# Detección y tratamiento de las señales EEG que permitan definir tareas de un drone

**Ivan Daniel Villegas Isasa**

Trabajo de grado presentado como requisito parcial para optar al título de:
**Ingeniero Electronico**

Director:
Cesar Andrey Charry Perdomo

Línea de Investigación:
Bioingenieria
Grupo de Investigación:
LASER / SEROB

Universidad Distrital Francisco José de Caldas
Facultad de ingeniería, Ingeniería Electrónica
Bogotá, Colombia
2020

# Agradecimientos

Ha sido mucha la ayuda que he recibido durante el desarrollo de este, mi proyecto de grado; por eso considero que lo mínimo que puedo hacer para mostrar mi agradecimiento a los involucrados es darles un eterno reconocimiento en este documento.

Primero agradezco a Dios, me ha permitido llegar hasta este punto y me ha llenado de fuerza y fe en momentos muy difíciles a lo largo de toda la carrera. No menos importante, a mis padres, Jorge Iván Villegas y Dila Angélica Isasa, quienes me han acompañado y animado a lo largo de toda mi vida, poseedores de paciencia, son testigos de todos los cambios que se realizaron en este documento, ayudaron en su actualización y complemento de cada sugerencia, lo han vuelto a leer, al igual que mis hermanas, a quienes también les expreso mi cariño y respeto por su dedicación y compromiso conmigo.

César Perdomo, mi director de tesis, profesor y codirector del grupo de investigación al que pertenezco (LASER), quien confió en mí, para hacer realidad este proyecto y me brindó todas las herramientas necesarias para desarrollarlo junto a su hermano Oscar Perdomo, por lo que se constituyeron en mi guía y sin su asesoría y apoyo, no había sido posible la construcción del mismo. Razones por las cuales, les manifiesto, sincero e importante reconocimiento.

Fabián Barreto, Nicolás Daza y Cristian Sánchez, con quienes inicié el proyecto y ahora lo terminan junto conmigo, les agradezco mucho su amistad y la ayuda que me brindaron en muchas ocasiones cuando me encontraba sin ninguna idea de cómo avanzar. De igual manera al resto de integrantes del grupo LASER quienes han estado dispuestos a ayudarme, dentro de los que resalto al profesor Jutinico con sus muy gratas realimentaciones y Jorge Lozano con un gran apoyo a lo largo no solo de este proyecto, sino a lo largo de toda la carrera junto con Cristian y Nicolás.

Fueron fundamentales, todos mis amigos y familiares que se sometieron a la tortura de ponerse el casco y grabar los gestos para poder levantar la base de datos, los cuales me permito nombrar a continuación: Catalina Cruz, David Medina, David Socha, Jonatán Ricaurte, Mónica Palencia, David Salazar, José Perea, Juan Amaya, Mario Villegas, Johanan Villegas, Sara Villegas y Valentina Villegas.

Por último, quiero reconocer la ayuda obtenida de Carlos Marín, cuando surgieron los problemas con la conexión del drone y a Juan Carlos Polo, quien en un inicio me ayudó a dilucidar este proyecto.

# Contenido

**Agradecimientos** III

**Lista de símbolos** VII

**1. Introducción** 2

**2. Objetivos** 3
2.1. Objetivo Principal . . . 3
2.2. Objetivos secundarios . . . 3

**3. ¿Donde ubicar los electrodos?** 4
3.1. Herramientas . . . 4
3.1.1. Electroencefalografía . . . 4
3.1.2. Python . . . 5
3.2. Posicionamiento de los electrodos . . . 5

**4. Análisis exploratorio de los datos EEG** 14
4.1. Filtrado de los datos . . . 14
4.2. Preprocesamiento de los datos . . . 16

**5. Desarrollo del modelo de inteligencia computacional** 19
5.1. Definición . . . 19
5.2. Levantamiento de datos . . . 19
5.2.1. Selección de gestos . . . 19
5.2.2. Toma de señales . . . 21
5.3. Indicador de éxito y protocolo de evaluación . . . 21
5.4. Preparación de los datos . . . 21
5.4.1. Datos categóricos . . . 22
5.5. Desarrollo del modelo base . . . 27
5.6. Desarrollo de un modelo mejor y más ajustado . . . 30
5.6.1. Modelo para seis gestos e inactividad . . . 31
5.6.2. Modelo para cuatro gestos e inactividad . . . 34
5.6.3. Modelo para tres gestos e inactividad . . . 34

**6. Manejo del Drone** **40**
6.1. Parrot Drone Mambo . . . . . 40
6.2. Algoritmo diseñado para el manejo . . . . . 40
6.3. Implementación del algoritmo . . . . . 43
6.4. Ejecución del GUI . . . . . 43
6.4.1. Subproceso en Windows . . . . . 45
6.4.2. Ejecución en Linux . . . . . 45

**7. Análisis de resultados** **47**

**8. Conclusiones** **51**

**A. Anexo: Base de datos usada para el entrenamiento de los clasificadores.** **53**

**B. Anexo: Gráficas de precisión y perdida, además de las matrices de confusión de las pruebas realizadas con las redes CNN** **54**
B.1. Gestos en la matriz de confusión . . . . . 54
B.2. Modelo para seis gestos e inactividad . . . . . 54
B.3. Modelo para cuatro gestos e inactividad . . . . . 55
B.4. Modelo para tres gestos e inactividad . . . . . 56

