

# UNIVERSIDAD SANTO TOMÁS

---

## DESARROLLO DE UN SISTEMA BCI BASADO EN EL DISPOSITIVO MINDWAVE MOBILE 2 PARA EL CONTROL DE LAS FUNCIONES DE MOVIMIENTO DEL ROBOT PEPPER

---

Realizado por

Johan Sebastián Martínez Burgos



Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)  
Facultad de Ingeniería Electrónica  
División de Ingenierías

2023

---

# **DESARROLLO DE UN SISTEMA BCI BASADO EN EL DISPOSITIVO MINDWAVE MOBILE 2 PARA EL CONTROL DE LAS FUNCIONES DE MOVIMIENTO DEL ROBOT PEPPER**

---

Realizado por

**Johan Sebastián Martínez Burgos**

Proyecto de Grado presentado en cumplimiento del requisito parcial  
para optar por el grado de Ingeniería Electrónica

Dirigido por

**Armando Mateus Rojas**

Codirigido por

**Juan Manuel Calderón Chávez**

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)  
Facultad de Ingeniería Electrónica  
División de Ingenierías

2023

## **Autoridades de la Universidad**

### **RECTOR GENERAL**

R.P. FRAY ÁLVARO JOSÉ ARANGO RESTREPO, O.P.

### **VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL**

R.P. FRAY LUIS FRANCISCO SASTOQUE POVEDA, O.P.

### **VICERRECTOR ACADÉMICO GENERAL**

R.P. FRAY EDUARDO GONZALEZ GIL, O.P.

### **SECRETARIO GENERAL**

Dra. INGRID LORENA CAMPOS VARGAS

### **DECANO DIVISIÓN DE INGENIERÍAS**

R.P. FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

### **SECRETARIA DE DIVISIÓN**

E.C. LUZ PATRICIA ROCHA CAICEDO

### **DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA**

ING. CARLOS ANDRÉS TORRES PINZÓN

## Nota de aceptación

---

---

---

---

---

---

---

Firma del autor

---

Firma del jurado

---

Firma del jurado

BOGOTÁ D.C. — DE 2023



## **Advertencia**

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

**Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.**

## **Dedicatoria**

Dedico este proyecto a mis padres por brindarme la oportunidad de poder estudiar, así como el constante apoyo recibido durante toda mi carrera, a mi hermana por su gran apoyo durante mi formación como futuro ingeniero electrónico.

Johan Sebastián Martínez Burgos

## **Agradecimientos**

Agradezco a todas las personas que de una u otra manera me apoyaron en el desarrollo y culminación de este proyecto de grado.

Johan Sebastián Martínez Burgos

## Resumen

El presente proyecto de grado se describe el diseño, implementación y desarrollo de un sistema BCI *Brain Computer Interface* basado en la recepción de ondas cerebrales el cual dota al robot *Pepper* de diferentes funcionalidades, permitiendo realizar una interacción inalámbrica con el entorno y utilizando al robot como medio de comunicación. El proyecto está enfocado hacia la robótica social y el apoyo a personas en condición de discapacidad las cuales tiene dificultades comunicativas.

Para la recepción de ondas se utiliza la diadema *MindWave Mobile 2* desarrollada por la empresa *Neurosky*. Esta diadema permite realizar la toma de muestras de los parpadeos voluntarios, estos parpadeos son clasificados mediante una red neuronal de tipo *Long-Short Term Memory* (LSTM), la cual es efectiva en entrenamientos de clasificación en series de tiempo. A su vez, el robot *Pepper* se programó con funcionalidades como el habla correspondiente a una pequeña descripción del proyecto y dos acciones de saludo diferentes, saludo con la mano y apretón de mano. Estas funcionalidades es una muestra de la capacidad de comunicación del robot con el entorno.

Por último, se realiza el sistema BCI mediante el entorno ROS *Robot Operating System*, el cual es un meta sistema operativo diseñado para robots, donde finalmente se obtienen los resultados de la clasificación de los parpadeos y las funcionalidades que adopta el robot dependiendo de los parpadeos voluntarios.

**Palabras clave:** BCI, EEG, Pepper, LSTM, MindWave Mobile 2, Parpadeos.

## Abstract

The present degree project describes the design, implementation, and development of a BCI system based on the reception of brain waves which enables the functionalities of the *Pepper* robot, allowing wireless interaction with the environment and using the robot as a means of communication. The project is focused on social robotics and support for people with disabilities who have communication difficulties.

For the reception of waves, the *MindWave Mobile 2* headband developed by the company *Neurosky* is used. This headband allows sampling of voluntary blinks, these blinks are classified using a *Long-Short Term Memory* (LSTM) neural network, which is effective in time series classification training. In turn, the *Pepper* robot was programmed with different corresponding speech functionalities a short description of the project and two different greeting actions, thus demonstrating communication with the robot's communication with the robot's environment.

Finally, the BCI system is carried out through the ROS environment, which is a meta operating system designed for robots, where finally the results of the classification of the blinks and the functionalities that the robot adopts depending on the voluntary blinks are obtained.

**Keywords:** BCI, EEG, Pepper, LSTM, MindWave Mobile 2, Blinks.

# Índice general

|                                                            |             |
|------------------------------------------------------------|-------------|
| <b>Lista de Figuras</b>                                    | <b>X</b>    |
| <b>Lista de Tablas</b>                                     | <b>XII</b>  |
| <b>Glosario</b>                                            | <b>XIII</b> |
| <b>1. Introducción</b>                                     | <b>1</b>    |
| 1.1. Planteamiento del Problema . . . . .                  | 2           |
| 1.2. Objetivos . . . . .                                   | 3           |
| 1.3. Objetivos Específicos . . . . .                       | 3           |
| 1.4. Justificación . . . . .                               | 4           |
| 1.5. Impacto Social . . . . .                              | 5           |
| <b>2. Estado del Arte</b>                                  | <b>7</b>    |
| 2.1. Técnicas de los Sistemas BCI . . . . .                | 7           |
| 2.2. Trabajos Relacionados . . . . .                       | 9           |
| 2.3. Adquisición, Interpretación y Procesamiento . . . . . | 11          |
| <b>3. Marco Teórico</b>                                    | <b>12</b>   |
| 3.1. Electroencefalograma EEG . . . . .                    | 12          |
| 3.2. Interfaz BCI . . . . .                                | 13          |
| 3.2.1. Diadema Recepción EEG . . . . .                     | 13          |
| 3.3. <i>Deep Learning</i> . . . . .                        | 14          |
| 3.3.1. Redes Neuronales . . . . .                          | 15          |
| 3.3.2. Redes Neuronales Recurrentes RNN . . . . .          | 17          |
| 3.3.3. Redes LSTM (Long-Short Term Memory) . . . . .       | 18          |
| 3.3.4. Características de Red Neuronal LSTM . . . . .      | 19          |
| 3.4. ROS . . . . .                                         | 20          |
| <b>4. Diseño Metodológico</b>                              | <b>21</b>   |
| 4.1. Investigación . . . . .                               | 22          |
| 4.2. Ejecución . . . . .                                   | 22          |
| 4.3. Construcción . . . . .                                | 22          |
| 4.4. Prueba y Puesta en Marcha del Proyecto . . . . .      | 23          |
| 4.5. Cronograma . . . . .                                  | 23          |

---

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>5. Desarrollo Conceptual</b>                                     | <b>24</b> |
| 5.1. Conexión de la Diadema . . . . .                               | 24        |
| 5.2. Programación de Recepción . . . . .                            | 25        |
| 5.3. Análisis de Datos . . . . .                                    | 26        |
| 5.4. Implementación de Red Neuronal . . . . .                       | 27        |
| 5.5. Entrenamiento de Red Neuronal LSTM . . . . .                   | 28        |
| 5.6. Movimientos del Robot . . . . .                                | 29        |
| 5.7. Entorno ROS . . . . .                                          | 32        |
| 5.8. Evaluación del Sistema BCI . . . . .                           | 33        |
| <b>6. Resultados y Discusión</b>                                    | <b>34</b> |
| 6.1. Recepción y Análisis de Ondas . . . . .                        | 34        |
| 6.2. Implementación de Red neuronal . . . . .                       | 37        |
| 6.3. Funciones de Movimiento Pepper . . . . .                       | 41        |
| 6.4. Sistema BCI . . . . .                                          | 45        |
| 6.5. Escenarios de Prueba . . . . .                                 | 46        |
| 6.5.1. Escenario 1: Entrenamiento . . . . .                         | 46        |
| 6.5.2. Escenario 2: Clasificación en Otros Entornos . . . . .       | 46        |
| 6.5.3. Escenario 3: Toma de Resultados de la Interfaz BCI . . . . . | 47        |
| 6.6. Resultados . . . . .                                           | 48        |
| <b>7. Conclusiones y Trabajos Futuros</b>                           | <b>51</b> |
| 7.1. Conclusiones . . . . .                                         | 51        |
| 7.2. Trabajos Futuros . . . . .                                     | 53        |
| <b>Bibliografía</b>                                                 | <b>54</b> |
| <b>8. Anexos</b>                                                    | <b>59</b> |
| <b>Anexos</b>                                                       | <b>59</b> |
| 8.1. Anexo A: . . . . .                                             | 59        |
| 8.2. Anexo B: . . . . .                                             | 60        |
| 8.2.1. Nodo Comunicación ( <i>Publisher</i> ) . . . . .             | 60        |
| 8.2.2. Nodo Clasificación ( <i>Publisher/Suscriber</i> ) . . . . .  | 60        |
| 8.2.3. Nodo de Funcionalidades ( <i>Suscriber</i> ) . . . . .       | 62        |
| 8.3. Anexo C: . . . . .                                             | 63        |
| 8.4. Anexo D: . . . . .                                             | 65        |
| 8.5. Anexo E: . . . . .                                             | 66        |

# Lista de Figuras

|     |                                                                           |    |
|-----|---------------------------------------------------------------------------|----|
| 1.  | Modelo red neuronal [41]. . . . .                                         | 16 |
| 2.  | Modelo red neuronal perceptrón (adaptado de [21]). . . . .                | 17 |
| 3.  | Red Neuronal Recurrente RNN [45]. . . . .                                 | 18 |
| 4.  | Diagrama ejecución ROS (adaptado de [50]). . . . .                        | 20 |
| 5.  | Metodología implementada. . . . .                                         | 21 |
| 6.  | Diagrama de posicionamiento MindWave Mobile 2 (adaptado de [52]). . . . . | 25 |
| 7.  | Esquema recepción de ondas. . . . .                                       | 26 |
| 8.  | Esquema script modelo de entrenamiento. . . . .                           | 27 |
| 9.  | Esquema de Red Neuronal. . . . .                                          | 28 |
| 10. | Esquema del script de conexión del robot. . . . .                         | 30 |
| 11. | Actuadores de movimientos Pepper [55]. . . . .                            | 31 |
| 12. | Funcionalidades propuestas de Pepper. . . . .                             | 31 |
| 13. | Implementación de nodos en ROS. . . . .                                   | 32 |
| 14. | Muestra EEG movimientos de extremidades. . . . .                          | 35 |
| 15. | Muestra de movimientos oculares. . . . .                                  | 35 |
| 16. | EEG 3 parpadeos voluntarios. . . . .                                      | 36 |
| 17. | Parámetros de exactitud y pérdida del modelo entrenado. . . . .           | 38 |
| 18. | Matriz de confusión modelo LSTM. . . . .                                  | 39 |
| 19. | Detección parpadeos. . . . .                                              | 40 |
| 20. | Predicción de los parpadeos. . . . .                                      | 41 |
| 21. | Parpadeos detectados. . . . .                                             | 41 |
| 22. | Funcionalidad descripción del Proyecto. . . . .                           | 42 |
| 23. | Funcionalidad saludo. . . . .                                             | 43 |
| 24. | Funcionalidad saludo de mano. . . . .                                     | 44 |
| 25. | Ejecución de nodos en ros. . . . .                                        | 45 |
| 26. | 1 Parpadeo. . . . .                                                       | 63 |
| 27. | 2 Parpadeos. . . . .                                                      | 63 |
| 28. | 3 Parpadeos. . . . .                                                      | 64 |
| 29. | Señal con Ruido 1. . . . .                                                | 65 |
| 30. | Señal con Ruido 2. . . . .                                                | 65 |
| 31. | Señal con Ruido 3. . . . .                                                | 66 |



|     |                               |    |
|-----|-------------------------------|----|
| 32. | Señal con Ruido 4. . . . .    | 66 |
| 33. | Movimientos Pepper 1. . . . . | 67 |
| 34. | Movimientos Pepper 2. . . . . | 67 |
| 35. | Movimientos Pepper 3. . . . . | 68 |

# Lista de Tablas

|     |                                                                  |    |
|-----|------------------------------------------------------------------|----|
| 1.  | Técnicas de adquisición, procesamiento e interpretación. . . . . | 11 |
| 2.  | Características de la diadema <i>Mindwave Mobile 2</i> . . . . . | 14 |
| 3.  | Cronograma de Actividades. . . . .                               | 23 |
| 4.  | Modelo 1 Red Neuronal. . . . .                                   | 29 |
| 5.  | Modelo 2 Red Neuronal. . . . .                                   | 29 |
| 6.  | Modelo 3 Red Neuronal. . . . .                                   | 29 |
| 7.  | Modelo 4 Red Neuronal. . . . .                                   | 29 |
| 8.  | Características red neuronal LSTM. . . . .                       | 37 |
| 9.  | Características red neuronal LSTM. . . . .                       | 38 |
| 10. | Parámetros de la red neuronal LSTM. . . . .                      | 38 |
| 11. | Evaluación de 0 parpadeos. . . . .                               | 48 |
| 12. | Evaluación de 1 parpadeo. . . . .                                | 49 |
| 13. | Evaluación de 2 parpadeos. . . . .                               | 49 |
| 14. | Evaluación de 3 parpadeos. . . . .                               | 50 |
| 15. | Efectividad total de la interfaz BCI. . . . .                    | 50 |

# Glosario

- **EEG:** Electroencefalografía.
- **BCI:** Brain Computer Interface.
- **IA:** Inteligencia Artificial.
- **LSTM:** Long Short-Term Memory.
- **RNN:** Red Neuronal Recurrente.
- **SSH:** Secure Shell.
- **ROS:** Robot Operating System.
- **ODS:** Objetivos de desarrollo sostenible.
- **GED:** Grupo de Estudio y Desarrollo en Robótica.
- **RFComm:** Radio Frecuency Communication.

# Capítulo 1

## Introducción

Una de las facultades más importantes de los seres vivos es la comunicación, permite la interacción entre dos o más individuos ya sea de manera verbal y no verbal mediante el envío, recepción e interpretación de la información que se quiere dar a entender. Con los avances tecnológicos se han creado diferentes canales de comunicación como lo son Internet, las redes de telecomunicaciones y la interacción mediante interfaces con la tecnología. En los últimos años, se han desarrollado nuevas técnicas que permiten una conexión más eficiente y compleja con el entorno tecnológico, creando nuevas tecnologías que permiten un mejor desarrollo y una inmersión con la tecnología. Además, estos nuevos desarrollos dan paso a la salud, proporcionando herramientas a personas con algún tipo de condición y llevando una mejor calidad de vida, desde el control de enfermedades mentales y motoras.

Unos de los desarrollos más importantes de la última década son los sistemas BCI (*Brain Computer Interface* por sus siglas en inglés). Este tipo de tecnología consta de la adquisición de las ondas cerebrales, el preprocesamiento de estas señales, la extracción de los datos, la clasificación y finalmente salida para la aplicación del sistema el cual permiten la conexión y comunicación con diferentes dispositivos tecnológicos. El primer acercamiento de los sistemas BCI es de 1924 con el primer electroencefalograma EEG desarrollado por Hans Berger que permitía detectar ondas cerebrales; esta técnica se ha utilizado en los diferentes desarrollos de sistemas BCI. Mediante el EEG se leen los datos de las ondas cerebrales con electrodos dispuestos alrededor de la cabeza. Mediante esta técnica logra interpretar las diferentes ondas existentes en el cerebro, lo que permite realizar una clasificación de la actividad cerebral de los seres humanos. Para procesar los datos arrojados por el EEG, se realizan técnicas como la inteligencia artificial, en concreto las redes neuronales recurrentes que no permiten analizar los datos para su posterior

implementación en diferentes aplicaciones. Este tipo de redes neuronales tienen la posibilidad de aprender con cada medida realizada, obteniendo una mejor validez de respuesta.

En el presente documento se realiza la descripción del proyecto de grado para un sistema BCI que permite dar movilidad a un robot. En este se realiza la pregunta problema de la cual parte el proyecto y la importancia de su desarrollo. Se proponen los objetivos a desarrollar junto con la información detallada con antecedentes y conceptos de lo que se pretende realizar, con una metodología descrita y un cronograma para realizar un desarrollo conceptual y las conclusiones finales del proyecto de grado. Dentro del desarrollo de proyecto de grado se realiza la implementación de una interfaz BCI, en la cual se lleva a cabo la recepción de ondas cerebrales mediante el dispositivo MindWave Mobile 2 el cual es un dispositivo portable que proporciona información acerca del estado mental, emocional y los parpadeos voluntarios de una persona. Con un algoritmo se realiza la toma de los datos de la diadema que luego interpretará una red neuronal recurrente de tipo LSTM, donde se realiza la implementación de esta. La implementación de esta red neuronal se realiza para obtener buenos resultados a partir de datos secuenciales adquiridos. Finalmente, para la conexión con el robot se realiza mediante el meta sistema operativo ROS, con el cual se realiza el algoritmo que permite dar movilidad al robot.

## 1.1. Planteamiento del Problema

Desde hace un tiempo se han venido desarrollando prototipos de *Brain computer interface* (BCI), con el fin de buscar soluciones mediante la integración de áreas como la robótica y la salud. Dentro de la robótica social se tiene que las interfaces de interacción humano – robot están desarrolladas para una comunicación tradicional (voz, imágenes y botones) excluyendo a las personas con algunas discapacidades [1]. La discapacidad es una barrera que denota limitaciones físicas y actitudinales, la convención sobre los derechos de las personas con discapacidad, buscan promover, proteger y asegurar el goce pleno y la igualdad [2]. La infraestructura utilizada en la interacción con la tecnología desampara parcialmente a esta población, concretamente a personas que no pueden valerse por sí mismas para interactuar directamente con la tecnología [3]. Para el año 2020 en Colombia, el 2,6 % de su población en su mayoría adulta, sufre de algún tipo de discapacidad, afectando así el desarrollo social [4]. Por otro lado, el boletín técnico de discapacidad, entre los meses de enero a marzo del año 2022, indica que la tasa de desempleo para esta población es del 16,9 % desde diciembre del año 2021. Este número ha ido en crecimiento, comparado con el desempleo general en Colombia, que se encuentra tres puntos porcentuales por debajo [5].

En el territorio colombiano se padece de una insuficiente infraestructura en el sistema de salud, ya que no se cuenta con la mejor organización administrativa, ni la infraestructura adecuada para prestar un servicio hospitalario, por ello la asistencia para las personas discapacitadas es muy limitada y costosa ya que se requiere personal de la salud altamente calificado para tratar con estas personas [6]. Por eso, para ayudar a suplir este déficit hospitalario y enfatizar desarrollos para la robótica social, se proponen técnicas de comunicación con robots para implementar implementaciones y dar apoyo y asistencia robótica en el sector de la salud [7].

