de la mano, para brindar una mejor asistencia y forma de vida a las personas.

Sin embargo, de acuerdo a [Chiu et al., 2020] se ha estimado que a nivel mundial alrededor de 47 millones de personas podrían sufrir algún tipo de enfermedad neurodegenerativa para el año 2050, lo que representaría un impacto directo en el estatus socio-económico de cualquier país.

Ante este panorama, existe la necesidad de acoplar las nuevas tecnologías predominantes del mundo actual, como el Big Data, Internet de las Cosas, Inteligencia Artificial, etc., hacia los denominados sistemas BCI, que se han constituido en alternativas de vida para aquellas personas que sufren algún tipo de enfermedad neurodegenerativa o que presentan algún tipo de limitación física en sus movimientos, el habla, etc., permitiendo que de alguna forma continúen con su ritmo de vida lo mejor posible.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema BCI basado en redes neuronales y movimientos de la cabeza, para el manejo de un ordenador en personas con movimiento reducido en sus extremidades superiores.

1.2.2. Objetivos específicos

- Investigar las interfaces cerebro computador, para la determinación del hardware y software que se empleará en el desarrollo del sistema, en base a la recopilación de fuentes bibliográficas.
- Diseñar una red neuronal para el entrenamiento de la interfaz cerebro computador, mediante técnicas de Inteligencia Artificial.
- Desarrollar el sistema BCI integrando el software inteligente, ordenador y sensores, para la obtención de las señales de control que permitan comandar una determinada acción.
- Validar el sistema BCI desarrollado mediante pruebas de funcionamiento, para una correcta manipulación del mismo en personas con movimiento reducido en sus extremidades superiores.

1.3. Contribuciones

El presente trabajo promueve un aporte hacia el desarrollo de los sistemas BCI, mediante la integración de una de las principales tecnologías del momento como es la Inteligencia Artificial, contribuyendo de manera significativa al campo de la medicina, permitiendo que personas con algún tipo de movimiento reducido en sus extremidades superiores, puedan tener acceso al manejo de un ordenador.

1.4. Organización del manuscrito

El presente manuscrito se compone de 5 capítulos, organizados de la siguiente manera:

- Capítulo 1. Introducción
- Capítulo 2. Marco Referencial
- Capítulo 3. Marco Conceptual
- Capítulo 4. Desarrollo del Sistema
- Capítulo 5. Pruebas y Resultados

Capítulo 2

Marco Referencial

Este capítulo se ha estructurado de la siguiente manera:

- **Estado del Arte:** Se hace una revisión literaria de los antecedentes que dan origen a la aparición de los sistemas BCI, sus técnicas, sus fases que lo componen y finalmente se hace un análisis de trabajos realizados por distintos autores relacionados a los sistemas BCI.
- **Pregunta de Investigación:** Se estructura la idea de investigación que origina la problemática a resolver.
- **Justificación:** Se indican los motivos e importancia del desarrollo del presente trabajo.
- **Propuesta de Solución:** Se menciona la metodología a emplearse para la obtención de los objetivos planteados en el presente trabajo.
- **Alcance:** Se indica el resultado que se pretende obtener con el diseño del sistema BCI propuesto.
- **Delimitación:** Se realiza una delimitación espacial, temporal e institucional.

2.1. Estado del Arte

En la actualidad, las enfermedades que más expansión han tenido son las enfermedades neurodegenerativas, por lo que de acuerdo a [Fernández-Cruz and Reynaud, 2020], más del 1% de la población mundial sufre de algún tipo de enfermedad neurodegenerativa, misma que tiende a afectar principalmente a individuos de edades avanzadas, reduciendo su calidad y expectativa de vida. Se espera que esta cifra se duplique en las próximas décadas debido a que este grupo de individuos se encuentra en constante aumento.

Dentro de este tipo de enfermedades se encuentran el Alzheimer, la enfermedad de Parkinson, la enfermedad de Huntington y la Esclerosis Lateral Amiotrófica, teniendo en común dichas enfermedades la pérdida progresiva de la estructura y función neuronal, llegando a afectar principalmente al deterioro de la motricidad humana [Zhang et al., 2020, Athisakthi and Rani, 2017].

La comunicación constituye la base fundamental con la que el ser humano transmite sus ideas hacia su entorno que lo rodea, pero esta destreza propia de todo ser humano se ha visto afectada por el desarrollo de las enfermedades de tipo neurodegenerativas, por lo que de acuerdo a [Michahial et al., 2012], ante una situación en la que algún paciente con incapacidad severa de hablar o moverse, la posibilidad de aprovechar sus tareas mentales haciendo uso de sus propias señales cerebrales, constituye una forma de establecer una comunicación. Es así que, dichas señales pueden llegar a ser convertidas en acciones de control sobre dispositivos físicos, a través de los denominados sistemas BCI (Brain Computer Interface, por sus siglas en inglés).

El autor [Sorbello et al., 2017] manifiesta que un sistema BCI representa un método directo de comunicación entre un cerebro humano y un ordenador, siendo la técnica de señales EEG (Electroencefalografía), la que más ha sido explorada para el desarrollo de sistemas BCI, la misma que se basa en medir la actividad cerebral de la persona, siendo una técnica segura, económica y que no representa ningún tipo de riesgo para la persona.

Debido a que los sistemas BCI utilizan dispositivos capaces de captar señales EEG a fin de lograr una transferencia óptima de información del cerebro a la máquina, el método de señales EEG es considerada la más confiable al momento de medir la actividad cerebral, por lo que es fundamental que el proceso de adquisición de estas señales sea de alta fidelidad y calidad, siendo los electrodos el componente principal para la medición de las señales EEG. Al ser los electrodos los encargados de estar en contacto con el cuero cabelludo del usuario, a fin de obtener información de la actividad cerebral y su posterior envío a un ordenador para su monitoreo

respectivo, existen técnicas para su colocación pudiendo ser de tipo invasiva (electrodos colocados internamente mediante una intervención quirúrgica) o no invasiva (electrodos colocados sobre la cabeza). Actualmente existen diferentes empresas que comercializan dispositivos de EEG para captura de señales cerebrales, sea de forma invasiva o no invasiva, entre ellas se destacan: Emotiv, Neurosky, Neuroscan, etc., pero la mayoría de las empresas principalmente de Europa y Asia están enfocadas en desarrollar dispositivos de EEG de tipo no invasivo, dado que representa mayor rapidez, facilidad, sin necesidad de algún tipo de intervención quirúrgica para su colocación [Ramadan and Vasilakos, 2017, Lee et al., 2013].

De igual manera, a nivel de software los sistemas BCI requieren de entornos y lenguajes de desarrollo, para que a través de algoritmos sofisticados se puedan realizar las distintas fases que intervienen dentro de los sistemas BCI, al punto de satisfacer alguna aplicación específica; es por eso que dentro de los lenguajes de programación más utilizados para el desarrollo de los sistemas BCI se encuentran: C++, MATLAB, Python, etc.

Uno de los puntos importantes al momento de captar las señales provenientes del cerebro humano a través de algún dispositivo de EEG, es utilizar estrategias denominadas señales de control, las cuales de acuerdo a [Ramadan and Vasilakos, 2017], se dividen en 3 grupos: señales evocadas, señales espontáneas y señales híbridas; permitiendo dichas señales de control crear estímulos para la activación del cerebro, siendo estos estímulos representados mediante sonidos, imágenes, comandos mentales, expresiones faciales, etc.

Al ser los sistemas BCI una herramienta que se basa en traducir la actividad cerebral del usuario para convertirlos finalmente en una acción de control, es necesario que estos sistemas pasen por determinadas fases como son: adquisición de señales, extracción de características y clasificación.

La incursión en el mundo de los sistemas BCI implica tener en claro cada una de las fases que intervienen en este tipo de sistemas, por lo que de acuerdo a lo mencionado por [Tiwari et al., 2018], la primera fase de todo sistema BCI es la adquisición de la señal, la misma que se realiza mediante la ayuda de dispositivos de EEG, utilizando esta fase procedimientos de tipo invasivos o no invasivos. Algunos autores como por ejemplo [Graumann et al., 2009, Hassanien and Azar, 2015], mencionan una fase previa a la extracción de características denominada preprocesamiento, la cual se basa en realizar tareas de filtrado de las señales adquiridas a fin de eliminar cualquier tipo de artefactos (componentes no deseados) que se encuentren durante la fase de adquisición de la señal de EEG, por ejemplo, movimientos oculares, latidos del corazón, actividades musculares, etc.

Posteriormente, se menciona una fase de extracción de características, que se basa en recibir señales filtradas para que a través de algún tipo de análisis se pueda determinar las señales más sobresalientes y de esta forma facilitar la información a la última fase denominada clasificación, siendo esta la encargada de asociar distintas actividades cerebrales generadas por el usuario en comandos de acciones de control, a través del empleo de algoritmos de clasificación sofisticados.

En base a investigaciones realizadas por [Irtiza and Farooq, 2016], se menciona que la fase de clasificación es la más trascendental en los sistemas BCI, dado que dependiendo del tipo de algoritmo que se emplee se tendrá excelentes resultados o por el contrario se generarán acciones de control equivocadas. Bajo esta premisa, actualmente existen diferentes algoritmos de clasificación que son utilizados ampliamente para el análisis de señales EEG, entre ellas se destacan: análisis discriminante lineal (LDA, por sus siglas en inglés), clasificador bayesiano, árbol de decisiones, etc.

Complementando a los mencionados algoritmos de clasificación, el autor [Tiware et al., 2018] aporta con otros algoritmos como son: máquinas de vectores de soporte (SVM, por sus siglas en inglés), redes neuronales artificiales (ANN, por sus siglas en inglés), k vecinos más cercanos (kNN, por sus siglas en inglés), aprendizaje híbrido, etc. Al utilizarse señales EEG como pieza fundamental de los sistemas BCI, se buscan clasificadores óptimos para el manejo de datos, por lo que el mismo autor menciona que el clasificador LDA es estable y no se ve afectado por variaciones en el conjunto de datos de entrenamiento.

Por su parte, el algoritmo SVM es utilizado en problemas de clasificación y regresión, siendo los datos ubicados en un hiper-plano al punto de ser separados mediante vectores que permitan diferenciar entre una clase y otra. En cambio, el clasificador bayesiano representa ser un algoritmo enfocado a calcular una probabilidad a priori que posee un vector característico de pertenecer a una clase determinada, por lo que no representa ser muy utilizado en aplicaciones BCI; por contraparte, [Hassanien and Azar, 2015] ratifica que investigaciones en el campo de los sistemas BCI consideran al algoritmo de redes neuronales artificiales, como el más utilizado por la variedad de arquitecturas que existen, siendo fundamental elegir el tipo de arquitectura para una correcta clasificación en el caso de tareas mentales que se requieran distinguir para ejecutar una determinada acción de control.

Una vez realizado un compendio breve y conciso acerca de los sistemas BCI, a continuación, se mencionan algunos de los trabajos realizados hasta el momento por distintos autores, a fin de ser una guía para el desarrollo del presente proyecto a realizarse.

De acuerdo a [Grimann et al., 2009, Rodríguez-Miranda et al., 2016], el rango de aplicaciones de los sistemas BCI es muy amplio, abarcando desde tareas simples hasta tareas complejas. Es así que, tradicionalmente se lo ha utilizado en aplicaciones en las que al usuario a través de un computador se le asiste para que seleccione un determinado caracter, de paso a una siguiente diapositiva, interactúe con juegos que entrenen sus habilidades aritméticas, mejorando su nivel de atención y a la vez creando distracción y relajación; de igual manera, en el campo de la Domótica se lo ha utilizado para tener el control de encendido y apagado sobre luminarias, televisión, radio, etc.

El autor [O'Regan et al., 2010] mediante su investigación denominada, "*Automatic detection of EEG artefacts arising from head movements*", señala la necesidad de identificar y separar cualquier tipo de artefacto generado al momento de realizar algún tipo de movimiento con la cabeza, por lo que propone el uso de pruebas estadísticas no paramétricas y del algoritmo de análisis de discriminante lineal (LDA), siendo este último el que mejor rendimiento posee al momento de separar las clases correspondientes a señales con artefactos y señales sin artefactos.

Dado que el propósito de los sistemas BCI, es aprovechar la facultad de una persona de utilizar sus señales cerebrales más allá de que tenga algún tipo de discapacidad severa, es fundamental el tipo de señales de control que se busca asociar para una aplicación específica, por lo que en la investigación realizada por [Campbell et al., 2010] denominada, "*NeuroPhone: Brain-Mobile Phone Interface using a Wireless EEG Headset*", propone el control de un teléfono móvil mediante señales cerebrales, utilizando como señales de control el uso guiños de ojos y pensamientos, a fin de asociar dichas señales al manejo de un teléfono móvil; en el que a través del empleo de un clasificador bayesiano, obtuvo una precisión de clasificación del 95.58 %, siendo uno de los inconvenientes presentados la presencia de artefactos (pestañeos involuntarios), haciendo que el rendimiento en el proceso de clasificación se vea afectado, con lo que se puede apreciar que las otras fases de los sistemas BCI también requieren de una óptima atención.

La investigación hecha por [Shedeed and Issa, 2016] denominada, "*Brain-EEG Signal Classification Based on Data Normalization for Controlling a Robotic Arm*", controla un brazo robótico en la que se evidencia que muchas veces el proceso de clasificación se ve afectado debido a una incorrecta fase de extracción de características, por lo que se mencionan algunas técnicas como posibles soluciones para esta fase por ejemplo: la transformada wavelet, la transformada rápida de fourier (FFT, por sus siglas en inglés), análisis de componentes principales (PCA, por sus siglas en inglés), etc., en la que mediante la técnica de la transformada wavelet

facilitó el proceso de aprendizaje a la red neuronal empleada tipo perceptrón multicapa, la misma que obtuvo una tasa de clasificación del 92.2 %.

A su vez, a nivel de software la posibilidad de utilizar la herramienta de programación Python ha demostrado ser factible, por lo que el autor [Patrón and Monje, 2016] en su investigación denominada, “*Emotiv EPOC BCI with Python on a Raspberry pi*”, hace uso del dispositivo Emotiv EPOC para la captura de señales EEG y de los valores de giroscopios que son entregados por dicho dispositivo, en la que a través del uso de librerías propias de Python se procesó el conjunto de datos recibidos y se realizó cada una de las fases del sistema BCI a fin de manipular un robot. Además, se empleó el algoritmo de clasificación SVM dando buenos resultados en la identificación de si las señales de control pertenecen a determinadas clases, y a su vez, se utilizó una tarjeta Raspberry pi para la conexión con el brazo robótico.

Finalmente, indicar que las señales de control que son utilizadas en la búsqueda de asociar a determinadas acciones de control, son importantes que estas sean diferenciadas por un algoritmo clasificador, por lo que el autor [Abiyev et al., 2016] propone en su trabajo de investigación denominado, “*Brain-Computer Interface for Control of Wheelchair Using Fuzzy Neural Networks*”, el empleo de una red neuronal difusa que permita clasificar tareas mentales a fin de comandar una silla de ruedas en distintas direcciones, obteniendo una precisión del modelo superior al 92%, demostrando la potencialidad que tiene la integración de este algoritmo a la hora de crear aplicaciones con un alto grado de complejidad en los sistemas BCI.

Con estos antecedentes mencionados en esta sección, el presente proyecto busca acoplar una de las tecnologías más dominantes de la actualidad como es la Inteligencia Artificial a los sistemas BCI, en la que a través de una elección adecuada del software y hardware se pueda controlar el manejo de un ordenador basado en redes neuronales y movimientos de la cabeza.

2.2. Pregunta de Investigación

¿Es posible emplear el algoritmo de redes neuronales para la clasificación de señales EEG, que permitan a personas con movilidad reducida en sus extremidades superiores tener el control en el manejo de un ordenador?

2.3. Justificación

En la actualidad, los sistemas BCI están ganando terreno en diversos campos como la robótica, videojuegos, pero su principal enfoque va hacia

el campo de la medicina, dado que representa ser una pieza tecnológica clave capaz de cambiar la vida de pacientes con algún tipo de limitación física en sus extremidades superiores. Es así que, dicho sistema tiene como objetivo traducir la actividad cerebral y transformar las señales generadas en comandos de control hacia un dispositivo externo del mundo real.

Por tal motivo, se busca aportar al desarrollo de los sistemas BCI hacia un enfoque educativo y de rehabilitación, tal es el caso de que a través de movimientos de la cabeza una persona tenga la posibilidad de controlar acciones básicas en el manejo de un ordenador, por ejemplo, tener el control sobre el movimiento del cursor del mouse y a su vez realizar acciones complementarias como el clickeo del mouse a través de señales de control como comandos mentales o gestos faciales, etc.

El propósito principal del presente trabajo es brindar la posibilidad al usuario de interactuar con el manejo básico de un ordenador, empleando técnicas y algoritmos de Inteligencia Artificial como son las redes neuronales, a fin de clasificar las señales cerebrales provenientes del usuario de forma correcta y poder asociarlas a cuando la persona tiene la intención de realizar un click izquierdo, click derecho o ningún click; por lo que el sistema BCI propuesto constituye una alternativa de una nueva forma de vida para aquellas personas que se encuentran imposibilitadas parcial o totalmente en cuanto a la movilidad de sus extremidades superiores, y si a esto se le suma la existencia de interfaces amigables de variedades de programas, juegos, etc., ayuda a que este grupo de personas puedan tener a través del manejo de un ordenador, una herramienta de distracción y comunicación con su entorno, más allá de la condición física en la que se encuentren.

2.4. Propuesta de Solución

El presente proyecto se orienta al desarrollo de un sistema BCI que detecte señales de control provenientes de la actividad eléctrica del cerebro humano en tiempo real, de tal manera que dichas señales de control puedan ser procesadas a través de técnicas de procesamiento y clasificadas mediante el uso de redes neuronales, a fin de comandar el manejo de un ordenador.

Por lo tanto, el sistema estará compuesto por tres elementos principales descritos a continuación:

- Un dispositivo comercial de señales EEG (diadema sensorial), que permita la captura de señales cerebrales en tiempo real y envíe dicha información de manera inalámbrica al computador para su posterior análisis.

- Un software con entorno de desarrollo, que se encargue del análisis de las señales cerebrales recibidas por parte de la diadema sensorial, que a su vez permita la integración, procesamiento y desarrollo de cada una de las fases del sistema BCI propuesto.
- Un ordenador, que a más de integrar el software de desarrollo permita validar el sistema desarrollado en su totalidad, a través de una interfaz de usuario para la navegación sobre la misma.

2.5. Alcance

Al finalizar el desarrollo del proyecto propuesto se pretende que el sistema BCI desarrollado, muestre la factibilidad que existe de manejar un ordenador a partir de una correcta selección de hardware (diadema sensorial), software (entorno de desarrollo) y el algoritmo de clasificación (redes neuronales), dotando a través de la integración de dichos componentes, la obtención de un sistema BCI que ofrezca fiabilidad al momento de manejar un ordenador, sobre todo en personas que poseen movimientos reducidos en sus extremidades superiores.

2.6. Delimitación

2.6.1. Temporal

El diseño y desarrollo del proyecto se lo realizó entre los meses de junio de 2020 y febrero de 2021.

2.6.2. Espacial

El proceso de obtención de señales EEG para el desarrollo y validación del sistema propuesto, se lo realizó en la zona centro de la ciudad de Latacunga, donde el grupo de prueba fue en personas sanas como en personas con algún tipo de limitación física en sus extremidades superiores.

2.6.3. Institucional

El presente proyecto sigue las directrices estipuladas por la Universidad Politécnica Salesiana, entorno a la normativa para la opción de titulación de proyectos de desarrollo; por lo que la parte investigativa se la realiza mediante una revisión literaria de artículos científicos de revistas de alto impacto, cuyo contenido sea de relevancia para la presente investigación.

Capítulo 3

Marco Conceptual

Este capítulo abarca de forma breve y concisa la fundamentación teórica necesaria a tener en cuenta para el desarrollo del proyecto, por lo que se ha estructurado de la siguiente manera:

- **Fundamentos Fisiológicos:** Se hace una revisión del cerebro humano como órgano principal para el control de la mayoría de funciones del cuerpo y de la mente, a su vez, se mencionan algunas enfermedades de tipo neurodegenerativas producidas por algún daño en el cerebro y finalmente se revisa la forma de medir la actividad eléctrica del cerebro mediante las señales electroencefalográficas (EEG).
- **Sistemas BCI:** Se indican los métodos empleados para medir la actividad cerebral, además se abordan las fases que componen este tipo de sistemas y las estrategias de señales de control para captar la actividad cerebral.
- **Dispositivos y Software Comerciales de Adquisición de Señales EEG:** Se realiza una tabla comparativa entre algunas empresas que comercializan dispositivos de EEG y a su vez se mencionan las principales características del dispositivo y software comercial seleccionado para el presente trabajo.
- **Redes Neuronales Artificiales:** Se proporciona una información breve en base a las generalidades, elementos y mecanismos de aprendizaje que intervienen dentro de las redes neuronales artificiales.
- **Entornos de Programación y Algoritmo de Kalman:** Se indican las principales características del lenguaje de programación seleccionado, y generalidades del Filtro de Kalman.

3.1. Fundamentos Fisiológicos

3.1.1. El Cerebro Humano

Un órgano que es esencial dentro del sistema nervioso es el cerebro, el cual continuamente recibe información de su entorno a través de los sentidos, para llegar a comandar acciones sobre los movimientos corporales, el habla, y de igual manera controlar funciones cognitivas de aprendizaje, memoria, etc.

El cerebro está dividido en dos partes denominadas hemisferios, siendo el hemisferio izquierdo el encargado de controlar el lenguaje, al igual que la habilidad para realizar cálculos matemáticos y pensar de forma analítica; por su parte, el hemisferio derecho se encarga de controlar las emociones, habilidades creativas, etc. A su vez, ambos hemisferios se encuentran recubiertos por un manto de células sobre su superficie, conocida como corteza cerebral [Gómez and Sanz, 2018].

En el campo de los sistemas BCI, la corteza cerebral es considerada el centro de investigación en este tipo de sistemas, ya que controla el procesamiento sensoriomotor y las funciones de nivel superior como el procesamiento del lenguaje, el reconocimiento de patrones, la planificación y el razonamiento [Ramadan and Vasilakos, 2017].

Por su parte, cada hemisferio está dividido en 4 lóbulos llamados: frontal, temporal, parietal y occipital, como se muestra en la Figura 3.1.