**C. Anexo: Pruebas realizadas para determinar la información suministrada por cada par de electrodos** **60**

**D. Anexo: Vídeo evidencia del manejo del drone con señales EEG** **62**

**Bibliografía** **63**

# Lista de símbolos

## Superíndices

| Superíndice | Término |
|---|---|
| F | Frontal |
| Fc | Frontral central |
| C | Central |
| Cp | Centro parietal |
| P | Parietal |
| Po | Parietal occipital |
| O | Occipital |
| T | Temporal |
| Nz | Nasion |
| Iz | Inion |

## Abreviaturas

| Abreviatura | Término |
|---|---|
| $EEG$ | Electroencefalográficas |
| $EMG$ | Electromiográficas |
| $BCI$ | Interfaz Cerebro Computador (Brain to Computer Interface) |
| $GUI$ | Interfaz Gráfica de Usuario (Grafic User Interface) |
| $KNN$ | k vecinos más próximos (K-Nearest Neighbor) |
| $RF$ | Bosques aleatorios (Random Forest) |
| $CNN$ | Redes neuronales convolucionales (Convolutional Neural Network) |

## Términos

| Abreviatura | Término |
|---|---|
| Precision | Métrica para evaluar un clasificador que compara la cantidad de verdaderos positivos con la cantidad de falsos positivos. |
| Recall | Métrica para evaluar un clasificador que compara la cantidad |

| Abreviatura | Término |
|---|---|
| | de verdaderos positivos con la cantidad de falsos negativos. |
| F1-score | Métrica para evaluar un clasificador que promedia las métricas precision y recall. |
| Overall Accuracy | Métrica para evaluar un clasificador que promedia las métricas precision y recall teniendo en cuenta la cantidad de datos utilizados para evaluar en cada clase. |
| Exactitud | Cantidad de datos clasificados de manera correcta. |
| Perdida | Distancia entre el dato clasificado y el dato real. |

# Lista de Figuras

**3-1**. Distribución de electrodos en el Ultracortex Mark IV, la imagen es tomada de [OpenBCI, 2020]. . . . . 6
**3-2**. Lóbulos en el encéfalo. Tomada de [Purves, 2007]. . . . . 6
**3-3**. Corteza motora. Tomada de [Álvaro González Velasco, 2012]. . . . . 8
**3-4**. Disposición de electrodos sugeridas, donde los círculos sin relleno de color corresponden a puntos de la configuración 10/20 que no pueden ser usadas debido al diseño del ultracortex, los círculos que tienen un color crema corresponde a los sitios donde el ultracortex deja ubicar los electrodos y los círculos rojos corresponden a la ubicación de electrodos sugerida; la imagen (a) pertenece a la distribución por defecto recomendada por el fabricante, mientras que la imagen (b) es la recomendada para este proyecto en particular teniendo como referencia la zonas cerebrales relacionadas con el movimiento. . . . . 9
**3-5**. Señales capturadas con la disposición de electrodos sugerida por OpenBCI filtradas para que se observa la banda $\alpha$ (8-13 Hz), con el eje $y$ en micro voltios y el eje $x$ en número de muestra . . . . . 11
**3-6**. Señales capturadas con la disposición sugerida para el proyecto con base en la teoría filtradas para que se observa la banda $\alpha$ (8-13 Hz), con el eje $y$ en micro voltios y el eje $x$ en número de muestra . . . . . 12
**3-7**. Comparación de las señales obtenidas con los electrodos que se cambiaron de posición, donde la imagen (a) corresponde a los canales 5, 6, 7 y 8 de la configuración sugerida por OpenBCI, mientras que la imagen (b) corresponde a los mismos canales pero con la configuración extraída del estudio teórico. La posición de los canales según el estándar 10/20 se muestra en la tabla **3-1**. 13

**4-1**. Señal EEG del levantamiento de la mano derecha tomada por el electrodo ubicado en C3 donde la gráfica superior es la señal captada sin filtrar seguida por las señales filtradas en orden de menor rango de frecuencias a mayor rango de frecuencias . . . . . 15
**4-2**. Señal EEG filtrada en la banda $\alpha$ perteneciente al levantamiento de un brazo captada por el electrodo 3 (C3). La imagen (a) es la señal filtrada que no se ha sometido a ningún procesamiento. La imagen (b) se obtiene tras estandarizar la señal de (a). La imagen (c) se obtiene tras escalar a un rango específico la señal de (a). Por ultimo la imagen (d) se obtiene tras Normalizar la señal de (a) 18

**5-1**. Homúnculo motor de Penfield. Imagen tomada de [Oscar, 2019] . . . . . 20
**5-2**. Características en el tiempo de la señal EEG tomada por el electrodo 1 perteneciente a cinco parpadeos del ojo derecho . . . . . 23
**5-3**. Parámetros de Hjorth de la señal EEG tomada por el electrodo 1 perteneciente a cinco parpadeos del ojo derecho . . . . . 24
**5-4**. Gráficas de la media y varianza de la señales captadas para entrenar el modelo con los diferentes pre-procesamientos propuestos. La imagen (a) son las características de la señal filtrada que no se ha sometido a ningún procesamiento. La imagen (b) muestra las características tras estandarizar la señal de (a). La imagen (c) se obtiene de las características tras escalar a un rango especifico la señal de (a). Por último la imagen (d) presenta las características tras Normalizar la señal de (a) . . . . . 25
**5-5**. Gráficas de los parámetros Hjorth de la señales captadas para entrenar el modelo con los diferentes pre-procesamientos propuestos. La imagen (a) son las parámetros de la señal filtrada que no se ha sometido a ningún procesamiento. La imagen (b) muestra los parámetros tras estandarizar la señal de (a). La imagen (c) se obtiene de las parámetros tras escalar a un rango especifico la señal de (a). Por último la imagen (d) presenta las características tras Normalizar la señal de (a) . . . . . 26
**5-6**. Resultados de las pruebas con las redes Base con siete gestos además de la inactividad . . . . . 28
**5-7**. Resultados de las pruebas con las redes Base con cuatro gestos además de la inactividad . . . . . 28
**5-8**. Resultados de las pruebas con las redes Base con tres gestos además de la inactividad . . . . . 29
**5-9**. Resultados de las pruebas con las redes Base con tres gestos además de la inactividad con datos diferentes . . . . . 30
**5-10**.Mapas de calor que muestran las pruebas realizadas para determinar los valores de la topología de la red con mejores resultados en la exactitud . . . . 32
**5-11**.Red CNN implementada . . . . . 33
**5-12**.Resultados de las pruebas de validación con la red CNN con seis gestos además de la inactividad . . . . . 34
**5-13**.Resultados de las pruebas de validación con la red CNN con cuatro gestos además de la inactividad . . . . . 35
**5-14**.Resultados de las pruebas con la red CNN con tres gestos además de la inactividad . . . . . 36
**5-15**.Resultados de las pruebas con la red CNN con tres gestos además de la inactividad usando datos nunca vistos por el clasificador . . . . . 36