A raíz de este tipo de problemáticas y pensando en un desarrollo investigativo para la aplicación de BCI, se plantea la siguiente pregunta de investigación: ¿Cómo desarrollar un sistema BCI para la recepción de ondas cerebrales con el cual poder controlar las funciones básicas de movimiento del robot *Pepper*?

## 1.2. Objetivos

Desarrollar un sistema BCI (*Brain Computer Interface*) basado en el dispositivo *MindWave Mobile 2* para el control de movimiento del robot *Pepper* a través de ROS, llevando a cabo técnicas de tratamiento de señales basadas en Machine Learning.

## 1.3. Objetivos Específicos

- Investigar el estado del arte sobre técnicas para la adquisición, procesamiento e interpretación de señales EEG (electroencefalograma), mediante la revisión de bases de datos científicas y la consulta de repositorios de códigos y *datasets*.
- Caracterizar la *MindWave Mobile 2* para recepción del EEG de modo que se implemente un algoritmo para la interpretación de los datos adquiridos del usuario.
- Implementar un software que permita el procesamiento de los datos haciendo uso de *Machine Learning* para dar movimiento al robot mediante la recepción y clasificación las de funcionalidades a través de ROS.
- Evaluar el funcionamiento de la interfaz BCI desarrollada mediante escenarios de prueba, y así observar el comportamiento de los movimientos del robot estableciendo métricas de desempeño.

## 1.4. Justificación

Las interacciones humano-máquina dentro de la robótica social cumplen un rol esencial que caracteriza su utilización; esto se convierte en un requerimiento fundamental para el apoyo por medios tecnológicos a personas en condición de discapacidad [8]. Lo anterior genera la necesidad de nuevas formas de comunicación humano - máquina, brindando nuevas posibilidades en el desarrollo y así ayudar a poblaciones vulneradas. Las interfaces BCI impactan en la creación de nuevas formas de comunicación; la tecnología BCI potencia posibilidades basadas en la interacción del cerebro con el mundo exterior [9].

Los sistemas como el reconocimiento de voz, el reconocimiento de imágenes y los nuevos desarrollos de BCI, brindan la oportunidad de potenciar la comunicación entre personas y dispositivos en la sociedad moderna.

Adicionalmente el proyecto presenta un interés particular por la captación de ondas cerebrales por medio del dispositivo *MindWave Mobile 2*. De igual forma, se tomó como base el proyecto de Andrés Reyes, “Herramienta para la generación de texto basado en una interfaz cerebro - computador” [10], el cual como su nombre lo indica es una interfaz basada también en el dispositivo *MindWave Mobile 2*, obteniendo resultados con una efectividad del 91,26 % en la tarea de escritura mediante un teclado virtual. Sin embargo, se busca un enfoque directo hacia la robótica, generando una manera de conexión inalámbrica para lograr la comunicación entre el humano y el robot, adoptándolo de funcionalidades que permitan evidenciar la interacción en diversos escenarios. Esto implica la utilización del entorno de trabajo ROS (*Robot Operating System*), el cual es un meta-sistema operativo *Open Source* de programación robótica, que permite realizar enlaces de comunicación con diferentes dispositivos [11] y de esta manera adoptar al robot de funcionalidades que permitan demostrar la comunicación.

La complejidad del desarrollo de la obtención de EEG (electroencefalograma) aplicado al movimiento de un robot, justifica el desarrollo de un proyecto de grado. En él, se ven reflejados diferentes campos disciplinarios aplicados dentro la electrónica, como lo son los conceptos básicos, el estudio y tratamiento de señales, la programación, la electrónica análoga, la conexión entre diferentes interfaces y la robótica móvil.

Finalmente, y dado que en el entorno local los robots no son accesibles para toda persona, con el proyecto de grado se busca brindar apoyo a futuros desarrollos donde esta tecnología esté al alcance de todos.

## 1.5. Impacto Social

El desarrollo del proyecto de grado busca tener un impacto positivo en el ámbito investigativo y social, es por ello que se apuntan a la integración de los ODS (Objetivos de desarrollo sostenible). En 2015 la asamblea general de las Naciones Unidas propuso estos objetivos para realizar un plan de mejora y asegurar un mejor futuro. [12].

Mediante el estudio de los ODS, el proyecto se encamina dentro de 2 de estos 17 objetivos:

- Salud y bienestar.
- Reducción de las desigualdades.

En el mundo, según la ONU [13], unos 3500 millones de personas en los próximos años por enfermedades tempranas y el envejecimiento natural humano, requerirán un recurso de asistencia tecnológica que le sirva de soporte para preservar su salud. El desarrollo de este proyecto de grado considera estas falencias que afectan a la población en condición de discapacidad, brindando igualdad y ayudando en su integración social. En todo el mundo según el Banco mundial [14], el 15% de la población sufre de algún tipo de discapacidad, siendo aproximadamente 1000 millones de habitantes con dicha condición. Conociendo esta información, se enfrenta una gran problemática social, generando una necesidad que conlleva a nuevos desarrollos e investigaciones para brindar ayuda humanitaria.

Dentro de la Universidad Santo Tomás existen lineamientos de proyección social, los cuales brindan una ruta para el desarrollo que integra la enseñanza, el aprendizaje con la investigación. Teniendo en cuenta la conceptualización, el proyecto de investigación apunta hacia el desarrollo comunitario y la apuesta por la preservación de la vida, según el marco de proyección social de la Universidad [15]: “El desarrollo comunitario es posible en la universidad gracias a la construcción y promoción de las comunidades a través de la realización de proyectos de investigación, apoyo científico y técnico que surgen de la confluencia de intereses académicos y las dinámicas políticas, sociales, culturales y económicas de las comunidades”. La descripción del proyecto se enfoca en la interacción del cerebro humano con un robot (*Pepper*) a través de una computadora, este desarrollo permite que personas que padezcan dificultades de movimiento puedan controlar un robot utilizando únicamente su cerebro, por tanto, se logra la captación del EEG mediante la diadema *MindWave Mobile 2*. El desarrollo investigativo puede tener varias aplicaciones, pero ahora el prototipo propuesto se diseña solo como asistente. Con este modelo de interfaz se logran otros tipos de desarrollo capaces de desempeñar tareas de alta



complejidad que requieren de otras tecnologías complementarias y servir con mayor eficacia para tareas de asistencia en actividades cotidianas.

El proyecto es realizado dentro del grupo de investigación GED de la facultad de ingeniería electrónica de la Universidad Santo Tomás, aportando en desarrollos que permitan apoyar e incentivar a los estudiantes a realizar proyectos de robótica enfocados en sistemas BCI.

## Capítulo 2

# Estado del Arte

En esta sección, se realiza la revisión de artículos y libros de investigación relacionados a las técnicas de las interfaces cerebro-computador BCI. Se abordan los avances, aplicaciones, desafíos y trabajos relacionados en este campo.

### 2.1. Técnicas de los Sistemas BCI

De acuerdo a [16], la decodificación de la actividad cerebral para la interacción con la tecnología se conoce como los sistemas BCI. En este artículo se describen las aplicaciones de esta tecnología y como estas dependen de factores relacionados a la percepción de cada usuario, la adaptación al manejo de estas interfaces y su psique. A su vez, en el artículo [17], se realizó una descripción de las características, la evolución y las aplicaciones más importante de estos sistemas. El trabajo resalta la importancia del sistema internacional 10-20 [18] en la adquisición de ondas cerebrales. Este sistema es un método ampliamente utilizado para la ubicación de los electrodos en la toma del EEG. En este artículo se concluyó la importancia de integrar las diferentes ciencias como la medicina y la electrónica para estos desarrollos, también se concluyó como la toma de ondas por medio de métodos no invasivos, depende en gran medida de la calidad de los electrodos y su ubicación de acuerdo al sistema internacional 10-20.

La investigación de la neurociencia, está dirigida a comprender el funcionamiento del cerebro humano como se menciona en el libro [19]. En el trabajo realizado en [20], se presenta la descripción de los tipos de electrodos utilizados para la recepción del EEG como los electrodos basales, los electrodos superficiales y los electrodos quirúrgicos. A su vez, se propuso una fase de procesamiento basada en la adquisición de ondas, segmentación de la señal, extracción de ondas y

la finalmente la clasificación. Los métodos propuestos de clasificación fueron la transformada de Fourier, la expansión de Gabor y la transformada de wavelets.

A pesar de los avances significativos en las interfaces Cerebro-Computadora, aún existen desafíos prácticos y técnicos que deben abordarse [9]. La aplicación práctica de la encefalografía basada en interfaces cerebro-computadora enfrenta dificultades en la precisión y confiabilidad de los resultados. Además, el uso de señales EEG presenta ciertas limitaciones, como la presencia de ruido durante la recepción y la necesidad de reducir dicho ruido para mejorar la calidad de las señales [21] [22].

En el artículo [23], se realizó un análisis de las señales EEG mediante un programa llamado *CHECAsEN*, el cual permite eliminar segmentos de ruido de una señal EEG. La eliminación del ruido se realizó a través de la inspección visual, lo que permitió al usuario final tener una mejor calidad de la señal. A su vez, en el trabajo [24], se realizó un análisis de señales EEG enfocado en la reducción de ruido mediante la transformada wavelet, eliminando artefactos visibles en la señal. La transformada de wavelet es comúnmente utilizada en la clasificación de ondas, como se propone en el artículo [25]. En este estudio, se llevó a cabo la adquisición de ondas EEG mediante el dispositivo *MindWave Mobile 2*, se propuso el método de la transformada Wavelet para analizar dichas ondas. Como resultado, se concluyó que se disponen de las características necesarias para realizar una clasificación de las ondas adquiridas por la diadema.

En el trabajo realizado en [26], se presenta la clasificación de las ondas EEG durante las fases de sueño. La clasificación se propuso mediante un árbol de decisiones y el método de los vecinos más cercanos (*K-Means*). El experimento se realizó con 4 sujetos sanos donde se tuvo puntaje de clasificación de 51,41 %. En contraste, en [27] se realizó la clasificación de EEG durante el habla imaginada mediante el dispositivo *EMOTIV*. En este artículo, se realizó la clasificación de ondas mediante el programa *Weka*, el cual es un programa de aprendizaje automático. Se obtuvo un porcentaje de exactitud promedio de 70,94 %, se propuso en trabajos futuros la implementación de métodos más avanzados que permitan una mejor clasificación.

Enfocado en las interfaces BCI, el artículo [28], propuso el control de un brazo robótico mediante ondas EEG. Para la clasificación, se implementaron algoritmos de aprendizaje automático basados en un clasificador bayesiano lineal y cuadrático. En el artículo [29], los autores presentan una encuesta sobre técnicas de *Machine Learning* para sistemas BCI basadas en la toma del EEG.

En el artículo [30], se realizó una revisión sobre los sistemas BCI, los resultados de esta revisión demostraron que las arquitecturas más utilizadas en estos sistemas son redes neuronales convolucionales y las redes neuronales recurrentes en un 47 % de los casos, se propone a su vez la

combinación de las redes neuronales recurrentes y una red de creencia profunda. Dentro del campo de estudio de la inteligencia artificial, en [31] se describen diferentes sub ramas como el aprendizaje automático (*Machine learning*), el aprendizaje profundo (*Deep learning*), base de datos de conocimiento (*Data Base Knowledge*) y la minería de datos (*Data Mining*). Por otra parte, en el artículo [32], se planteó un tipo de red neuronal recurrente llamada memoria a corto plazo LSTM (*Long Short-Term Memory*). Este tipo de red es ampliamente utilizada en aplicaciones como *Google Translate*, *Siri*, *Alexa*, entre otras. Los resultados del artículo concluyen que la red LSTM es eficaz en la clasificación de series de tiempo.

## 2.2. Trabajos Relacionados

En la Universidad Santo Tomás en el artículo [10], se desarrolló un sistema de obtención de EEG mediante el dispositivo *Mindwave Mobile 2*. En este trabajo se realizó el desarrollo de una herramienta la cual ayuda a las personas a comunicarse mediante los parpadeos voluntarios. Esto se logró obteniendo el análisis del EEG proporcionado por la *MindWave Mobile 2* para así detectar los parpadeos voluntarios. Para el procesamiento de los datos se implementó una red neuronal recurrente RNN de tipo LSTM, la red neuronal implementada clasifica los tipos de parpadeos para así dar una instrucción. Se implementó la red neuronal dentro de una aplicación móvil Android, esta tiene un teclado virtual de 27 letras, los comandos de borrar y enter. Esta información se presenta de manera visual en la aplicación. Finalmente, para el funcionamiento se hizo un experimento con una población de 8 personas, este arrojó un 91,26 % de efectividad en la lectura de las letras.

La *EMOTIV EPOC+* es una diadema de recepción de ondas cerebrales, la cual cuenta con 14 canales de EEG. En el *dataset* [33], se llevó a cabo una medición continua de EEG con la diadema *EMOTIV*, con una duración de 117 segundos. Durante la medición se detectaron los estados de los ojos abiertos o cerrados. Además, se registró el estado de los ojos mediante una cámara mientras se realizaba la recepción; posteriormente se agregaron los datos al *dataset* del EEG en orden cronológico. El propósito de este *dataset* es clasificar el estado de los ojos, donde 0 indica ojos abiertos y 1 ojos cerrados.

En la fuente de referencia [34], se describe el funcionamiento de un robot de Arduino utilizando la diadema *MindWave Mobile 2*. En este repositorio se encuentran los scripts de recopilación de datos y los códigos de procesamiento de las ondas adquiridas. El funcionamiento del sistema se basa en la captura en tiempo real de los niveles de concentración y los parpadeos detectados.

Posteriormente, estos datos son interpretados y procesados mediante lenguaje de programación c#. Además, dentro del repositorio a su vez se encuentra el demo del funcionamiento del sistema.

En el artículo [35], se lleva a cabo la recepción de ondas cerebrales utilizando la diadema *Mind-Wave Mobile 2* para controlar a una silla de ruedas eléctrica. La recepción de ondas de la diadema se realizó con la targeta Arduino Mega 2560 R3 mediante el bluetooth HC-50. Mediante la programación de esta targeta, se adquieren las señales de los parpadeos y el nivel de atención en variables, las cuales posteriormente ejecutan comandos de movimiento y encendido o apagado de los motores. Los resultados obtenidos demostraron que la interfaz empleada funciona en algunas personas, ya que su eficacia depende del nivel de concentración de cada individuo.

En el repositorio [36], se encuentra una lista de diferentes *dataset* de muestras de EEG empleados en diferentes escenarios y aplicaciones. En este se encuentra secciones como el reconocimiento de emociones, música y EEG, parpadeos y movimientos oculares, EEG clínicos, etc.

Dentro de este repositorio se encuentra el artículo [37], en el cual se lleva a cabo la adquisición de señales EEG a partir de 14 electrodos. En este artículo se toman muestras de 20 personas con el fin de clasificar las señales de parpadeos voluntarios e involuntarios, con el propósito de evaluar los efectos de los parpadeos en componentes independientes del EEG. El *dataset* empleado en el artículo, consta de un conjunto de datos tomados de 14 electrodos distribuidos en diversas regiones cerebrales, se llevaron a cabo 3 sesiones con 20 ensayos cada una. Estos datos corresponden a muestras de 1 segundos que incluyen parpadeos voluntarios e involuntarios. Los resultados del artículo demostraron que los efectos distorsión de los parpadeos en el EEG se clasifican en las bandas de frecuencia delta y theta.

## 2.3. Adquisición, Interpretación y Procesamiento

En la tabla 1, se presentan las técnicas de adquisición, procesamiento e interpretación, basados en la investigación del estado del arte.

| Técnicas       | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adquisición    | <ul style="list-style-type: none"> <li>- Electrodo de EEG: Uso de electrodos colocados en el cuero cabelludo para capturar señales eléctricas cerebrales.</li> <li>- Dispositivos BCI comerciales: Utilización de dispositivos como <i>MindWave Mobile 2</i> o <i>EMOTIV</i> para adquirir señales EEG.</li> </ul>                                                                                                                                                                                                                                                                                                                                                        |
| Procesamiento  | <ul style="list-style-type: none"> <li>- Pre-procesamiento de señales: Filtrado, eliminación de ruido para mejorar la calidad.</li> <li>- Extracción de características: Identificación y extracción de patrones relevantes de las señales EEG.</li> <li>- Transformada de Fourier: Análisis en el dominio de frecuencia para representar la información en el espectro.</li> <li>- Transformada Wavelet: Descomposición multiresolución para detectar características en diferentes escalas.</li> <li>- Métodos de aprendizaje automático: Utilización de algoritmos como <i>SVM</i>, <i>K-NN</i>, o redes neuronales de tipo <i>LSTM</i> para clasificación.</li> </ul> |
| Interpretación | <ul style="list-style-type: none"> <li>- Reconocimiento de patrones: Clasificación de actividades cerebral a partir de patrones EEG.</li> <li>- Control de dispositivos: Decodificación de comandos mentales para controlar prótesis, sillas de ruedas o interfaces.</li> <li>- Interfaz cerebro-computador: Traducción de las intenciones del usuario en comandos para interactuar con sistemas tecnológicos externos.</li> </ul>                                                                                                                                                                                                                                        |

TABLA 1: Técnicas de adquisición, procesamiento e interpretación.

## Capítulo 3

# Marco Teórico

### 3.1. Electroencefalograma EEG

El electroencefalograma es una técnica con la cual se realiza el registro de señales eléctricas según la actividad cerebral por medio de electrodos dispuestos en el cuero cabelludo. Se transmiten como unas series de pulsos cuya frecuencia depende directamente de la amplitud [38]. La captación de la amplitud del EEG en un humano normal es de 10 a 100mv. Así también se tienen diferentes tipos de ondas medibles [19]:

- Ondas Delta: Este tipo de ondas van desde los 0,5 a 4 Hz de frecuencia, se registran durante el sueño profundo.
- Ondas Theta: Son ondas que están alrededor de los 4 a 7 Hz de frecuencia, presentes normalmente en humanos al realizar actividades de meditación y relajación profunda similar a un estado de sueño.
- Ondas Alfa: Tienen una frecuencia de entre 7 a 12 Hz, están presentes en estados de tranquilidad y relajación.
- Ondas Beta: entre 13 y 39 Hz en una frecuencia, están en una persona totalmente despierta y con actividad cerebral.
- Ondas Gamma: Estas ondas están por encima de los 40 Hz, se producen por estados de agresividad, pánico, miedos, preocupación, etc.

## 3.2. Interfaz BCI

Las interfaces BCI son una tecnología la cual establece canales de comunicación entre un humano y diferentes dispositivos mediante la captación de ondas cerebrales. Para la recepción de estas ondas cerebrales EEG, se realiza por medio de dispositivos que permiten adquirirlas ya sea de manera invasiva o no invasiva.

Existen diferentes opciones en el mercado, uno de estos dispositivos desarrollados para captación de EEG es la diadema *Mindwave Mobile 2*.

### 3.2.1. Diadema Recepción EEG

La *Mindwave Mobile 2* desarrollada por la empresa *Neurosky*, es una diadema utilizada para la obtención de EEG no invasivo, esta diadema cuenta con un biosensor en la parte frontal y un terminal en el lóbulo de la oreja y se alimenta por una batería de tipo AAA. La conexión de la diadema es inalámbrica mediante bluetooth v4. 0, se empareja con un software proporcionado por la empresa desarrolladora, que permite comunicarse diferentes dispositivos móviles y computadoras [39].

La diadema tiene un sensor dispuesto en la parte frontal, el cual realiza la adquisición del electroencefalograma del lóbulo frontal del cerebro; por ello, es importante conocer qué tipo de medición se realizará y el uso que se le dará. Esta diadema viene con un software que permite observar diferentes funcionalidades de la diadema, también dentro de su tienda existen diferentes aplicaciones pagas de entretenimiento para los entusiastas [39]. A continuación se presenta la caracterización de la diadema *Mindwave Mobile 2*.