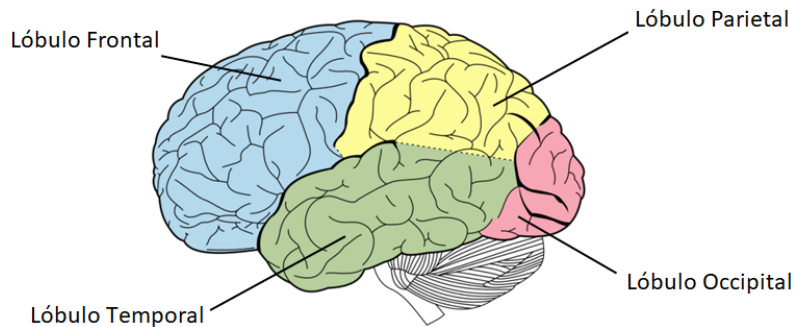


Figura 3.1: Lóbulos de la corteza cerebral

3.1.2. Enfermedades Neurodegenerativas

Son enfermedades progresivas que se caracterizan por la pérdida de la estructura o función neuronal asociadas con la muerte de las neuronas del

cerebro humano [Wu et al., 2020].

A continuación, se mencionan algunas enfermedades de este tipo, que tienden a afectar diversas capacidades intelectuales como movimientos, lenguaje, etc., derivando en una progresiva pérdida de autonomía.

- Enfermedad de Alzheimer
- Esclerosis Lateral Amiotrófica
- Enfermedad de Parkinson

Adicionalmente a este grupo de enfermedades, y al tratarse la presente investigación al desarrollo de un sistema BCI que reemplace la forma tradicional de manejar un ordenador, existen enfermedades que tienden a afectar principalmente a la movilidad de las manos, ocasionando a futuro deformaciones sobre la misma, distinguiéndose dos tipos de enfermedades como son:

- Tendinitis
- Artritis

3.1.3. Señales Electroencefalográficas (EEG)

La Electroencefalografía (EEG) es una técnica basada en el uso de imágenes médicas que registran la actividad eléctrica del cerebro a través del cuero cabelludo, generada por las estructuras del cerebro humano. Una de las principales características de esta técnica, es que emplea el uso de electrodos como medio para medir la actividad eléctrica del cerebro, y a su vez utiliza un procedimiento no invasivo, permitiendo que no haya riesgo o daño sobre el paciente durante el proceso de medición [Teplan et al., 2002].

Por otra parte, la actividad eléctrica medida a través de este tipo de técnicas es visualizada en forma de ondas a través de un ordenador para su respectivo análisis [Almira and Domingo, 2016].

En base a lo mencionado, existen tres aspectos muy importantes a tener en cuenta al momento de utilizar este tipo de técnicas, las mismas que se abordan a continuación.

(a) Electroodos

La actividad eléctrica recogida por los electrodos se encuentra en el rango de los microvoltios, por lo que las intensidades de las ondas cerebrales registradas desde la superficie del cuero cabelludo están alrededor de

0 a 200 microvoltios, y a nivel de frecuencias van desde una vez por segundo hasta 50 o más por segundo [Hall and Hall, 2020]. La medición de las señales EEG dependen de la calidad de contacto de los electrodos sobre la corteza cerebral. Una mala recepción por parte de los electrodos pueden afectar considerablemente la calidad de la señal del EEG, por lo que se distinguen dos tipos de electrodos: secos (poseen un metal que actúa como conductor entre la piel y el electrodo) y húmedos (utilizan de un gel conductor para establecer el contacto entre la piel y el electrodo), siendo estos últimos los más utilizados [Almira and Domingo, 2016].

(b) **Sistema Internacional 10-20**

Es un sistema de posicionamiento de electrodos que se basa en la relación que existe entre la localización de un electrodo y la región del cerebro que hay bajo la misma. Los números 10 y 20 indican la distancia de separación entre electrodos adyacentes que va desde el 10% o el 20% de la distancia total en relación a determinadas referencias del cráneo [Almira and Domingo, 2016].

El sistema internacional 10-20 maneja una nomenclatura indicada en la Tabla 3.1, en la que a cada lóbulo se le asigna una letra para su identificación, a su vez, a los hemisferios del cerebro se los representa mediante números.

Hemisferio Derecho: Números Pares [2,4,6,8]

Hemisferio Izquierdo: Números Impares [1,3,5,7]

Tabla 3.1: Nomenclatura Sistema Internacional 10-20

Electrodo	Lóbulo
F	Frontal
T	Temporal
C	Central
P	Parietal
O	Occipital
AF	Frontal Anterior

La letra Z también es utilizada para hacer referencia a la ubicación de un electrodo en la línea central de la corteza cerebral.

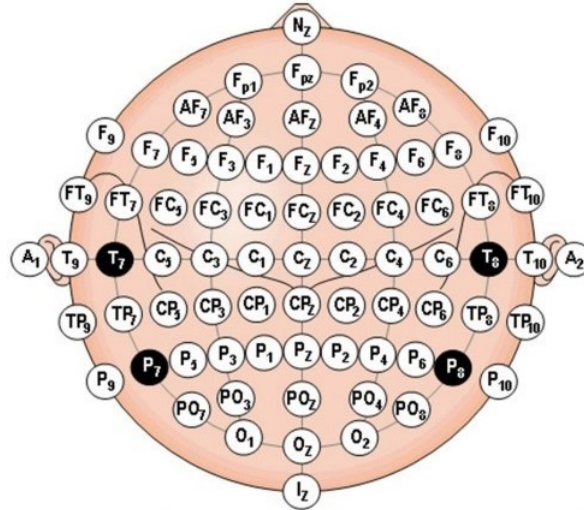


Figura 3.2: Sistema Internacional 10-20. Posición de los electrodos

En la Figura 3.2 se muestra la distribución de los electrodos mediante el sistema internacional 10-20. Se puede apreciar que las letras y números van acompañados, indicando de esta manera su posición precisa sobre el cuero cabelludo de la persona.

(c) Artefactos en las señales EEG

Son señales no deseadas producidas por diversas fuentes de ruido, principalmente generadas por actividades fisiológicas propias de cada persona, tales como: latidos del corazón, parpadeos, movimientos musculares, etc [Lee et al., 2013]. De igual manera, este tipo de señales pueden ser provocadas por fuentes externas, tales como: ruidos en los electrodos, interferencias electromagnéticas, el entorno, etc.

3.2. Sistemas BCI

Los Sistemas de Interfaz Cerebro Computadora, más conocidos como sistemas BCI (Brain Computer Interface, por sus siglas en inglés), son el resultado de una combinación de hardware y software que permiten traducir la actividad eléctrica del cerebro humano para convertirlos en comandos de control hacia los dispositivos externos del mundo real.

Entre los aspectos más importantes a tomar en cuenta durante el desarrollo de un sistema BCI, son los que se mencionan en la presente sección del documento.

3.2.1. Medición de la actividad cerebral

(a) Métodos invasivos

Este procedimiento de medición se basa en la implantación de electrodos directamente sobre la corteza cerebral, es decir, bajo el cráneo, requiriendo de un procedimiento quirúrgico, donde al estar los electrodos cercanos a las áreas del cerebro, la señal medida es de alta calidad, existiendo mínima afectación a cualquier tipo de artefacto.

(b) Métodos no invasivos

Este procedimiento de medición se basa en la colocación de electrodos sobre la superficie del cuero cabelludo, como se puede ver en la Figura 3.3, por lo que no requiere de ningún tipo de intervención quirúrgica sobre la persona.

Por su parte, la calidad de la señal medida es relativamente baja y está propensa a algún tipo de artefacto.



Figura 3.3: Medición de la actividad cerebral. (Método no invasivo)

3.2.2. Fases de los sistemas BCI

Para que un sistema BCI logre cumplir su propósito general, es necesario que este tipo de sistemas pasen por una serie de fases secuenciales, indicadas en la Figura 3.4, en la que cada fase tiene sus propias características e importancias sobre el mismo.

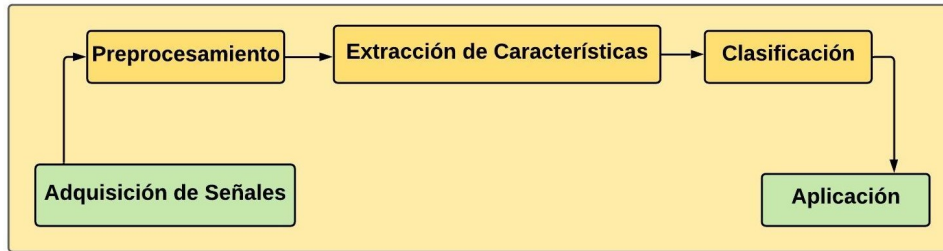


Figura 3.4: Fases de los sistemas BCI

(a) **Adquisición de señales**

La fase inicial de todo sistema BCI es la adquisición de señales, la misma que hace referencia al uso del protocolo del sistema internacional 10-20 para la ubicación de los electrodos sobre el cuero cabelludo, además, esta fase requiere la utilización de algún dispositivo comercial de captura de señales EEG.

(b) **Preprocesamiento de señales**

Realiza un tratamiento previo a las señales obtenidas, a fin de eliminar cualquier artefacto que se haya presentado durante el proceso de adquisición de las señales, está relacionada al uso de filtros que permitan eliminar ruidos y sobre todo identificar las señales de interés, preparando estas señales y facilitando los posteriores procesos a las siguientes fases de los sistemas BCI, conduciendo a tener una mejor separabilidad de características y una óptima clasificación [Hassanien and Azar, 2015].

(c) **Extracción de características**

Esta fase se centra en generar un vector de características que almacene información útil, a fin de proveer un conjunto relevante de características a la siguiente fase de clasificación. Estas características vienen representadas sea por medidas de la magnitud de la señal, medidas estadísticas de la frecuencia de los datos, medidas de conectividad entre diferentes regiones de interés del cerebro, etc. Por ende, esta fase conlleva a la maximización de la exactitud por parte del algoritmo de clasificación que se emplee, proporcionando un óptimo rendimiento en general sobre la totalidad del sistema BCI [Hassanien and Azar, 2015].

(d) Clasificación

La fase de clasificación es considerada la más trascendental en un sistema BCI, dado que a través de la misma el sistema asocia la intención del usuario a partir de estrategias de señales de control para comandar una acción específica.

De acuerdo a [Graumann et al., 2009], el rendimiento de un sistema BCI viene dado en base a la precisión de clasificación del mismo, el cual se basa en la relación que existe entre el número de ensayos clasificados de manera correcta (intentos exitosos de realizar las tareas mentales requeridas) y el número total de ensayos realizados. De igual manera, se puede determinar el error en base a la proporción de ensayos clasificados de manera incorrecta y el número total de ensayos.

En la actualidad, existen diversos algoritmos de clasificación empleados para procesos de clasificación de señales, entre ellos se pueden mencionar: análisis discriminante lineal (LDA, por sus siglas en inglés), máquinas de vectores de soporte (SVM, por sus siglas en inglés), clasificador bayesiano, árbol de decisiones, redes neuronales artificiales, k vecinos más cercanos (kNN, por sus siglas en inglés), etc.

(e) Aplicación

La integración de manera secuencial de cada una de las etapas anteriormente mencionadas, permiten llegar a esta última fase de los sistemas BCI, en la que se ejecuta una acción de control sobre un dispositivo físico del mundo real, por ejemplo: silla de ruedas, computadora, brazo robótico, etc.

3.2.3. Estrategias de señales de control

Un sistema BCI para que pueda descifrar la intención a realizar por parte de la persona, a más de medir la actividad del cerebro necesita apoyarse de ciertos patrones cerebrales conocidos como estrategias mentales, las mismas que constituyen una parte clave en el proceso de comunicación de todo sistema BCI, ya que determinan lo que la persona debe realizar para generar patrones cerebrales a fin de que el sistema BCI pueda interpretarlos y asociarlos para comandar una determinada acción de control [Graumann et al., 2009].

Las estrategias de señales de control por lo general se clasifican en tres grupos.

- **Señales Evocadas**

Son señales generadas de forma inconsciente por la persona, producidas al medir la actividad cerebral como respuesta a estímulos externos.

- **Señales Espontáneas**

Son señales generadas de forma voluntaria por la persona, por lo que no necesitan de algún tipo de estimulación externa, es decir, son producidas en base a la emisión de algún tipo de pensamiento o expresión facial.

- **Señales Híbridas**

Son utilizadas como un interruptor cerebral, cuando una de las tareas es indicar que la persona no tiene la intención de realizar alguna acción de control [Amiri et al., 2013].

3.3. Dispositivos y software comerciales de adquisición de señales EEG

En la Tabla 3.2 se establece una comparativa entre algunos dispositivos de adquisición de señales EEG, que son comercializados por diferentes fabricantes en la actualidad principalmente para usos investigativos.

Tabla 3.2: Dispositivos comerciales de adquisición de señales EEG

Fabricante	Dispositivo	Costo (\$)	Número de Electroodos	SDKs (Costo \$)
Neurosky	Mindwave	99,99	1+1	NeuroView (499,99)
Emotiv	EPOC+	799,99	14+2	Cortex API (29,00/mes)
Emotiv	Insight	299,99	5+2	Cortex API (29,00/mes)
Interaxon	Muse	299,00	4+3	MuseLab (Libre)
OpenBCI	Ganglion	199,99	4	OpenBCI GUI (Libre)

En base a esta información, para la selección del dispositivo de EEG del presente proyecto a cumplirse, se ha tomado en cuenta el número de electrodos necesarios, acceso al kit de desarrollo para manejo de flujos de datos y a su vez se ha considerado aspectos económicos y de accesibilidad, por lo que se ha seleccionado el dispositivo Emotiv Insight como instrumento para la adquisición de señales EEG para el desarrollo del proyecto.

3.3.1. Emotiv Insight

La empresa Emotiv actualmente comercializa diferentes dispositivos de captura de EEG, uno de los más populares es la diadema Emotiv Insight, la misma que integra su respectivo kit para establecer una comunicación hacia un computador, por lo que en la Figura 3.5 se muestra dicho kit.

Se trata de un auricular inalámbrico de 5 canales y 2 de referencia, fabricado para usos investigativos en señales EEG, ver Figura 3.5(a).

Este auricular está diseñado en base al sistema internacional 10-20, permitiendo que los electrodos se ubiquen en las posiciones correctas sobre el cuero cabelludo de la persona para un óptimo registro de las señales, siendo los electrodos de material hecho a base de polímeros. A su vez, el auricular dentro de su Electrónica integra sensores de movimiento de 9 ejes para detectar movimientos de la cabeza [Emotiv, 2020].

La forma de conexión de la diadema Emotiv Insight puede ser realizada de dos formas: vía Bluetooth o mediante la utilización de un receptor USB propio del fabricante, ver Figura 3.5(b).



(a) Diadema Emotiv Insight



(b) Receptor USB

Figura 3.5: Kit Emotiv Insight

Este receptor USB permite establecer una comunicación de forma inalámbrica, mediante una conexión Bluetooth entre la diadema Emotiv

Insight y el computador, permitiendo que las señales EEG puedan ser transmitidas hacia el computador para su visualización o su monitoreo respectivo a través de flujos de datos.

En la Tabla 3.3 se resume las principales características del dispositivo Emotiv Insight [Emotiv, 2020].

Tabla 3.3: Características técnicas de la diadema Emotiv Insight

Características	Emotiv Insight
Número de canales	5+2 de referencia
Nombre de los canales	AF3, AF4, T7, T8, Pz
Frecuencia de muestreo	128 Hz
Resolución de EEG	14 bits
Respuesta de frecuencia	0.5 - 43 Hz
Rango dinámico	8400 uV (pp)
Sensores de movimiento	Sensor de 9 ejes: 3x Giroscopios 3x Acelerómetros 3x Magnetómetros
Conectividad	Tecnología inalámbrica Bluetooth Receptor USB 2.4 GHz
Batería	Polímero de litio de 480 mAh 8 horas usando receptor USB
Detecciones	Comandos mentales Métricas de rendimiento Expresiones faciales

3.3.2. EmotivPRO

A nivel de software, la empresa Emotiv cuenta con su propio software llamado EmotivPRO, diseñado para tareas investigativas y de neurociencia, mediante la utilización de sus propios dispositivos tales como: INSIGHT, EPOC+ y EPOC Flex [Emotiv, 2020].

A continuación, se indican las principales características del software EmotivPRO en su versión gratuita [Emotiv, 2020].

- Flujos de datos (calidad de contacto de los sensores, señales EEG sin procesar, métricas de rendimiento, sensores de movimiento, paquetes de datos, análisis de frecuencias FFT y banda de potencia).
- Visualización en tiempo real de las señales EEG de manera gráfica.
- Acceso de múltiples perfiles de usuarios.
- Grabaciones de los flujos de datos para su posterior reproducción o exportación (licencia).
- Marcadores de eventos de interés sobre los flujos de datos.

EmotivPRO es un software licenciado, por lo que los desarrolladores que requieran acceder a los datos (valores) de forma individual de cada electrodo, sea en tiempo real o a través de grabaciones para su exportación en formatos de archivos .csv o .edf, en base a alguno de los flujos de datos mencionados anteriormente, es necesario que se disponga de alguna de las licencias que oferta la empresa Emotiv.

3.3.3. API Cortex

Emotiv en la actualidad a través de cada una de sus licencias, permite a desarrolladores acceder a su Kit de Desarrollo de Software (SDK, por sus siglas en inglés), conocido como API Cortex, el mismo que constituye ser la pieza central de la tecnología Emotiv que permite al usuario interactuar con los sistemas BCI.

La potencialidad de este servicio radica en la posibilidad que tiene el desarrollador de hacer uso de flujos de datos en tiempo real, principalmente de las señales EEG mediante los valores medidos por cada canal, o a su vez los valores registrados por los sensores de movimiento, permitiendo al desarrollador la creación de aplicaciones personalizadas.

El servicio API Cortex está construido sobre el protocolo WebSockets y formato JSON, permitiendo a través de estos que variedades de lenguajes y entornos de programación puedan acceder a los flujos de datos.

Uno de los requisitos básicos a más de tener acceso a una determinada licencia, es el de poseer conocimientos en lenguajes de programación, dado que a través de los mismos el desarrollador puede acoplar estos flujos de datos para su posterior desarrollo.

3.4. Redes neuronales artificiales

Dentro del campo de la Inteligencia Artificial, las redes neuronales artificiales se han constituido en herramientas con alto grado de aplicabilidad, las mismas que son modelos matemáticos inspiradas en emular el comportamiento del cerebro humano, basadas en el aprendizaje a través de la experiencia y a su vez en la extracción del conocimiento genérico a partir de un conjunto de datos [López et al., 2008].

Las redes neuronales artificiales son uno de los algoritmos de clasificación más utilizados en el mundo de los sistemas BCI al momento de buscar clasificar determinadas tareas mentales, expresiones faciales, etc.

Algunas de las características más sobresalientes de las redes neuronales artificiales se describen a continuación:

- **Aprendizaje adaptativo:** Uno de los aspectos claves y representativos en este tipo de algoritmos de clasificación, radica en su forma de aprendizaje, ya que el comportamiento de una red neuronal viene dada en función de un proceso de entrenamiento previo, conseguido a través de un conjunto de informaciones sean ejemplos, datos, imágenes, etc., al punto de permitir que la red neuronal tenga la capacidad de aprender y asociar nueva información.
- **Auto-organización:** Durante el proceso de aprendizaje las redes neuronales organizan la información recibida en base a su diseño, por lo que la característica de auto-organización se centra en la habilidad que tiene una red neuronal en responder de manera correcta, ante un conjunto de informaciones a las que no fue expuesto anteriormente, dando lugar a la generalización como medio para responder adecuadamente cuando la información de entrada es poco clara o incompleta.
- **Operación en tiempo real:** Una de las fortalezas de las redes neuronales artificiales, es la posibilidad de permitir que aplicaciones puedan manejar un conjunto de datos y estos puedan ser clasificados en tiempo real, ya que al ser mínimo la actualización de los pesos entre cada una de las conexiones posibilita que estas puedan trabajar en tiempo real.

3.4.1. Elementos de una red neuronal

La arquitectura de una red neuronal viene representada de manera general por capas, en las que se maneja un número determinado de neuronas,

siendo estas el mecanismo que permiten que se procese la información desde la entrada hasta la salida, realizando una interconexión entre las mismas formando la red neuronal.

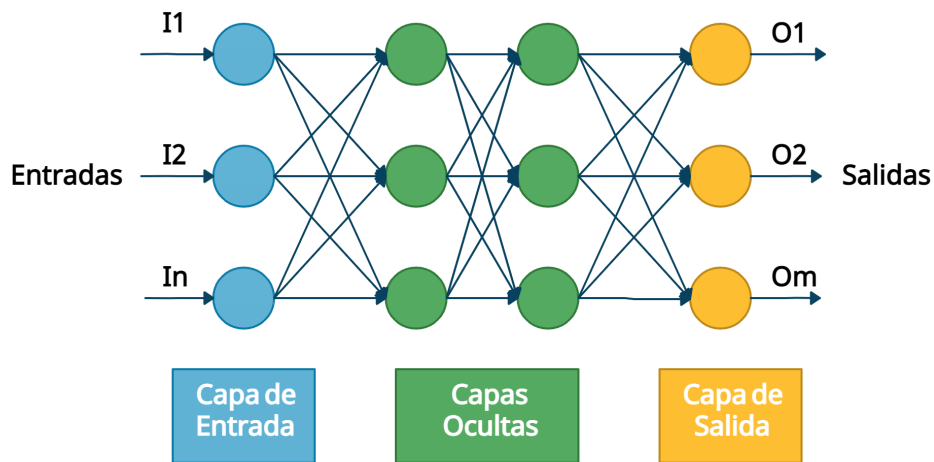


Figura 3.6: Estructura de una red neuronal

En la Figura 3.6 se puede apreciar la arquitectura básica de una red neuronal, en la que se distinguen 3 capas: de entrada, ocultas y de salida; en la que además se visualiza la interconexión que se crea entre cada una de las neuronas de una capa con las neuronas de las capas subsiguientes.

- **Capa de entrada:** Se encarga de recibir información de su entorno proveniente de sensores o de vectores que manejan patrones de aprendizaje.
- **Capas ocultas:** Se caracterizan por ser internas a la red, por lo que se las conoce también como capas intermedias, estas no tienen conexión directa con el entorno, a su vez pueden constar de múltiples capas dentro de la arquitectura de la red neuronal, a diferencia de la capa de entrada y de salida que solo poseen una capa.
- **Capa de salida:** Emite información de la red hacia el exterior, es decir, proporciona la respuesta de la red neuronal.