**5-16**.Gráficas de los clasificadores CNN únicamente con el filtrado en el ritmo $\alpha$, en la imagen (a) se evidencia la exactitud alcanzada superior al 85 % mientras que en (b) se muestra los resultados de la pérdida. El resultado se obtiene al entrenar con el gesto de la mordida, sin embargo, al entrenar con el gesto de las cejas el resultado es bastante similar. . . . . 37

**5-17**.Matrices de confusión de los clasificadores CNN únicamente con el filtrado en el ritmo $\alpha$, donde de arriba hacia abajo y de izquierda a derecha la primera fila y columna pertenece al parpadeo con el ojo izquierdo, la segunda fila y columna es el parpadeo con el ojo derecho y la cuarta fila y columna es la inactividad y donde el gesto 3 se cambia, en la imagen (a) este gesto pertenece a el movimiento de las cejas, mientras que en (b) el gesto pertenece a la mordida 39

**6-1**. Inicio del algoritmo . . . . . 41

**6-2**. Algoritmo del interruptor . . . . . 41

**6-3**. Algoritmo de dirección . . . . . 42

**6-4**. Interfaz gráfica de usuario para el control del drone con señales EEG . . . . 44

**7-1**. Comparación de las matrices de confusión con el conjunto de pesos 1 obtenidos. La imagen (a) pertenece a la evaluación utilizando los datos de test de la base de datos, mientras que (b) se obtiene de pruebas en tiempo real. . . . 48

**7-2**. Comparación de las matrices de confusión con el conjunto de pesos 2 obtenidos. La imagen (a) pertenece a la evaluación utilizando los datos de test de la base de datos, mientras que (b) se obtiene de pruebas en tiempo real. . . . 49

**B-1**. Resultados del entrenamiento de la red CNN con seis gestos e inactividad sin pre-procesamiento, únicamente con el filtrado en el ritmo $\alpha$ . . . . . 55

**B-2**. Resultados del entrenamiento de la red CNN con seis gestos e inactividad con los datos escalados . . . . . 55

**B-3**. Resultados del entrenamiento de la red CNN con seis gestos e inactividad con los datos escalados entre valores fijos 0 y 1 . . . . . 55

**B-4**. Resultados del entrenamiento de la red CNN con seis gestos e inactividad con los datos Normalizados . . . . . 56

**B-5**. Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad sin pre-procesamiento, únicamente con el filtrado en el ritmo $\alpha$ . . . . . 56

**B-6**. Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad con los datos escalados . . . . . 57

**B-7**. Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad con los datos escalados entre valores fijos 0 y 1 . . . . . 57

**B-8**. Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad con los datos Normalizados . . . . . 57

**B-9**. Resultados del entrenamiento de la red CNN con tres gestos e inactividad sin pre-procesamiento, únicamente con el filtrado en el ritmo $\alpha$ . . . . . 58
**B-10**Resultados del entrenamiento de la red CNN con tres gestos e inactividad con los datos escalados . . . . . 58
**B-11**Resultados del entrenamiento de la red CNN con tres gestos e inactividad con los datos escalados entre valores fijos 0 y 1 . . . . . 58
**B-12**Resultados del entrenamiento de la red CNN con tres gestos e inactividad con los datos Normalizados . . . . . 59

**D-1**. Captura del vídeo evidencia al que dirige el link. . . . . 62

# Lista de Tablas

**3-1**. Posición de los electrodos . . . . . 10

**7-1**. Métricas obtenidas de las matrices de confusión de las figuras **7-1** y **7-2**, donde se comparan los valores de las pruebas realizada con grabaciones de la base de datos con las pruebas realizadas en tiempo real obtenidos con dos diferentes conjuntos de pesos. . . . . 49

**7-2**. Promedio y desviación estándar de los datos mostradas en la tabla **C-1** obtenidos para determinar la importancia de cada par de electrodos en el modelo de clasificador organizados en orden de importancia según la información que brindan. . . . . 50

**B-1**. Distribución de las clases en las matrices de confusión . . . . . 54

**C-1**. Resultados de las pruebas realizadas para obtener el par de electrodos que brinda más información. . . . . 61

# 1. Introducción

Según la Organización Mundial de la Salud (OMS) más del 15 % de la población sufre de algún tipo de discapacidad [OMS, 2011] y en Colombia, según el último registro del DANE (Departamento Administrativo Nacional de Estadísticas) más de dos millones de personas presentan alguna característica que dificulta el desarrollo de actividades cotidianas [DANE, 2010]. Una actividad cada vez más común, es el uso de Vehículos aéreos no tripulados conocidos como drones, los cuales en algunos países incluso llegan a necesitar de certificados para poder ser utilizados. Con el ánimo de facilitar el acceso a esta tecnología a personas con dificultades motoras se propone utilizar las señales electroencefalográficas, para dar ordenes a un drone y así facilitar su manejo para estas personas.