En la tabla 2, se detallan las especificaciones técnicas del dispositivo, proporcionando información sobre las características de la diadema. Además, se detalla las medidas físicas y la estructura de la diadema. Esto permite conocer las capacidades y el diseño del dispositivo.



| Especificaciones técnicas                          | Valor                                                                                                    |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Sensor                                             | NeuroSky TGMA1                                                                                           |
| Alimentación                                       | Batería AAA                                                                                              |
| Duración de la batería                             | 8 horas                                                                                                  |
| Módulo Bluetooth                                   | BT/BLE dual mode module                                                                                  |
| Alcance                                            | 10 metros                                                                                                |
| ID estático de auricular                           | Identificación única de emparejamiento                                                                   |
| Compatibilidad con sistemas operativos             | Windows (XP/7/8/10), Mac (OSX 10.8 o anterior), iOS (iOS 8 o anterior), Android (Android 2.3 o anterior) |
| Medición                                           | Valor                                                                                                    |
| Emisión de ondas cerebrales sin procesar (12 bits) | 3 - 100 Hz con frecuencia de muestreo a 512 Hz                                                           |
| Emisión de espectros de potencia EEG               | Alfa, Beta, etc.                                                                                         |
| Salida del medidor eSense de NeuroSky              | Atención, Meditación y otros medidores futuros                                                           |
| Análisis de calidad de señal EEG/ECG               | Detecta un contacto deficiente y si el dispositivo está fuera de la cabeza                               |
| Estructura                                         | Valor                                                                                                    |
| Peso                                               | 90 gramos                                                                                                |
| Brazo del sensor hacia arriba                      | Alto: 225 mm, Ancho: 155 mm, Profundidad: 92 mm                                                          |
| Brazo del sensor hacia abajo                       | Alto: 225 mm, Ancho: 155 mm, Profundidad: 165 mm                                                         |

TABLA 2: Características de la diadema *Mindwave Mobile 2*

El módulo TGMA1, cuenta con el integrado ThinkGear. Este integrado realiza la toma del electrodo seco y calcula la señal cruda y los medidores eSense. Los datos de la medición de eSense corresponde al nivel concentración y meditación, estas variables toman valores de 1 a 100. Los datos crudos corresponden a la medición llamada RAW que son los datos de la señal sin procesar [35].

### 3.3. Deep Learning

El *deep learning* o en español aprendizaje y maquina es una técnica revolucionaria en el campo de la inteligencia artificial. Consiste en el aprendizaje autónomo mediante redes neuronales con múltiples capas, permitiendo el aprendizaje independiente mediante la experiencia. El deep learning permite interpretar y analizar patrones de datos complejos con una alta precisión [40].

Las principales ventajas del deep learning es su capacidad de comprensión de información compleja como imágenes, texto, voz, etc. El éxito de esta tecnología se debe a la gran cantidad

de información que se tiene actualmente, así también los recursos de cómputo que permiten el desarrollo de estas tecnologías.

### 3.3.1. Redes Neuronales

Las redes neuronales son un modelo matemático computacional basado en la emulación de características propias de la inteligencia humana para procesar la información. Su estructura es similar una neurona biológica, al igual que su funcionamiento que implica un aprendizaje adaptativo basado en el entrenamiento o en la experiencia. Estas redes son autoorganizadas ya que tiene la capacidad de organizar la información por sí mismas mediante etapas de aprendizaje.

Además, las redes neuronales tienen tolerancia a fallos ya que tienen la capacidad de retener información incluso si se sufre de daños en su estructura. También pueden reconocer patrones en presencia de ruido u distorsiones. Estas redes pueden realizar operaciones en tiempo real, lo que permite procesar datos de manera paralela y rápida. Además, se puede integrar fácilmente en diversas tecnologías. Las redes neuronales son un modelo matemático computacional basado en la emulación de características propias de la inteligencia humana para procesar la información. Su estructura es similar una neurona biológica, al igual que su funcionamiento que implica un aprendizaje adaptativo basado en el entrenamiento o en la experiencia. Estas redes son autoorganizadas ya que tiene la capacidad de organizar la información por sí mismas mediante etapas de aprendizaje [20].

Las redes neuronales tienen tolerancia a fallos ya que tienen la capacidad de retener información incluso si se sufre de daños en su estructura. También pueden reconocer patrones en presencia de ruido u distorsiones. Estas redes pueden realizar operaciones en tiempo real, lo que permite procesar datos de manera paralela y rápida. Además, se puede integrar fácilmente en diversas tecnologías.

Si bien existen diferentes tipos de redes neuronales, una red neuronal se compone de nodos o neuronas interconectadas, organizadas en varias capas que se encargan del procesamiento y la transmisión de datos. La capa de entrada es la etapa en la que se recibe la información del exterior. Las capas ocultas, por otro lado, no tienen contacto directo con el exterior y pueden o no estar interconectadas entre sí. Estas capas ocultas pueden distribuirse en diferentes tipologías. Finalmente, la capa de salida envía la información al exterior, proporcionando los resultados o las predicciones deseadas [41].

El flujo de información comienza con los datos de entrada en la capa de entrada, que luego se propagan a través de las capas ocultas hasta llegar a la capa de salida. Cada nodo en las capas ocultas realiza cálculos y transformaciones basados en las conexiones y los pesos asociados a cada una de ellas. Estos cálculos se realizan utilizando funciones de activación y algoritmos de aprendizaje, que permiten a la red neuronal aprender y ajustar sus parámetros a lo largo del tiempo [42].

La figura 1 muestra una representación gráfica que ilustra la composición básica de una red neuronal y sus componentes fundamentales, como los nodos, las conexiones y las capas. Esta figura brinda una descripción visual de cómo se estructura una red neuronal típica.

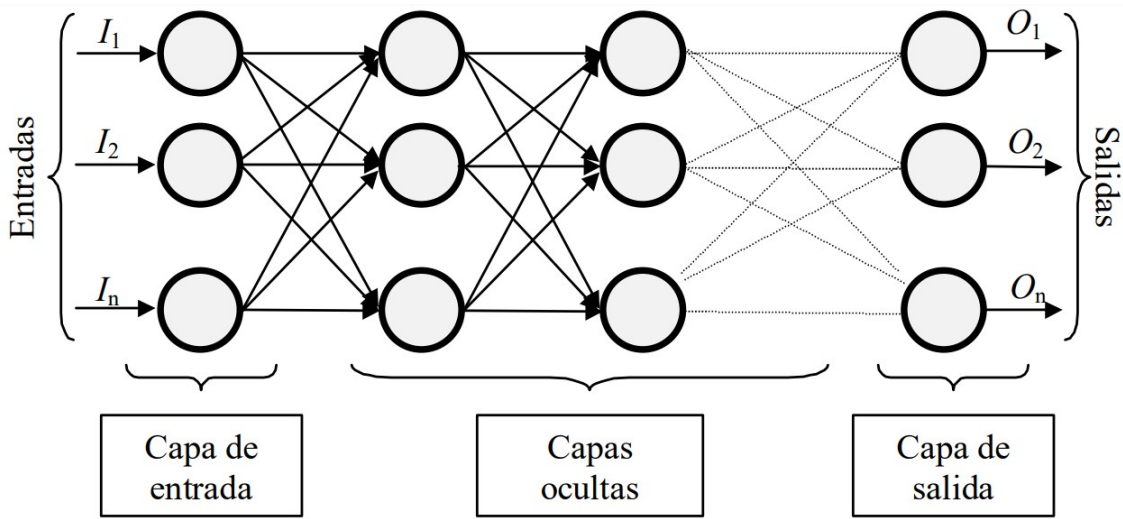


FIGURA 1: Modelo red neuronal [41].

El primero modelo matemático computacional de una red neuronal fue desarrollado en 1943 por McCulloch y Pitts, en el cual se describe en la ecuación 3.1, donde  $H$  corresponde a la función de activación con un umbral denotado por la letra  $u$ , la señal de entrada corresponde a  $x_j$  y sus pesos  $w_j$  donde  $j$  corresponde al número de entradas [43].

$$y = H\left(\sum_{n=1}^j w_j * x_j - u\right). \quad (3.1)$$

En la figura 2, se puede observar el modelo propuesto conocido como red neuronal perceptrón, el cual es un modelo básico de red neuronal. Este modelo se basa en único nodo o neurona que realiza la operación dentro de una arquitectura de una sola capa definida como capa perceptrón.

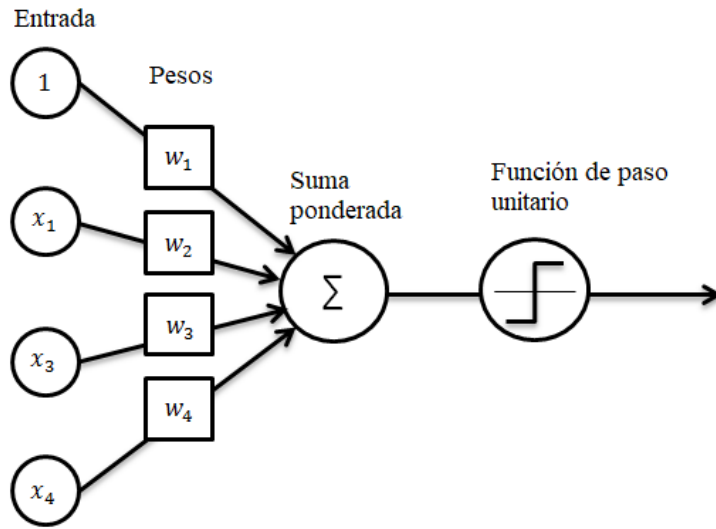


FIGURA 2: Modelo red neuronal perceptrón (adaptado de [21]).

### 3.3.2. Redes Neuronales Recurrentes RNN

Según [22], las redes neuronales recurrentes son un tipo de red neuronal que se caracteriza por su capacidad de tratamiento de datos secuenciales, son conocidas por tener una retroalimentación que permite el análisis de secuencias de tiempo. A diferencia de las redes neuronales convolucionales, las redes recurrentes reconocen relaciones y dependencias en los datos de entrada, permitiendo así, realizar temporalmente un análisis de información.

Este tipo de red neuronal puede realizar un entrenamiento autónomo por sí solas. Cada neurona corresponde a dos parámetros los cuales son el parámetro de *output* o en español salida y un parámetro de información el cual es adquirido por la siguiente neurona y así se realiza la recurrencia [44], como se observa en la figura 3.

El modelo de una red neuronal recurrente esta dado por la ecuación (3.2).

$$h^t = f(h^{t-1}, xt; \theta) \quad (3.2)$$

Donde  $h^t$  corresponde al estado actual de la neurona,  $h^{t-1}$  corresponde al estado anterior,  $xt$  es la entrada actual y  $\theta$  corresponde a los pesos de la neurona.

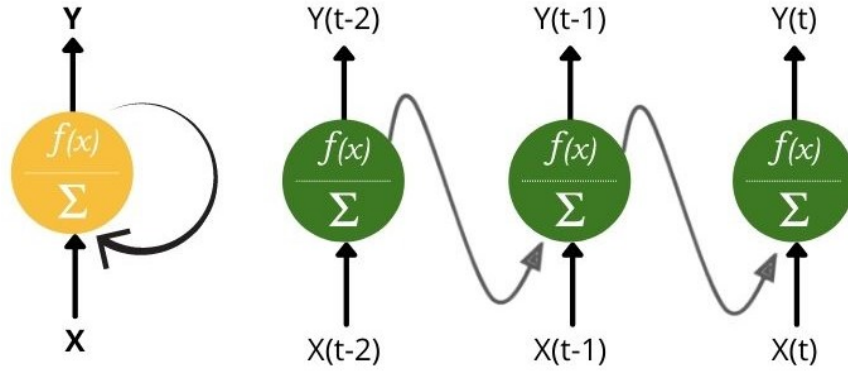


FIGURA 3: Red Neuronal Recurrente RNN [45].

### 3.3.3. Redes LSTM (Long-Short Term Memory)

Las redes neuronales LSTM son una red neuronal recurrente usada en el procesamiento de secuencias y datos en series temporales. Tienen la particularidad de solventar el problema que tienen las redes neuronales recurrentes de desaparición o degradación del gradiente, este ocurre cuando la propagación del error se vuelve más pequeño a través de cada capa. El principio de esta red neuronal es una celda de memoria que mantiene un estado diferente del flujo de la RNN [46]. Estas redes neuronales contienen 3 tipos de compuertas:

La compuerta de olvido que decide si desecha o mantiene la información éntrate descrita en la ecuación (3.3).

$$f_t = \sigma(W[xt, h_t, C_{t-1}] + b_f) \quad (3.3)$$

Donde  $W$  corresponde a los vectores de peso para cada entrada,  $\sigma$  es la función sigmoidea,  $h_{t-1}$  es la salida del bloque anterior,  $xt$  es la secuencia de entrada,  $ct-1$  corresponde a la memoria del bloque anterior y  $b_f$  es el vector de polarización.

La compuerta de entrada la cual decide acerca de la actualización de la información se describe como la ecuación (3.4) y (3.5).

$$i_t = \sigma(W[xt, h_t, C_{t-1}] + b_i) \quad (3.4)$$

$$C_t = \tanh(W_c[xt, h_{t-1}, ] + b_c) \quad (3.5)$$

la compuerta de salida la cual devuelve la respuesta final de las anteriores compuertas se define en la ecuación (3.6).

$$h_t = \tanh(c_t)o_t \quad (3.6)$$

### 3.3.4. Características de Red Neuronal LSTM

Según [47], las redes neuronales se componen de pesos y puertas. Estos parámetros son fundamentales en el funcionamiento de la red neuronal, ya que define cómo los datos son transmitidos y procesados a lo largo de las capas definidas.

Estos parámetros se definen como:

- Pesos de entrada que ponderan la entrada para el tiempo actual.
- Peso de salida, utilizado para ponderar el último paso de tiempo.
- Estado interno, calcula la salida para el paso del tiempo.
- Puerta de olvido, utilizada para descartar información no relevante.
- Puerta de entrada, actualiza el estado de la memoria decidiendo su valor de entrada.
- Puerta de salida, según la puerta de entrada y la memoria de la celda, evalúa cual es la salida.

Para especificar la forma de entrada de los datos para el entrenamiento, se define el parámetro *input\_shape*. Este parámetro indica la dimensión de los datos de entrada, lo cual es crucial para establecer el tamaño adecuado de la primera capa de la red neuronal. La forma de los datos de entrada dependerá del tipo de entrenamiento que se realice.

La clasificación se define dependiendo la función de pérdida, esta permite medir que tan bien esta implementada la red neuronal, esta función de pérdida depende del tipo de clasificación que se requiera. Estas funciones se definen comúnmente como la regresión lineal, clasificación binaria y clasificación multiclase. A su vez es necesario un algoritmo de optimización. Los algoritmos comúnmente más utilizados son el descenso del gradiente, Adam (*Adaptive Moment Estimation*) y RMSprop (*Root Mean Square Propagation*).

### 3.4. ROS

ROS (*Robot Operating System*), es un meta sistema operativo de código abierto desarrollado para software de robótica. Proporciona librerías y servicios que facilitan la programación en lenguajes como C++ y Python. Uno de los puntos importantes de ROS es la comunidad de desarrollo de sistemas para robots que tiene, los desarrollos van desde aficionados y profesionales [48].

ROS se basa en una arquitectura distribuida a base de componentes llamados Nodos. Estos nodos tienen la posibilidad de ejecutarse en diferentes entornos mediante mensajes entre sí. ROS cuenta herramientas que facilitan el control y la ejecución de los nodos [49]. Estas herramientas son:

- roscore: Inicializa el entorno de trabajo, se ejecuta como el nodo maestro.
- rosrund: Ejecuta los nodos especificados dentro de un paquete ROS.
- rostopic: Permite tener control sobre la ejecución de los nodos.
- rosparm: Se utiliza para acceder y manipular parámetros.

Cuando se inicializa el entorno de trabajo, las funciones están dirigidas mediante *ROS Master*. Este es un componente central de ROS que permite la comunicación y lleva un registro de mensajes y los servicios de ejecución como una base de datos. En la figura 4, se observa como compone la ejecución de los nodos.

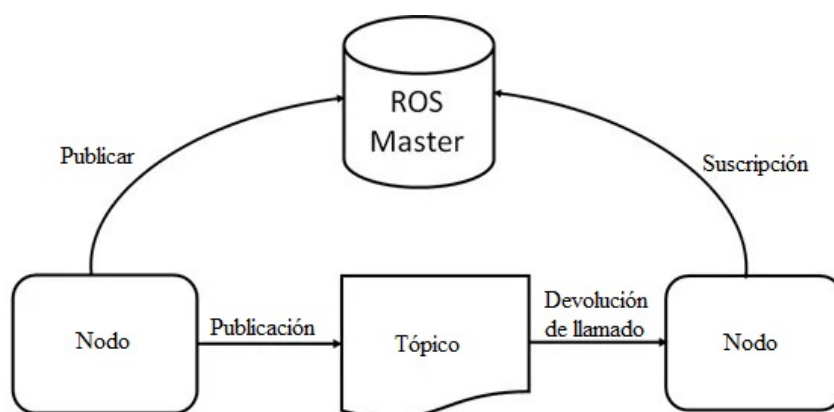


FIGURA 4: Diagrama ejecución ROS (adaptado de [50]).

## Capítulo 4

# Diseño Metodológico



FIGURA 5: Metodología implementada.



## 4.1. Investigación

Inicialmente se realiza la revisión bibliográfica tomando como base los repositorios de la Universidad y artículos investigativos dentro de las diferentes bases de datos científicas brindadas por el CRAI como *Scopus*, *IEEE Xplore* y *Science Direct* y búsquedas mediante sitios investigativos en internet. Se estudiarán artículos relacionados con enfoques en temas de sistemas BCI e interfaces de adquisición de EEG para diferentes aplicaciones, tomando técnicas que ayuden al desarrollo del proyecto de grado.

Se investigará sobre diferentes redes neuronales, procesos y técnicas de entrenamiento, y plataformas tecnológicas empleadas en los trabajos previos; se logrará validar lo anterior identificando y poniendo en marcha códigos disponibles en repositorios abiertos de investigación. Los resultados obtenidos en esta fase serán condensados en el estado del arte del proyecto.

## 4.2. Ejecución

Con los resultados de la investigación realizada se inicia el diseño conceptual para realizar un proceso de asignación y dimensionamiento tecnológico. Para esto, se definirán los requisitos tecnológicos de la solución a implementar con base en las características funcionales y estructurales encontradas en el estado del arte. En concurrencia con lo anterior se realiza el estudio de los diversos repositorios identificados en la fase previa definiendo métricas para determinar la eficacia y eficiencia de cada implementación. Se determinará en esta fase que implementaciones pueden ser reutilizadas y que algoritmos deben ser reprogramados.

## 4.3. Construcción

Se realiza la caracterización e implementación de la *MindWave Mobile 2* para la recepción de EEG, posteriormente se implementa un algoritmo para la interpretación de los datos arrojados por el dispositivo de recepción, los cuales son procesados y seccionados para ser interpretado mediante nodos y servicios ROS dando movilidad a los brazos del robot. En caso de no obtener buenos resultados en la construcción, se realiza la retroalimentación mediante procesos de ajuste y sintonización además de investigar nuevas fuentes que sirvan de apoyo al desarrollo del proyecto. En esta fase se realizan los procesos de entrenamiento necesarios que serán la base para la operación del sistema de interpretación de movimientos. Este entrenamiento requerirá

de procesos generales necesarios para la correcta identificación automática de las características EEG que serán relacionadas con los nodos en ROS.