Por su parte, en el proceso de aprendizaje se realizan una serie de ajustes y cálculos que se realizan internamente, por lo que en la Figura 3.7 se muestran los componentes que intervienen en dicho proceso:

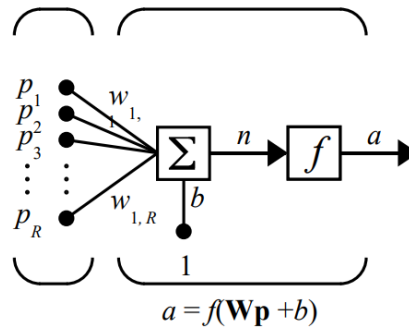


Figura 3.7: Componentes de una red neuronal

Donde:

$p_1, p_2, p_3, \dots, p_R$	Elementos de entrada
$w_{1,1}, w_{1,2}, \dots, w_{1,R}$	Pesos
b	Bias
f	Función de activación
a	Salida

Este proceso de ajuste y cálculos giran en torno a encontrar los pesos ideales entre las distintas conexiones que se crean en la red, distinguiendo dos funciones que son descritas a continuación:

- **Función de entrada:** Se calcula a partir de un vector de entrada, es decir, los valores de entrada de dicho vector se multiplican por los pesos anteriormente ingresados a la neurona.

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b \quad (3.1)$$

De acuerdo a la Figura 3.7, la neurona tiene un bias que se suma con las entradas ponderadas para formar la entrada de la red mediante n como se indica en 3.1. El mismo que viene a ser el argumento de la función de activación.

- **Función de activación:** La función de activación calcula el estado de actividad de una neurona, por lo que se basa en combinar la entrada

total a la n -ésima neurona obtenido a partir de los estímulos y pesos recibidos con el estado inicial de la neurona, para producir un nuevo estado de activación acorde con la nueva información recibida [López et al., 2008]. Se utiliza para normalizar la salida de una neurona en términos de nivel de actividad de su entrada y por ende permitir que se obtengan resultados lo más cercano posible a los esperados, por lo que es aplicada a la salida del bloque sumador, ver Figura 3.7.

En la Tabla 3.4 se indican algunas de las funciones de activación más utilizadas y sus respectivos rangos.

Tabla 3.4: Funciones de activación de las redes neuronales

	Función	Rango
Identidad	$f(x) = x$	$[-\infty, +\infty]$
Escalón	$f(x) = \begin{cases} 0, & \text{si } x < 0 \\ 1, & \text{si } x \geq 0 \end{cases}$	$[-1, +1]$ o $[0, +1]$
Lineal a tramos	$f(x) = \begin{cases} -1, & \text{si } x < -1 \\ x, & \text{si } -1 \leq x \leq +1 \\ +1, & \text{si } x > +1 \end{cases}$	$[-1, +1]$
Sigmoidea	$f(x) = \frac{1}{1 + e^{-x}}$	$[0, +1]$
Tangente Hiperbólica	$f(x) = tgh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$[-1, +1]$
ReLU	$f(x) = max(0, x) = \begin{cases} 0, & \text{si } x < 0 \\ x, & \text{si } x \geq 0 \end{cases}$	$[0, +\infty]$

3.4.2. Mecanismos de aprendizaje

El aprendizaje constituye un factor importante dado que representa el proceso con el que la red neuronal crea, modifica o destruye sus conexiones (pesos) en respuesta a una información de entrada [López et al., 2008]. En virtud de esto, los pesos de las conexiones entre cada una de las neuronas representan la forma en cómo se produce el conocimiento, por lo que de acuerdo al tipo de aprendizaje que se siga permitirá que el valor de los pesos

de cada neurona que forman la red se modifiquen, hasta que estos adopten un valor constante que permitan alcanzar la salida deseada, dando a entender que la red ha aprendido.

Por lo que el proceso de aprendizaje se basa en una serie iterativa de entrenamientos mediante patrones, distinguiéndose tres tipos de aprendizajes que se describen a continuación.

- **Aprendizaje supervisado:** Se caracteriza porque la red dispone de información tanto a la entrada como a la salida, es decir, conoce los patrones de entrada y sus respectivas salidas deseadas para dichas entradas, por lo que durante este tipo de aprendizaje el proceso de entrenamiento es controlado por un supervisor.
- **Aprendizaje no supervisado:** Se caracteriza porque la red dispone únicamente de información de sus entradas, por lo que al no disponer de salidas deseadas ante determinados patrones de entrada, permite que la red se autoorganice clasificando estas entradas en función de asociar parámetros tales como similitudes, características comunes, etc. Es así que, los pesos de la red fluctúan de manera libre al no estar sujetos a algún tipo de salida deseada, de tal manera que se genera una salida en la red cuando los pesos demuestran tener cierto grado de estabilidad.
- **Aprendizaje híbrido:** Es el resultado de la combinación del aprendizaje supervisado y no supervisado, programando a la red a que aprenda mediante un conjunto de datos de entrada basado en el aprendizaje supervisado, mientras que los pesos restantes son obtenidos a través del aprendizaje no supervisado.

3.5. Entornos de programación y el algoritmo de Kalman

Uno de los aspectos claves que radica en todo sistema BCI, es la manera en la que se va a procesar toda la información recopilada por los dispositivos de captura EEG, por lo que es necesario la integración de herramientas informáticas que manejen entornos y lenguajes de programación adecuados para cumplir dicho proceso. Es así que, haciendo referencia al uso de la diadema Emotiv Insight, se ha considerado utilizar un lenguaje de programación que sea compatible con la API Cortex de Emotiv, a fin de recibir los flujos de datos de las señales EEG en tiempo real para su respectivo procesamiento.

3.5.1. Python

Es un lenguaje de programación de alto nivel, con la particularidad de ser de tipo interpretado, multiparadigma, libre y gratuito; lo que hace que hoy en día sea uno de los más empleados por sus grandes capacidades en campos de la informática e ingeniería, por ejemplo, una de sus grandes potencialidades radica en el enfoque que tiene hacia la Inteligencia Artificial al poseer librerías que facilitan el uso del aprendizaje automático.

Las principales características que dispone Python son las siguientes:

- Lenguaje de programación de alto nivel (Programación orientada a objetos).
- Sintaxis clara y flexible.
- Software libre, multiplataforma.
- Manejo de librerías para cubrir diferentes aplicaciones por parte del programador.
- Librerías con enfoque hacia la Inteligencia Artificial para el uso del Machine Learning.
- Capacidad de integración con el SDK de Emotiv (API Cortex), para manejo de datos en tiempo real.
- Desarrollo de interfaces gráficas de usuario.

3.5.2. Filtro de Kalman

Es un método recursivo que se basa en la estimación de un sistema en el instante actual (t), a partir de la información disponible en el instante anterior ($t - 1$), a su vez, permite conciliar los datos proporcionados por los sensores, los mismos que vienen afectados por el denominado ruido blanco.

El Filtro de Kalman de tipo unidimensional viene definido por 3 parámetros, los mismos que intervienen dentro del algoritmo recursivo que utiliza en cada iteración que se realiza.

1. Ganancia de Kalman (K_n)

Representa cuánto cambia la estimación mediante una medición, su valor oscila entre 0 y 1, su funcionalidad radica en la de dar más importancia al valor medido o al valor estimado, mediante la comparación de los errores de ambos, viene dada por [3.2](#).

$$K_n = \frac{p_{n,n-1}}{p_{n,n-1} + r_n} \quad (3.2)$$

Donde:

$p_{n,n-1}$: Error de estimación previa

r_n : Error de medición

2. Estimación del estado actual ($\hat{x}_{n,n}$)

Es el valor de salida, es decir, el cálculo de la variable que rige el comportamiento del sistema, viene dada por [3.3](#).

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n(Z_n - \hat{x}_{n,n-1}) \quad (3.3)$$

Donde:

$\hat{x}_{n,n}$: Estimación de x en el tiempo n

$\hat{x}_{n,n-1}$: Estimación previa de x que se realizó en el tiempo $n - 1$

Z_n : Valor de la medición en el instante n

3. Error en la estimación actual ($p_{n,n}$)

Es un indicativo de cómo de próximos están los valores estimados a los medidos, viene dada por [3.4](#).

$$p_{n,n} = (1 - K_n)p_{n,n-1} \quad (3.4)$$

Donde:

K_n : Ganancia de Kalman

$p_{n,n-1}$: Error de estimación previa

$p_{n,n}$: Error en la estimación actual

Capítulo 4

Desarrollo del Sistema

En este capítulo se desarrollará el sistema BCI propuesto en su totalidad, en base al acoplamiento de herramientas de hardware y software seleccionadas, tomando como punto de partida el manejo de la diadema Emotiv Insight como dispositivo de captura de señales EEG, a su vez la familiarización con su respectivo software comercial EmotivPRO; para posteriormente abarcar la metodología empleada en cada una de las fases del sistema BCI propuesto, que permitirán relacionar las señales de control a utilizarse con las acciones de control buscadas (click izquierdo, click derecho y ninguna).

De igual manera, se indicarán los aspectos tomados en consideración para lograr el manejo del cursor del mouse a base de movimientos de la cabeza por parte del usuario.

Finalmente, se diseñará una interfaz de usuario de tipo educativa, orientada principalmente a personas que tengan algún tipo de limitación física en sus extremidades superiores, a fin de que permita validar el sistema BCI desarrollado.

4.1. Requerimientos previos

Para el desarrollo en su totalidad del sistema BCI, se hará uso de los siguientes componentes, los mismos que fueron seleccionados en base a las características técnicas y funcionalidades analizadas en el capítulo anterior, por lo que a continuación se justifica sus respectivas utilidades en el desarrollo del sistema.

- **Diadema Emotiv Insight:** Dispositivo que permitirá la captura de las señales EEG, durante todo el proceso de desarrollo del sistema.
- **EmotivPRO:** Software comercial propio de la diadema Emotiv Insight, que permitirá al usuario realizar grabaciones de las señales EEG, entrenamientos virtuales mediante detecciones de comandos mentales o expresiones faciales, además de la visualización en tiempo real de los flujos de datos de cada electrodo de manera gráfica.
- **Licencia Emotiv Student:** Se hará uso de esta licencia al ser la más económica que oferta actualmente la empresa Emotiv, cuyo precio se encuentra en los \$29 mensuales, además que las características que brinda son suficientes para el desarrollo del sistema a cumplirse, dado que la misma proporcionará el acceso a los datos de forma individual de cada electrodo, sean en tiempo real o mediante la exportación a través de grabaciones en formato .csv de archivos de Excel.
- **API Cortex:** El servicio SDK que ofrece la empresa Emotiv a todo desarrollador que posee algún tipo de licencia contratada, permitirá acceder al uso de los flujos de datos de los electrodos y sensores de movimiento en tiempo real, mediante la utilización de algún tipo de lenguaje de programación.
- **Python:** Lenguaje de programación necesario para la comunicación con la API Cortex de Emotiv, y a su vez, para el desarrollo de cada una de las fases del sistema BCI propuesto.

Como punto de partida del desarrollo del sistema, se ha considerado una etapa de preparación previa a la realización del experimento, en la que se requiere que los usuarios que sirvan de objeto de prueba para el desarrollo del experimento y validación del sistema, cumplan de dos requisitos fundamentales y necesarios, los mismos que se abordan en los apéndices del presente documento, ordenados de la siguiente manera:

Uso de la diadema Emotiv Insight (ver apéndice [A](#)).

Entrenamientos virtuales (ver apéndice [B](#))

- **Preparación:** El participante toma una postura firme, adoptando una posición cómoda y sobre todo manteniendo un estado de relajación.
- **Guiños de ojos:** El participante debe centrarse en generar los guiños de ojos correspondientes al tipo de grabación a realizarse, siendo una duración aproximada de 1 segundo por cada guiño de ojo, tratando de seguir en lo posible la secuencia mostrada en la Figura 4.1.
- **Neutral:** El participante durante este lapso de tiempo de alrededor de 5 segundos, debe dejar de hacer los guiños correspondientes y centrarse en mantener un estado neutral. En este tiempo el participante puede mover los ojos, pestañear involuntariamente, etc., tratando de seguir en lo posible la secuencia mostrada en la Figura 4.1.
- **Descanso:** El participante puede descansar a su manera, dado que la grabación del experimento finalizó.

4.3. Etapas y metodología de diseño

Las etapas a cumplirse para el desarrollo del sistema BCI propuesto se resume en la Figura 4.2, en la que se muestra cada una de las fases consideradas a seguir para la elaboración total del algoritmo a ejecutarse en el presente proyecto.

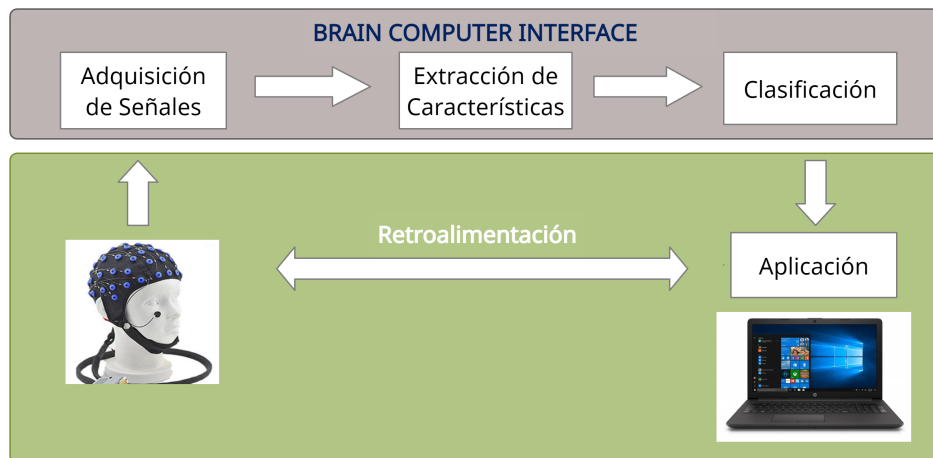


Figura 4.2: Etapas de diseño a cumplirse del sistema BCI

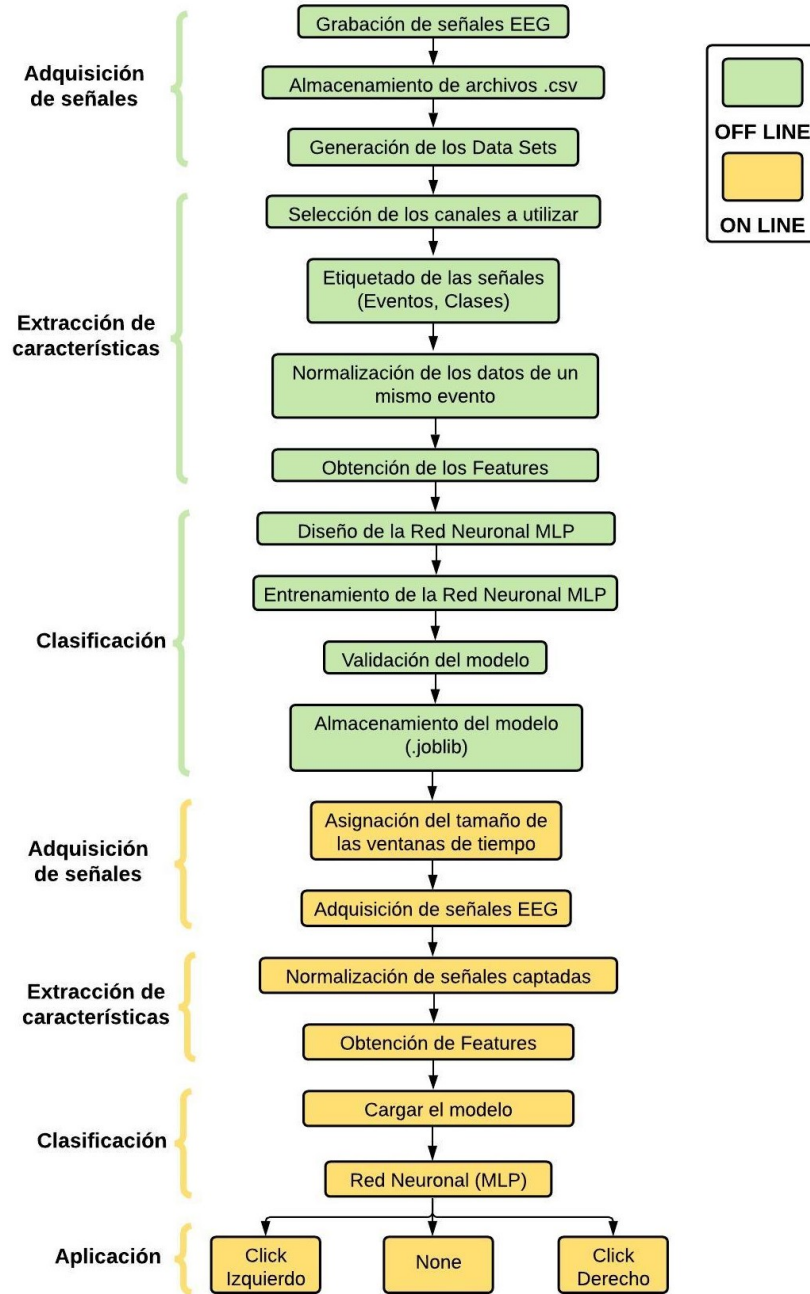


Figura 4.3: Diagrama general del algoritmo a desarrollarse

La metodología que se aplicará se centra en el uso del algoritmo de clasificación de redes neuronales artificiales, por lo que se aplicará el aprendizaje automático supervisado orientado a problemas de clasificación, con el objetivo de predecir a que clases pertenecen un conjunto de datos. Por lo que la clasificación será de tipo multiclase, a partir de conjuntos de datos que provienen de actividades en las que los participantes realicen con sus ojos tareas como: guiño izquierdo, guiño derecho y cualquier otra acción que será identificada como none (ninguna). En la Figura 4.3 se indica el flujo de trabajo a seguir durante el diseño del sistema BCI, a fin de desarrollar el modelo de clasificación que permitirá posteriormente identificar las tareas que realicen los participantes en tiempo real y convertirlas en acciones de control sobre los eventos del mouse como son click izquierdo y click derecho, dependiendo del guiño de ojo realizado por los participantes de prueba.

Posteriormente, se acoplará al sistema desarrollado el control sobre el cursor del mouse en base a los movimientos de cabeza que realice el usuario, cuya metodología se indicará en secciones posteriores.

4.4. Diseño del sistema BCI

En base a la metodología de trabajo diseñada y mostrada en la Figura 4.3, se hará uso de dos modos de trabajo los mismos que son descritos a continuación.

- **Modo OFF LINE:** Tiene como objetivo hacer un análisis de las señales obtenidas durante la realización de cada uno de los experimentos realizados sobre los participantes de prueba que se considere, haciendo uso de los archivos .csv (ver apéndice C), los mismos que permitirán desarrollar paso a paso las tres primeras etapas del sistema BCI, a fin de obtener el modelo de clasificación que pueda posteriormente clasificar nuevas señales provenientes de la actividad cerebral de los participantes captadas por la diadema.
- **Modo ON LINE:** Tiene como objetivo probar el modelo desarrollado en tiempo real, en la que a través de la herramienta de análisis de ventanas de tiempo, se irá tomando un determinado número de muestras sobre las señales EEG captadas a lo largo del tiempo, logrando que los participantes haciendo uso de la diadema puedan emitir los guiños de ojos y el modelo generado sea capaz de predecir en tiempo real las tres posibles acciones a clasificar: click izquierdo, click derecho o none.

4.4.1. Adquisición de señales

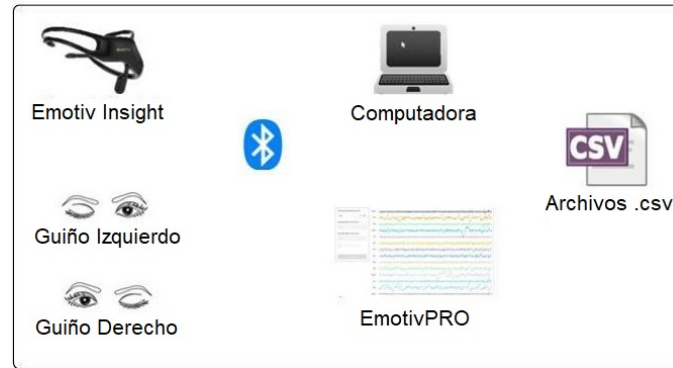


Figura 4.4: Proceso de adquisición de señales

El proceso de adquisición de señales se basa en lo mostrado en la Figura 4.4, en la que se aprecia las estrategias de señales de control definidas a los participantes (guiños de ojos), el tipo de comunicación (Bluetooth) que se realiza entre la diadema y el computador para la obtención de datos, la visualización en directo de forma gráfica de las señales (EmotivPRO), a su vez el formato de salida de las grabaciones realizadas (archivos .csv).

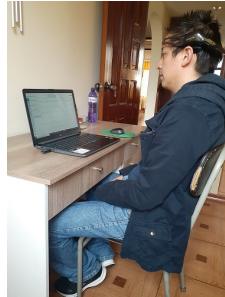


Figura 4.5: Participante durante la toma de señales

Para la adquisición de las señales se ha seleccionado un único participante, siendo el encargado de realizar las grabaciones de las señales EEG, por lo que al participante se le explica inicialmente el objetivo del experimento a efectuarse, y se le solicita cumplir la secuencia de dicho experimento, ver Figura 4.1, en la que el experimento se lo realiza en un total de 5 sesiones,

tanto para guiños izquierdos como para guiños derechos, a fin de tener una base de datos de respaldo.

Una vez recolectada las grabaciones obtenidas de cada una de las sesiones de grabación realizadas sobre el participante, se procede haciendo uso de EmotivPRO a exportar dichas grabaciones, las mismas que se exportan en formato de archivos .csv de Excel.

El apéndice C indica la información que contiene este tipo de archivos, posteriormente de todas las sesiones de grabación realizadas, se selecciona un solo data set de las cinco grabaciones de guiños derechos realizadas y de forma similar se obtiene un solo data set para guiños izquierdos.

Finalmente, estos dos data sets seleccionados (guiños izquierdos y guiños derechos), constituyen la materia prima de las demás etapas del sistema BCI, a razón de que sobre ambos data sets se trabajará de forma separada. Mediante Excel se procede a separar los datos de cada columna para una mejor interpretación, siendo la información más relevante las columnas (EEG.AF3, EEG.T7, EEG.Pz, EEG.T8, EEG.AF4), tal como se muestra en la Figura 4.6, en la que cada columna contiene la información recopilada por los cinco canales (electrodos) de la diadema (datos en crudo o señales puras), generándose n número de muestras cuyos valores están expresados en microvoltios (uV).