La electroencefalografía es una técnica de monitoreo cerebral, la cual recibe señales de la corteza del encéfalo a través de electrodos ubicados sobre el cuero cabelludo. Para que una persona llegue a controlar un objeto mediante su actividad cerebral se requiere de un sistema de comunicación que monitoriza la actividad cerebral y es capaz de traducirla para determinar las intenciones del usuario. A este sistema se le denomina Brain-Computer Interface (BCI), o interfaz cerebro-ordenador y se puede clasificar en dos grandes grupos dependiendo de la naturaleza de la señal de entrada, para este trabajo se utiliza un sistema BCI endógeno el cual depende de la capacidad del usuario para controlar su actividad electrofisiológica como por ejemplo lo son los sistemas BCI basados en imágenes motoras (ritmos sensoriomotores) o en potenciales corticales lentos (Slow Cortical Potentials, SCP).[ROBERTO HORNERO, 2012].

Con el actual desarrollo de la tecnología EEG y el crecimiento de los Drones, el presente trabajo se orienta a la detección y el tratamiento de las señales electroencefalográficas (EEG) obtenidas por interfaces de cerebro a computador (BCI), que permitan obtener una serie de comandos, que sirvan para enviar órdenes a un drone.

# 2. Objetivos

## 2.1. Objetivo Principal

Caracterizar señales electroencefalográficas, a través del uso de un dispositivo ultracortex IV con una tarjeta enlazada a computador entrenado para responder a diferentes comandos cerebrales, que permitan dar órdenes a un drone.

## 2.2. Objetivos secundarios

- Determinar las regiones que permita extraer la suficiente variedad de diferentes comandos de manera correcta haciendo uso del dispositivo ultracortex.
- Procesar las señales electroencefalográficas para determinar una serie de acciones viables en el control del drone.
- Identificar un paradigma de inteligencia computacional adecuado para clasificación de las señales electroencefalográficas.
- Caracterizar, implementar y probar un sistema que permita operar el drone, a través de señales electroencefalográficas capturadas por el ultracortex.

# 3. ¿Donde ubicar los electrodos?

Primero es necesario conocer las herramientas utilizadas para el desarrollo del proyecto, las cuales se muestran a continuación:

## 3.1. Herramientas

### 3.1.1. Electroencefalografía

La electroencefalografía es una técnica de registro de la actividad cerebral usando el cuero cabelludo para ubicar electrodos en posiciones estándar sin necesidad de invadir el cerebelo atravesando el cráneo. Dentro sus principales limitaciones se encuentra la baja resolución espacial para determinar las regiones activas. [Purves, 2007]. En el presente trabajo los dispositivos usados para conseguir el registro de las señales EEG (electroencefalográficas) son los siguientes:

#### Cyton Board

La Cyton board es una tarjeta, la que se encarga de muestrear, amplificar, y enviar la información recopilada por los electrodos. La placa necesita de tres a seis voltios de batería para funcionar, tiene un microcontrolador *PIC32MX250F128B* con un chipKIT *PIC32MX250F128B* para el arranque. EL amplificador operacional es el *ADS1299*, tiene un acelerómetro *LIS3DH* de tres ejes además de un RFduino BLE para comunicarse de manera inalámbrica con el USB Dongle el cual se conecta al computador por serial *IC (FT231XQ-R)*.La tarjeta cuenta con ocho canales de alta ganancia y bajo ruido donde cada uno muestrea a 250Hz y una resolución de 32 bits. [OpenBCI, 2020].

#### Deisy Board

El modulo Deisy permite ampliar los canales de 8 a 16, cuenta también con un amplificador *ADS1299*, se alimenta al conectarse a la tarjeta Cyton obteniendo también la señal de reloj de esta misma. AL conectar la Deisy a la tarjeta Cyton la frecuencia de muestreo por canal baja a 125Hz. Es necesario aclarar que estos no son dispositivos diseñados para un uso de diagnóstico médico, por lo que no hay una garantía que estos funcionen de manera correcta debido a que se pueden haber pérdidas de información o daños en el dispositivo.[OpenBCI, 2020].

#### Ultracortex Mark IV

Ultracortex es un set de cabeza de fuente abierta para impresión en 3D diseñado para funcionar con los sistemas OpenBCI. Es una herramienta para la adquisición de información de la actividad cerebral(EEG) con fines de investigación. El casco permite posicionar electrodos en: Fp1, Fpz, Fp2, AF3, AFz, AF4, F7, F3, Fz, F4, F8, FC5, FC1, FC2, FC6, T3, C3, Cz, C4, T4, CP5,CP1, CP2, CP6, T5, P3, Pz, P4, P6, PO3, POz, PO4, O1, Oz y O2. Esta distribución se muestra en la figura **3-1**. [OpenBCI, 2020].

#### Electrodos

Los electrodos usados, son electrodos secos con un diámetro máximo de 1/4 de pulgada y protuberancias de 5mm de largo los usados en la cabeza, mientras que los usados en la frente y orejas son planos; los electrodos están hechos a base de plata y cloruro de plata (Ag-AgCl) y tienen una duración media de aproximadamente 20 usos.[OpenBCI, 2020].