#### 4.4. Prueba y Puesta en Marcha del Proyecto

Una vez realizada la construcción se procede a evaluar el movimiento del robot con diferentes señales, verificando el funcionamiento y cuantificando la información para realizar un contraste de los datos y validar la efectividad de la interfaz BCI. La evaluación propuesta será conducida a través de la medición de indicadores estándar como la matriz de confusión y de otros indicadores específicos que se definan durante el desarrollo conceptual del proyecto.

#### 4.5. Cronograma

En la tabla 3 se muestra el cronograma de actividades a realizar durante el proyecto de grado. Para su construcción, se analizó los temas abarcados durante el desarrollo, así también el tiempo empleado en cada actividad propuesta y lograr la finalización exitosa del proyecto.

| ACTIVIDAD                                                                                | MESES   |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
|------------------------------------------------------------------------------------------|---------|---|---|-------|---|---|-------|---|---|------|----|----|-------|----|----|----|----|----|
|                                                                                          | FEBRERO |   |   | MARZO |   |   | ABRIL |   |   | MAYO |    |    | JUNIO |    |    |    |    |    |
|                                                                                          | SEMANAS |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
|                                                                                          | 1       | 2 | 3 | 4     | 5 | 6 | 7     | 8 | 9 | 10   | 11 | 12 | 13    | 14 | 15 | 16 | 17 | 18 |
| Revisión de datasets.                                                                    |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Caracterizar la Diadema.                                                                 |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Implementar un algoritmo que permita la toma de datos.                                   |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Realizar la implementación en ROS para el movimiento del robot.                          |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Implementar una Red Neuronal para procesamiento y validación.                            |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Realizar la conexión de ROS con la red neuronal.                                         |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Retroalimentar la interfaz mediante procesos de nuevas fuentes, ajuste y sincronización. |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Evaluar el funcionamiento de la interfaz BCI.                                            |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Realizar la matriz de confusión.                                                         |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |
| Elaboración documento final                                                              |         |   |   |       |   |   |       |   |   |      |    |    |       |    |    |    |    |    |

TABLA 3: Cronograma de Actividades.

## Capítulo 5

# Desarrollo Conceptual

El objetivo principal del proyecto de grado es realizar un sistema BCI enfocado en la robótica social, realizando la adquisición de ondas EEG mediante la diadema *MindWave Mobile 2*, clasificándolas en funcionalidades de movimiento del robot *Pepper*. El desarrollo conceptual se basa en la metodología implementada.

### 5.1. Conexión de la Diadema

La diadema *MindWave Mobile 2* es un dispositivo de recepción de ondas cerebrales EEG de emparejamiento inalámbrico desarrollado por la empresa *Neurosky*, según el fabricante, las características principales de esta diadema es la medición del nivel de concentración, el nivel de meditación y los parpadeos voluntarios. La recepción de las ondas se realiza mediante un biosensor *ThinkGear TGAM* que está dispuesto para colocarse en la frente del usuario, a su vez cuenta con una tierra la cual es la referencia neutral de voltaje en relación con el sensor. La referencia es un gancho que se coloca en el lóbulo de la oreja. En la figura 6, se observa cómo es el posicionamiento correcto de la diadema.

Teniendo en cuenta estas características, se pretende realizar la conexión inalámbrica de la diadema mediante bluetooth con un computador con sistema operativo *Linux*. Aunque el instructivo de la diadema nos dice que no ofrece compatibilidad con este sistema operativo, la conexión se realizara de forma manual. Por tal motivo, es necesario utilizar una aplicación de gestión de dispositivos bluetooth como lo es *Blueteman*. Esta aplicación permite administrar y controlar diferentes dispositivos bluetooth. Para poder emparejar este dispositivo es necesario

realizar una conexión manual mediante el protocolo RfComm (*Radio Frequency Communication*), teniendo en cuenta la dirección MAC del dispositivo y asignándolo a un canal RfComm [51].

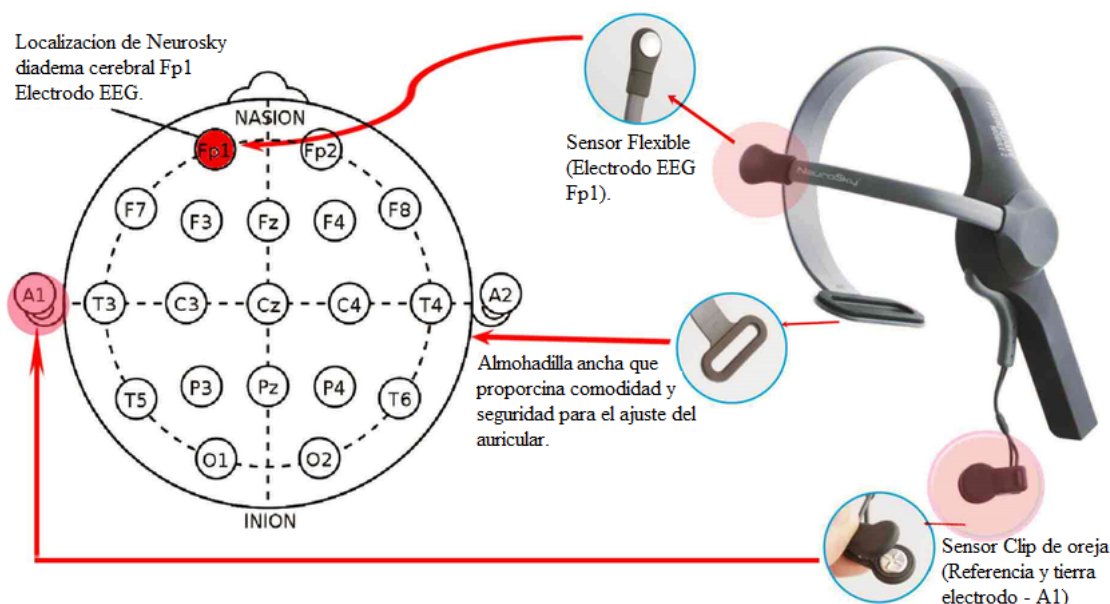


FIGURA 6: Diagrama de posicionamiento MindWave Mobile 2 (adaptado de [52]).

## 5.2. Programación de Recepción

Para realizar la programación e iniciar la recepción y conexión, se hace uso de la librería de código abierto *NeuroPy*. Esta librería está disponible en *GitHub* [53], desarrollada en el lenguaje Python. Esta basada en el protocolo de comunicación publicado por *NeuroSky*.

Para realizar la conexión de la diadema, se instancia una clase de la librería *NeuroPy*, la cual realiza la conexión del dispositivo mediante el puerto serial bluetooth RfComm, así también la velocidad de transmisión que se establece según el fabricante de 57600 baudios.

Esta librería al no tener una amplia documentación, es necesario realizar el estudio detallado de la misma para conocer las propiedades de recepción que incorpora la librería. En la figura 7, se define el esquema de conexión de la diadema y el pseudocódigo en el anexo 8.1.

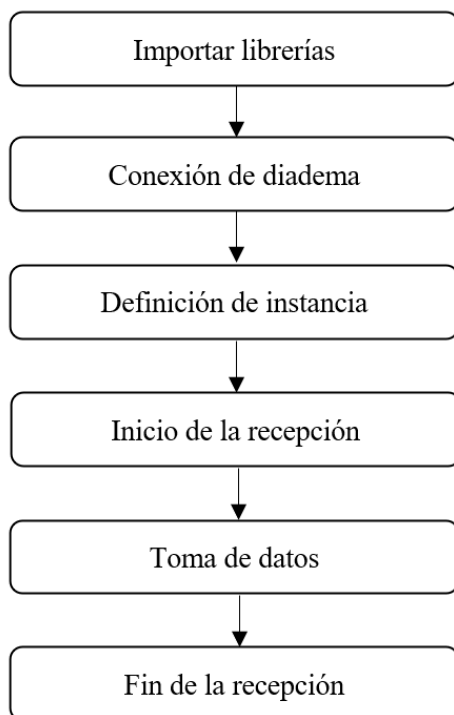


FIGURA 7: Esquema recepción de ondas.

### 5.3. Análisis de Datos

Una vez implementado el código, se procede a analizar la clase correspondiente a la señal proporcionada por la diadema. Se evalúan diferentes movimientos, capturando y revisando gráficamente el comportamiento de los resultados y sus características. Es importante, a su vez tener en cuenta las limitaciones de la diadema, ya que de esta manera se pueden obtener mejores resultados al clasificar los tipos de ondas.

Teniendo en cuenta lo anterior y realizando la recepción de la señal, se realiza el análisis de la señal donde se define la ventana de tiempo con la que se trabaja dependiendo de los datos adquiridos. Para realizar la clasificación es importante crear un *dataset* después de realizada la recepción el cual contenga la información de la ventana predefinida con los datos correspondientes. Es importante visualizar y estudiar el *dataset* creado ya que de este dependerá el resultado.

## 5.4. Implementación de Red Neuronal

Teniendo en cuenta que se requiere realizar un clasificador de series de tiempo, se opta por implementar una red neuronal de tipo LSTM. Este tipo de red neuronal tiene la particularidad de tener memoria a corto y largo plazo. Esta característica le permite retener información reciente del flujo de datos y a su vez almacena la información relevante a largo plazo, lo que resulta altamente efectivo en el procesamiento de secuencias y la captura de dependencias a largo plazo.

Para la implementación de la red neuronal, inicialmente se tiene que realizar la preparación del *dataset* antes del entrenamiento. Por ello, es recomendable realizar la normalización de los datos de entrada para mejorar el rendimiento y la eficiencia del modelo, aunque no en todos los casos. El entrenamiento se realiza bajo un aprendizaje supervisado, por lo que se debe tener en cuenta el dimensionamiento de los datos y la clasificación de los grupos, en donde la variable de entrada y la variable de salida correspondiente al clasificador, cabe recalcar que estas tienen que tener la misma dimensión para efectuar el entrenamiento. En la figura 8 se muestra el esquema de desarrollo para una red neuronal LSTM.

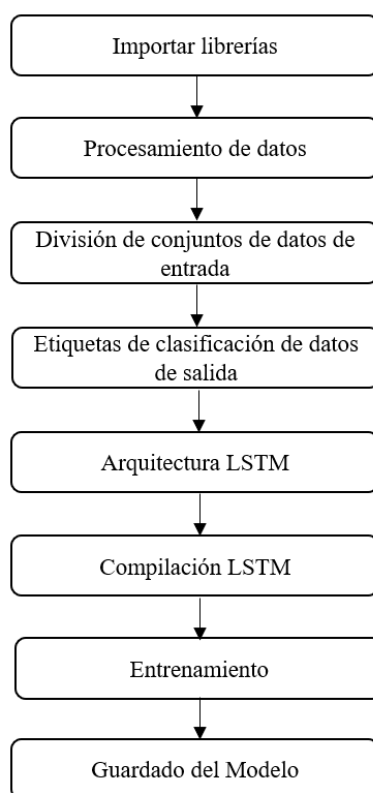


FIGURA 8: Esquema script modelo de entrenamiento.

## 5.5. Entrenamiento de Red Neuronal LSTM

Para realizar el entrenamiento de la red neuronal se utilizan las librerías de *Tensor Flow* y *Keras* en el lenguaje de programación Python. Para el entrenamiento se propone el siguiente esquema de red neuronal:

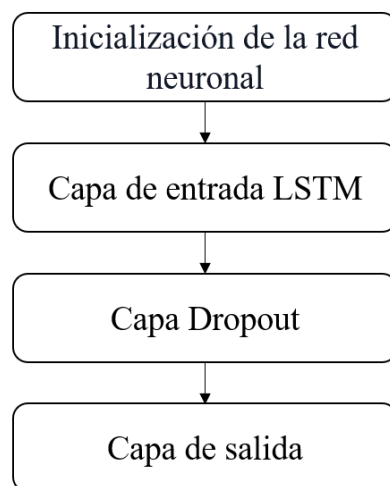


FIGURA 9: Esquema de Red Neuronal.

Como se observa en la figura 9, se describe el esquema de la estructura de entrenamiento donde se inicializan las librerías *Tensor Flow* y *keras* y se hace el llamado de la función *Sequential()*. Esta función se utiliza para agregar varias capas secuencialmente sin la necesidad de especificar el orden.

En la capa de entrada se define el tipo de red neuronal que se utilizará junto con las neuronas de entrada. Como se menciona en el marco teórico, es crucial definir el parámetro *input\_shape* correspondiente a la dimensión de los datos de entrada. Este parámetro garantiza que la configuración de la red neuronal es la correcta y de este modo se adquieren los datos de entrada para realizar el entrenamiento. A su vez, dentro de la universidad se han realizado proyectos donde se implementan redes neuronales como [10] y [54], donde se propone la implementación de una capa de *Dropout*. Esta capa, es una técnica efectiva para evitar el sobre entrenamiento de la red neuronal, desactivando un porcentaje de neuronas aleatoriamente.

Finalmente, en cuanto a la capa de salida, se definen las neuronas correspondientes al número de etiquetas o clasificaciones requeridas. A su vez se especifica la función de activación adecuada para el modelo desarrollado dependiendo de cómo se requiera la clasificación.

Para la red neuronal a implementar se proponen 4 modelos representados en las tablas 4, 5, 6 y 7:

| Modelo 1 |                       |
|----------|-----------------------|
| LSTM     | 64 neuronas           |
| Dropout  | 0.1                   |
| Dense    | (#Etiquetas) neuronas |

TABLA 4: Modelo 1 Red Neuronal.

| Modelo 2 |                       |
|----------|-----------------------|
| LSTM     | 128 neuronas          |
| Dropout  | 0.1                   |
| Dense    | (#Etiquetas) neuronas |

TABLA 5: Modelo 2 Red Neuronal.

| Modelo 3 |                       |
|----------|-----------------------|
| LSTM     | 128 neuronas          |
| Dropout  | 0.1                   |
| LSTM     | 64 neuronas           |
| Dropout  | 0.1                   |
| Dense    | (#Etiquetas) neuronas |

TABLA 6: Modelo 3 Red Neuronal.

| Modelo 4 |                       |
|----------|-----------------------|
| LSTM     | 128 neuronas          |
| Dropout  | 0.1                   |
| LSTM     | 64 neuronas           |
| Dropout  | 0.1                   |
| LSTM     | 32 neuronas           |
| Dropout  | 0.1                   |
| Dense    | (#Etiquetas) neuronas |

TABLA 7: Modelo 4 Red Neuronal.

## 5.6. Movimientos del Robot

En primer lugar, para la programación del robot *Pepper* es necesario conocer cada una de sus funcionalidades, la conexión del robot y las librerías de programación. Según la documentación de *Aldebaran*, la empresa responsable de la creación y fabricación del robot *Pepper*, ha desarrollado la tecnología y las capacidades con las que el robot posee, como el control de movimiento, la comunicación verbal, el reconocimiento de voz, etc. Para programar las funcionalidades del robot, es esencial utilizar la librería *NAO*.

Para establecer la conexión mediante el sistema operativo *Linux* y el robot, se utiliza el protocolo de comunicación SSH. Este protocolo permite establecer una conexión remota a través de la red. Para llevar a cabo esta conexión es necesario que ambos dispositivos estén conectados a la misma red WIFI y se ingresa mediante la dirección IP proporcionada por el robot. Una vez se realiza la conexión se puede acceder a la computadora interna de *Pepper*, esta computadora



interna es un sistema operativo basado en Linux llamado NAOqi, la programación de las funcionalidades del robot se realiza en un script de Python. En la tabla 10, se observa el esquema de programación propuesto.

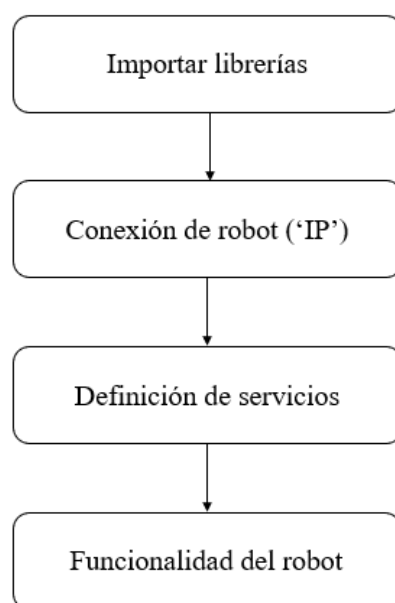


FIGURA 10: Esquema del script de conexión del robot.

Estas funcionalidades del robot se pueden dar de diferente manera, la librería trae preestablecidos diferentes movimientos los cuales son llamados mediante servicios, a su vez se puede realizar funciones de movimiento totalmente programadas manualmente, para ello el robot cuenta servicios de movilidad para cada parte de su cuerpo como se describen en la figura 11. Estas funciones se encuentran en la documentación, así como la descripción de los grados de libertad de cada actuador que contiene el robot.

Se plantea la programación de funcionalidades que faciliten la comunicación de personas en condición de discapacidad. Estas funcionalidades se proponen con el objetivo de brindar una alternativa de comunicación verbal, así como la capacidad de realizar movimientos predefinidos y programados. Para la comunicación verbal se hace una pequeña descripción del proyecto de grado, la función de movimiento predeterminada corresponde a la acción de saludar y para el movimiento programado, saludar dando la mano. A continuación, en la figura 12, se observa la clasificación correspondiente a cada funcionalidad propuesta para el robot.

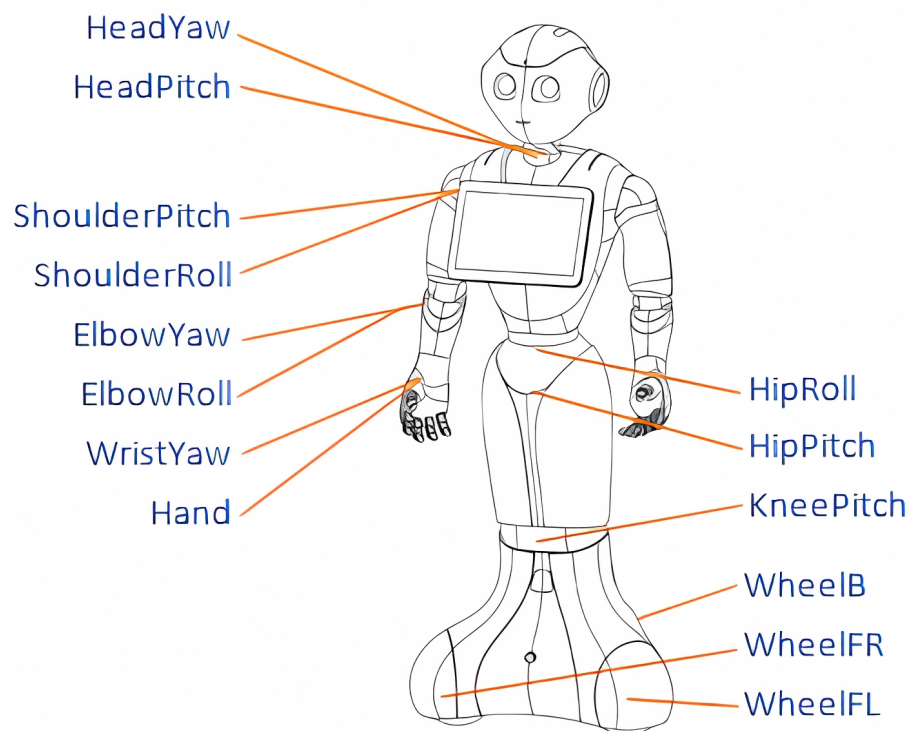


FIGURA 11: Actuadores de movimientos Pepper [55].