Column4	Column5	Column6	Column7	Column8
headset type	headset serie	headset firm	channels:88	sampling rat
EEG.AF3	EEG.T7	EEG.Pz	EEG.T8	EEG.AF4
4209.743652	4230.256348	4152.820313	4153.846191	4084.615479
4207.692383	4218.974121	4151.282227	4158.974121	4088.205078
4201.538574	4205.128418	4147.179688	4158.461426	4090.769287
4198.974121	4212.820313	4150.256348	4153.846191	4093.846191
4199.487305	4206.666504	4153.846191	4135.384766	4098.974121

Figura 4.6: Separación de los datos de los cinco canales EEG

4.4.2. Extracción de características

Una vez generado los data sets correspondientes para ambos tipos de guiños, el siguiente paso constituye en generar los features (patrones) más representativos de las señales obtenidas en cada data set.

Por lo que el objetivo principal de esta etapa es generar un data set final que reúna un determinado número de ejemplos con sus respectivos features

(atributos) de las señales procesadas, el mismo que servirá como información de entrada para la red neuronal durante la fase de entrenamiento de la misma.

En primera instancia, previo al desarrollo del experimento considerado, mediante exploración al visualizar de forma gráfica el comportamiento de los cinco canales ante la acción de movimientos de ojos, se determinó que se hará uso de dos canales (AF3 y AF4), dado que los mismos son los que más información aportan sobre las señales a procesarse ante estos estímulos.

Dichos estímulos giran en torno a la acción de guiños de ojos para una mejor aplicabilidad en el manejo del ordenador, por lo que en esta etapa se indicará el procedimiento tomando como referencia el data set de guiños derechos, dado que el procedimiento es similar para el data set de guiños izquierdos.

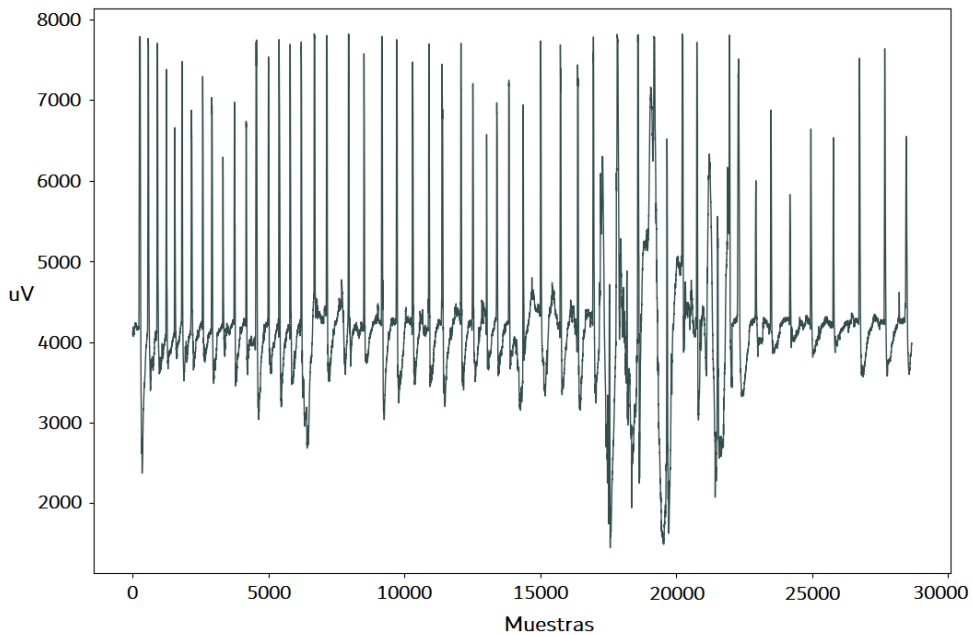


Figura 4.7: Gráfica resultante del data set de guiños derechos

Utilizando un script de Python se procede a graficar todo el data set, obteniendo la Figura 4.7, de la que se deduce que está compuesta de cerca de 30000 muestras, y a su vez caracterizada la señal por varios picos que representan los instantes en que se realizaron guiños derechos y a su vez instantes cuando el participante estuvo en reposo.

Si bien la clasificación a realizarse sigue el aprendizaje supervisado, es necesario en la actual etapa identificar desde un inicio el tipo y número de clases totales a predecir, por lo que se procede a identificar los eventos generados durante toda la señal. Es así que, aplicando un zoom sobre la figura en mención, se puede apreciar de mejor manera que el comportamiento de la señal a lo largo del total de las muestras generadas, posee dos tipos de eventos: guiño derecho (intervalos donde se producen los picos, generados por la acción de que el participante en ese instante hizo un guiño derecho) y de reposo (intervalos donde la señal tiene la apariencia de estabilizarse, tomando la forma de ruidos por parpadeos involuntarios, los mismos que corresponden a los 5 segundos destinados en la secuencia de grabación en la que el participante mantiene un estado neutral), esto se puede apreciar en la Figura 4.8.

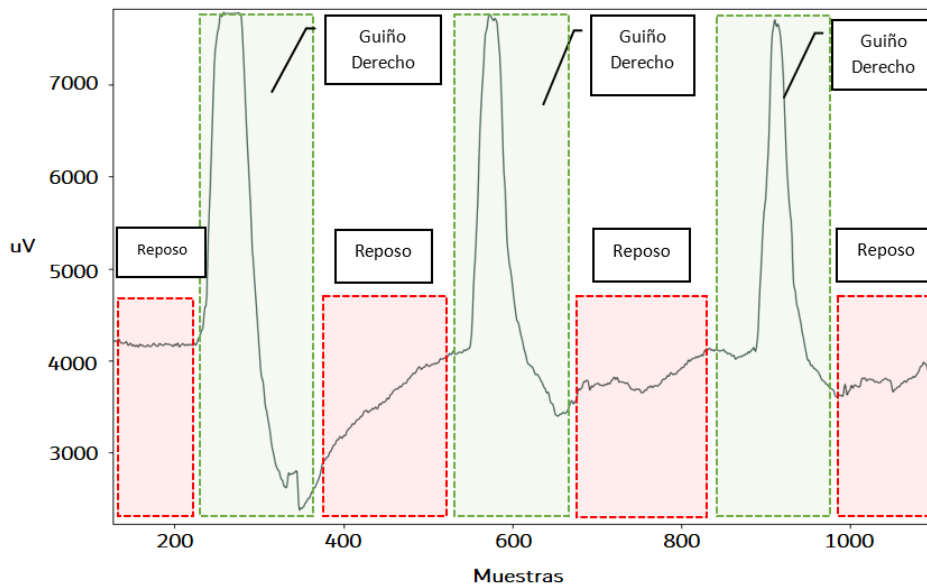


Figura 4.8: Determinación de eventos sobre un tramo de la señal

De igual manera, haciendo este mismo análisis para el data set de guiños izquierdos, de forma similar se determinó dos tipos de eventos: guiño izquierdo y reposo; por lo que, en base a este análisis realizado a ambos data sets, se determina que son un total de 3 clases las que se pretende predecir, por lo que, en la Tabla 4.1 se indica dichas clases y sus respectivos identificadores asignados para cada uno de ellos.

Tabla 4.1: Identificadores de las clases a predecir

Clase	Identificador	Observación
Guiño derecho	1	Acción de guiño
Guiño izquierdo	2	Acción de guiño
None	3	Pestañeos involuntarios

Adicional a estos identificadores, se hace uso de contadores los mismos que permitirán conocer la cantidad total de eventos que se identificó durante la totalidad de las muestras correspondientes al data set de análisis.

Estos eventos representan la cantidad de transiciones que existió de un estado a otro (reposo - guiño derecho - reposo, ...), en la que un evento puede contener n números de muestras dentro de las cuales pueden estar un guiño derecho o corresponder a un estado de reposo (none), como ejemplo utilizando la Figura 4.8, se puede apreciar que en ese intervalo seleccionado existen 7 eventos, distribuidos de la siguiente manera: 4 eventos de reposo y 3 eventos de guiño derecho.

Posteriormente, teniendo en cuenta los respectivos identificadores que se les asignó a las tres clases a predecir, se utiliza el archivo correspondiente al data set de guiños derechos y se crean dos nuevas columnas, ver Figura 4.9, asignándolas los nombres de Evento y Clase, las mismas que se irán llenando en el proceso de etiquetado.

1	Column4	Column5	Column6	Column7	Column8	Column9	Column10	
2	headset type headset serie headset firm channels:88 sampling rate:eeg_128;mot_64;p							
3	EEG.AF3	EEG.T7	EEG.Pz	EEG.T8	EEG.AF4	Evento	Clase	
4	4209.743652	4230.256348	4152.820313	4153.846191	4084.615479			
5	4207.692383	4218.974121	4151.282227	4158.974121	4088.205078			
6	4201.538574	4205.128418	4147.179688	4158.461426	4090.769287			
7	4198.974121	4212.820313	4150.256348	4153.846191	4093.846191			
8	4199.487305	4206.666504	4153.846191	4135.384766	4098.974121			

Figura 4.9: Columnas de Evento y Clase para el proceso de etiquetado

■ Series temporales y Ventanas de tiempo

Las series temporales son una sucesión de datos recopilados mediante observaciones de una variable recogidas en determinados periodos

secuenciales de tiempo, con la característica de que los datos pueden estar separados a intervalos iguales o desiguales.

Por otra parte, las ventanas de tiempo representan una herramienta eficaz para el análisis de señales o datos, donde su funcionalidad radica en la posibilidad de rastrear un determinado grupo de datos y ver como estos cambian a lo largo de un periodo de tiempo.

Haciendo uso de la metodología en la que se basan las series temporales y las ventanas de tiempo, se hace uso de ambas herramientas para el proceso de etiquetado de las columnas creadas denominadas Evento y Clase, en los dos data sets generados anteriormente.

La Figura 4.7 representa una serie temporal, dado que, el data set generado correspondiente al de guiños derechos de dicha figura, contiene un total de 28665 muestras recopiladas a lo largo del tiempo que duró la grabación realizada.

De igual manera, para determinar con exactitud los tramos sobre los cuales de las 28665 muestras generadas en su totalidad en el data set de análisis (guiños derechos), desde que número de muestra a que número de muestra corresponde un determinado evento sea de acción de guiño derecho o acción de reposo (none), se hace uso de ventanas de tiempo.

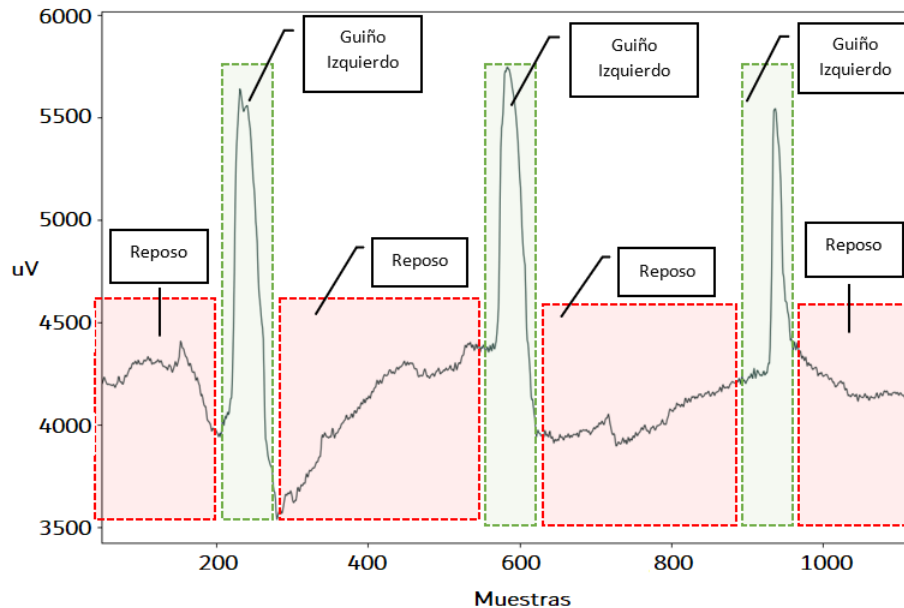


Figura 4.10: Ventanas de tiempo sobre un tramo de la señal

Por lo que, la Figura 4.10 muestra la manera en la que se utiliza las ventanas de tiempo para rastrear determinados tramos de la señal, en las que se visualizan ventanas para acciones de guiño izquierdo y de reposo.

■ Etiquetado de las señales

En base a lo mencionado de la utilidad de las ventanas de tiempo, este procedimiento permitirá ir determinando el número total de eventos generados a lo largo del total de muestras del data set de análisis, es decir, mediante una observación gráfica basada en la exploración de que al efectuar un guiño sea con el ojo izquierdo o con el ojo derecho, se generan señales pronunciadas con la característica de tener picos, por su parte, cuando el participante está en etapa de reposo, las señales tienen la apariencia de estabilizarse y ser tipo ruidosas; por lo que permite tener una referencia para saber desde que muestra a que muestra corresponde un evento y por ende su respectiva clase de acuerdo al identificador que se le asignó anteriormente.

205	4221.025879	4220.000000	4136.922852	4163.077148	4178.974121	0	3
206	4213.333496	4222.051270	4136.410156	4154.358887	4167.692383	0	3
207	4207.692383	4207.692383	4135.897461	4146.666504	4162.051270	0	3
208	4230.256348	4238.974121	4157.436035	4178.974121	4185.641113	0	3
209	4237.436035	4267.692383	4155.897461	4195.897461	4188.717773	0	3
210	4212.307617	4222.563965	4122.563965	4155.897461	4157.948730	0	3
211	4204.102539	4185.128418	4113.333496	4122.051270	4147.692383	1	1
212	4218.461426	4210.769043	4128.717773	4133.846191	4162.563965	1	1
213	4230.769043	4222.051270	4144.102539	4161.538574	4172.307617	1	1
214	4243.077148	4213.333496	4147.692383	4174.358887	4179.487305	1	1
215	4249.743652	4238.974121	4154.358887	4167.179688	4184.615234	1	1
216	4234.871582	4244.615234	4141.538574	4161.538574	4174.358887	1	1
217	4224.102539	4211.282227	4140.000000	4167.692383	4162.563965	1	1

Figura 4.11: Transición del evento 0 al evento 1 en el proceso de etiquetado

El etiquetado de las señales consiste en ir generando las dos columnas creadas anteriormente denominadas: Evento y Clase; por lo que haciendo referencia a la Figura 4.8 se puede apreciar que las primeras 210 muestras corresponden a la clase none (reposo), cuyo identificador es 3 correspondiéndole el contador de 0, indicando que es el primer evento encontrado de todo el conjunto del data set de análisis.

La Figura 4.11 muestra un ejemplo de la transición que ocurre de un evento a otro, donde se pasa del evento 0 al evento 1, por lo que de forma simultánea se procede a colocar el identificador de 1 correspondiente a la clase de guiño derecho. Dicho procedimiento se lo realiza a lo largo del total

de muestras de los dos data sets generados al inicio, de esta manera se puede conocer con certeza el número de eventos que componen cada data set.

■ Normalización de datos

La normalización consiste en ajustar un grupo de valores medidos en diferentes escalas, respecto a una nueva escala que sea común para todo el mismo grupo de valores.

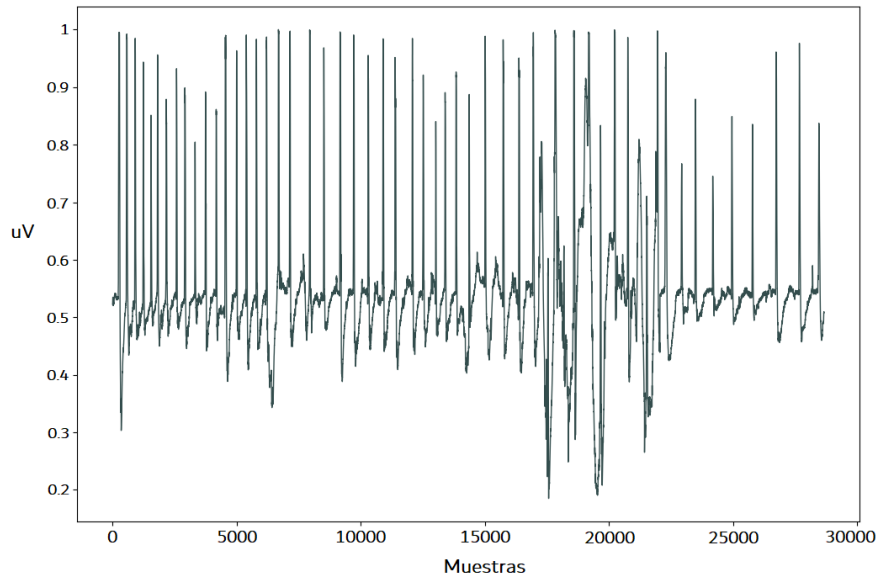


Figura 4.12: Gráfica del data set de guiños derechos normalizado

Tomando en cuenta que los valores promedios medidos por los canales considerados para el experimento (AF3 y AF4) bordean los 4200 uV, y que ante las acciones de guiños de ojos tienden a subir o bajar en referencia al valor promedio mencionado, se hace uso de la normalización sobre los datos recopilados que componen un mismo evento, tomando como referencia los canales de interés, a fin de mantener en una misma escala los datos de ambos canales y garantizar uniformidad sobre los mismos.

La Figura 4.12 tiene como finalidad mostrar una representación de como el data set original de guiños derechos ha sido normalizado, encontrándose en un rango de 0 a 1. La forma de utilizar la normalización se basa en ir analizando evento por evento a lo largo del total del data set de análisis, en la que en cada evento se procede a encontrar el valor máximo de todos

los datos que componen el canal AF3 en ese evento y de forma simultánea encontrar el valor máximo de todos los datos que componen el canal AF4 de ese mismo evento de análisis, a fin de entre ambos valores determinados identificar cual es el mayor y de esta manera utilizar dicho valor para que cada uno de los datos de ambos canales que componen un mismo evento sean divididos para el valor mayor encontrado, garantizando de esta forma que los datos de las columnas de los canales AF3 y AF4 de cada evento en cuestión se mantengan en un rango de 0 a 1.

■ Generación de los Features

Con los datos de ambos canales normalizados correspondientes a un mismo evento, el siguiente y último paso dentro de la etapa de extracción de características, consiste en generar los features para cada uno de los eventos que componen el data set de análisis, los mismos que proporcionarán la información más relevante extraída sobre las señales que fueron procesadas.

Por lo que, en esta etapa para buscar los features se hará uso de medidas de centralización (media, mediana, percentil 25) y medidas de dispersión (varianza, asimetría); las mismas que serán aplicadas sobre los datos normalizados respecto a un mismo evento, sobre los canales de interés (AF3 y AF4), dado que ambos presentan variaciones en forma simultánea ante efectos de guiños izquierdos o derechos.

Dichos features se explican a continuación, tomando como referencia el procedimiento a cumplirse sobre cada evento de análisis.

Media: Es el valor promedio de todos los valores de las muestras que definen un tramo de la señal en la que está identificado un evento.

Mediana: Es una medida central que divide en dos partes iguales la cantidad de muestras que forman un evento, es decir, su valor se encuentra por debajo del 50% de las muestras que componen un evento y a su vez se encuentra por encima del 50% de las demás muestras que componen dicho evento.

Percentil 25: Representa el cuartil del grupo de datos que componen un evento, es decir, es la medida tal que el 25% de datos son inferiores a su valor y el 75% de los datos son superiores.

Varianza: Es una medida de dispersión utilizada para representar como varían los datos correspondientes a un evento respecto al valor de la media de dichos datos del mismo evento en cuestión.

Asimetría: Mide el grado en que los valores comprendidos en el evento, tienden a distribuirse hacia un determinado lado con respecto a la media, es decir, si la distribución está sesgada a la derecha posee un (sesgo positivo) o si la distribución está sesgada a la izquierda tendrá un (sesgo negativo).

Por lo tanto, en base a estas cinco herramientas estadísticas mencionadas, se obtendrán cinco features por cada canal de interés respecto a un mismo evento. La Figura 4.13 indica una parte de los features obtenidos para el data set de guiños derechos, en la que cada fila está representada por un total de diez features obtenidos en cada evento del data set de análisis, definidos con la letra A para el canal AF3 y con la letra B para el canal AF4; a su vez la columna Tipo indica la clase a la que pertenece dicho evento.

1	MediaA	MedianaA	Percentil25A	VarianzaA	AsimetriaA	MediaB	MedianaB	Percentil25B	VarianzaB	AsimetriaB	Tipo
2	0,931826	0,932763	0,928761984	7,865E-05	0,5725683	0,899092	0,910814	0,877361641	0,0013342	-0,664339	None
3	0,553402	0,554089	0,548851763	0,0001156	1,1288146	0,578793	0,493923	0,489222419	0,0234421	1,6059256	ClicDer
4	0,880596	0,88551	0,868412585	0,0002986	-0,52211	0,873376	0,889839	0,86359704	0,0016327	-1,286792	None
5	0,56276	0,554904	0,552516716	0,0002564	1,4094948	0,604487	0,516846	0,483085532	0,0281475	1,2617615	ClicDer

Figura 4.13: Features obtenidos para el data set de guiños derechos

De esta manera, al obtener los features sobre la totalidad de eventos de ambos data sets por separado, como último paso se procede a unir ambos data sets de features obtenidos, en un solo archivo de Excel al que se le denominará "FeatureNet", el mismo que será utilizado en la siguiente etapa de clasificación.

4.4.3. Clasificación

Una vez obtenido los patrones o features más representativos sobre las señales EEG que sirvieron de objeto de estudio para formar ambos data sets, en esta etapa del sistema BCI se hará uso de una de las herramientas de clasificación que integra la Inteligencia Artificial como es el algoritmo de redes neuronales artificiales.

- **Consideraciones de diseño**

La arquitectura de la red neuronal a emplearse es la de una red Perceptrón Multicapa (MLP, por sus siglas en inglés), por lo que su composición en cuanto al número de capas, neuronas en las mismas, etc., se detallará con mayor precisión dicha información en el capítulo 5 del presente documento.

Para el diseño de la red neuronal se hará uso como ha sido hasta el momento del lenguaje de programación Python, donde mediante librerías de Machine Learning como es scikit-learn se construirá el modelo de clasificación, el mismo que se utilizará posteriormente para predecir los eventos de clicks del mouse en tiempo real.