### 3.1.2. Python

Python es un lenguaje de programación multiparadigma ya que soporta orientación a objetos, programación imperativa además de la programación funcional, aunque esta ultima en menor medida. Entre las aplicaciones de python se encuentran los desarrollos web, la ciencia y la matemática, el desarrollo de interfaces gráficas de usuario para escritorio, el desarrollo de software y las aplicaciones de negocios. La razón por la que se utiliza en este proyecto es porque cuenta con las librerías: Scipy para matemáticas e ingeniería, Pandas para el análisis de datos, Keras para la implementación de redes neuronales y deep learning, pyopenbci para la conexión con la tarjeta Cyton y pyparrot para el control del drone. [Python, 2020].

## 3.2. Posicionamiento de los electrodos

El encéfalo es el órgano del cuerpo humano en el cual el presente documento centra su atención, siendo un órgano tan complejo, este se puede dividir en varias secciones. La primera división y la más evidente debido al surco longitudinal que separa las dos masas más voluminosas del sistema nervioso central denominadas como hemisferios izquierdo y derecho. A su vez cada uno de estos hemisferios se puede dividir en cuatro lóbulos: el frontal que se ubica en la parte delantera al surco central y por encima del surco lateral , el surco temporal que se ubica entre el surco central y el surco parietoccipital, el lóbulo temporal ubicado bajo el surco lateral y el lóbulo occipital situado debajo del surco parietoccipital y por detrás del lóbulo temporal. La figura **3-2**, muestra de manera más clara de división de estos en el encéfalo.

Los hemisferios cerebrales están formados por tres capas dependiendo de su profundidad [Universidad a Distancia de Madrid, 201],en la capa interior se encuentran las estructuras