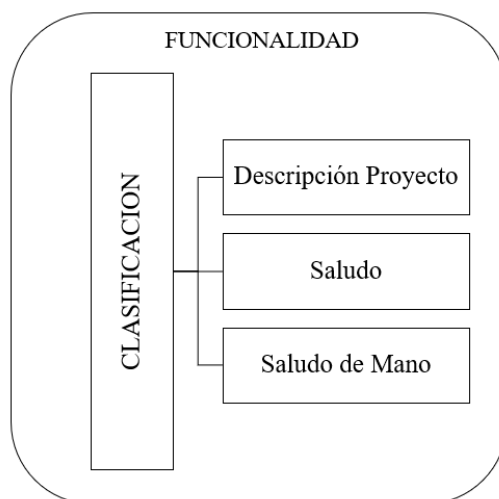


FIGURA 12: Funcionalidades propuestas de Pepper.

## 5.7. Entorno ROS

Como se describe en el marco teórico, ROS es un meta sistema operativo de código libre pensado para el desarrollo de software de robots. Se pretende realizar la implementación del proyecto en este entorno de trabajo. Para esto, se definen los cada uno de los nodos a implementar:

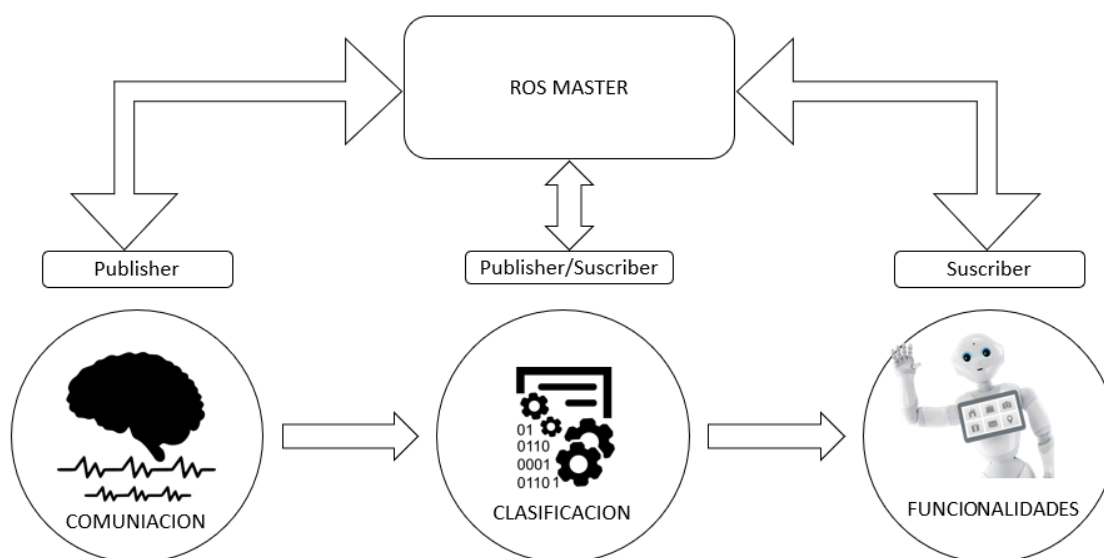


FIGURA 13: Implementación de nodos en ROS.

Como se observa en la figura 13, se tienen los nodos de comunicación, clasificación y funcionalidades. El primer nodo es de comunicación, en este nodo se realiza la conexión de la diadema y se hace recepción de las ondas, este nodo es un *Publisher* ya que realiza el envío de las ondas capadas mediante un mensaje.

El segundo nodo corresponde a la clasificación, este nodo adquiere los datos del nodo de comunicación por ende es un nodo *Suscriber*. Teniendo los datos, este nodo realiza la carga del modelo donde se realiza la clasificación de las ondas recibidas, la respuesta a esta clasificación se envía al próximo nodo por lo que a su vez es un nodo *Publisher*.

Finalmente, el tercer nodo de funcionalidades es un nodo *Suscriber* que recibe la clasificación del nodo de clasificación. En este nodo se realiza la conexión y la programación de las funcionalidades que adopta el robot Pepper y así observar los movimientos este.

## 5.8. Evaluación del Sistema BCI

Es importante realizar una evaluación del sistema para validar la funcionabilidad y conocer las limitaciones y posibles mejoras que pudiese llegar a tener. En el proceso evaluativo del sistema se pretende realizar tanto la matriz de confusión como el análisis del porcentaje de efectividad del sistema.

La matriz de confusión es utilizada para problemas de clasificación por lo que se plantea para evaluar el funcionamiento de la red neuronal y así comprobar la eficacia del modelo de clasificación de ondas. Esta matriz muestra la cantidad de veces de las predicciones correctas e incorrectas.

El análisis de porcentaje de efectividad se utiliza como método evaluativo de confiabilidad del sistema, se realiza mediante la toma de muestras simuladas junto con la clasificación correspondiente al funcionamiento esperado del robot. Esto también permite identificar las fuentes de error que pueden afectar el sistema.

## Capítulo 6

# Resultados y Discusión

En esta sección se presentan los resultados en relación al desarrollo conceptual, donde se describen los pasos para cumplir los objetivos propuestos. De esta manera se muestra los resultados y el funcionamiento del sistema BCI implementado en ROS basado en los escenarios de prueba. A su vez, se proporcionan métricas y estadísticas que permiten la validación de los resultados obtenidos durante el desarrollo.

### 6.1. Recepción y Análisis de Ondas

En un principio se realizó un script en Python que realiza la conexión de la diadema y realiza la recepción de las ondas. En la figura 14, se observaron los resultados del comportamiento de la señal EEG durante una muestra de prueba.

Al tomar las muestras de EEG realizando diferentes acciones corporales, los resultados no cumplieron las expectativas debido a la presencia evidente de ruido. Este ruido puede atribuirse al hecho de ser un método no invasivo, lo cual tiende a generar mayores niveles de interferencias, así como las limitaciones de hardware de la diadema. Además, resultaba muy difícil comprender y definir una acción específica y lograr realizar la clasificación efectiva. A su vez, durante el análisis de los datos y la evaluación del movimiento, se obtuvieron datos erróneos que no correspondían a una clasificación aceptable. Por tal motivo, se realizó una retroalimentación y la investigación se enfocó en las ondas que proporciona el fabricante por defecto como lo son los niveles de concentración, meditación y los parpadeos voluntarios.

Este tipo de ondas se presentan como una propiedad de la Librería *NeuroPy* llamado *rawValue* permitiendo la toma de la señal y los datos del EEG. Con estos datos se realiza el experimento de toma de los parpadeos voluntarios. A su vez se consideran los movimientos de los ojos y las cejas como se observa en la figura 15.

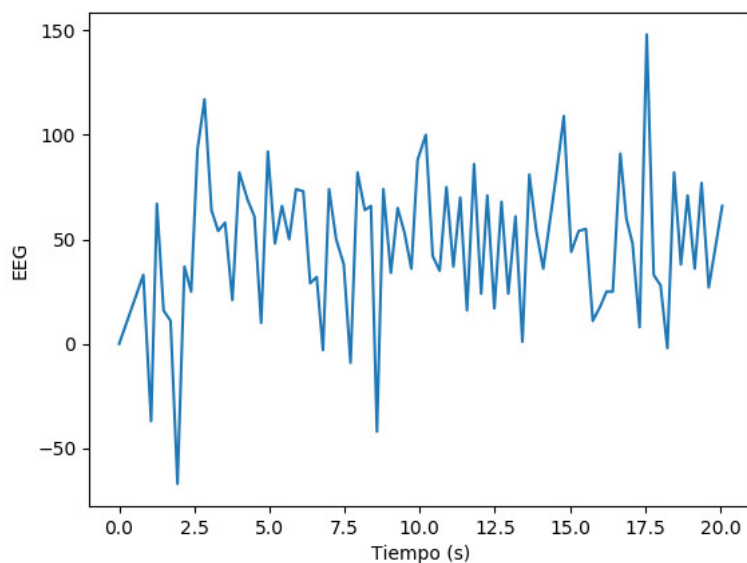


FIGURA 14: Muestra EEG movimientos de extremidades.

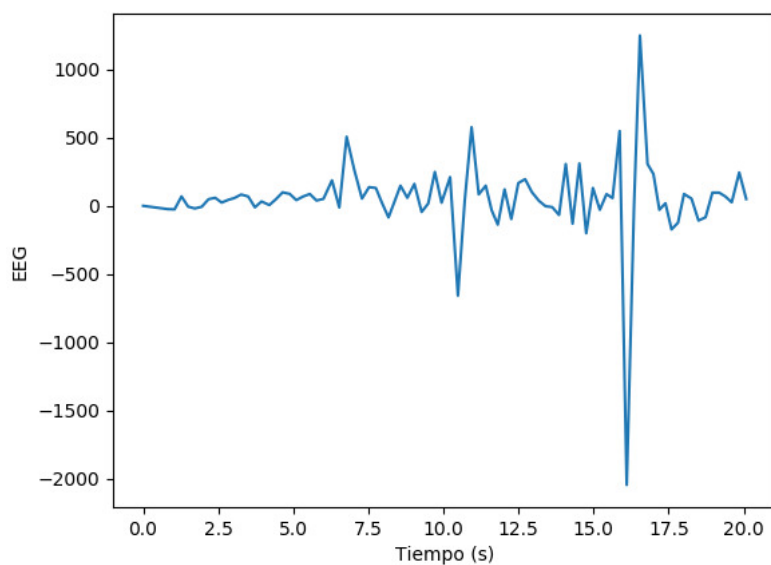


FIGURA 15: Muestra de movimientos oculares.

Teniendo en cuenta el análisis de los movimientos realizados con anterioridad y las ondas que proporciona el fabricante por defecto, se define como acción principal los parpadeos voluntarios teniendo en cuenta la interpretación de la figura 15, donde se observa el comportamiento de la señal.

Una vez definida la acción, se realizó un *dataset* siguiendo el pseudocódigo del anexo 8.1. En dicho proceso, se tomaron múltiples muestras de datos guardadas en un archivo *csv*, que corresponden periodos sin parpadeos y periodos con parpadeos de las ondas capturadas por la diadema. Para la clasificación, se definió muestras de 6 segundos donde se obtengan 3 parpadeos, considerando que cada parpadeo voluntario dura aproximadamente más de 300 ms como se define en [56]. Se establece este intervalo de tiempo para evitar la fatiga causada por los parpadeos continuos.

Con el fin de obtener una clasificación precisa y minimizar cualquier posible confusión con parpadeos involuntarios, las muestras se tomaron con parpadeos voluntarios fuertes. Esto permite maximizar la diferencia entre parpadeos voluntarios e involuntarios, asegurando así un mejor proceso de clasificación. En la figura 16, se obtiene un ejemplo de una muestra de la señal con 3 parpadeos en una muestra de 6 segundos.

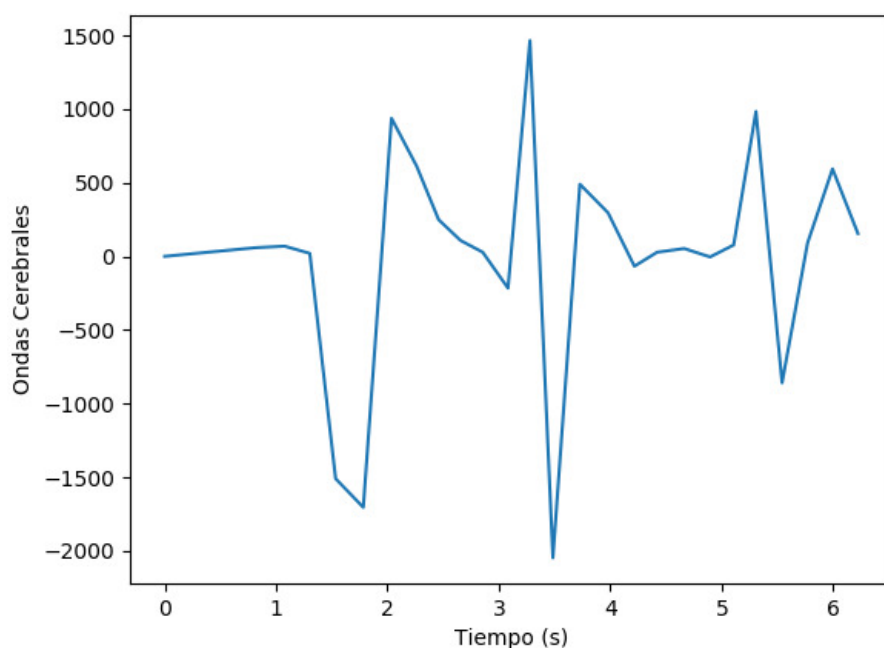


FIGURA 16: EEG 3 parpadeos voluntarios.

En el anexo 8.3, se observan muestras de las diferentes señales correspondientes a 0 parpadeos, 1 parpadeo, 2 parpadeos y 3 parpadeos. Como se observa en el anexo 8.4, no siempre se obtiene una buena señal debido al ruido que se genera en diferentes condiciones y al ambiente que se realizan las muestras. Por ello se plantearon métodos que permitan el filtrado de este ruido y así poder obtener un mejor resultado en la predicción de los parpadeos voluntarios.

## 6.2. Implementación de Red neuronal

Para implementar la red neuronal primero se definieron diferentes grupos de datos, el *dataset* contiene 9600 muestras donde se dividen en mitad parpadeos y la otra mitad no parpadeos. De estas muestras se realizó una iteración de 20 muestras por grupo, con el fin de realizar un barrido de todos los datos y así obtener una mejor interpretación del modelo.

La red neuronal implementada realiza una clasificación de aprendizaje supervisado. Durante el entrenamiento, se utilizaron grupos de datos en donde no se presentan parpadeos y otro grupo donde sí los presentan. La clasificación de estos dos grupos se definió como *binary\_crossentropy* o clasificación binaria.

De los modelos propuestos, el que mejores resultados obtuvo, fue el modelo 3 con las siguientes características:

| Red Neuronal LSTM |         |              |
|-------------------|---------|--------------|
| Capa1             | LSTM    | 128 neuronas |
| Capa 2            | Dropout | 0.1          |
| Capa 3            | LSTM    | 64 neuronas  |
| Capa 4            | Dropout | 0.1          |
| Capa de salida    | Dense   | 2 neuronas   |

TABLA 8: Características red neuronal LSTM.



Como se puede observar en la tabla 9, se llevaron a cabo pruebas con los diferentes modelos propuestos en el desarrollo conceptual. Se realizó el entrenamiento y la validación teniendo en cuenta el tiempo por época y la época en la que la variable de pérdida fue inmediatamente menor al 1 % y la exactitud se acercó a 1.

| Modelo | Tiempo/época | Épocas | Pérdida | Exactitud |
|--------|--------------|--------|---------|-----------|
| 1      | 16s          | 50     | 0.0093  | 0.9976    |
| 2      | 33s          | 32     | 0.0059  | 0.9981    |
| 3      | 57s          | 15     | 0.0083  | 0.9966    |
| 4      | 67s          | 16     | 0.0081  | 0.9974    |

TABLA 9: Características red neuronal LSTM.

Si bien algunos de los modelos realizaban una clasificación relativamente acertada, este modelo demostró obtener mejores resultados en términos de tiempo de entrenamiento y parámetros de rendimiento. Considerando que un entrenamiento se puede considerar correcto cuando la variable de pérdida *Loss* fue inferior al 1 % y la variable de precisión *Accuracy* es lo más cercano a 1. Los parámetros de entrenamiento para este modelo se definieron según la tabla 10.

| Parámetros LSTM |                     |
|-----------------|---------------------|
| Groups          | 9600                |
| Input shape     | 10                  |
| Activación      | Softmax             |
| Loss            | Binary_crossentropy |
| Optimizer       | Adam                |
| Epochs          | 50                  |
| Bach size       | 10                  |

TABLA 10: Parámetros de la red neuronal LSTM.

Los resultados de las métricas de exactitud y pérdida del entrenamiento en 50 épocas obtenidos se observan en la figura 17.

```
Epoch 50/50
960/960 [=====] - 56s 58ms/step - loss: 1.3550e-04 - accuracy: 1.0000
```

FIGURA 17: Parámetros de exactitud y pérdida del modelo entrenado.

Uno de los métodos de evaluación propuestos para el sistema BCI, fue la matriz de confusión. Este método permite evaluar la precisión de la red neuronal previamente entrenada con datos de pruebas y cuantificando el porcentaje de las predicciones.

Teniendo en cuenta el modelo previamente entrenado, en la figura 18 podemos observar cómo es el comportamiento de las predicciones, donde se obtiene una de efectividad de no parpadeos de un 100 % y si parpadeos de un 93.75 %. Hay que tener en cuenta que se filtró el ruido de los datos, definiendo un rango experimental teniendo en cuenta el análisis de datos realizado con anterioridad.

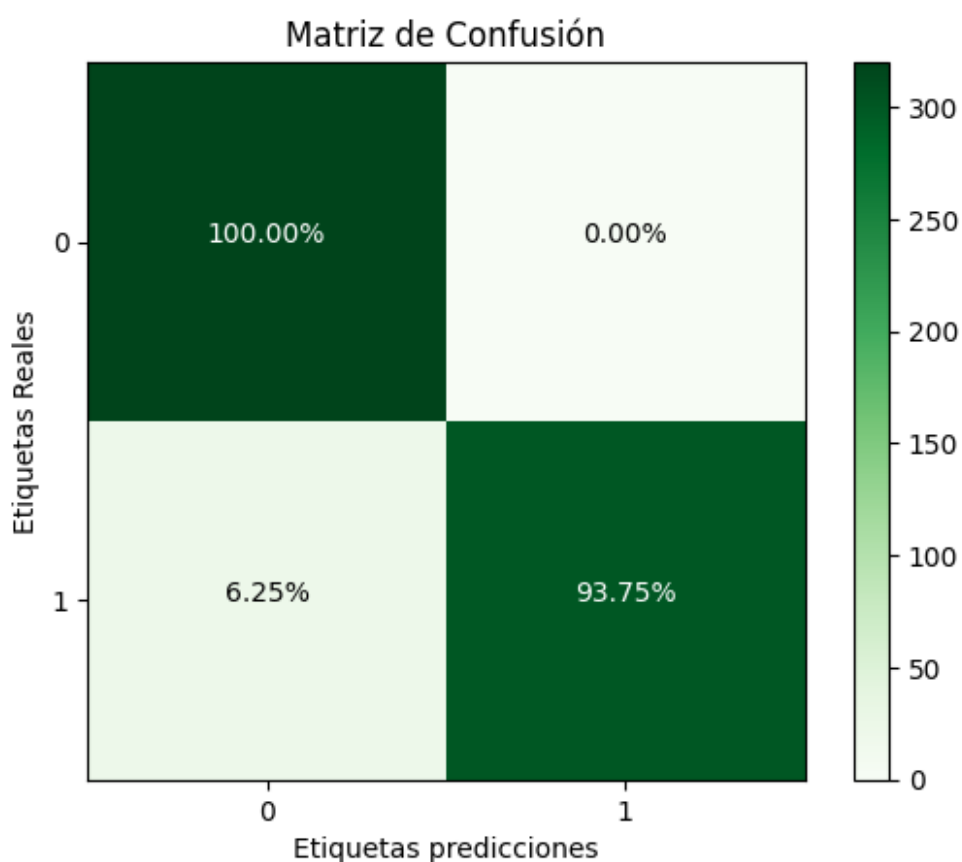


FIGURA 18: Matriz de confusión modelo LSTM.