Se utilizará el clasificador MLPClassifier que integra la librería scikit-learn, el mismo que está basado en el algoritmo de retropropagación del error denominado "Backpropagation", este a su vez hace uso de la técnica de optimización del descenso del gradiente estocástico, para la actualización de los pesos durante el proceso de entrenamiento de la red neuronal.

A su vez, para el proceso de entrenamiento de la red neuronal se utilizará un validador cruzado, a fin de distribuir adecuadamente la información en grupos de entrenamiento y de prueba, por ende, evitar que el modelo se vea afectado por el sobreajuste.

- Creación de la Red Perceptrón Multicapa (MLP)

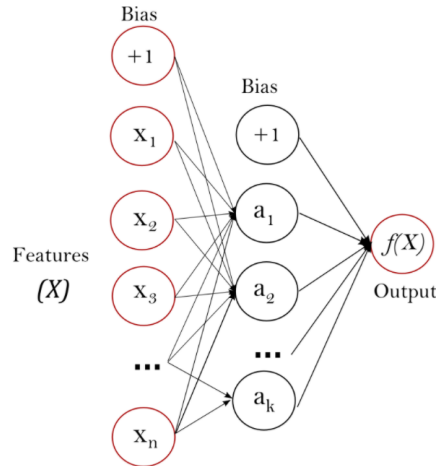


Figura 4.14: Arquitectura general de una red neuronal MLP

La Figura 4.14 muestra una representación general de una red neuronal MLP, la misma que se caracteriza por estar formada por una capa de entrada, una o varias capas ocultas y una capa de salida, en la que cada capa maneja de forma independiente un determinado número de neuronas.

Tomando en cuenta que a la capa de entrada de la red neuronal se le proporciona un conjunto de datos patrones, se hará uso del archivo de Excel al que se le denominó anteriormente como "FeatureNet", en la que siguiendo la metodología del aprendizaje supervisado, cada fila de dicho archivo representa las instancias o ejemplos con sus respectivas salidas esperadas (clases), por lo que el número de neuronas correspondientes a la capa de entrada será determinado por el número de features que maneje

cada ejemplo del archivo "FeatureNet", a su vez, el número de neuronas en la capa de salida dependerá del número de clases que se tenga a predecir.

El número total de ejemplos que contenga dicho archivo, será dividido en dos partes: la primera para la fase de entrenamiento y la segunda para la fase de prueba o validación de la red neuronal diseñada.

```
import pandas as pd
df = pd.read_excel('FeatureNet.xlsx', sheet_name='Hoja de datos')
a=df.loc[:, 'MediaA': 'AsimetriaB']
X=np.array(a).astype(np.float)
b=df.loc[:, 'Tipo']
Y=np.array(b)
```

Figura 4.15: Lectura del archivo FeatureNet mediante la librería Pandas

Las líneas de programación mostradas en la Figura 4.15, permiten leer el archivo "FeatureNet" y asignar a la variable X los datos extraídos de los features para los dos electrodos de análisis (AF3 y AF4) y a su vez asignar a la variable Y las salidas correspondientes de cada ejemplo.

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(activation='relu',hidden_layer_sizes=(5,5,5),
                    solver='sgd',learning_rate_init=0.01,shuffle=True,
                    n_iter_no_change=10,verbose=True, max_iter=3000)
```

Figura 4.16: Parámetros de configuración de la red MLPClassifier

Por su parte, la Figura 4.16 muestra la importación del clasificador MLP perteneciente a la librería sklearn.neural_network, de esta manera, en la siguiente línea se definen los parámetros de configuración que manejará la red neuronal MLP, descritos a continuación:

activation: Función de activación que devuelve una salida que será generada por la neurona dada una entrada o conjunto de entradas.

hidden_layer_sizes: Número de capas ocultas y neuronas en la misma.

solver: Técnica de optimización para el cálculo de los pesos durante el entrenamiento, "sgd" hace referencia al optimizador del descenso de gradiente estocástico.

learning_rate_init: Valor de la tasa de aprendizaje, controla el tamaño del paso en la actualización de los pesos, durante las iteraciones de cada entrenamiento de la red.

shuffle: Toma ejemplos aleatorios para formar el set de entrenamiento durante las iteraciones de entrenamiento.

n_iter_no_change: Número de iteraciones consecutivas en las cuales, si la función de pérdida o puntuación no mejora, se considera que se ha alcanzado la convergencia y se detiene el entrenamiento.

verbose: Visualizar el progreso del entrenamiento de la red neuronal.

max_iter: Número máximo de iteraciones a ser cumplidas durante la etapa de entrenamiento de la red neuronal, puede ser cumplida en su totalidad o no, dependiendo de si durante las iteraciones de entrenamiento no alcanzó la convergencia en el número de iteraciones consecutivas definidas en n_iter_no_change.

■ Fase de entrenamiento de la red neuronal

El proceso de entrenamiento de la red neuronal implica separar el total de muestras (ejemplos) contenidas en el archivo "FeatureNet" en dos grupos, de tal manera que, el un grupo contenga un determinado número de muestras destinadas para la fase de entrenamiento de la red neuronal y el otro grupo contenga el resto de muestras destinadas para la fase de prueba.

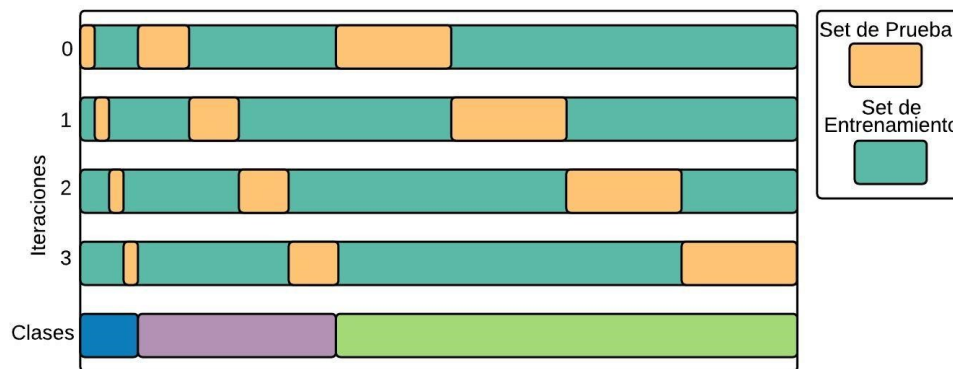


Figura 4.17: Validación cruzada estratificada

En este proceso interviene el método denominado validación cruzada, el cual se basa en dividir los datos en dos particiones (folds), de acuerdo a un determinado número de k iteraciones conocido como k-fold, a fin de probar la capacidad del modelo para predecir nuevos datos que no se utilizaron durante su entrenamiento.

La librería scikit-learn permite preparar los datos de entrenamiento y prueba haciendo uso de la validación cruzada, disponiendo de algunas variantes a la hora de tomar los datos, una de ellas es la denominada StratifiedKFold (validación cruzada estratificada), la misma que de acuerdo a la Figura 4.17 se centra en buscar que exista proporcionalidad al momento de tomar las muestras de entrenamiento y de prueba, de acuerdo a las clases que se tenga a predecir.

```

from sklearn.model_selection import StratifiedKFold
rkf = StratifiedKFold(n_splits=5)

for train, test in rkf.split(X,Y):
    modelo=mlp.fit(X[train], Y[train])
    score_train=modelo.score(X[train], Y[train])
    score_test=modelo.score(X[test], Y[test])
    print('Precisión en el set de Entrenamiento: ', score_train)
    print('Precisión en el set de Test: ', score_test)
    if score_train>=score_trainFinal:
        score_trainFinal=score_train
        Trainbest=train
        Testbest=test
    s = pickle.dumps(modelo)

```

Figura 4.18: Entrenamiento de la red neuronal

En virtud de esto, haciendo uso del validador cruzado estratificado se procede a entrenar la red neuronal, a fin de por cada iteración de entrenamiento que realice el validador cruzado se obtenga su tasa de precisión en los sets de entrenamiento y prueba, y de esta forma determinar la precisión de clasificación final del entrenamiento realizado de la red neuronal.

Se debe tomar en cuenta que, en cada iteración de entrenamiento se ejecutará un determinado número máximo de iteraciones, definido previamente en los parámetros de configuración del clasificador MLPClassifier.

La Figura 4.18 muestra el código empleado para la fase de entrenamiento de la red neuronal, en la que a través del método StratifiedKFold se organizan las muestras para entrenamiento y prueba, tomando como argumento el comando n_splits, el mismo que indica la cantidad de iteraciones de entrenamiento a cumplirse por parte del validador cruzado estratificado.

Durante cada iteración de entrenamiento se irá obteniendo la precisión alcanzada en el set de entrenamiento y prueba para la actual iteración de entrenamiento; por su parte, el condicional if permite ir comparando los scores de precisión obtenidos en cada una de las iteraciones de

entrenamiento cumplidas, con la finalidad de que una vez que haya terminado el entrenamiento de la red neuronal, se obtenga un modelo, el mismo que va utilizar el set de prueba de la iteración de entrenamiento que mejor puntuación haya alcanzado para su posterior validación.

■ Almacenamiento y validación del modelo

Una vez finalizado el entrenamiento de la red neuronal y haber determinado la mejor de iteración de entrenamiento cumplida, se procede a almacenar dicho modelo haciendo uso de la librería joblib, la misma que permite almacenar el modelo con un nombre definido por el usuario con la extensión .joblib.

```
from joblib import dump, load

modelo_final = pickle.loads(s)
dump(modelo_final, 'ModeloNet.joblib')

score_train=modelo_final.score(X[Trainbest], Y[Trainbest])
score_test=modelo_final.score(X[Testbest], Y[Testbest])
print('Precisión en el modelo final: set de Entrenamiento: ', score_train)
print('Precisión en el modelo final: set de Test: ', score_test)
predictions = modelo_final.predict(X[Testbest])
print(confusion_matrix(Y[Testbest],predictions))
print(classification_report(Y[Testbest],predictions))
```

Figura 4.19: Validación del modelo mediante métricas de clasificación

La fase de validación se basa en evaluar al modelo obtenido, a través del set de prueba correspondiente a la mejor iteración de entrenamiento, por lo que, las líneas de código mostradas en la Figura 4.19, permiten evaluar dicho modelo en base a la utilización de métricas de clasificación, tales como: precisión del modelo final, matriz de confusión o error, reporte de clasificación, etc.

De igual manera, los resultados obtenidos en la fase de validación del modelo generado, se mostrarán en el capítulo 5 del presente documento.

■ Uso del modelo en tiempo real

Hasta el momento todo lo desarrollado ha sido en modo OFF LINE, por lo que siguiendo el flujo de trabajo mostrado en la Figura 4.3, la siguiente etapa consiste en trabajar el desarrollo del algoritmo en modo ON LINE.

Es así que, una vez obtenido y seleccionado el modelo de clasificación, este modelo será consumido en tiempo real, es decir, tomará como nuevas entradas las señales que vaya captando la diadema a lo largo del tiempo, de acuerdo a las estrategias de señales de control que emita el usuario (guiños de ojos), por lo que, el modelo deberá ser capaz de clasificar dichas señales permitiendo que el sistema pueda traducir en acciones correspondientes a los eventos de click del mouse (click derecho o click izquierdo) o dependiendo de algún estado neutral por parte del usuario, el sistema sea capaz de no realizar ninguna acción de click del mouse.

En el modo ON LINE, a fin de captar los valores medidos por los canales de interés (AF3 y AF4) en tiempo real, en primera instancia se define el tamaño de la ventana de tiempo, la misma que se encargará de ir tomando un determinado número de muestras captadas por la diadema a lo largo del tiempo que permanezca activo el sistema, a fin de que el número de muestras captadas por la ventana de tiempo sean procesadas siguiendo la metodología empleada en la etapa de extracción de características, es decir, que estas señales pasen por el proceso de normalización hasta obtener sus respectivos features, para finalmente consumir el modelo obtenido en el modo OFF LINE y de esta forma usar las salidas que prediga el modelo para ejecutar las acciones de clicks del mouse o en su defecto ninguna acción.

4.5. Control del cursor del mouse

Como etapa complementaria al sistema BCI desarrollado hasta el momento, en esta parte del sistema se hará uso de los sensores de movimiento que tiene integrado internamente la diadema Emotiv Insight, a fin de utilizar dichos sensores para comandar el control sobre el cursor del mouse.

■ Objetivo a desarrollarse

Tomando en cuenta que el presente proyecto se centra en el manejo de un ordenador haciendo uso exclusivo de la diadema Emotiv Insight, el usuario a más de generar los clicks derecho e izquierdo en base a las estrategias de señal de control establecidas anteriormente (guiños de ojos), tendrá la posibilidad de controlar la posición del cursor del mouse a lo largo de la pantalla del ordenador. Es así que, haciendo uso de movimientos de su cabeza, el usuario podrá visualizar como el cursor del mouse se desplaza de forma simultánea hacia las mismas direcciones que apunta su cabeza.

- **Consideraciones**

De igual manera, al ser el control sobre el cursor del mouse en tiempo real, se hace uso del servicio de la API Cortex de Emotiv, a fin de captar los datos medidos en tiempo real por los sensores de movimiento.

4.5.1. Diseño

El diseño está basado en hacer uso de los sensores de movimiento de la diadema Emotiv Insight, el servicio API Cortex y un ordenador.

Los sensores se utilizan para seguir y captar los movimientos de la cabeza por parte del usuario, por su parte, la API Cortex actúa como medio de enlace para recoger y enviar los datos de los sensores en tiempo real hacia el ordenador para su respectivo procesamiento, y este último permitirá visualizar como el cursor del mouse se desplaza sobre la pantalla en función de los datos recibidos por los sensores de movimiento de la diadema.

En esta parte se indicarán aspectos tomados en cuenta para el diseño del control del cursor del mouse, por lo que en el capítulo 5 se mostrarán los resultados obtenidos durante este proceso.

- **Uso del acelerómetro para control del mouse**

Los acelerómetros se encargan de medir la aceleración de un cuerpo en una dirección determinada sobre un sistema de referencia, así como la posición de un objeto con respecto a la superficie de la Tierra.

Este tipo de sensores son capaces de medir la aceleración en sus tres ejes de coordenadas, por lo que el diseño del control del cursor del mouse se lo realizará en base a los datos proporcionados por los acelerómetros de la diadema.

El espacio de trabajo del cursor del mouse sobre la pantalla del ordenador se centra en desplazamientos en X y Y, por lo que, el valor del eje Z del acelerómetro (*AccZ*) no se lo considera para el diseño propuesto; por ende, como punto de partida es necesario determinar cómo están referenciados los tres ejes del acelerómetro de la diadema, dado que al momento de recibir los datos de movimiento del acelerómetro, se desplegarán tres columnas con los datos medidos en ese instante de tiempo por los acelerómetros en sus tres ejes, siendo necesario identificar cada columna a su respectivo eje de acelerómetro.

Si bien la aceleración y la velocidad están relacionados entre sí, por lo general, la mayor parte de sensores inerciales presentan algún tipo de

desviación, sesgo, produciendo error durante la medición, por lo que se ve en la necesidad de buscar alguna herramienta que minimice estas desviaciones.

- **Uso del Filtro de Kalman**

Teniendo en cuenta que, un pequeño error durante las mediciones de aceleración realizadas por los acelerómetros considerados (AccX y AccY), pueden llevar a desencadenar errores considerables sobre la velocidad del cursor del mouse; por lo tanto, se hará uso del Filtro de Kalman como herramienta que permita minimizar y suavizar los datos de aceleración provenientes de la diadema.

Además, el Filtro de Kalman al ser un algoritmo recursivo permitirá realizar estimaciones más precisas a partir de mediciones tomadas en tiempo real por los ejes de los acelerómetros considerados, de esta manera, facilitando al usuario a tener un mayor control y precisión sobre la posición del cursor del mouse en la pantalla del ordenador.

En base a estos dos aspectos considerados dentro del diseño, se desarrollará el algoritmo de programación necesario para asociar los movimientos de cabeza del usuario al cursor del mouse, por lo que, en el capítulo 5 se indicarán detalles observados durante el desarrollo del mismo.

4.6. API Cortex para el modo ON LINE

Dado que el funcionamiento del manejo de un ordenador es en tiempo real, en esta etapa la herramienta de la API Cortex de Emotiv juega un papel trascendental en el desarrollo del sistema BCI propuesto, ya que este servicio integra varios métodos (funciones de programación), permitiendo al usuario acceder a distintas funcionalidades propias de la Cortex, al punto de a través de lenguajes de programación que manejen el protocolo websocket como es el caso de Python, dichos métodos puedan ser llamados directamente desde un script y ocupados para un propósito específico.

En virtud de esto, a continuación se dará a conocer los aspectos principales a tomar en cuenta de la API Cortex, a fin de que sus métodos puedan ser utilizados en el script de programación de Python, permitiendo poner en marcha y en tiempo real las dos partes que conforman el sistema.

La Figura 4.20 muestra las líneas de código correspondientes al método de inicialización, la cual se basa en hacer uso de una conexión websocket, la misma que es el medio de enlace entre la API Cortex y Python.


```
def __init__(self, user, debug_mode=False):  
    url = "wss://localhost:6868"  
    self.ws = websocket.create_connection(url,  
        sslopt={"cert_reqs": ssl.CERT_NONE})  
    self.user = user  
    self.debug = debug_mode
```

Figura 4.20: Comunicación websocket entre la API Cortex y Python

A más de utilizar el método de inicialización, es necesario hacer uso de algunos métodos de la API Cortex, a fin de poder crear una sesión a través de una aplicación; por lo que los demás métodos a utilizarse son los siguientes:

- **requestAccess()**: Solicita la aprobación del usuario para la aplicación actual a través de EMOTIV App.
- **controlDevice()**: Este método sirve para conectar la diadema a la aplicación.
- **queryHeadsets()**: Muestra los detalles de la diadema conectada a la aplicación, ID, conectado por receptor USB, sensores de movimiento, canales EEG, estado= conectado, etc.
- **authorize()**: Este método sirve para generar un token (código) de acceso a Cortex, que se crea al momento de establecer la conexión con la API Cortex. Algunos métodos requieren este parámetro para validar las peticiones o solicitudes que se les haga.
- **createSession()**: Permite abrir y activar una sesión con una diadema de EMOTIV.
- **subscribe()**: Permite suscribirse a uno o varios streams, mediante este método Cortex enviará flujos de datos.

El contenido de estos y demás métodos que maneja la API Cortex, se encuentra disponible en el gitbook de la página oficial de Emotiv.

El método subscribe() de la API Cortex para el desarrollo del sistema BCI propuesto, representa ser una pieza fundamental; ya que permite recibir los flujos de datos en tiempo real tanto de las señales EEG (5 canales) y de los sensores de movimiento (giroscopios, acelerómetros y magnetómetros), devolviendo en forma continua una trama de datos en formato JSON de lo que la diadema está midiendo en ese instante de tiempo.

Por su parte, las líneas de programación mostradas en la Figura 4.21, indican los streams utilizados como parámetros del método `subscribe()` para el desarrollo en general del sistema, en la que la API Cortex asocia la información correspondiente a los flujos de datos de los sensores de movimiento el stream 'mot', y a los flujos de datos de las señales EEG captadas por los cinco electrodos el stream 'eeg', para su respectiva utilización.