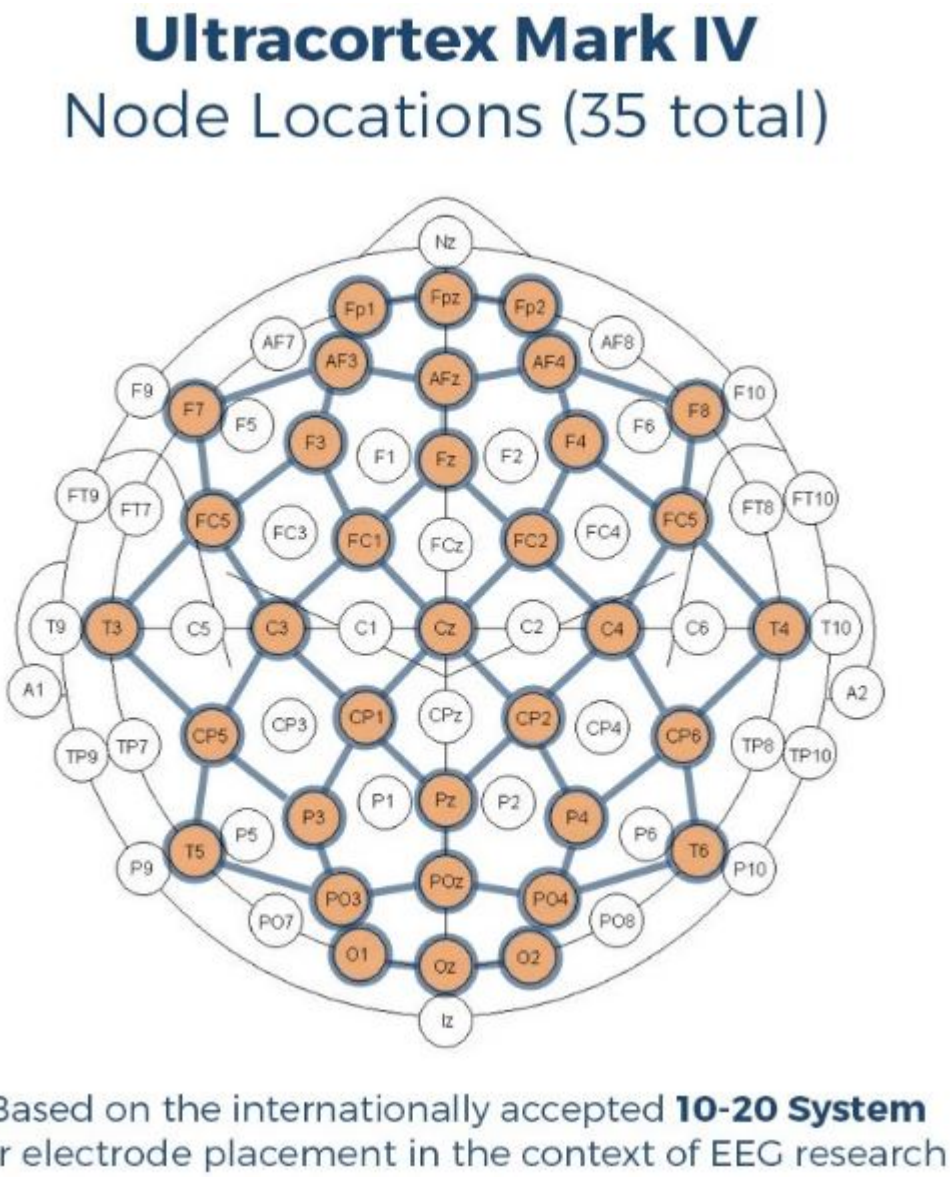



**Figura 3-1**.: Distribución de electrodos en el Ultracortex Mark IV, la imagen es tomada de [OpenBCI, 2020].

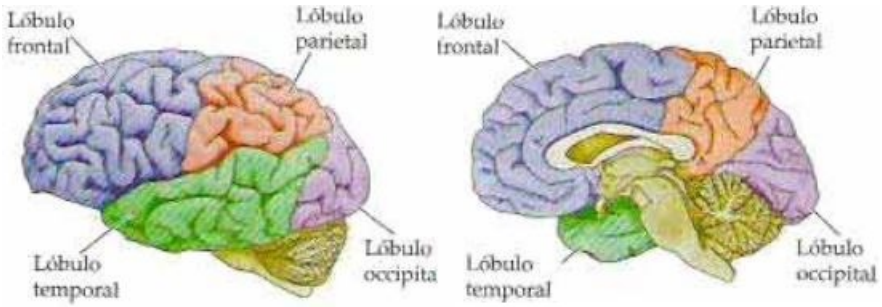



**Figura 3-2**.: Lóbulos en el encéfalo. Tomada de [Purves, 2007].

subcorticales, la capa intermedia se compone por la sustancia blanca, la cual son los axones de las neuronas que se encuentran en la corteza cerebral y capa exterior compuesta por materia gris. Para el presente estudio es de interés sobre todo lo ocurrido en la corteza cerebral, debido a que el dispositivo usado para capturar las señales no se adentra en lo profundo del encéfalo, ni siquiera atraviesa el cráneo, por lo que las señales que se van a recibir provienen de la región más superficial del encéfalo.

Se decide controlar el drone con señales que sean generadas a voluntad del usuario, por ende, las señales generadas por el movimiento voluntario son adecuadas para el desarrollo del proyecto. Teniendo en cuenta lo anterior, es necesario identificar la región del cerebro donde se detecten las señales producidas por el movimiento voluntario; Según [Álvaro González Velasco, 2012] la corteza motora se divide en tres partes, la corteza motora primaria, área motora suplementaria y corteza premotora. La corteza motora primaria está situada justo delante del surco central y se expande de manera paralela a este. "*El área motora suplementaria es una región de la corteza motora de asociación situada en las zonas dorsal y dorsomedial del lóbulo lateral*"[Álvaro González Velasco, 2012]. "*La corteza premotora está situada inmediatamente delante de la corteza primaria motora y es una región de la corteza motora de asociación situada en la zona lateral del lóbulo frontal.*"[Álvaro González Velasco, 2012]. Estas zonas se aprecian mejor en la figura **3-3**.

Identificadas las zonas donde tomar las señales, lo siguiente es tomar señales para comprobar lo que la teoría dice. De acuerdo a lo anterior, se utiliza el estándar adoptado por la federación en electroencefalografía y neurofisiología clínica llamado sistema 10/20 para colocación de electrodos; en cuanto a la ubicación de los electrodos, el sistema 10/20 divide la cabeza en distancias proporcionales teniendo como referencias partes prominentes del cráneo. La cabeza se divide en dos, siendo la mitad el punto preauricular y denominado C (Central), cada 20 % adicional hacia adelante o hacia atrás se ubica un electrodo, los más alejados del punto central están alejados un 10 % del Nasion y el Inion, la ubicación de los electrodos reciben el nombre de la región en la que se encuentran F (frontal), T (temporal), P (posterior), y O (occipital) acompañado de un número que cumple con la misma proporción que las letras, con los mismos puntos de referencia, pero indicando que tan alejado se encuentra el electrodo del surco longitudinal. Los números también indican el hemisferio en el que se encuentran, los pares el hemisferio derecho y los impares el hemisferio izquierdo. [Bratislava, 2002].

Como el dispositivo usado para tomar las medidas tiene capacidad para obtener información hasta de 16 electrodos, se sugiere ubicar estos de manera que se pueda extraer la mayor cantidad de información útil para el objetivo del proyecto. Por este motivo y con base en la información previamente suministrada, se presenta una configuración de electrodos diferente a la presentada por la empresa proveedora del equipo de electroencefalografía mostrada en la figura **3-4**, la cual sugiere distribuir electrodos abarcando la mayor cantidad de partes