Se realizó un script el cual permite la visualización de cómo se clasifican los parpadeos. Para observar esta clasificación de parpadeos voluntarios, inicialmente se cargó el modelo entrenado previamente, seguidamente se propuso un nuevo *dataset* de prueba el cual contiene diferentes parpadeos en una muestra de 10 segundos, a la cual posteriormente se le aplica el filtrado del ruido de la señal, donde finalmente se realiza la predicción basados en el modelo propuesto y se grafica el resultado obtenido. En la figura 19, se observar cómo se realiza la clasificación de los parpadeos, realizando la predicción de donde son detectados.

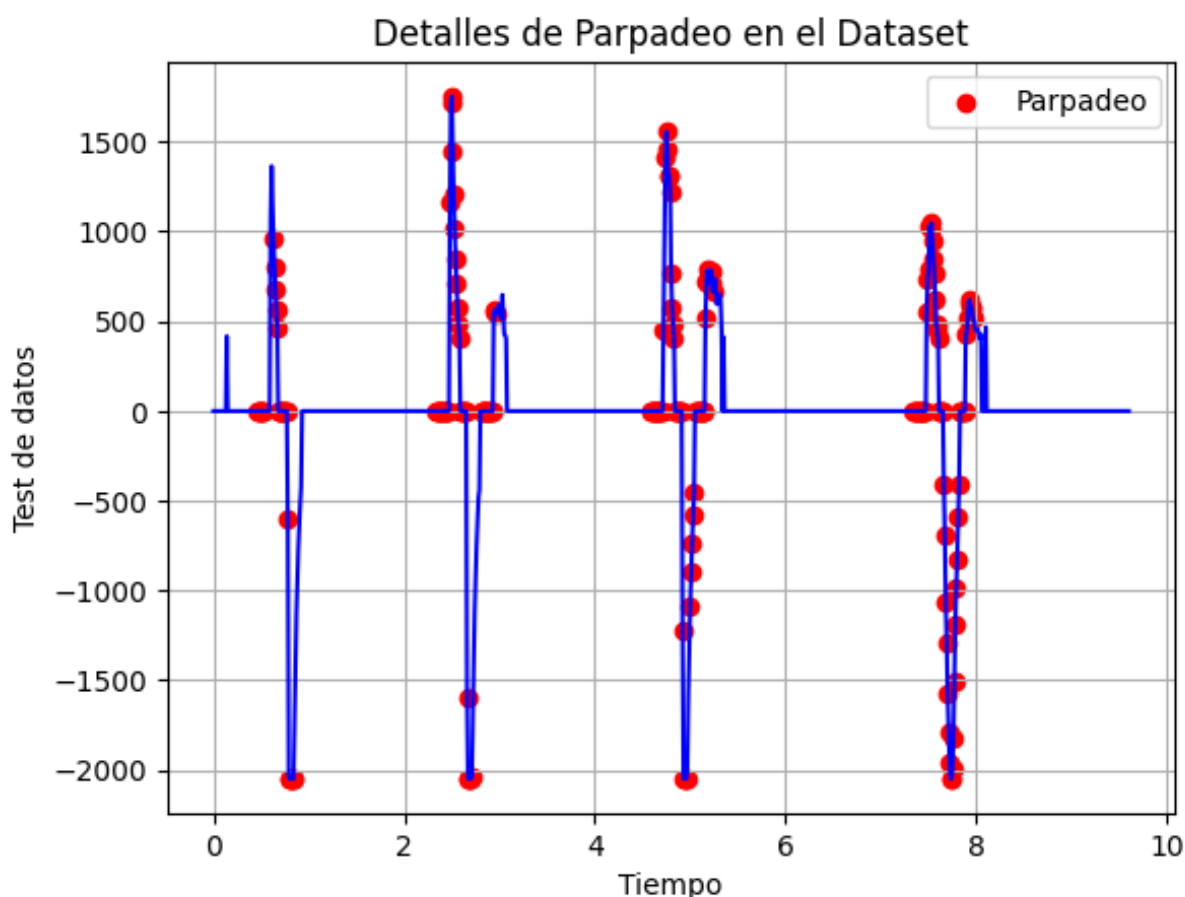


FIGURA 19: Detección parpadeos.

Para el conteo de los parpadeos, inicialmente se carga el *dataset* y se realiza el filtrado del ruido; Posteriormente se inicializa un contador de parpadeos, después se realiza la predicción y se evalúa el arreglo de las predicciones de la clasificación binaria, donde si en el arreglo existe un valor *True* o 1 consecutivamente en una iteración mayor a 30 y que a su vez finalice cuando el siguiente dato sea un *False* o 0, realiza el conteo de los parpadeos y así se obtiene la predicción.

En la figura 20, se observa cómo realiza la predicción para cada uno de los grupos de parpadeos y en la figura 21, se calculan los parpadeos totales detectados en el *dataset* de test.

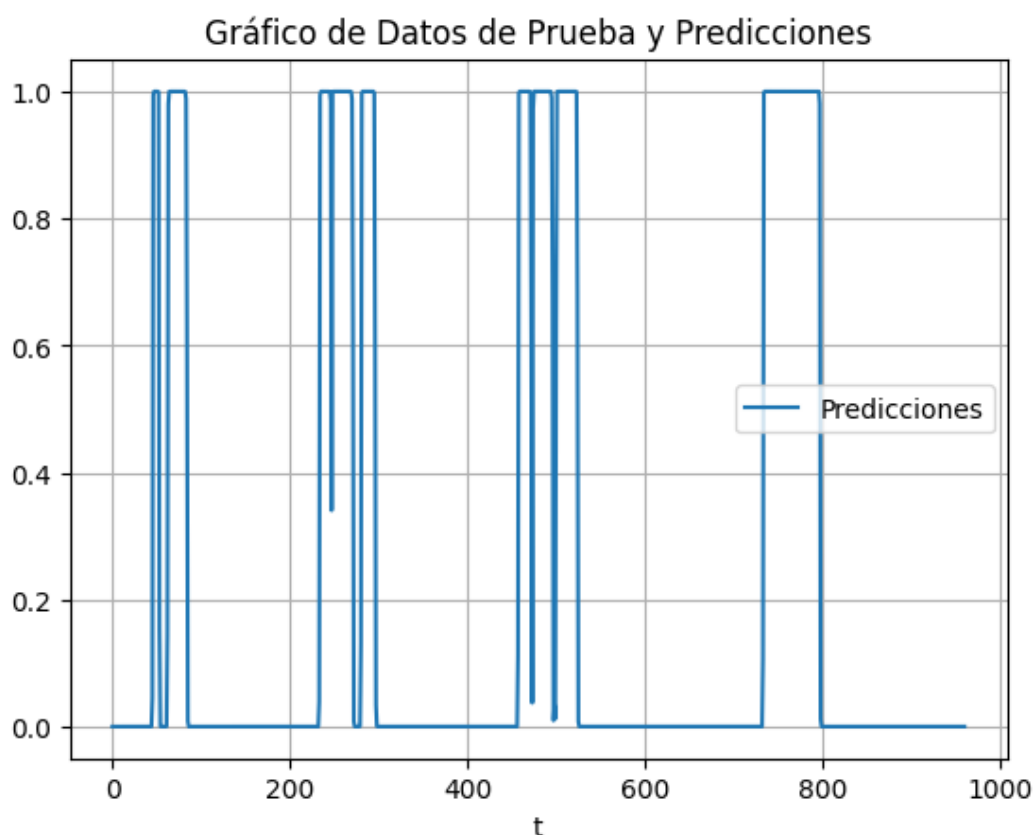


FIGURA 20: Predicción de los parpadeos.

**Número de parpadeos detectados: 4**

FIGURA 21: Parpadeos detectados.

### 6.3. Funciones de Movimiento Pepper

Para la programación del robot Pepper se tuvieron en cuenta las diferentes funcionalidades que se definieron en el desarrollo conceptual. Las funciones del robot fueron programadas mediante la computadora interna de Pepper. Se programaron diferentes funcionalidades como se aprecia en el Anexo 8.5, de estas se tomaron 3; una funcionalidad de habla y dos funcionalidades de acciones del robot.

Como se puede apreciar en la figura 22, cuando el robot realiza la funcionalidad de hablar su color de ojos cambia, esto indica que se está ejecutando la función de habla correspondiente a la descripción del proyecto.

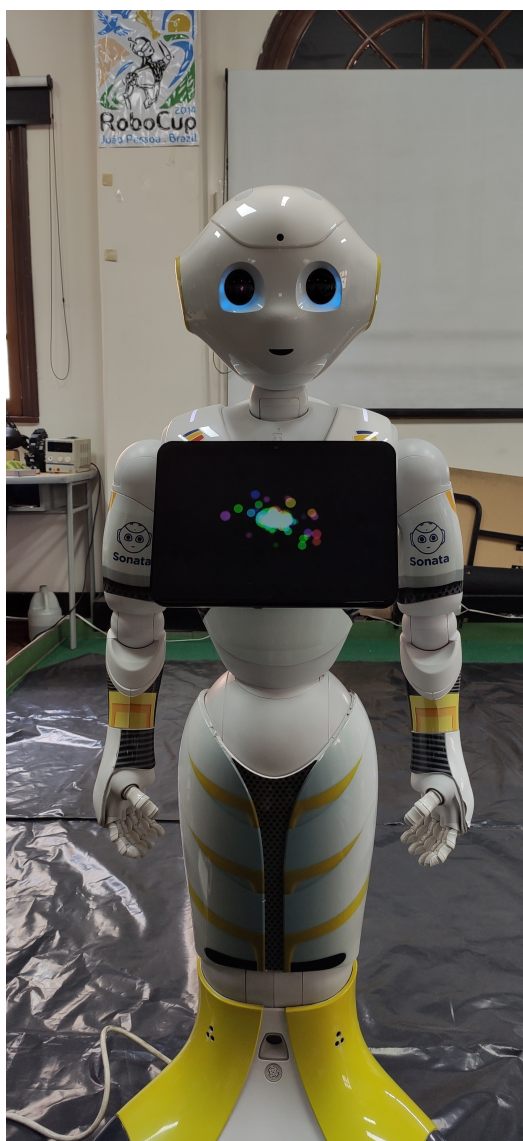


FIGURA 22: Funcionalidad descripción del Proyecto.

En la figura 23, se observa la acción correspondiente al saludo del robot. Esta acción fue tomada de las acciones predefinidas obtenidas de la librería de Pepper.

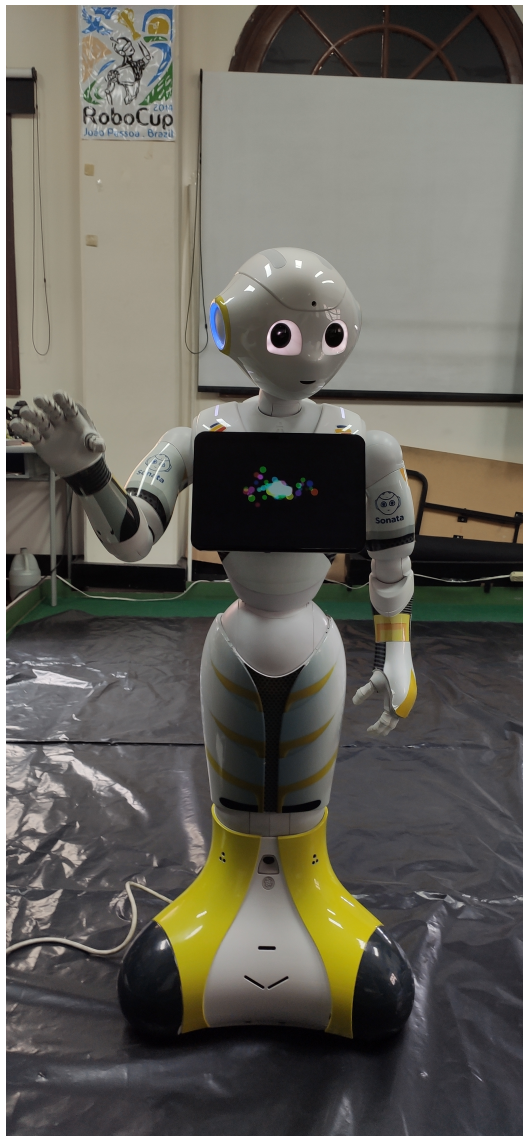


FIGURA 23: Funcionalidad saludo.

Se observa en la figura 24, como se realiza la acción de saludo de mano adoptado por el robot. Esta acción se programó manualmente teniendo en cuenta las extremidades y actuadores del robot, así como sus grados de libertad.

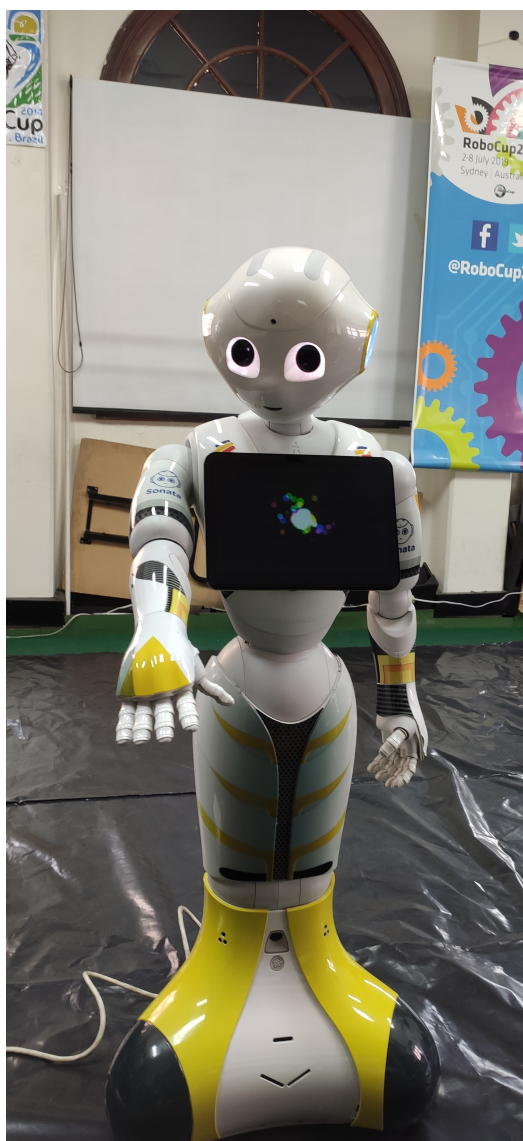


FIGURA 24: Funcionalidad saludo de mano.

## 6.4. Sistema BCI

Finalmente, se realiza la implementación de los nodos en ROS considerando sus características propuestas en el desarrollo conceptual. Para ello, se realizó un *workspace* el cual es el directorio de los diferentes paquetes y archivos de compilación. Una vez se implementaron los nodos, se procede a realizar la conexión de cada uno de ellos mediante los nodos descritos anteriormente; se realizó la comprobación de la conexión y envío de datos mediante el comando *rostopic*. Los pseudocódigos de los nodos implementados se describen en el anexo 8.2. En la figura 25, se observa los nodos implementados durante la ejecución del sistema BCI.

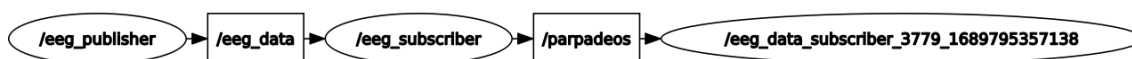


FIGURA 25: Ejecución de nodos en ros.

En el código, se muestra el nodo en ROS de clasificación de parpadeos en la terminal de *Linux*. Se pueden observar las impresiones de texto correspondientes a las funcionalidades asociadas a las tres clasificaciones de parpadeos previamente definidas.

---

```

johan@johan-Lenovo-ideapad-320-14ISK:~$ rosrunc Reception Red.py
[INFO] [1689191984.217013]: Número total de parpadeos: 1
[INFO] [1689192016.025431]: Número total de parpadeos: 2
[INFO] [1689192034.279648]: Número total de parpadeos: 3
  
```

---

Posteriormente, en el código se observa el nodo de ROS correspondiente a las funcionalidades del robot. Estas funcionalidades se clasificaron dependiendo de los parpadeos clasificados anteriormente, podemos evidenciar como con cada muestra de 1, 2 y 3 parpadeos, se clasifica la funcionalidad correspondiente del robot.

---

```

johan@johan-Lenovo-ideapad-320-14ISK:~$ rosrunc Reception Pepper.py
Funcionalidad 1
Funcionalidad 2
Funcionalidad 3
  
```

---



## 6.5. Escenarios de Prueba

En esta sección, se presentan los escenarios de prueba diseñados para evaluar el funcionamiento y la efectividad del sistema BCI desarrollado en ROS. Cada escenario de prueba se enfoca en el entrenamiento, clasificación y la toma de resultados. La información de los códigos y los *dataset* se encuentra disponible en el repositorio [57].

### 6.5.1. Escenario 1: Entrenamiento

En este escenario, se realizaron pruebas para la adquisición de ondas y el posterior entrenamiento de la red neuronal con las siguientes condiciones:

1. Locación de pruebas: laboratorio de robótica.
2. Conexión de la diadema: bluetooth.
3. Nivel de ruido: entre -200 y 200 mv.
4. Batería de dispositivo: 100 %.

Teniendo en cuenta las condiciones anteriormente mencionadas, se llevó a cabo la toma de muestras de señales EEG correspondientes a los parpadeos voluntarios en horarios de poca afluencia de personas. Una muestra de las ondas capturadas se encuentra adjunto en el anexo 8.3, donde se puede observar el comportamiento de la señal bajo estas condiciones. Los datos obtenidos se almacenaron en conjuntos de datos.

Una vez obtenidos los dataset, se llevó a cabo el entrenamiento de la red neuronal, teniendo en cuenta los parámetros especificados en la tabla 10.

### 6.5.2. Escenario 2: Clasificación en Otros Entornos

En este escenario se realizaron las pruebas de clasificación de parpadeos voluntarios mediante la adquisición de nuevas ondas bajo las siguientes condiciones:

1. Locación de prueba: laboratorio ETM 10.
2. Conexión de la diadema: bluetooth.

3. Nivel de ruido: entre -400 y 400 mv registrado por la diadema.
4. Batería del dispositivo: 70 %.

Esta clasificación se realizó bajo condiciones con alta afluencia de personas durante los horarios pico de estudio. En este escenario, la clasificación se realizó con señales sin filtrado, lo que resultó en fallos debido a los niveles de ruido como se observó en el anexo 8.4. Por esta razón se propuso filtrar los niveles de ruido.

### **6.5.3. Escenario 3: Toma de Resultados de la Interfaz BCI**

Finalmente, en este escenario se realizaron las pruebas finales, registrando los resultados de la clasificación de la interfaz con las siguientes condiciones:

1. Locación de pruebas: laboratorio de robótica.
2. Conexión de la diadema: bluetooth.
3. Batería de dispositivo: entre 100 % y 10 %.
4. Nivel de ruido: inferior a 200 mv registrado por la diadema.
5. Red WIFI: 'CIEGO (red propia)'.
6. Conexión del robot: SSH.
7. Batería del robot: 50 %.