```
streams = ['mot', 'eeg']  
s.sub(streams)
```

Figura 4.21: Streams utilizados para el modo ON LINE

Por ende, en el desarrollo del algoritmo de programación se hace uso de ambos streams dentro de un ciclo infinito 'while (True)', a fin de obtener los flujos de datos de forma simultánea tanto de los dos acelerómetros (AccX y AccY), como de los dos canales de interés (AF3 y AF4).

4.7. Interfaz gráfica de usuario

Si bien la funcionalidad del sistema BCI desarrollado, es la de permitir que el usuario pueda navegar libremente sobre todo el entorno de Windows, haciendo uso de sus propias señales de control (guiños de ojos) y movimientos de la cabeza.

Se ha considerado oportuno diseñar una interfaz gráfica de usuario, a fin de que la misma permita validar el sistema BCI desarrollado, por lo que en Figura 4.22 se puede apreciar la pantalla principal de la interfaz, teniendo en consideración que su diseño y funcionalidad se basa en un enfoque educativo, principalmente como herramienta de distracción y comunicación.

De tal manera que, personas que tienen alguna limitación física en sus extremidades superiores, puedan poner a prueba el sistema desarrollado a través de la utilización de dicha interfaz, seleccionando botones a través de guiños de ojos y desplazando el cursor del mouse hacia distintos objetivos (botones) a través de movimientos de cabeza, y de esta manera, ayudar a que este tipo de personas tengan una forma de entretenimiento y comunicación con su entorno.



Figura 4.22: Pantalla principal de la interfaz gráfica de usuario

4.7.1. Librerías utilizadas

Python para el diseño de interfaces gráficas de usuario integra la librería tkinter, adicionalmente posee diversas librerías para diferentes utilidades por parte del programador, por lo que, a continuación se indican las librerías utilizadas para el diseño de la interfaz gráfica de usuario.

- **tkinter:** Es una librería multiplataforma completa que posibilita el diseño de interfaces gráficas de usuario, permitiendo la creación de ventanas, botones, slider, etiquetas, textos, etc.
- **tkfont:** Es una librería complementaria de tkinter, que permite hacer uso de distintos tipos y tamaños de letras.
- **time:** Es una librería utilizada para manejar tareas de tiempo, es decir, para generar pausas al momento de ejecutar una determinada acción.
- **playsound:** Es una librería que permite la reproducción de archivos de audio.

4.7.2. Funcionalidades de la interfaz

La interfaz gráfica de usuario diseñada mediante Python cuenta con dos funcionalidades.

1. Teclado BCI

Permite al usuario interactuar con un teclado virtual, el mismo que fue diseñado a través de Python a base de las funciones que integra la librería tkinter, teniendo como finalidad permitir que el usuario pueda ir seleccionando letras del alfabeto y de esa manera ir formando palabras o frases que le permitan establecer una comunicación con su entorno, ver Figura 4.23.

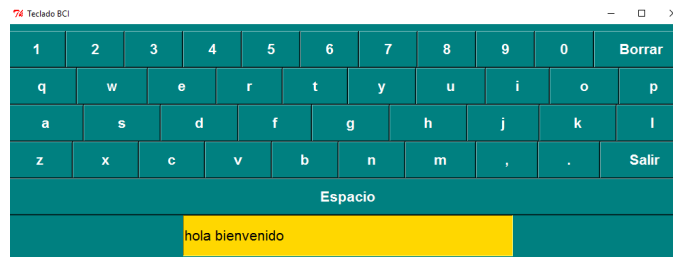


Figura 4.23: Teclado BCI

2. Sonidos BCI

Esta funcionalidad tiene un enfoque educativo, en la que se hace uso de la librería playsound y time, permitiendo al usuario que tenga la posibilidad de seleccionar letras del alfabeto o números, con la finalidad que se le despliegue una nueva ventana de tipo informativa, con la imagen del caracter seleccionado y su respectivo audio, ver Figura 4.24.

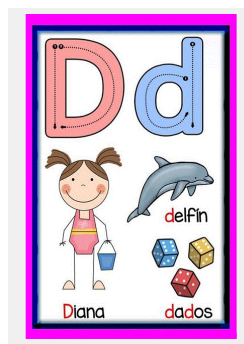


Figura 4.24: Sonidos BCI

Capítulo 5

Pruebas y Resultados

En este capítulo se detallarán los resultados obtenidos durante cada una de las etapas del sistema BCI desarrollado, así como también se indicarán las pruebas consideradas para la validación del sistema en general sobre algunos participantes, indicando sus respectivos resultados alcanzados; para finalmente realizar una discusión de resultados, tomando como referencia los logros alcanzados por los autores guía de la presente investigación.

5.1. Resultados en las etapas de desarrollo del sistema BCI

En esta sección se indicará de forma precisa como está constituido y desarrollado el sistema BCI en cada una de sus etapas, por lo que se hará uso de tablas, figuras, que resuman los aspectos considerados o generados en cada una de estas etapas, de acuerdo a la metodología de trabajo seleccionada e indicada en el capítulo anterior.

5.1.1. Etapa de adquisición de señales

El proceso de adquisición de señales a través de la funcionalidad de grabaciones de señales EEG que permite EmotivPRO, se lo efectuó en su totalidad sobre un único participante, siguiendo la secuencia del experimento mostrada anteriormente en la Figura 4.1.

En la Tabla 5.1 se resume los aspectos considerados en la etapa de adquisición de señales.

Tabla 5.1: Consideraciones tomadas durante la adquisición de señales

Descripción	Cantidad
Participantes para la toma de datos	1
Grabación para el data set (guiños derechos)	1
Grabación para el data set (guiños izquierdos)	1
Duración de cada grabación	8 minutos
Total de data sets generados	2

5.1.2. Etapa de extracción de características

Los dos data sets generados durante la etapa de adquisición de señales por parte del participante seleccionado, constituyó la base con la que se trabajó las demás etapas del sistema BCI, en la que al disponer de dos data sets por separado correspondientes a grabaciones de guiños derechos y grabaciones de guiños izquierdos; el proceso de etiquetado de las señales sobre ambos data sets mostró los siguientes resultados, los mismos que se resumen en la Tabla 5.2 para el data set de guiños derechos y en la Tabla 5.3 para el data set de guiños izquierdos.

Tabla 5.2: Eventos identificados en el data set de guiños derechos

Descripción	Cantidad
Eventos totales	101
Eventos de guiños derechos	54
Eventos de none	47
Total de muestras en el data set	28665

Tabla 5.3: Eventos identificados en el data set de guiños izquierdos

Descripción	Cantidad
Eventos totales	101
Eventos de guiños izquierdos	50
Eventos de none	51
Total de muestras en el data set	24900

Por su parte, durante el proceso de generación de features sobre la totalidad de eventos previamente normalizados en ambos data sets, se pudo apreciar que todas las muestras que correspondían a un mismo evento se reducían a una única muestra por evento, siendo representadas por un total de 10 features.

Tabla 5.4: Información en el data set de Features de guiños derechos

Acción	Nº Muestras
Click derecho	54
None	47
Total	101

En virtud de esto, la Tabla 5.4 indica el número de muestras correspondientes a las acciones de click derecho y de none, en la que se debe tomar en cuenta que únicamente se diferencian estas dos acciones, dado que durante el proceso de grabación de las señales el participante se centró en cumplir la secuencia del experimento, realizando estas dos acciones a lo largo del experimento.

De igual manera, siguiendo esta misma lógica en la Tabla 5.5 se indica el número de muestras correspondientes a las acciones de click izquierdo y de none.

Tabla 5.5: Información en el data set de Features de guiños izquierdos

Acción	Nº Muestras
Click izquierdo	50
None	51
Total	101

Finalmente, el archivo al que se le denominó anteriormente como “FeatureNet”, que abarca la información de la Tabla 5.4 y de la Tabla 5.5, obtuvo como resultado la información que se menciona en la Tabla 5.6.

Tabla 5.6: Información recopilada del archivo FeatureNet

Acción	Nº Muestras
Click derecho	54
Click izquierdo	50
None	98
Total	202

5.1.3. Etapa de clasificación

Si bien, durante la fase de extracción de características desarrollado en el capítulo anterior, el uso de ventanas de tiempo aplicadas sobre las señales generadas en ambos data sets de análisis, permitió identificar desde un inicio el número de clases objetivo a predecir por parte del sistema, las mismas que fueron indicadas en la Tabla 4.1.

- **Arquitectura de la red neuronal empleada**

Haciendo referencia al aprendizaje automático supervisado, en el que a la red se le proporciona información tanto a la entrada como a la salida, la Figura 5.1 muestra el esquema general utilizado en la etapa de clasificación.

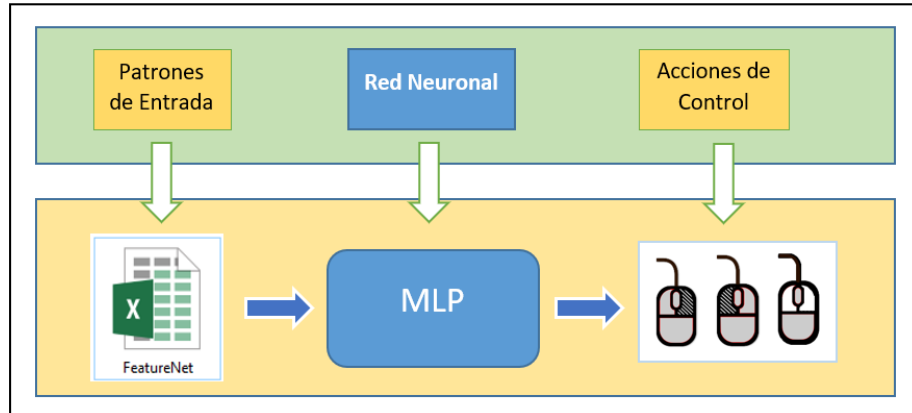


Figura 5.1: Esquema general de la etapa de clasificación

En La Figura 5.1 se aprecia que los patrones de entrada fueron definidos por el archivo denominado “FeatureNet”, para que en base a estos features característicos la red neuronal genere a su salida las acciones de control esperadas (eventos de clicks del mouse).

Tabla 5.7: Composición de la red neuronal MLP empleada

Descripción	Selección
Red neuronal	MLP
Capas de la red	1 capa de entrada 3 capas ocultas 1 capa de salida
Nº de neuronas (Capa de entrada)	10
Nº de neuronas (Capas ocultas)	5-5-5
Nº de neuronas (Capa de salida)	3

Finalmente, la Tabla 5.7 indica como está definida la arquitectura de la red neuronal en sus distintas capas.

▪ Validación cruzada estratificada

Para garantizar un buen entrenamiento de la red neuronal, evitando en caer en un sobreajuste, el validador cruzado estratificado empleado tiene la

característica de asegurarse que el set de entrenamiento y el set de prueba sea formado de forma proporcional por ejemplos de las 3 posibles clases, por lo que se definió un `n_splits=5`, con la finalidad de que el validador cruzado estratificado agrupe el total de muestras (202) del archivo "FeatureNet" de la siguiente manera: 80 % de muestras para la fase de entrenamiento de la red y 20 % de muestras para la fase de prueba, a su vez, dicho valor indica que se realizó dicho agrupamiento en cada una de las 5 iteraciones de entrenamiento cumplidas.

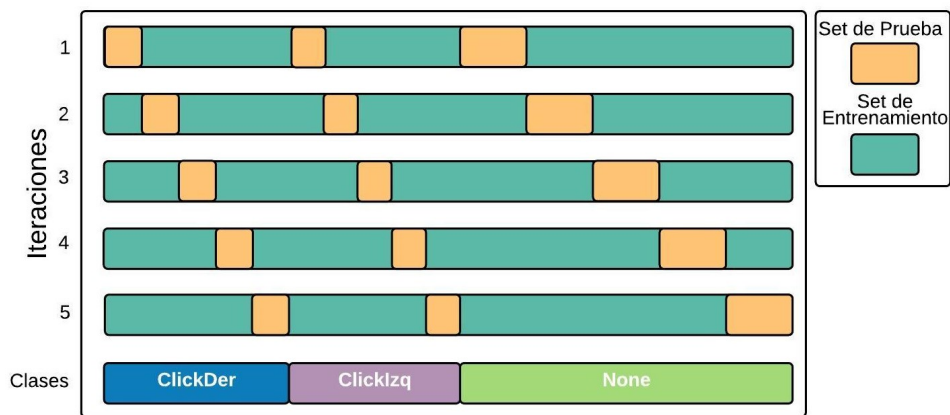


Figura 5.2: Validación cruzada durante las iteraciones de entrenamiento

La Figura 5.2 es una representación de como la validación cruzada estratificada permite que en cada iteración de entrenamiento realizada, se vaya obteniendo de forma proporcional un 80 % y un 20 % de muestras para el set de entrenamiento y de prueba respectivamente sobre cada clase.

Tabla 5.8: Distribución de las muestras por cada clase para ambos sets

Clase	Set Entrenamiento	Set Prueba
ClickDer: 54 (100 %)	43 (80 %)	11 (20 %)
ClickIzq: 50 (100 %)	40 (80 %)	10 (20 %)
None: 98 (100 %)	78 (80 %)	20 (20 %)
Total: 202 (100 %)	161 (80 %)	41 (20 %)

Por su parte, la Tabla 5.8 resume el número de muestras exactas tomadas de cada clase por el validador cruzado estratificado, para la conformación final del set de prueba y de entrenamiento, durante cada una de las iteraciones de entrenamiento cumplidas.

■ Resultados obtenidos en el entrenamiento de la red neuronal

Una vez que se efectuó el proceso de entrenamiento de la red neuronal haciendo uso del validador cruzado estratificado, se obtuvieron los scores de precisión en los sets de entrenamiento y prueba de cada iteración de entrenamiento cumplida, indicados a través de la Figura 5.3.

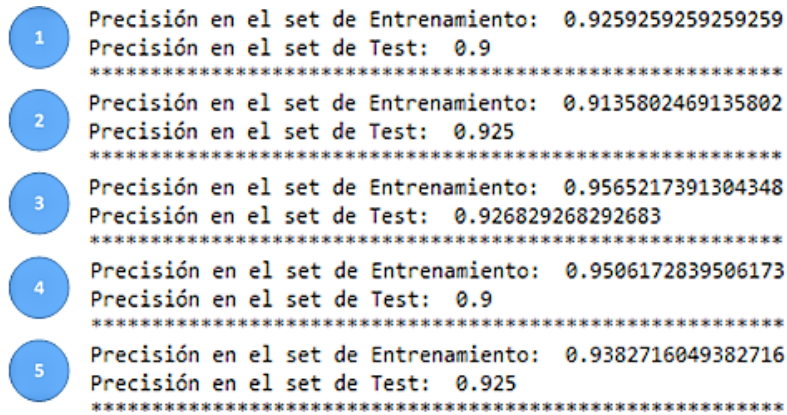


Figura 5.3: Scores de precisión durante el proceso de entrenamiento

Durante el proceso de entrenamiento de la red neuronal, el criterio de parada que detuvo cada iteración de entrenamiento dependió del número máximo de iteraciones a ejecutarse (3000) o a su vez si durante estas 3000 iteraciones se cumple que 10 iteraciones consecutivas no mejoran su función de pérdida, dando a entender que se alcanzó la convergencia.

Los resultados mostrados en la Figura 5.3, permiten determinar la precisión de clasificación obtenida por el clasificador durante el proceso de entrenamiento, la misma que se obtiene mediante la media aritmética de los resultados de precisión del set de prueba de cada iteración de entrenamiento, obteniendo una precisión de clasificación del 91.53 %.

■ Métricas de clasificación del modelo empleado

En el algoritmo de programación desarrollado se fue almacenando los datos de entrenamiento y prueba, de la iteración de entrenamiento que

alcanzó los mejores scores de precisión, de tal manera que el modelo que se obtuvo fue validado en base al set de prueba correspondiente a la iteración 3 de la Figura 5.3.

Para el caso de la iteración de entrenamiento 3, su criterio de parada fue determinado por el número de iteraciones consecutivas, es así que, su entrenamiento finalizó en la iteración 1778, alcanzando un error de 0.116, por lo que en la Figura 5.4 se muestra la gráfica de la función de pérdida.

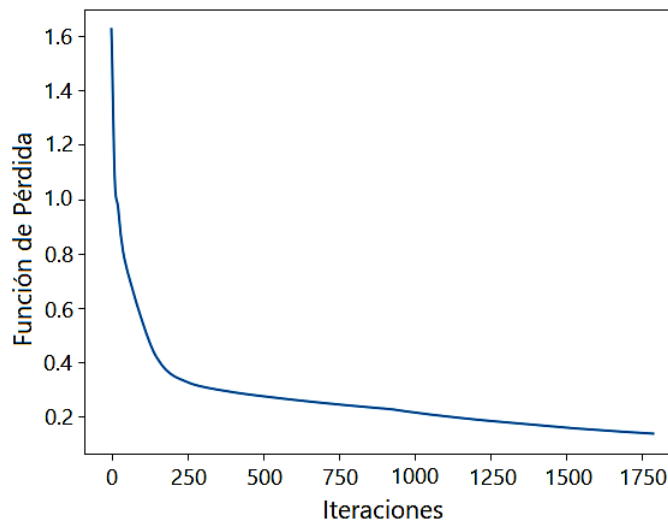


Figura 5.4: Función de pérdida vs Iteraciones

En base al modelo obtenido, el proceso de validación del mismo se lo realizó utilizando métricas de clasificación de Machine Learning.

En la Figura 5.5 se observa la matriz de confusión obtenida para el set de prueba utilizado una vez que la red neuronal quedó entrenada. La diagonal principal indica los patrones que fueron clasificados correctamente para cada clase, y el resto de valores distintos a los de la diagonal principal representan la cantidad de patrones que fueron clasificados de forma errónea en cada clase.

Por su parte, la Figura 5.6 indica el reporte de clasificación obtenido del set de prueba utilizado basado en los resultados generados por la matriz de confusión, en la que se aprecia que la columna support distribuyó los patrones de prueba de acuerdo a lo mencionado en la Tabla 5.8.

ClicDer	9	0	2
ClicIzq	1	9	0
None	0	0	20
	ClicDer	ClicIzq	None

Figura 5.5: Matriz de confusión resultante para el set de prueba

	precision	recall	f1-score	support
ClicDer	0.90	0.82	0.86	11
ClicIzq	1.00	0.90	0.95	10
None	0.91	1.00	0.95	20
accuracy			0.93	41
macro avg	0.94	0.91	0.92	41
weighted avg	0.93	0.93	0.93	41

Figura 5.6: Reporte de clasificación obtenido del set de prueba

La precisión de cada clase viene dada por su número de patrones clasificados correctamente, sobre el número total de patrones que exista en su columna.

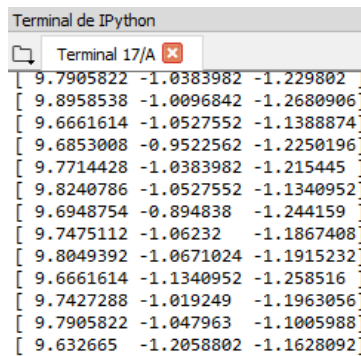
El recall de cada clase viene dado por su número de patrones clasificados correctamente, sobre el número total de patrones que exista en su fila.

La métrica f1-score de cada clase es la media aritmética del valor de precisión y de recall de su respectiva clase.

Finalmente, la precisión total (accuracy) viene dado por la suma de todos los patrones clasificados correctamente que se encuentran en la diagonal principal, sobre la suma de todos los patrones que conforman la matriz de confusión, dando como resultado un accuracy del 93%.

5.2. Pruebas del control del cursor del mouse

Dado que el control del cursor del mouse se lo realizó en base a la información proveniente de los acelerómetros de la diadema, el primer aspecto a considerar fue determinar como están posicionados los tres ejes del acelerómetro cuando el usuario está colocado la diadema. Es así que se hizo uso de la API Cortex de Emotiv, a fin de a través de la petición del stream “mot”, imprimir en tiempo real a través de la consola de Python el flujo de datos correspondientes al acelerómetro en sus tres ejes.



```
Terminal de IPython
Terminal 17/A
[ 9.7905822 -1.0383982 -1.229802 ]
[ 9.8958538 -1.0096842 -1.2680906 ]
[ 9.6661614 -1.0527552 -1.1388874 ]
[ 9.6853008 -0.9522562 -1.2250196 ]
[ 9.7714428 -1.0383982 -1.215445 ]
[ 9.8240786 -1.0527552 -1.1340952 ]
[ 9.6948754 -0.894838 -1.244159 ]
[ 9.7475112 -1.06232 -1.1867408 ]
[ 9.8049392 -1.0671024 -1.1915232 ]
[ 9.6661614 -1.1340952 -1.258516 ]
[ 9.7427288 -1.019249 -1.1963056 ]
[ 9.7905822 -1.047963 -1.1005988 ]
[ 9.632665 -1.2058802 -1.1628092 ]
```

Figura 5.7: Valores medidos por los tres acelerómetros de la diadema

La Figura 5.7 permitió identificar como están referenciados los ejes del acelerómetro, determinando rápidamente que la columna 1 correspondía a los datos medidos por el AccZ, ya que está apuntando en dirección a la gravedad, es decir, al momento de colocarse la diadema esta columna marcaba valores casi fijos cercanos al 9.8 que corresponde a la gravedad ($1g = 9,8 m/s^2$), por su parte, haciendo movimientos verticales con la cabeza se pudo apreciar que en la columna 2 los datos variaban dando a entender que dicha columna correspondía al AccY, de igual manera, haciendo movimientos horizontales con la cabeza se pudo apreciar que en la columna 3 los datos variaban dando a entender que dicha columna correspondía al AccX.

Por otro lado, si bien es cierto matemáticamente la aceleración está relacionada a la velocidad, en este caso un mínimo sesgo o error por parte del acelerómetro puede ocasionar un error considerable sobre la velocidad del cursor del mouse.

Por lo que, en el desarrollo del algoritmo de programación se tomó en consideración este aspecto, siendo necesario utilizar la herramienta del Filtro de Kalman a fin de aplicarlo sobre los datos de aceleración medidos por parte

del AccX y AccY, ya que se trabajó únicamente sobre estos, dado que el cursor del mouse se mueve sobre la pantalla del ordenador en las direcciones X y Y, por lo que el AccZ se lo omitió su uso.

De tal manera que se utilizó las ecuaciones del Filtro de Kalman unidimensionales mencionadas en el capítulo 3, dado que el desplazamiento que sigue el cursor del mouse es en torno a direcciones horizontales y verticales.