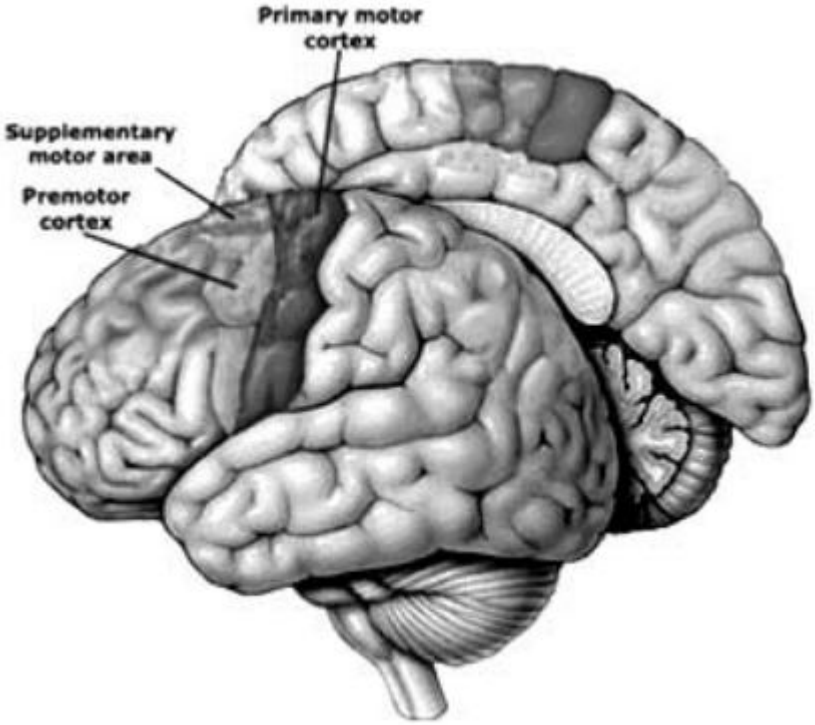
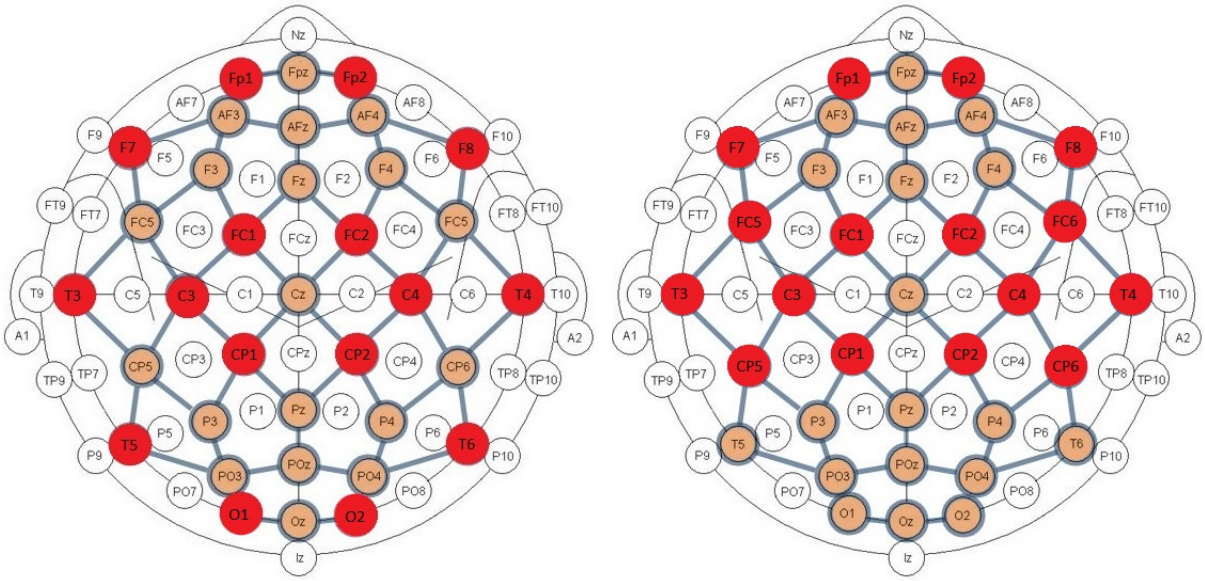



**Figura 3-3**.: Corteza motora. Tomada de [Álvaro González Velasco, 2012].

de la corteza cerebral, con el fin de hacer observaciones generales. En este documento se propone mover los electrodos ubicados en el lóbulo occipital, hacia una posición que tenga una mayor cercanía con las regiones motoras como muestra la figura **3-4**; también se dejan una línea de electrodos detrás del surco central, debido a que los electrodos al estar en la superficie del cuero cabelludo y no dentro del encéfalo capturan un promedio de señales de puntos cercanos a su ubicación, más no la actividad exacta de la zona sobre la que se ubican.

Debido a que la tarjeta Cyton y Deisy manejan cada uno de los canales en un determinado orden, en el que posteriormente son guardados y mostrados en el ordenador, la tabla **3-1** muestra el número de canal/electrodo y su respectiva ubicación según el estándar 10/20.

Con el objetivo de comparar los cambios se tomaron diferentes muestras de señales EEG realizando una acción motora y se observó la actividad cerebral en el ritmo $\alpha$ (8-13Hz) en cada una de las configuraciones presentadas anteriormente. Las figuras **3-5** y **3-6** evidencian una muestra de señales con la configuración del fabricante y con la configuración sugerida en el proyecto respectivamente, obtenida al levantar el brazo derecho de una misma persona. Si bien a simple vista se observan bastantes diferencias incluso entre los electrodos que no se cambiaron de posición, estos cambios pueden deberse al cambio de la resistencia entre los electrodos y la superficie del cuero cabelludo.




(a) Configuración del fabricante

(b) Configuración sugerida para el proyecto

**Figura 3-4.**: Disposición de electrodos sugeridas, donde los círculos sin relleno de color corresponden a puntos de la configuración 10/20 que no pueden ser usadas debido al diseño del ultracortex, los círculos que tienen un color crema corresponde a los sitios donde el ultracortex deja ubicar los electrodos y los círculos rojos corresponden a la ubicación de electrodos sugerida; la imagen (a) pertenece a la distribución por defecto recomendada por el fabricante, mientras que la imagen (b) es la recomendada para este proyecto en particular teniendo como referencia la zonas cerebrales relacionadas con el movimiento.

| Electrodo | Posición OpenBCI | Electrodo | Posición Sugerida |
|---|---|---|---|
| 1 | Fp1 | 1 | Fp1 |
| 2 | Fp2 | 2 | Fp2 |
| 3 | C3 | 3 | C3 |
| 4 | C4 | 4 | C4 |
| 5 | T5 | 5 | Cp5 |
| 6 | T6 | 6 | Cp6 |
| 7 | O1 | 7 | Fc5 |
| 8 | O2 | 8 | Fc6 |
| 9 | F7 | 9 | F7 |
| 10 | F8 | 10 | F8 |
| 11 | Fc1 | 11 | Fc1 |
| 12 | Fc2 | 12 | Fc2 |
| 13 | T3 | 13 | T3 |
| 14 | T4 | 14 | T4 |
| 15 | Cp1 | 15 | Cp1 |
| 16 | Cp2 | 16 | Cp2 |

**Tabla 3-1**.: Posición de los electrodos

Como no todos los canales se cambiaron de posición, no es necesario comparar la actividad cerebral sino en cuatro de ellos, los canales 5, 6, 7 y 8. La figura **3-7**, extrae de las figuras **3-5** y **3-6** los canales que se cambiaron de lugar y permite comparar de manera más simple las señales obtenidas por los electrodos correspondientes.