La toma de resultados se realizó en condiciones similares al escenario de entrenamiento. Inicialmente se ejecutó el entorno ROS junto con los nodos de recepción, clasificación y funcionalidad. Se realizaron 6 pruebas cada una con 50 intentos. La evaluación se realizó para 0, 1, 2 y 3 parpadeos. Por cada intento se registró el número de aciertos, el número de fallos y el porcentaje de efectividad.

## 6.6. Resultados

En esta sección se muestran los resultados de la evaluación de la interfaz BCI, donde se comprueba su funcionamiento y efectividad. Esto permite sacar conclusiones y realizar posibles mejoras en caso de no obtener la clasificación correspondiente.

En las tablas 11, 12, 13 y 14 , se muestra el resultado de la evaluación experimental teniendo en cuenta el escenario de toma de resultados de la interfaz BCI .

| 0 PARPADEOS |                    |                    |                  |                        |
|-------------|--------------------|--------------------|------------------|------------------------|
| PRUEBA      | NUMERO DE INTENTOS | NUMERO DE ACIERTOS | NUMERO DE FALLAS | PORCENTAJE EFECTIVIDAD |
| 1           | 50                 | 49                 | 1                | 98                     |
| 2           | 50                 | 50                 | 0                | 100                    |
| 3           | 50                 | 49                 | 1                | 98                     |
| 4           | 50                 | 50                 | 0                | 100                    |
| 5           | 50                 | 50                 | 0                | 100                    |
| 6           | 50                 | 48                 | 2                | 96                     |
|             | 300                | 296                | 4                | 98,66                  |

TABLA 11: Evaluación de 0 parpadeos.

En la tabla 11, se realizó la evaluación con 0 parpadeos para verificar el funcionamiento del filtro propuesto. Se observa cómo se obtiene un alto porcentaje de efectividad teniendo en cuenta a su vez que el ruido está presente entre -400 y 400 mV.

| 1 PARPADEO |                    |                    |                  |                        |
|------------|--------------------|--------------------|------------------|------------------------|
| PRUEBA     | NUMERO DE INTENTOS | NUMERO DE ACIERTOS | NUMERO DE FALLAS | PORCENTAJE EFECTIVIDAD |
| 1          | 50                 | 40                 | 10               | 80                     |
| 2          | 50                 | 36                 | 14               | 72                     |
| 3          | 50                 | 42                 | 8                | 84                     |
| 4          | 50                 | 47                 | 3                | 94                     |
| 5          | 50                 | 48                 | 2                | 96                     |
| 6          | 50                 | 46                 | 4                | 92                     |
|            | 300                | 259                | 41               | 86,33                  |

TABLA 12: Evaluación de 1 parpadeo.

Se observa en la tabla 12, el porcentaje de efectividad de la simulación de ondas con 1 parpadeo. Se evidencia que, en 300 intentos, se tuvieron 259 aciertos correspondiente a un porcentaje de efectividad del 86.33 %.

| 2 PARPADEOS |                    |                    |                  |                        |
|-------------|--------------------|--------------------|------------------|------------------------|
| PRUEBA      | NUMERO DE INTENTOS | NUMERO DE ACIERTOS | NUMERO DE FALLAS | PORCENTAJE EFECTIVIDAD |
| 1           | 50                 | 47                 | 3                | 94                     |
| 2           | 50                 | 41                 | 9                | 82                     |
| 3           | 50                 | 45                 | 5                | 90                     |
| 4           | 50                 | 49                 | 1                | 98                     |
| 5           | 50                 | 47                 | 3                | 94                     |
| 6           | 50                 | 48                 | 2                | 96                     |
|             | 300                | 277                | 23               | 92,33                  |

TABLA 13: Evaluación de 2 parpadeos.

En la tabla 13, se logra un porcentaje total de efectividad del 92.33 % al evaluar el rendimiento del sistema. De los 300 intentos, solo se obtuvieron 23 predicciones fallidas, siendo la mejor clasificación de las tres que contienen parpadeos.

| 3 PARPADEOS |                    |                    |                  |                        |
|-------------|--------------------|--------------------|------------------|------------------------|
| PRUEBA      | NUMERO DE INTENTOS | NUMERO DE ACIERTOS | NUMERO DE FALLAS | PORCENTAJE EFECTIVIDAD |
| 1           | 50                 | 42                 | 8                | 84                     |
| 2           | 50                 | 48                 | 2                | 96                     |
| 3           | 50                 | 44                 | 6                | 88                     |
| 4           | 50                 | 40                 | 10               | 80                     |
| 5           | 50                 | 49                 | 1                | 98                     |
| 6           | 50                 | 46                 | 4                | 92                     |
|             | 300                | 269                | 31               | 89,66                  |

TABLA 14: Evaluación de 3 parpadeos.

Teniendo en cuenta la toma de 3 parpadeos, se obtuvo una efectividad del 89.66 % como se evidencia en la tabla 14.

| PARPADEOS | PORCENTAJE EFECTIVIDAD |
|-----------|------------------------|
| 0         | 98,66                  |
| 1         | 86,33                  |
| 2         | 92,33                  |
| 3         | 89,66                  |
|           | 91.75                  |

TABLA 15: Efectividad total de la interfaz BCI.

En la tabla 15, se observan los resultados experimentales de los parpadeos adquiridos, los cuales reflejan una efectividad total del 91.75 % para la interfaz desarrollada. Estos indicadores proporcionan una visión detallada del desempeño del sistema, permitiendo identificar posibles áreas a mejorar.

## Capítulo 7

# Conclusiones y Trabajos Futuros

### 7.1. Conclusiones

Gracias a la investigación sobre técnicas para la adquisición, procesamiento e interpretación de señales EEG, se identificaron diferentes metodologías como se muestra en la tabla 1. Estas técnicas desempeñaron un papel fundamental en la orientación del proyecto.

Se evidenció como el modelo implementado obtiene un alto nivel de predicción como se observa en la tabla 15, ya que la manera en que se realizó la toma de los datos haciendo un barrido por toda la gráfica ayudó al modelo a interpretar de mejor manera como era la clasificación. Por el contrario, si se hubiese simplemente dado una ventana fija donde se le indicaba donde estaban los diferentes parpadeos clasificando cada uno, el sistema tendría que ser más robusto y no se tendría un buen nivel de predicción.

Un aspecto que mejoró la predicción fue filtrar el ruido de la señal gracias a la toma de muestras en otros ambientes y el análisis de datos como se observa en la figura 19. A su vez no se realizó una normalización de los datos ya que el sistema realizó la clasificación de mejor manera sin implementar la normalización.

Al tener una red neuronal más robusta como los modelos propuestos, se realiza un mejor aprendizaje en un menor número de épocas como se describe en la tabla 9. Sin embargo, se corre el riesgo de tener un overfitting o sobre entrenamiento de la red neuronal, es por esta razón que es importante tener en cuenta el número de épocas, el tiempo de entrenamiento y las variables de exactitud y pérdida.

Gracias al escenario de prueba de evaluación de la interfaz, se ha podido evidenciar claramente el contraste en términos de efectividad del modelo entrenado a partir de la matriz de confusión 18. Estos resultados obtenidos han sido fundamentales para el análisis de la interfaz a partir de las métricas de desempeño y evaluación. Se ha podido evidenciar la capacidad de la interfaz para clasificar correctamente las distintas categorías de parpadeos. Estos hallazgos proporcionan una base sólida para comprender el rendimiento del sistema BCI desarrollado.

## **7.2. Trabajos Futuros**

Teniendo en cuenta los resultados obtenidos en la clasificación de EEG, como trabajos futuros se abren interesantes posibilidades para explorar futuros proyectos que involucren adquisición y clasificación de ondas cerebrales utilizando dispositivos EEG de última generación. De esta manera se podrían realizar estudios e investigaciones más exhaustiva y enriquecedora de las ondas cerebrales y su correlación con diferentes condiciones y procesos mentales para interactuar con sistemas BCI que permitan una inmersión directa con la tecnología. Estos avances podrían tener un impacto significativo en diferentes áreas, ampliando un sin fin de nuevas tecnologías y posibilidades.



# Bibliografía

- [1] Q. Miguel y D. Rafael, «Escuela Politécnica Superior Reconocimiento de emociones humanas y su aplicación a la Robótica Social», Escuela Politécnica Superior, 2019.
- [2] C. de la República, «Ley 1346 de 2009 - Convención sobre los Derechos de las Personas con Discapacidad», 1346.
- [3] L. N. Morales y C. A. Rotela, «Types of disabilities in a community of Caazapá», *Anales de la Facultad de Ciencias Médicas (Asunción)*, vol. 52, págs. 69-76, 3 dic. de 2019, ISSN: 18168949. DOI: 10.18004/anales/2019.052.03.69-076.
- [4] C. Alzate, J. César, C. Perea y A. Santiago, «Boletines Poblacionales: Personas con Discapacidad-PCD 1 Oficina de Promoción Social I-2020»,
- [5] B. T. (GEIH), «Mercado laboral de personas con discapacidad Trimestre móvil enero-marzo 2022», 2022.
- [6] E. Trujillo U, «El problema hospitalario», Español, *Revista Facultad Nacional de Salud Pública*, 2014, ISSN: 0120-386X. dirección: <https://www.redalyc.org/articulo.oa?id=12058127002>.
- [7] G. Ozkil Ali, Z. Fan, S. Dawids, H. Aanes, J. K. Kristensen y K. H. Christensen, «Service robots for hospitals: A case study of transportation tasks in a hospital», en *2009 IEEE international conference on automation and logistics*, IEEE, 2009, págs. 289-294.
- [8] B. Ignacio González, G. T. David y V. H. Nallely. «Actividad motora: Una aproximación integral - Google Libros». (), dirección: [https://books.google.com.co/books?hl=es&lr=&id=DgBMEAAQBAJ&oi=fnd&pg=PA307&dq=discapacidades+motoras+ejemplos&ots=GdMqieZQ3c&sig=nwF722nV3-skAAvgUWUxFC707NM&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.co/books?hl=es&lr=&id=DgBMEAAQBAJ&oi=fnd&pg=PA307&dq=discapacidades+motoras+ejemplos&ots=GdMqieZQ3c&sig=nwF722nV3-skAAvgUWUxFC707NM&redir_esc=y#v=onepage&q&f=false).
- [9] M. Xu, F. He, T. P. Jung, X. Gu y D. Ming, «Current Challenges for the Practical Application of Electroencephalography-Based Brain-Computer Interfaces», *Engineering*, vol. 7, págs. 1710-1712, 12 dic. de 2021, ISSN: 20958099. DOI: 10.1016/j.eng.2021.09.011.

- 
- [10] A. Felipe y R. Fernández, «Herramienta para la generación de texto basada en una interfaz cerebro-computador», Universidad Santo Tomás, 2020.
  - [11] C. Camilo y C. Castañeda, «Ros-Gazebo. una Valiosa Herramienta de Vanguardia para el desarrollo de la Robótica Ros-Gazebo.», Universidad Nacional Abierta y a Distancia, 2015.
  - [12] «Objetivos de Desarrollo Sostenible (ODS) | Comisión Económica para América Latina y el Caribe». (2022), dirección: <https://www.cepal.org/es/temas/agenda-2030-desarrollo-sostenible/objetivos-desarrollo-sostenible-ods>.
  - [13] *Casi mil millones de personas con discapacidades carecen de acceso a la tecnología de asistencia | Noticias ONU*. dirección: <https://news.un.org/es/story/2022/05/1508712>.
  - [14] Q. Wodon, C. Male, C. Montenegro y A. Nayihouba, «The Challenge of Inclusive Education in Sub-Saharan Africa», *The Challenge of Inclusive Education in Sub-Saharan Africa*, dic. de 2018. DOI: 10.1596/31005. dirección: <https://www.bancomundial.org/es/topic/disability>.
  - [15] G. Acosta, J. José, M. Mendoza et al., «Proyección Social USTA COLOMBIA». dirección: <https://repository.usta.edu.co/handle/11634/19319>.
  - [16] L. M. Alonso-Valerdi, M. A. Arreola-Villarruel, J. Argüello-García, L. M. Alonso-Valerdi, M. A. Arreola-Villarruel y J. Argüello-García, «Interfaces Cerebro-Computadora: Conceptualización, Retos de Rediseño e Impacto Social», *Revista mexicana de ingeniería biomédica*, vol. 40, págs. 1-18, 3 sep. de 2019, ISSN: 0188-9532. DOI: 10.17488/RMIB.40.3.8. dirección: [http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S0188-95322019000300008&lng=es&nrm=iso&tlng=es%20http://www.scielo.org.mx/scielo.php?script=sci\\_abstract&pid=S0188-95322019000300008&lng=es&nrm=iso&tlng=es](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0188-95322019000300008&lng=es&nrm=iso&tlng=es%20http://www.scielo.org.mx/scielo.php?script=sci_abstract&pid=S0188-95322019000300008&lng=es&nrm=iso&tlng=es).
  - [17] I. Moreno, E. Batista, S. Serracín et al., «Vista de Los sistemas de interfaz cerebro-computadora basado en EEG: características y aplicaciones», 2019. dirección: <https://revistas.utp.ac.pa/index.php/id-tecnologico/article/view/2230/3137>.
  - [18] E. M. H. Gonzalez, «MANUAL SISTEMA 10/20 INTERNACIONAL ANEXO», 2020.
  - [19] S. En, L. A. Educación, E. A. Sciotto y E. B. Niripil, «ONDAS CEREBRALES, CONCIENCIA Y COGNICIÓN».
  - [20] X. B. Olabe, «REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES», 2008.
  - [21] A. Patil, «History of Artificial Neural Network and Deep Learning | by Ashish Patil | Medium», dirección: <https://medium.com/@ashishpatil199199/history-of-artificial-neural-network-and-deep-learning-b5f6cb0c296b>.

- 
- [22] S. Kamilya y P. J. Alex, «A survey on LSTM memristive neural network architectures and applications», *European Physical Journal: Special Topics*, vol. 228, págs. 2313-2324, 10 oct. de 2019, ISSN: 19516401. DOI: 10.1140/epjst/e2019-900046-x.
  - [23] A. De y Original, «CHECAsEN: programa para revisar señales EEG fuera de línea», 2010, págs. 135-141. dirección: [www.medigraphic.org.mx](http://www.medigraphic.org.mx)
  - [24] G. L. Cristian, «Dialnet-AnalisisDeReduccionDeRuidoEnSenalesEegOrientadoAIR-5062974», 2008.
  - [25] F. Torres, C. Sánchez y K. Palacio-Baus, «Adquisición y análisis de señales cerebrales utilizando el dispositivo MindWave», 2014. dirección: [www.neurosky.com](http://www.neurosky.com).
  - [26] M. L.-G. Gerardo y H. S. David, «ANALISIS MULTIRESOLUCION EN SEÑALES DE ELECTROENCEFALOGRAMA CON APLICACIONES AL MAPEO CEREBRAL. VALIDACION: EVENTO VASCULAR CEREBRAL», 2013.
  - [27] A. A. T. García, C. A. R. García, I. L. V. Pineda e I. G. G. Aguilar, «Análisis y clasificación de electroencefalogramas (EEG) registrados durante el habla imaginada Por», 2016.
  - [28] I. J. David y A. Galeano, «Sistema BCI basado en el control de un brazo robótico de 6 grados de libertad para el tratamiento del TDAH», 2021.
  - [29] S. Aggarwal y N. Chugh, «Review of Machine Learning Techniques for EEG Based Brain Computer Interface», *Archives of Computational Methods in Engineering*, vol. 29, págs. 3001-3020, 2022. DOI: 10.1007/s11831-021-09684-6. dirección: <https://doi.org/10.1007/s11831-021-09684-6>.
  - [30] N. A. Alzahab, L. Apollonio, A. Di Iorio et al., «Hybrid Deep Learning (hDL)-Based Brain-Computer Interface (BCI) Systems: A Systematic Review», *Brain Sciences*, vol. 11, n.º 1, 2021, ISSN: 2076-3425. dirección: <https://www.mdpi.com/2076-3425/11/1/75>.
  - [31] S. Ana Blasi y F. M. Jordi, «Diseño de un modelo Deep Learning de clasificación de imágenes para la detección de la neumonía MEMORIA», 2021.
  - [32] Z. Cao, «A review of artificial intelligence for EEG-based braincomputer interfaces and applications», *Brain Science Advances*, vol. 6, n.º 3, págs. 162-170, 2020. DOI: 10.26599/BSA.2020.9050017. eprint: <https://doi.org/10.26599/BSA.2020.9050017>. dirección: <https://doi.org/10.26599/BSA.2020.9050017>.
  - [33] O. Roesler, *EEG Eye State*, UCI Machine Learning Repository, DOI: <https://doi.org/10.24432/C57G7J>, 2013.

- 
- [34] M. Fernandes, S. K. Singh, K. Prabhu y D. Sharma, «GitHub - MexsonFernandes/EEG-controlled-Robot-using-Neurosky-Mindwave-Mobile2: Mobile Robot controlled using Neurosky Mindwave headset.», 2021. dirección: <https://github.com/MexsonFernandes/EEG-controlled-Robot-using-Neurosky-Mindwave-Mobile2#readme>.
  - [35] D. Toapanta, «DESARROLLO DE UNA SILLA DE RUEDAS ELÉCTRICA CONTROLADO POR ONDAS CEREBRALES USANDO EL DISPOSITIVO NEUROSKY MINDWAVE MOBILE 2 AUTOR: CRISTIAN DANIEL TOAPANTA VEGA TUTOR: Mg. RENÉ ERNESTO CORTIJO LEYVA»,
  - [36] M. Agarwal, «GitHub - meagmohit/EEG-Datasets: A list of all public EEG-datasets», dirección: <https://github.com/meagmohit/EEG-Datasets>.
  - [37] S. Kanoga, M. Nakanishi e Y. Mitsukura, «Assessing the effects of voluntary and involuntary eyeblinks in independent components of electroencephalogram», *Neurocomputing*, vol. 193, págs. 20-32, 2016, ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2016.01.057>. dirección: <https://www.sciencedirect.com/science/article/pii/S0925231216001569>.
  - [38] J. Osselton, C. Shaw y R. Cooper, *EEG Technology - Google Libros*. 1974. dirección: [https://books.google.com.co/books?hl=es&lr=&id=Zgg6AwAAQBAJ&oi=fnd&pg=PP1&dq=eeg&ots=tVc1KA7o2Z&sig=wnbeU4z9SEtJNihlzf7QkSANU7g&redir\\_esc=y#v=onepage&q=eeg&f=false](https://books.google.com.co/books?hl=es&lr=&id=Zgg6AwAAQBAJ&oi=fnd&pg=PP1&dq=eeg&ots=tVc1KA7o2Z&sig=wnbeU4z9SEtJNihlzf7QkSANU7g&redir_esc=y#v=onepage&q=eeg&f=false).
  - [39] «MindWave Mobile 2: User Guide»,
  - [40] C. Janiesch, P. Zschech y K. Heinrich, «Machine learning and deep learning», DOI: 10.1007/s12525-021-00475-2 / Published. dirección: <https://doi.org/10.1007/s12525-021-00475-2>.
  - [41] C. A. R. M. S. B. Autor y D. J. Matich, «Cátedra: Informática Aplicada a la Ingeniería de Procesos-Orientación I Redes Neuronales: Conceptos Básicos y Aplicaciones», 2001.
  - [42] J. Fedjaev, «Decoding eeg brain signals using recurrent neural networks», 2017.
  - [43] D. D. Trabajo y C. Arana, «UNIVERSIDAD DEL CEMA Buenos Aires Argentina Serie», 2021. dirección: [www.cema.edu.ar/publicaciones/doc\\_trabajo.html](http://www.cema.edu.ar/publicaciones/doc_trabajo.html).
  - [44] F. Bueno, «Redes Neuronales: Entrenamiento y Comportamiento», 2019.
  - [45] C. Ruben, «Redes neuronales recurrentes | Qué son las RNN», 2021. dirección: <https://abdatum.com/tecnologia/redes-neuronales-recurrentes>.
  - [46] J. P. G. Sevilla, J. de, T. por y R. P. Mejías, «REDES RECURRENTEs», 2020.