```
def FiltroKalman(data,preData,var,Q):
    preVar = var + Q
    K = preVar / (preVar + Q)
    outputData = preData + K * (data - preData)
    var = (1- K) * preVar
    return (outputData,var)
```

Figura 5.8: Filtro de Kalman unidimensional aplicado a datos de aceleración

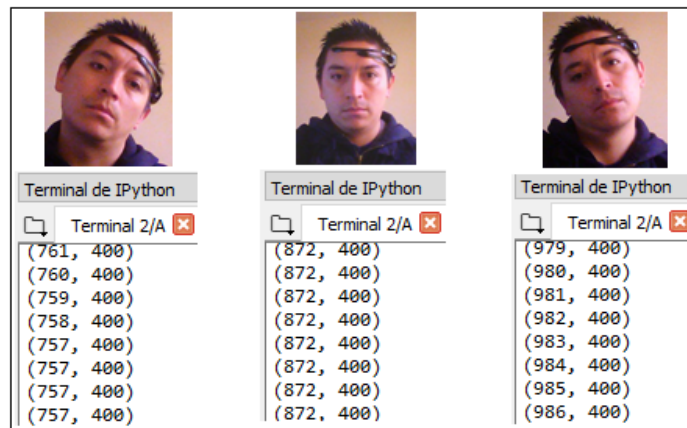
La Figura 5.8 muestra las líneas de código que hacen referencia a dichas ecuaciones 3.2, 3.3, 3.4, las mismas que están dentro del método denominado FiltroKalman(), que básicamente se centra en un método iterativo de predicción del siguiente estado haciendo uso de errores de estimación de estados anteriores, por lo que en el algoritmo de programación desarrollado los nuevos datos captados por los acelerómetros AccX y AccY pasan siempre por la función FiltroKalman(), retornando esta dos valores: la estimación del estado actual del dato AccX y el error en la estimación actual que sirve para la siguiente iteración, mismo proceso para el caso de los datos del AccY.

Tomando como referencia de que al mandar a imprimir los valores del acelerómetro AccX una vez pasados por el filtro, toman valores positivos mientras la persona tenga inclinada su cabeza al lado derecho, por su parte, toman valores negativos si la persona tiene inclinada su cabeza al lado izquierdo, pero si la persona mantiene su cabeza en una posición recta y firme los valores medidos son aproximados a cero.

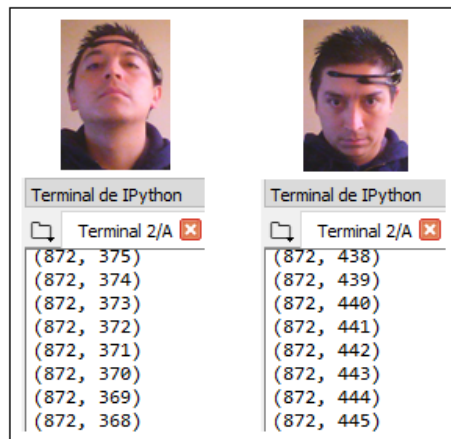
De igual manera, para el acelerómetro AccY toman valores positivos mientras la persona tiene inclinada su cabeza hacia arriba, en cambio, toman valores negativos si la persona tiene inclinada su cabeza hacia abajo.

En base a esto, se aplicó el valor absoluto a los datos de los AccX y AccY, y se fijó un mismo valor de umbral para ambos con la finalidad que si dichos datos superan un valor umbral definido en 1.5, significa que la persona se encuentra con su cabeza inclinada en alguna de las direcciones indicadas anteriormente, caso contrario significa que la persona se encuentra con su cabeza en una posición recta y firme.

Para que el cursor del mouse experimente algún desplazamiento hacia alguna dirección, dependió de si los datos de aceleración en cualquiera de sus dos ejes de análisis superaban dicho valor de umbral, caso contrario el cursor del mouse permanecía estático sobre la pantalla del ordenador, por lo que, los desplazamientos del cursor en cualquiera de las dos direcciones giran en torno al valor del acelerómetro respectivo, multiplicado por un valor de una constante la misma que mientras más grande sea su valor ocasionaba que el cursor se desplace con mayor velocidad.



(a) Variación en el eje X



(b) Variación en el eje Y

Figura 5.9: Variación de la posición del cursor ante movimientos de la cabeza

Finalmente, haciendo uso de la librería win32api se puede obtener con precisión las coordenadas (X,Y) del cursor sobre la pantalla del ordenador, la misma que devuelve dos valores, en la cual al primer dato se le adiciona el valor del desplazamiento en X obtenido, y al segundo dato se le resta el valor del desplazamiento en Y obtenido; por lo que, en la Figura 5.9 se puede apreciar como varían las coordenadas en (X,Y) del cursor del mouse sobre la pantalla del ordenador, dependiendo de la dirección a la que apunte la cabeza de la persona, de esta manera, el cursor del mouse está asociado a los acelerómetros de la diadema en función de los movimientos de la cabeza que realice la persona.

5.3. Funcionamiento del sistema

La Figura 5.10 indica los dos modos de funcionamiento del sistema desarrollado, el mismo que está compuesto de: generación de los eventos de clicks del mouse mediante la acción de guiños por parte del usuario y el control sobre el cursor del mouse mediante los movimientos de cabeza que realice el usuario; en la que ambas partes del sistema desarrollado se ejecutan de forma simultánea en tiempo real.

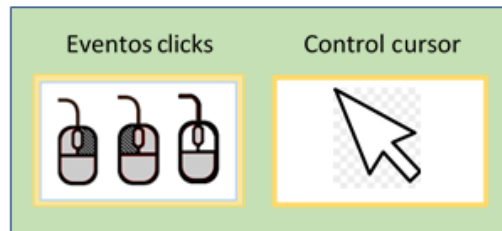


Figura 5.10: Modos de funcionamiento del sistema BCI

Como se indicó al inicio del capítulo anterior, es necesario que todo usuario previo a hacer uso del sistema desarrollado considere los aspectos mencionados en los apéndices A y B, a fin de familiarizarse con las estrategias de señales de control establecidas (guiños de ojos y movimientos de cabeza), permitiendo de esta manera al usuario lograr una correcta manipulación del sistema y que los resultados sean los esperados.

El sistema BCI trabaja en su totalidad en modo ON LINE, es decir, el usuario a través del uso de la diadema Emotiv Insight tiene la posibilidad de manejar un ordenador, por lo tanto, una vez que se le da la indicación

de la puesta en marcha del sistema, el usuario debe permanecer alrededor de unos 6 segundos con una posición cómoda, relajante, manteniendo su cabeza recta y firme con la mirada hacia el frente, con la finalidad de que en el transcurso de este tiempo, se realice el proceso de calibración sobre los sensores de movimiento de la diadema, a fin de que el sistema pueda determinar una posición de referencia cuando el usuario pretende mantener el cursor del mouse estático sobre la pantalla.

Transcurrido este tiempo el usuario ya está listo para hacer uso del sistema, si bien el uso es en general sobre todo el entorno de Windows del computador, se diseñó previamente una interfaz de usuario de tipo educativa para validar el sistema.

5.4. Pruebas sobre el sistema BCI

En esta sección se mostrarán las pruebas realizadas al sistema BCI desarrollado, en la que se utilizarán cinco participantes (dos participantes sanos y tres participantes con problemas de movilidad), sobre quienes se validará el rendimiento del sistema BCI desarrollado.

- **Consideraciones previas**

Se realizaron tres modos de prueba del sistema: (prueba test, prueba sobre la interfaz y prueba general), siendo todos los participantes evaluados en cada prueba considerada de forma separada.

La Tabla 5.9 muestra las características físicas de los cinco participantes seleccionados para el proceso de validación del sistema.

Tabla 5.9: Características físicas de los participantes de prueba

Id	Edad	Género	Participación grabaciones	Observación
Participante 1	29	Masculino	Si	Ninguna
Participante 2	25	Masculino	No	Brazo inmovilizado
Participante 3	24	Masculino	No	Brazo inmovilizado
Participante 4	36	Masculino	No	Brazo inmovilizado
Participante 5	63	Masculino	No	Ninguna

Nótese que, de acuerdo a la Tabla 5.9 las pruebas fueron sobre participantes que se encontraban en condiciones normales y en participantes

que al momento de realizar dichas pruebas presentaban alguna limitación física en sus extremidades superiores (brazos inmovilizados), de igual manera, de los cinco colaboradores para las pruebas, únicamente el participante 1 participó durante la fase de adquisición de señales a través de las grabaciones, por lo que los otros cuatro participantes son ajenos al desarrollo del sistema.

Los resultados obtenidos en las tres pruebas consideradas sobre cada uno de los participantes seleccionados, se indicarán en la siguiente sección.

5.4.1. Prueba test

Esta prueba consiste en dar la orden al participante en que evento de click del mouse debe intentar ejecutar, es decir, el participante debe emitir el guiño de ojo respectivo sea para click izquierdo o click derecho, según la indicación que se le dé; además, al no especificarle al participante alguno de estos dos eventos de clicks, se sobreentiende que el sistema no debe ejecutar ningún click, ya que el participante se encuentra en modo de reposo al que se le denominó (none).



(a) Participante 1

```
subscribe request -----  
accion final: ClicDer  
subscribe request -----  
accion final: None  
subscribe request -----  
accion final: ClicIzq  
subscribe request -----  
accion final: None  
subscribe request -----  
accion final: ClicDer  
subscribe request -----  
accion final: None  
subscribe request -----  
accion final: None  
subscribe request -----  
accion final: ClicIzq
```

(b) Predicción del sistema

Figura 5.11: Prueba test

La Figura 5.11(a) muestra al participante 1 realizando la prueba test, la misma que se efectúa haciendo uso de un sonido corto que indica al participante que debe intentar realizar un click izquierdo y mediante otro tipo de sonido corto que indica al participante que debe intentar realizar un click derecho; por su parte, la Figura 5.11(b) muestra la respuesta del sistema de la clase predicha ante la acción de guiño o estado neutral del participante, visto directamente desde la consola de Python.

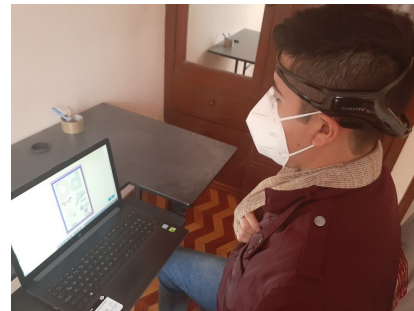
De igual manera, una vez que el participante finalizó la prueba propuesta, como segunda parte se le pide al participante que intente desplazar el cursor del mouse a las distintas esquinas de la pantalla (superior, central e inferior), a fin de verificar si el participante tiene facilidad o dificultad de direccionar el cursor del mouse a determinadas posiciones de la pantalla.

5.4.2. Prueba sobre la interfaz de usuario

Esta prueba consiste en que los participantes navegan haciendo uso de la interfaz de usuario diseñada.



(a) Participante 2



(b) Participante 3

Figura 5.12: Prueba sobre la interfaz de usuario

Los objetivos a cumplirse por parte de los participantes son indicados a continuación:

1. Desplazar el cursor del mouse hacia el botón “Teclado BCI”.
2. Seleccionar el botón “Teclado BCI”.
3. Escribir la palabra “hola mundo”.
4. Cerrar el Teclado BCI, seleccionando el botón “Salir”.
5. Desplazar el cursor del mouse hacia el botón “Sonidos BCI”.
6. Seleccionar el botón “Sonidos BCI”.
7. Desplazar el cursor del mouse hacia la tecla “D” y seleccionarla.

8. Cerrar la ventana informativa de la letra D, seleccionando el botón “Salir”.
9. Cerrar la ventana de Sonidos BCI, seleccionando el botón “Salir”.
10. Cerrar la pantalla principal de la interfaz de usuario, seleccionando el botón “Salir”.

La Figura 5.12(a) muestra al participante 2 el instante en el que escribió la palabra “hola mundo”, por su parte, la Figura 5.12(b) muestra al participante 3 el instante en el que una vez que seleccionó la tecla “D”, se le despliega la ventana informativa de la letra “D”.

5.4.3. Prueba general sobre el sistema

Esta prueba consiste en que los participantes navegan sobre el entorno de Windows del computador.



(a) Participante 4



(b) Participante 5

Figura 5.13: Prueba general sobre el sistema

Los objetivos a cumplirse por parte de los participantes son indicados a continuación:

1. Crear 5 carpetas sobre el escritorio de Windows.
2. Abrir el explorador de archivos, desde la barra de tareas.
3. Ingresar a las propiedades del Disco Local (C:).
4. Abrir Microsoft Word desde su respectivo icono ubicado en el escritorio del computador.

5. Abrir el teclado virtual de la computadora desde la barra de tareas.
6. Escribir en el documento de Word la frase “mouse”.
7. Cerrar Microsoft Word.

La Figura 5.13(a) muestra al participante 4 el instante en que abrió las propiedades del Disco Local (C:), por su parte, la Figura 5.13(b) muestra al participante 5 el instante en el que una vez que ingresó a Microsoft Word escribió la frase “mouse”.

5.5. Resultados generales sobre el sistema BCI

La forma de evaluar el rendimiento del sistema BCI desarrollado en las tres pruebas consideradas, se basa en lo mencionado por el autor [Graumann et al., 2009], quien manifiesta que la precisión de clasificación del mismo se basa en la relación que existe entre el número de ensayos clasificados de manera correcta (intentos exitosos de realizar las tareas mentales requeridas) y el número total de ensayos realizados.

■ Prueba test

En esta prueba se evaluó 20 acciones de clicks, distribuyéndose de forma equitativa para que los participantes emitan aleatoriamente 10 clicks izquierdos y 10 clicks derechos, por lo que los resultados obtenidos en esta prueba se indican en la Tabla 5.10.

Tabla 5.10: Resultados de la prueba test

Id	ClickIzq	ClickDer	% Acierto Clicks
Participante 1	10/10	10/10	100 %
Participante 2	10/10	9/10	95 %
Participante 3	9/10	8/10	85 %
Participante 4	9/10	9/10	90 %
Participante 5	9/10	8/10	85 %
Promedio	—	—	91 %

La forma de interpretación de los resultados es la siguiente:

$$ParticipanteX = \frac{ClicksY \text{ detectados por el sistema}}{Intentos de clicksY} \quad (5.1)$$

Donde, en 5.1 la letra Y representa el correspondiente evento de click del mouse a realizarse, sea click izquierdo o click derecho.

De acuerdo a la Tabla 5.10, se obtuvo un promedio total del 91 % en torno al porcentaje de aciertos sobre los eventos de clicks solicitados a los cinco participantes en dicha prueba.

Por su parte, para evaluar el rendimiento sobre los movimientos del cursor del mouse, se consideró como acierto si el participante es capaz de posicionar el cursor del mouse sobre 5 posiciones de la pantalla (esquina superior izquierda, esquina superior derecha, centro de la pantalla, esquina inferior izquierda y esquina inferior derecha), por lo que los resultados obtenidos para la prueba del cursor del mouse se indican en la Tabla 5.11.

Tabla 5.11: Resultados sobre el control del cursor del mouse

Id	Cursor	% Acierto Mov.Cursor
Participante 1	5/5	100 %
Participante 2	5/5	100 %
Participante 3	5/5	100 %
Participante 4	5/5	100 %
Participante 5	5/5	100 %
Promedio	—	100 %

La Tabla 5.11 muestra que todos los participantes cumplieron satisfactoriamente dicha parte de la prueba, obteniendo un promedio total del 100 % sobre los movimientos del cursor del mouse a las distintas posiciones de la pantalla solicitadas.

▪ Prueba sobre la interfaz de usuario

En esta prueba se evidenció anticipadamente que únicamente los eventos de click izquierdo actúan sobre los distintos botones o caracteres que posee la interfaz de usuario desarrollada, por lo que, la forma de evaluar dicha prueba se basó en la cantidad de guiños izquierdos que necesitó utilizar cada participante para cumplir la totalidad de la prueba que se indicó anteriormente. En un inicio de forma experimental se determinó que con un total de 17 clicks izquierdos realizados, dicha prueba puede ser cumplida en su totalidad, asumiendo que no se equivocó sobre alguna tecla o caracter mientras cumplía la prueba solicitada.

Se incluyó la métrica de tiempo para determinar cuántos minutos les tomó a los participantes alcanzar los objetivos planteados en dicha prueba.

Tabla 5.12: Resultados de la prueba sobre la interfaz de usuario

Id	ClickIzq	% Acierto Click	Tiempo
Participante 1	19/21	90.47 %	2' 58"
Participante 2	21/24	87.5 %	3' 12"
Participante 3	19/22	86.36 %	3' 24"
Participante 4	22/25	88 %	3' 08"
Participante 5	23/27	85.18 %	3' 52"
Promedio	—	87.50 %	—

La forma de interpretación de los resultados es la siguiente:

$$ParticipanteX = \frac{ClicksIzq\ detectados\ por\ el\ sistema}{N^{\circ}\ de\ guinosIzq\ utilizados\ en\ la\ prueba} \quad (5.2)$$

Dicha prueba fue cumplida es su totalidad por todos los participantes, por lo que, en base a los resultados mostrados en la Tabla 5.12, se puede apreciar que todos los participantes para cumplir la totalidad de la prueba planteada, emitieron un número mayor de clicks izquierdos de los necesarios (17), siendo el participante 5 a quién le costó mayor trabajo y por ende se demoró más en cumplir la prueba en relación al resto de participantes.

De tal manera que, el promedio general de aciertos de la prueba de la interfaz de usuario fue de: 87.50 %.

■ Prueba general sobre el sistema

En esta prueba, a fin de dar uso a los dos eventos de clicks del mouse, se definió una secuencia de objetivos a ser cumplidos en el entorno de Windows del computador; por lo que, la forma de evaluar dicha prueba se basó en la cantidad de guños izquierdos y derechos que necesitó utilizar cada participante para cumplir la totalidad de la prueba. En un inicio de forma experimental se determinó que con un total de 27 clicks izquierdos y 7 clicks derechos realizados, dicha prueba puede ser cumplida en su totalidad, asumiendo que no se equivocó sobre algún icono, tecla, caracter mientras cumplía la prueba solicitada. La Tabla 5.13 indica los resultados obtenidos en esta prueba.

La forma de interpretación de los resultados es la siguiente:

$$ParticipanteX = \frac{ClicksY\ detectados\ por\ el\ sistema}{N^{\circ}\ de\ guinosY\ utilizados\ en\ la\ prueba} \quad (5.3)$$

Tabla 5.13: Resultados de la prueba del sistema

Id	ClickIzq	ClickDer	% Acierto	Tiempo
Participante 1	29/31	7/8	90.52%	4' 02"
Participante 2	32/35	7/8	89.46%	4' 10"
Participante 3	29/33	7/9	82.82%	4' 35"
Participante 4	31/35	7/8	88.03%	4' 22"
Participante 5	35/41	7/10	77.68%	4' 58"
Promedio	—	—	85.70%	—

De igual manera, dicha prueba fue cumplida es su totalidad por todos los participantes, evidenciando de acuerdo a lo mostrado en la Tabla 5.13, que todos los participantes para cumplir la totalidad de la prueba planteada, emitieron un número mayor de clicks izquierdos (27) y clicks derechos (7) de los necesarios, siendo el participante 1 quién mejor rendimiento obtuvo en esta prueba. Por lo que el promedio general de aciertos de la prueba sobre el sistema fue de: 85.70%.

■ Resultados generales

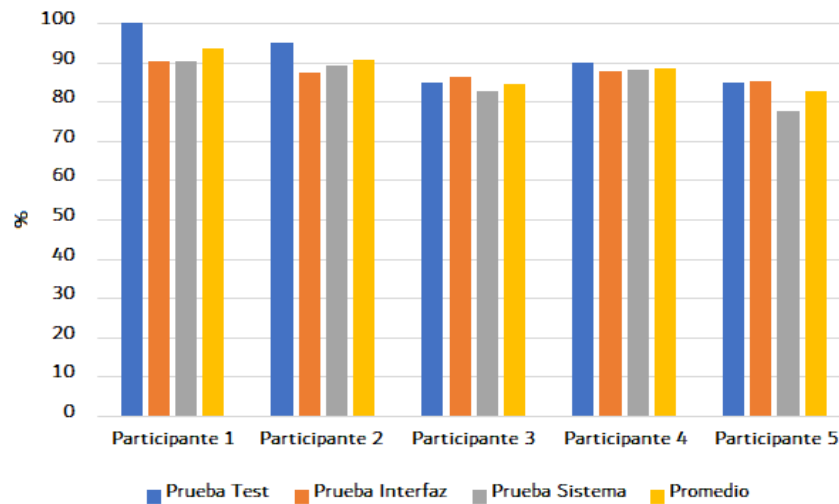


Figura 5.14: Resultados generales de los cinco participantes

La Figura 5.14 permite tener una referencia de los resultados obtenidos por los cinco participantes en las tres pruebas consideradas, por lo que, en la Tabla 5.14 se resume de forma individual los puntajes alcanzados por cada participante en sus respectivas pruebas, además del promedio total obtenido en sus pruebas, y el promedio general alcanzado por el sistema desarrollado.

Tabla 5.14: Resultados generales del sistema desarrollado

Id	Prueba Test	Prueba Interfaz	Prueba Sistema	Promedio
Participante 1	100 %	90.47 %	90.52 %	93.66 %
Participante 2	95 %	87.5 %	89.46 %	90.65 %
Participante 3	85 %	86.36 %	82.82 %	84.72 %
Participante 4	90 %	88 %	88.03 %	88.67 %
Participante 5	85 %	85.18 %	77.68 %	82.62 %
Promedio	91 %	87.50 %	85.70 %	88.06 %

En base a los resultados mostrados en la Tabla 5.14, se llega a determinar que el rendimiento del sistema desarrollado obtuvo una precisión del 88.06 %, tomando como referencia que la fase de pruebas fue hecha con cinco participantes, de los cuales, dos se encontraban en condiciones físicas normales y los otros tres presentaban alguna limitación en sus brazos.

5.5.1. Discusión de resultados

En primera instancia, se va establecer una comparativa con los resultados obtenidos de las investigaciones que fueron base del presente proyecto, las mismas que se mencionaron en la revisión literaria del estado del arte, por lo que, en la Tabla 5.15 se indica la precisión de clasificación obtenida de acuerdo al algoritmo de clasificación empleado por diversos autores.

Tabla 5.15: Resultados obtenidos por otros autores

Autor	Algoritmo	% Accuracy
[Campbell et al., 2010]	Clasificador Bayesiano	95.58 %
[Shedeed and Issa, 2016]	MLP	92.2 %
[Abiyev et al., 2016]	Redes Neuronales Difusas	>92 %

Si bien, el eje central del presente proyecto se basó en hacer uso de una de las técnicas de la Inteligencia Artificial como es el algoritmo de clasificación de redes neuronales, en la que se empleó la arquitectura de una red perceptrón multicapa (MLP), con la cual durante la fase de entrenamiento se obtuvo un promedio de precisión de clasificación por parte del clasificador del 91.53 %. Es así que, haciendo referencia a los resultados mostrados en la Tabla 5.15 por los autores guía de la presente investigación, se puede determinar que la precisión del clasificador obtenido es aceptable encontrándose muy cercano a dichos resultados.

Respecto a los resultados generales obtenidos en el sistema, si se toma en cuenta que uno de los objetivos planteados en un inicio fue el de validar el sistema en personas con movimiento reducido en sus extremidades superiores, en este caso de los cinco participantes considerados, fueron tres de ellos los que al momento de efectuar las pruebas presentaban movilidad reducida en sus brazos, encontrándose vendados, en virtud de esto, considerando previamente que las pruebas fueron cumplidas en su totalidad, si se analiza la Tabla 5.14 únicamente los resultados de los participantes 2, 3 y 4, y se determina el promedio de dichos participantes, se obtiene que el rendimiento del sistema sobre ellos obtuvo una precisión del 88.01 %, y a su vez, si se toma en cuenta que dichos participantes en mención fueron ajenos al proceso de obtención de señales EEG durante las grabaciones; permite responder a la pregunta de investigación planteada en un inicio, determinando que el algoritmo de clasificación de redes neuronales es una vía válida para clasificar señales EEG provenientes de personas que presentan algún tipo de movilidad reducida en sus extremidades superiores, en este caso de estudio permitió clasificar dichas señales para asociarlas a los eventos de clicks del mouse.

Por su parte, se evidenció sobre los cinco participantes que el control sobre la posición del cursor del mouse en base a los movimientos de su cabeza, no representó ningún obstáculo; prueba de ello fue que todos alcanzaron el máximo puntaje, comprobándose que a medida que se hace uso del sistema en tiempo prolongado, genera en los participantes cansancio y fatiga. Es así que, el participante 5 más allá de que no tenía ningún tipo de lesión, fue a quién le costó mayor dominio en el sistema, en la que factores como su edad hacía que le cueste tener control sobre sus guiños, lo que generaba que el sistema emita falsos clicks o identifique como ninguno.

Finalmente, si se quiere mejorar la precisión del sistema se puede considerar probar otros algoritmos de clasificación, dispositivos de captura EEG más robustos, estrategias de señales de control más cómodas para el usuario, etc.

Conclusiones y Recomendaciones

En esta sección se mencionarán las respectivas conclusiones y recomendaciones a las que se llegaron luego del desarrollo del sistema BCI basado en redes neuronales y movimientos de la cabeza para el manejo de un ordenador.

Para culminar la presente investigación, se mencionarán las conclusiones generales a las que se ha llegado durante el proceso de desarrollo del sistema BCI, tomando como referencia el cumplimiento de los objetivos que dieron lugar a la investigación realizada.

La selección de la diadema Emotiv Insight como dispositivo de captura de señales EEG, resultó eficiente para el propósito establecido, dado que el número de canales y sensores de movimiento que posee internamente, fueron determinantes para el procesamiento de las señales.

A nivel de software, el hacer uso del lenguaje de programación Python fue fundamental, gracias a la compatibilidad que tiene con la API Cortex de Emotiv, permitiendo obtener los flujos de datos de los electrodos y acelerómetros en tiempo real.

La red neuronal diseñada fue una red perceptrón multicapa (MLP), en la cual, para el proceso de aprendizaje de la red, se aplicó la herramienta de validación cruzada estratificada al data set que contenía los features extraídos, la misma que permitió agrupar las muestras en 80 % para entrenamiento y 20 % para prueba, de tal manera que, se garantizó que haya proporcionalidad al momento de la toma de muestras sobre las tres clases y de esta manera evitar el sobreajuste del modelo.

Durante el proceso de entrenamiento de la red neuronal diseñada, se obtuvo durante las cinco iteraciones de entrenamiento realizadas un promedio de precisión de clasificación por parte del clasificador del 91.53 %. A su vez, el proceso de validación del modelo se lo realizó en base al set de prueba de la iteración de entrenamiento que mejor score de precisión alcanzó en los sets de entrenamiento y prueba, en la que se utilizó métricas de clasificación utilizadas en Inteligencia Artificial (Machine Learning), alcanzando el modelo de clasificación obtenido, un accuracy del 93 %.

Al sistema BCI desarrollado se acopló los sensores de movimiento que dispone la diadema, en la que a través de los acelerómetros (AccX y AccY), se asoció los desplazamientos del cursor del mouse con los movimientos de la cabeza del usuario. Además, se utilizó la herramienta del Filtro de Kalman para suavizar, minimizar algún ruido durante las mediciones obtenidas por los acelerómetros utilizados, y a su vez estimar la posición real del cursor sobre la pantalla del ordenador.

Para que el cursor del mouse adquiriera un cambio de velocidad al momento que se encuentra desplazándose hacia alguna dirección, dependerá del grado de inclinación que realice el usuario con su cabeza.

Respecto al funcionamiento ON LINE del sistema, se manejó ventanas de tiempo con un tamaño de 200, es decir, el sistema fue recogiendo 200 muestras a lo largo del tiempo, las mismas que posteriormente fueron procesadas para

obtener sus respectivos features, estos a su vez se convertían en información de entrada para que el modelo generado emita sus respectivas predicciones en ese instante de tiempo; dicho valor se lo eligió tomando como referencia el número de muestras promedio que compone una acción de guiño.

Los tiempos de procesamiento del sistema fueron los siguientes: tiempo de adquisición de la información a procesarse: 1.56 segundos, tiempo de procesamiento (cálculos): 0.05 segundos; por lo tanto, el tiempo de respuesta del sistema fue de aproximadamente 1.6 segundos.

Por otra parte, en cuanto al funcionamiento del sistema BCI desarrollado, el proceso de validación se lo realizó sobre 5 participantes, determinando que el rendimiento del sistema obtuvo una precisión del 88.06 %.

Finalmente, es oportuno mencionar que el sistema BCI desarrollado no sería aplicable en personas que poseen algún tipo de bloqueo total sobre la movilidad de su cuerpo, pero representa una alternativa interesante para aquellas personas que están imposibilitadas parcial o totalmente de sus brazos, por lo que, al disponer de la movilidad de su cabeza y el control sobre sus gestos faciales, pueden cumplir con el manejo básico de un ordenador.

Así mismo, es necesario mencionar ciertas recomendaciones a tomarse en cuenta del sistema BCI desarrollado.

Uno de los aspectos a considerar, es verificar que exista un buen contacto por parte de los electrodos de la diadema sobre el cuero cabelludo del usuario, por lo que es necesario hidratar los electrodos previamente a la utilización del sistema; además, considerar que si el sistema empieza a disminuir su grado de efectividad en torno a la detección de clicks del mouse ante las acciones de guiños correspondientes, se debe principalmente a que los electrodos han perdido su grado de conductividad, por lo que la solución radica en volver a hidratarlos. En virtud de esto, factores como calidad de contacto de los electrodos, cantidad de cabello del usuario, representan aspectos a considerarse para que el sistema tenga el rendimiento esperado.

Para obtener los resultados que se indicó en la Tabla 5.14, fue de suma importancia permitir que los participantes realicen entrenamientos con sus guiños de ojos, dado que en un inicio todos los participantes a excepción del participante 1, les costaba dominar la acción de guiño sobre sus ojos, por lo que una vez que se verificó que los participantes ya demostraban tener control sobre sus guiños, se procedió a evaluarlos en las distintas pruebas.

El uso del sistema BCI desarrollado, requiere de acceso a internet en la computadora a utilizarse para establecer la conexión a la API Cortex de Emotiv; y la gran limitante encontrada, disponer alguna licencia de Emotiv, a fin de tener acceso a los flujos de datos en tiempo real de las señales EEG.

Bibliografía

- R. H. Abiyev, N. Akkaya, E. Aytac, I. Günsel, and A. Çağman. Brain-computer interface for control of wheelchair using fuzzy neural networks. *BioMed research international*, 2016, 2016.
- R. Alazrai, H. Alwanni, and M. I. Daoud. Eeg-based bci system for decoding finger movements within the same hand. *Neuroscience letters*, 698:113–120, 2019.
- J. Almira and M. Domingo. *Neuromatemáticas: el lenguaje eléctrico del cerebro. ¿Qué sabemos de?* Consejo Superior de Investigaciones Científicas, 2016. ISBN 9788400101152. URL <https://books.google.com.ec/books?id=KVpPvgAACAAJ>.
- S. Amiri, R. Fazel-Rezai, and V. Asadpour. A review of hybrid brain-computer interface systems. *Advances in Human-Computer Interaction*, 2013, 2013.
- A. Athisakthi and M. P. Rani. Statistical energy values and peak analysis (sep) approach for detection of neurodegenerative diseases. In *2017 World Congress on Computing and Communication Technologies (WCCCT)*, pages 240–245. IEEE, 2017.
- A. Campbell, T. Choudhury, S. Hu, H. Lu, M. K. Mukerjee, M. Rabbi, and R. D. Raizada. Neurophone: brain-mobile phone interface using a wireless eeg headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, pages 3–8, 2010.
- H.-F. Chiu, K. Venkatakrisnan, and C.-K. Wang. The role of nutraceuticals as a complementary therapy against various neurodegenerative diseases: A mini-review. *Journal of Traditional and Complementary Medicine*, 2020.

- E. Emotiv. *Emotiv Inc. Website*, 2020. URL <https://www.emotiv.com/about-emotiv/>.
- I. Fernández-Cruz and E. Reynaud. Proteasome subunits involved in neurodegenerative diseases. *Archives of Medical Research*, 2020.
- J. Gómez and M. Sanz. *El cerebro del siglo XXI*. Editorial El Manual Moderno Colombia S.A.S., 2018. ISBN 9789588993348. URL <https://books.google.com.ec/books?id=6F5-DwAAQBAJ>.
- B. Graimann, B. Allison, and G. Pfurtscheller. Brain-computer interfaces: A gentle introduction. In *Brain-computer interfaces*, pages 1–27. Springer, 2009.
- J. E. Hall and M. E. Hall. *Guyton and Hall textbook of medical physiology e-Book*. Elsevier Health Sciences, 2020.
- A. E. Hassanien and A. Azar. Brain-computer interfaces. *Switzerland: Springer*, 2015.
- N. Irtiza and H. Farooq. Classification of brain states using subject-specific trained classifiers. *University of Engineering and Technology Taxila. Technical Journal*, 21(2):96, 2016.
- S. Lee, Y. Shin, S. Woo, K. Kim, and H.-N. Lee. Dry electrode design and performance evaluation for eeg based bci systems. In *2013 International Winter Workshop on Brain-Computer Interface (BCI)*, pages 52–53. IEEE, 2013.
- R. F. López, J. M. Fernández, and J. M. F. Fernández. *Las redes neuronales artificiales*. Netbiblo, 2008.
- S. Michahial, R. R. Kumar, P. H. Kumar, and A. P. Kumar. Hand rotate eeg signal feature extraction by second order daubechies wavelet transform (dwt). In *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, pages 1–6. IEEE, 2012.
- S. O'Regan, S. Faul, and W. Marnane. Automatic detection of eeg artefacts arising from head movements. In *2010 annual international conference of the ieee engineering in medicine and biology*, pages 6353–6356. IEEE, 2010.

- J. S. Patrón and C. R. B. Monje. Emotiv epoc bci with python on a raspberry pi. *Sistemas y Telemática*, 14(36):27–38, 2016.
- R. A. Ramadan and A. V. Vasilakos. Brain computer interface: control signals review. *Neurocomputing*, 223:26–44, 2017.
- G. Rodríguez-Miranda, R. Santos-Osorio, N. Ledesma-Uribe, and J. Camacho-Ramírez. Estudio de las señales electroencefalográficas (eeg) desarrollando una aplicación basada en interfaz cerebro-computadora (bci), para el apoyo del aprendizaje infantil. *Revista de Sistemas Computacionales y TIC's*, 2(6):1–4, 2016.
- H. A. Shedeed and M. F. Issa. Brain-eeg signal classification based on data normalization for controlling a robotic arm. *Int. J. Tomogr. Simul.*, 29(1):72–85, 2016.
- R. Sorbello, S. Tramonte, M. E. Giardina, V. La Bella, R. Spataro, B. Allison, C. Guger, and A. Chella. A human–humanoid interaction through the use of bci for locked-in als patients using neuro-biological feedback fusion. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2):487–497, 2017.
- M. Teplan et al. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.
- N. Tiwari, D. R. Edla, S. Dodia, and A. Bablani. Brain computer interface: A comprehensive survey. *Biologically inspired cognitive architectures*, 26:118–129, 2018.
- A.-G. Wu, X.-G. Zhou, G. Qiao, L. Yu, Y. Tang, L. Yan, W.-Q. Qiu, R. Pan, C.-L. Yu, B. Y.-K. Law, et al. Targeting microglial autophagic degradation in nlrp3 inflammasome-mediated neurodegenerative diseases. *Ageing research reviews*, page 101202, 2020.
- X. Zhang, D. Hu, Y. Shang, and X. Qi. Using induced pluripotent stem cell neuronal models to study neurodegenerative diseases. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1866(4):165431, 2020.

Apéndice A

Uso de la diadema Emotiv Insight

Como punto inicial para llevar a cabo los distintos experimentos que permitirán poner en marcha el desarrollo del sistema, se indicarán aspectos a tomar en cuenta respecto al manejo de la diadema Emotiv Insight.

(a) **Conexión de la diadema**

La diadema Emotiv Insight utiliza como medio de enlace hacia el computador el uso de la tecnología inalámbrica Bluetooth, por lo que es necesario el emparejamiento entre la diadema y el computador para que exista una comunicación entre ambos, la misma que se la realiza mediante la utilización de un receptor USB (ver Figura 3.5(b)) propio de la empresa Emotiv, que es colocado en un puerto USB del equipo a utilizarse, seguido a esto se procede a encender la diadema a través del pulsador que se encuentra ubicado en el extremo inferior de la misma, esperando alrededor de unos 5 segundos hasta que el led que integra el receptor USB comience a parpadear rápidamente, indicando que la conexión entre la diadema y el computador ya se encuentra listo.

(b) **Hidratación de los electrodos**

Es necesario que los 5 electrodos y los 2 de referencia que dispone la diadema, sean hidratados previamente mediante la utilización de un gel (solución salina), a fin de que estos puedan obtener un buen contacto con el cuero cabelludo de la persona a colocarse, de tal manera que logren captar las señales cerebrales de forma eficiente.

(c) **Colocación de la diadema**

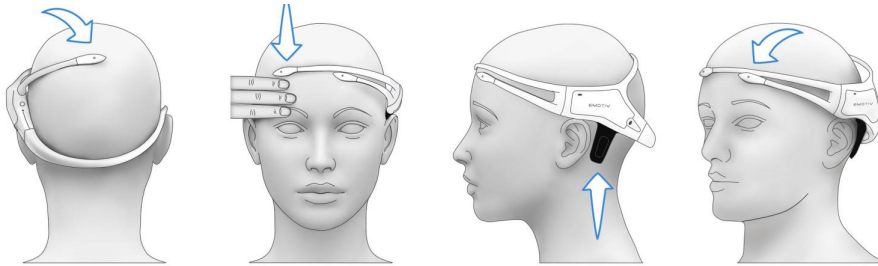


Figura A.1: Colocación de la diadema Emotiv Insight

El protocolo a seguir para la colocación de la diadema sobre la persona, es mostrado en la Figura A.1, siendo necesario identificar como están situados los 2 electrodos de referencia en la diadema, los mismos que

deben hacer contacto con la piel justo detrás de la oreja izquierda, posteriormente los sensores frontales deben ser colocados tomando en cuenta que se sitúen unos tres dedos por encima de las cejas.

En este punto, es conveniente comprobar el estado del nivel de calidad de conexión de cada uno de los 5 electrodos de la diadema. El software EmotivPRO dispone entre sus funcionalidades la posibilidad de verificar en tiempo real el nivel de contacto de cada electrodo sobre el cuero cabelludo de la persona, por lo que en la Figura A.2 se muestra la representación típica al momento de verificar la calidad de conexión de los electrodos haciendo uso del software EmotivPRO.

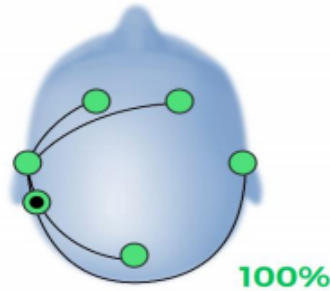


Figura A.2: Calidad de conexión de los electrodos

De igual manera, es necesario tomar en cuenta que el estado del nivel de calidad de conexión de cada electrodo, puede variar dependiendo del nivel de contacto que tenga sobre el cuero cabelludo de la persona, por lo que en la Tabla A.1 se indican las posibles representaciones que se pueden tener al momento de colocarse la diadema sobre la persona.

Tabla A.1: Representación del nivel de contacto de los electrodos

Color	Representación
Verde	Contacto bueno
Naranja	Contacto moderado
Rojo	Contacto pobre
Negro	Contacto demasiado pobre o nulo

Apéndice B

Entrenamientos virtuales

La herramienta EmotivBCI será utilizada principalmente para ayudar a que el usuario pueda familiarizarse con el control de sus propias señales EEG hacia un objeto virtual. Es así que dispone de un cubo virtual en 3D como se indica en la Figura B.1, para que el usuario busque tener el control sobre el mismo, mediante la realización de múltiples entrenamientos en base a detecciones de comandos mentales o expresiones faciales.

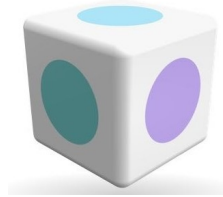


Figura B.1: Cubo virtual para entrenamientos mediante EmotivBCI

(a) **Creación de un perfil de entrenamiento**

Es necesario crear un perfil de entrenamiento, a fin de que el usuario pueda acceder a los distintos entrenamientos en un futuro sobre su respectivo perfil.

(b) **Objetivo del entrenamiento**

El usuario debe entrenar distintos comandos mentales, que le permitan ejercer una acción de control sobre el cubo virtual que aparecerá en pantalla durante la fase de entrenamiento. Algunas de las acciones de control que se pueden realizar sobre dicho cubo se encuentran: empujar el cubo, mover a distintas direcciones el cubo, rotar el cubo, etc.

(c) **Entrenamiento neutral**

Es necesario que el usuario entrene previamente un estado neutral, a fin de tratar de mantener su mente en blanco y en un estado de relajación durante los 8 segundos que dura el entrenamiento, permitiendo contrastar la actividad cerebral de los demás entrenamientos que se realicen.

(d) **Entrenamiento 1: Empujar el cubo**

Una vez realizado el entrenamiento neutral, el usuario durante 8 segundos debe emitir algún tipo de comando mental, a fin que dicho comando mental quede asociado para la acción de empujar el cubo, tal

como se puede apreciar en la Figura B.2(a), es recomendable que los entrenamientos se lo realicen al menos unas 5 veces.

(e) **Entrenamiento 2: Mover el cubo a la derecha**

El usuario realiza el procedimiento de forma similar al entrenamiento 1, con la diferencia de que el usuario debe emitir un nuevo comando mental totalmente distinto al utilizado en el entrenamiento 1, logrando que el cubo se mueva hacia el lado derecho de la pantalla como se puede apreciar en la Figura B.2(b).



(a) Empujar el cubo

(b) Cubo a la derecha

Figura B.2: Entrenamientos realizados sobre el cubo virtual

(f) **Modo en vivo**

Permite al usuario verificar si el cubo virtual responde en tiempo real ante los comandos mentales entrenados. La Figura B.3 muestra el diagrama del espacio cerebral, que permite identificar a través de puntos de colores si existe diferenciación entre ambos pensamientos para cada una de las acciones de control fijadas, siendo que mientras más alejadas estén mejor será la diferenciación entre los dos comandos mentales entrenados.

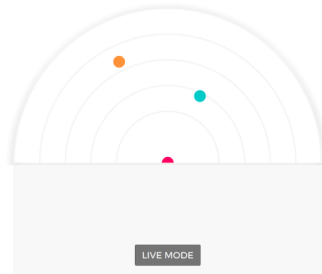


Figura B.3: Diagrama del espacio cerebral de los entrenamientos realizados

Apéndice C

Manejo de archivos .csv

El disponer activa la licencia Emotiv Student, proporciona acceso a realizar grabaciones ilimitadas de las señales EEG y su posterior exportación en archivos de formato .csv de Excel, para de esta manera poder utilizar la información recopilada en dichos archivos para el desarrollo del sistema BCI.

	A	B	C	D	E	F	G	H	I
1	title:Tder20, mot_64		pm_0.1	pow_8, samples:1152		, version:2.0			
2	Timestamp,EEG.Counter,EEG.Interpolated,EEG.AF3,EEG.T7,EEG.Pz,EEG.T8,EEG.AF4,EEG.RawCq,EEG.Battery,Mar								
3	1596321532.066845,	39.000000,	0.000000,	4282.051270,	4250.256348,	4181.538574,	4218.461426,	4272.307617,	0.000000
4	1596321532.074540,	40.000000,	0.000000,	4302.563965,	4244.615234,	4193.333496,	4222.051270,	4287.692383,	0.000000
5	1596321532.082234,	41.000000,	0.000000,	4298.461426,	4226.153809,	4182.051270,	4205.128418,	4289.230957,	0.000000
6	1596321532.090029,	42.000000,	0.000000,	4279.487305,	4227.179688,	4172.307617,	4200.512695,	4281.538574,	0.000000
7	1596321532.097823,	43.000000,	0.000000,	4281.025879,	4230.769043,	4183.077148,	4211.282227,	4272.307617,	0.000000
8	1596321532.105517,	44.000000,	0.000000,	4282.563965,	4214.358887,	4178.974121,	4199.487305,	4263.077148,	0.000000
9	1596321532.113212,	45.000000,	0.000000,	4276.922852,	4213.846191,	4169.230957,	4187.179688,	4260.000000,	0.000000
10	1596321532.121006,	46.000000,	0.000000,	4285.128418,	4240.512695,	4177.436035,	4194.871582,	4264.615234,	0.000000
11	1596321532.128701,	47.000000,	0.000000,	4288.205078,	4253.333496,	4177.436035,	4191.794922,	4265.641113,	0.000000
12	1596321532.136396,	48.000000,	0.000000,	4283.589844,	4251.282227,	4169.743652,	4186.666504,	4264.102539,	0.000000
13	1596321532.144190,	49.000000,	0.000000,	4280.000000,	4237.436035,	4165.128418,	4190.769043,	4257.436035,	0.000000
14	1596321532.151985,	50.000000,	0.000000,	4266.666504,	4215.384766,	4146.666504,	4178.974121,	4247.692383,	0.000000
15	1596321532.159679,	51.000000,	0.000000,	4257.436035,	4210.256348,	4128.205078,	4163.077148,	4241.538574,	0.000000
16	1596321532.167474,	52.000000,	0.000000,	4263.077148,	4213.846191,	4134.871582,	4168.717773,	4235.384766,	0.000000

Figura C.1: Ejemplo de formato de archivos .csv

En la Figura C.1 se puede apreciar un ejemplo de archivo .csv, el mismo que resultó después de haber realizado la exportación de una grabación, en la que se puede visualizar una serie de valores numéricos que corresponden a diversos parámetros captados durante el lapso de tiempo que duró una determinada grabación realizada.

Para el desarrollo del sistema BCI, se hará uso de cada uno de los archivos .csv que se hayan recopilado durante los procesos de grabación realizados, dado que estos archivos constituyen la herramienta que permite dar inicio al desarrollo de cada una de las fases del sistema BCI, siendo la fase de adquisición de señales la encargada de leer la información contenida por cada uno de estos archivos a través del lenguaje de programación que se utilice.

En la Tabla C.1 se indican las etiquetas más importantes contenidas por cada columna de cualquier archivo .csv que se exporte.

Tabla C.1: Etiquetas y descripción de los archivos .csv

Etiqueta	Descripción
Timestamp	Tiempo dado en segundos después que la diadema se conectó a la aplicación de la que se tomó la muestra.
EEG.Counter	Contador de paquetes de datos recibidos por parte de la diadema, sus valores van desde 0 hasta 128.
EEG.Interpolated	Muestra si se descartó un paquete asignando el valor de 1, por su parte, si asigna el valor de 0 indica que la muestra es buena en ese instante de tiempo.
EEG. (AF3,T7,Pz,T8,AF4)	Datos de cada uno de los electrodos de la diadema medidos en microvoltios (uV), a lo largo del tiempo.
EEG.Battery	Indica el nivel de batería que se tuvo en cada muestra recopilada, en un rango de 4 a 1, donde 4 representa una máxima señal obtenida y 1 una mínima señal.
CQ. (AF3,T7,Pz,T8,AF4)	Muestra el color del nivel de contacto que se obtuvo por parte de los electrodos en cada muestra, siendo: 1 = negro, 2 = rojo, 3 = naranja, 4 = verde
MOT. (Q0, Q1, Q2, Q3 AccX, AccY, AccZ MagX, MagY, MagZ)	Muestran los datos de movimiento obtenidos para los cuatro cuaterniones, los tres acelerómetros y los tres magnetómetros a lo largo del tiempo.