- 
- [47] J. B. Disclaimer, «Long Short-Term Memory Networks With Python Develop Sequence Prediction Models With Deep Learning», 2017.
  - [48] A. G. C. Director y J. L. M. Lozano, «ROS: Robot Operating System», Universidad Politécnica de Cartagena, 2013.
  - [49] J. T. Quesada, «Captura, visualización y seguimiento de objetos TONDA QUESADA JORGE», Universidad de Alicante, 2017.
  - [50] Ş. Vural, «ROS 101», 2021. dirección: <https://medium.com/move-on-ai/ros-101-30b8ca5c0c7a>.
  - [51] O. F. C. Camargo, L. F. P. Martinez y C. A. Hernandez, «TECNOLOGÍA BLUETOOTH: ALTERNATIVA PARA REDES CELULARES DE VOZ Y DATOS BLUETOOTH TECHNOLOGY: ALTERNATIVE TO CELLULAR NETWORKS OF VOICE AND DATA Palabras clave», 2009.
  - [52] R. Shaw, C. Mohanty, B. Patra y A. Pradhan, «1D Multi-Point Local Ternary Pattern: A Novel Feature Extraction Method for Analyzing Cognitive Engagement of students in Flipped Learning Pedagogy», *Cognitive Computation*, mayo de 2022. DOI: 10.1007/s12559-022-10023-5.
  - [53] sahil, «lihas/NeuroPy: NeuroPy library written in python to connect, interact and get data from neurosky's MindWave EEG headset.», 2017. dirección: <https://github.com/lihas/NeuroPy>.
  - [54] M. P. S. BELTRÁN, «APRENDIZAJE PROFUNDO PARA LA DETECCIÓN DE CAÍDAS EN PERSONAS VULNERABLES», Universidad Santo Tomás, 2023.
  - [55] *Motors — Aldebaran 2.4.3.28-r2 documentation*. dirección: [http://doc.aldebaran.com/2-4/family/pepper\\_technical/motors\\_pep.html](http://doc.aldebaran.com/2-4/family/pepper_technical/motors_pep.html).
  - [56] M. Varela, «Raw EEG signal processing for BCI control based on voluntary eye blinks», en *2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)*, 2015, págs. 1-6. DOI: 10.1109/CONCAPAN.2015.7428477.
  - [57] J. S. M. Burgos, «JohanSMB/DESARROLLO-DE-UN-SISTEMA-BCI-BASADO-EN-EL-DISPOSITIVO-MINDWAVE-MOBILE-2: DESARROLLO DE UN SISTEMA BCI BASADO EN EL DISPOSITIVO MINDWAVE MOBILE 2 PARA EL CONTROL DE LAS FUNCIONES DE MOVIMIENTO DEL ROBOT PEPPER», 2023. dirección: <https://github.com/JohanSMB/DESARROLLO-DE-UN-SISTEMA-BCI-BASADO-EN-EL-DISPOSITIVO-MINDWAVE-MOBILE-2/tree/master>.

# Capítulo 8

## Anexos

### 8.1. Anexo A:

Código de recepción de EEG de la diadema *MindWave Mobile 2*.

---

**Algorithm 1:** Obtención de datos EEG y generación de gráfica

---

**Importar** : NeuroPy, time, pandas as pd, matplotlib.pyplot as plt, os.path

**Declarar** : neuropy, eeg\_data

**Inicializar:** neuropy  $\leftarrow$  NeuroPy(/dev/rfcomm0")

eeg\_data  $\leftarrow$  []

neuropy.start() start\_time  $\leftarrow$  time.time()

elapsed\_time  $\leftarrow$  0

**Mientras** : elapsed\_time < 10

current\_time  $\leftarrow$  time.time()

elapsed\_time  $\leftarrow$  current\_time - start\_time

EEG  $\leftarrow$  neuropy.rawValue

eeg\_data.append(EEG)

time.sleep(0.01)

neuropy.stop()

**Crear** : df  $\leftarrow$  pd.DataFrame('EEG': eeg\_data)

**Guardar** : df.to\_csv('.csv', index=False)

**Guardar** : plt.savefig('.png')

---

## 8.2. Anexo B:

### 8.2.1. Nodo Comunicación (*Publisher*)

Código de recepción de ondas EEG.

---

**Algorithm 2:** Adquisición y publicación de datos EEG
 

---

```

Importar : NeuroPy, time, rospy, Float32MultiArray, MultiArrayLayout,
            MultiArrayDimension, os, matplotlib.pyplot as plt, numpy as np
Función : publish_eeg_data()
            rospy.init_node('eeg_publisher')
            pub = rospy.Publisher('eeg_data', Float32MultiArray, queue_size=10)
            eeg_values = []
            Imprimir : "Presiona ENTER para iniciar..."
            Esperar : Tecla
            neuropy.start()
            start_time = time.time()
            elapsed_time = 0
            Mientras : elapsed_time < 6
                current_time = time.time()
                elapsed_time = current_time - start_time
                eeg_value = neuropy.rawValue
                eeg_values.append(eeg_value)
                time.sleep(0.01)
            neuropy.stop()
            eeg_msg
            pub.publish(eeg_msg)
Intentar : publish_eeg_data()
Excepto : rospy.ROSInterruptException:
            Pasar
  
```

---

### 8.2.2. Nodo Clasificación (*Publisher/Suscriber*)

Código de clasificación de parpadeos.

---

**Algorithm 3:** Procesamiento de datos EEG y publicación de parpadeos

---

```

Importar : rospy, Int32, Bool, Float32MultiArray, os, numpy as np,
            tensorflow.keras.models as load_model

Declarar : ruta_modelo, model, publicador_parpadeos
ruta_modelo = os.path.expanduser(' /Descargas/Modelo_Parpadeos128_64_50.h5')
model = load_model(ruta_modelo, compile=False)
publicador_parpadeos = rospy.Publisher('parpadeos', Int32, queue_size=10)

Función : eeg_callback(msg)
    Declarar : input_data, mask, input_grup, predictions, blinks, blink_count,
                total_blinks, in_blink
    input_data = np.array(msg.data)
    mask = ((input_data >= 0) & (input_data <= 400)) | ((input_data >= -400) &
(input_data <= 0))
    input_data = np.where(mask, 0, input_data)
    input_grup = [input_data[i:20+i] para i en rango(len(input_data)-20+1)]
    input_data = np.array(input_grup).reshape(-1, 20, 1)
    predictions = model.predict(input_data)
    blinks = np.argmax(predictions, axis=1) == 1
    blink_count = 0
    total_blinks = 0
    in_blink = Falso

    Para      : blink en blinks
        Si      : blink diferente in_blink
            in_blink = Verdadero
            blink_count = 1
        SinoSi   : blink y in_blink
            blink_count += 1;
        SinoSi   : no blink y in_blink
            in_blink = Falso

    Si      : blink_count ≥ 24
        total_blinks += 1

    Imprimir : Numero total de parpadeos: total_blinks
    publicador_parpadeos.publish(total_blinks)

Función : eeg_subscriber()
    rospy.init_node('eeg_subscriber')
    rospy.Subscriber('eeg_data',Float32MultiArray, eeg_callback)
    rospy.spin()

Si      : __name__ == '__main__'
    Intentar : eeg_subscriber()
    Excepto  : rospy.ROSInterruptException
                Pasar

```

---



### 8.2.3. Nodo de Funcionalidades (*Suscriber*)

Pseudocódigo de programación funcionalidades del robot.

---

**Algorithm 4:** Manejo de datos EEG y ejecución de acciones
 

---

**Importar** : rospy, qi, argparse, sys, almath, math, motion, time, os, ALProxy, Int32,  
Float32MultiArray

session = qi.Session()

session.connect("tcp://192.168.0.103")

animated\_speech\_service = session.service(^LAnimatedSpeech")

behavior\_service = session.service(^LBehaviorManager")

**Función** : eeg\_data\_callback(msg)

**Declarar** : i, motion\_service, joint\_names, handshake\_angles

i = msg.data

**Si** : i == 1

animated\_speech\_service.say(Descripción Proyecto)

**SinoSi** : i == 2

behavior\_service.runBehavio

(animations/Stand/Gestures/Hey\_6")

**SinoSi** : i == 3

motion\_service = session.service(^LMotion");

joint\_names = [RShoulderPitch", RHand", RElbowYaw",  
RElbowRoll"]

handshake\_angles = [0.4, 1.0, 0.0, 0.0]

motion\_service.angleInterpolationWithSpeed(joint\_names,

handshake\_angles, 0.1)

**Función** : eeg\_data\_subscriber()

rospy.init\_node('eeg\_data\_subscriber', anonymous=True)

rospy.Subscriber('parpadeos', Int32, eeg\_data\_callback)

rospy.spin()

**Si** : \_\_name\_\_ == '\_\_main\_\_'

**Intentar** : eeg\_data\_subscriber();

**Excepto** : rospy.ROSInterruptException

Pasar

---

### 8.3. Anexo C:

Señales de clasificación parpadeos voluntarios.

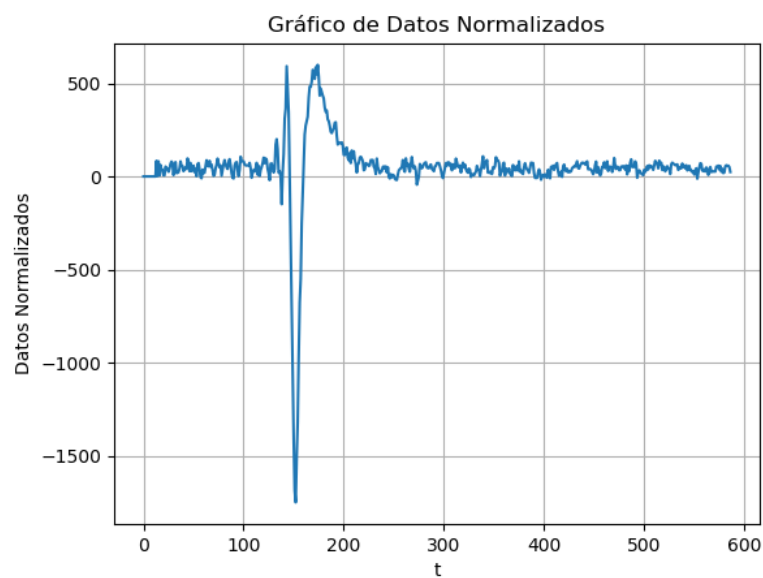


FIGURA 26: 1 Parpadeo.

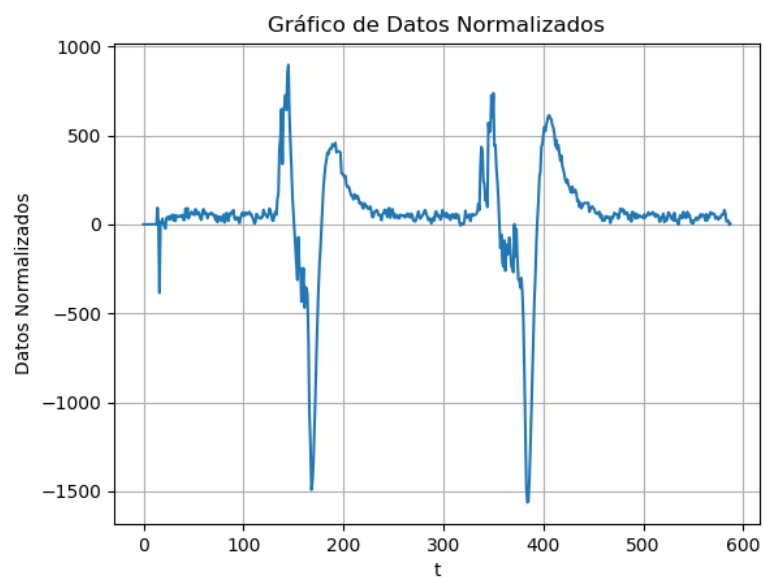


FIGURA 27: 2 Parpadeos.

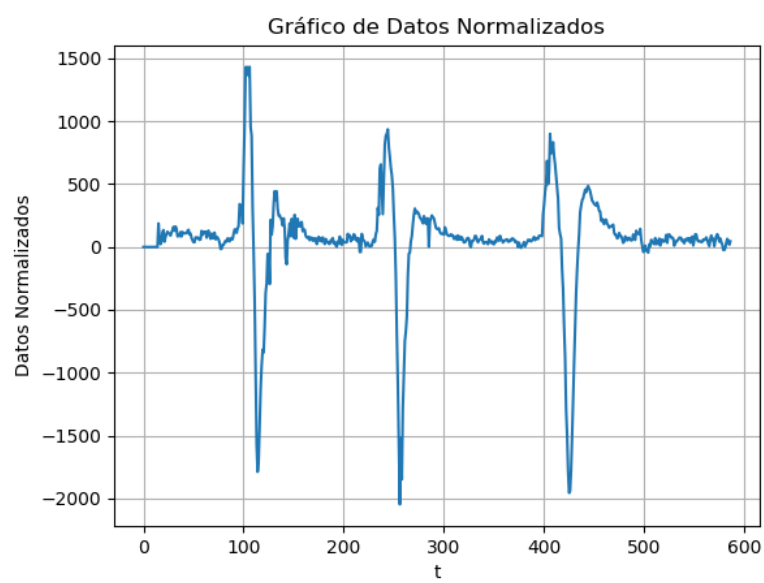


FIGURA 28: 3 Parpadeos.

## 8.4. Anexo D:

Señales adquiridas con ruido.

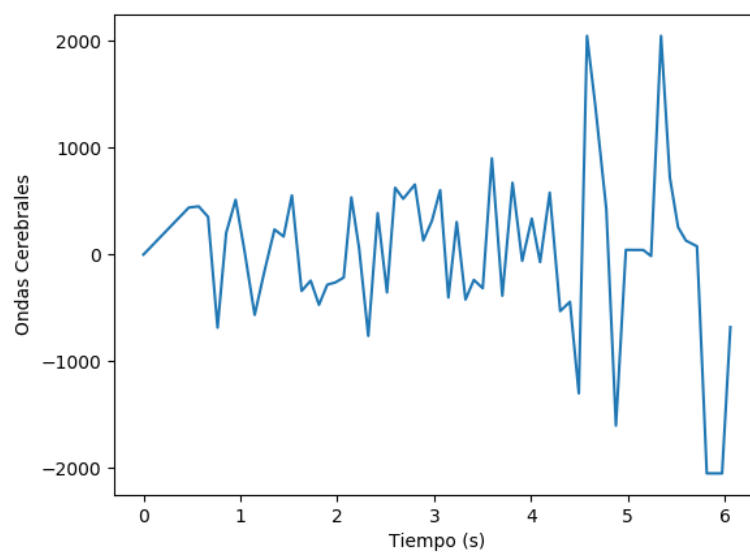


FIGURA 29: Señal con Ruido 1.

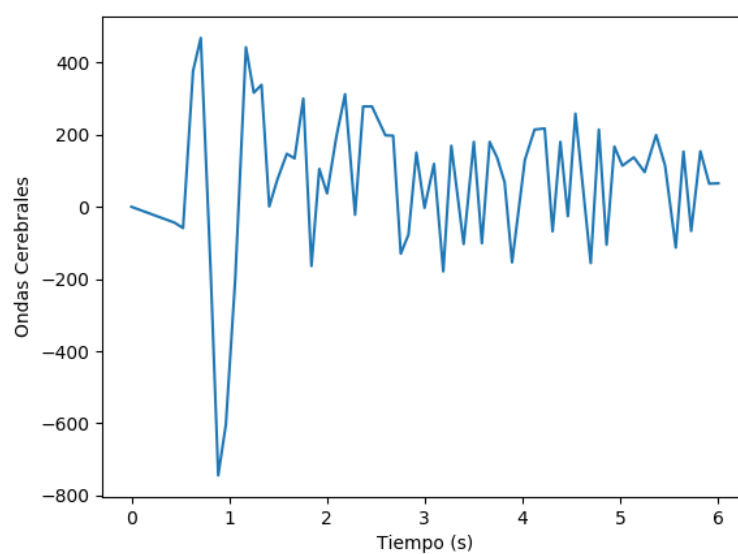


FIGURA 30: Señal con Ruido 2.

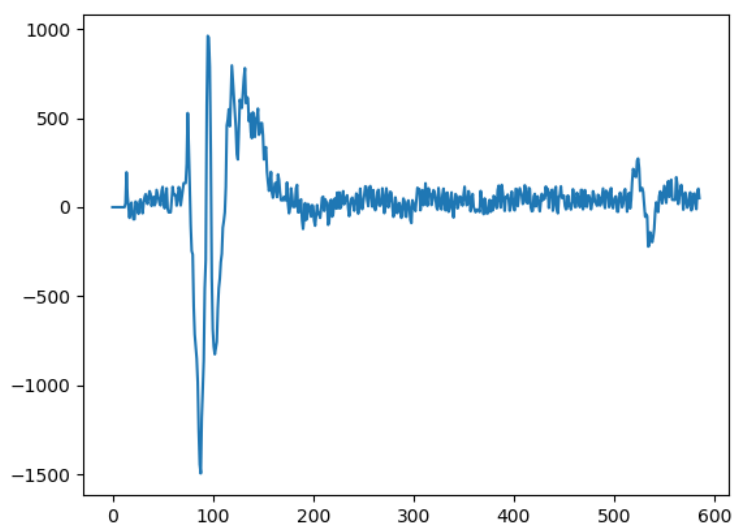


FIGURA 31: Señal con Ruido 3.

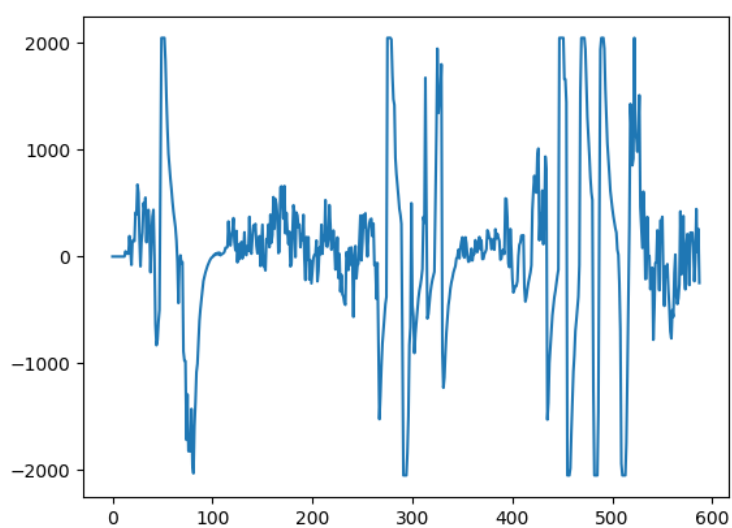


FIGURA 32: Señal con Ruido 4.

## 8.5. Anexo E:

En este anexo se muestran las imágenes del robot *Pepper* y sus diferentes acciones.



FIGURA 33: Movimientos Pepper 1.



FIGURA 34: Movimientos Pepper 2.



FIGURA 35: Movimientos Pepper 3.