Al comparar las señales en la figura **3-7**, lo primero que se observa es una clara diferencia en la actividad cerebral, con una señal más homogénea en la configuración sugerida por OpenBCI lo cual indica una falta de actividad cerebral relevante en la zona, contrastando con los electrodos 5 y 6 de la configuración elegida para el desarrollo del proyecto, en las cuales es evidente el momento en el que se desarrolla la acción debido a un incremento en la actividad cerebral mostrada por el incremento en la amplitud. En el caso de los electrodos 7 y 8, aunque para este caso no se evidencia un incremento de la amplitud en intervalos de tiempo específicos tan notorio, las señales en general tienen una amplitud con mayor relevancia frente a su contra parte en la configuración por defecto.

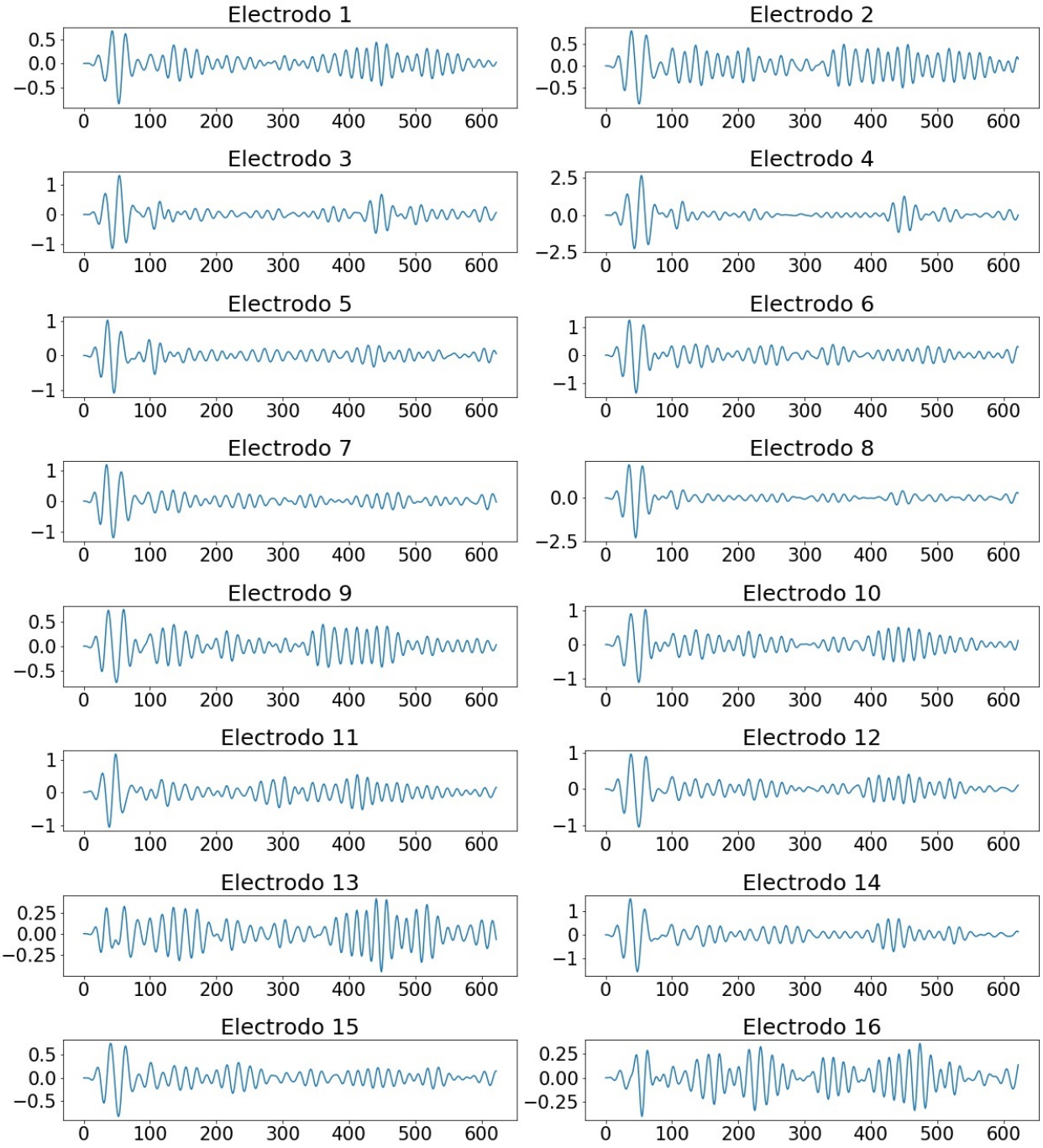



**Figura 3-5**.: Señales capturadas con la disposición de electrodos sugerida por OpenBCI filtradas para que se observa la banda $\alpha$ (8-13 Hz), con el eje $y$ en micro voltios y el eje $x$ en número de muestra

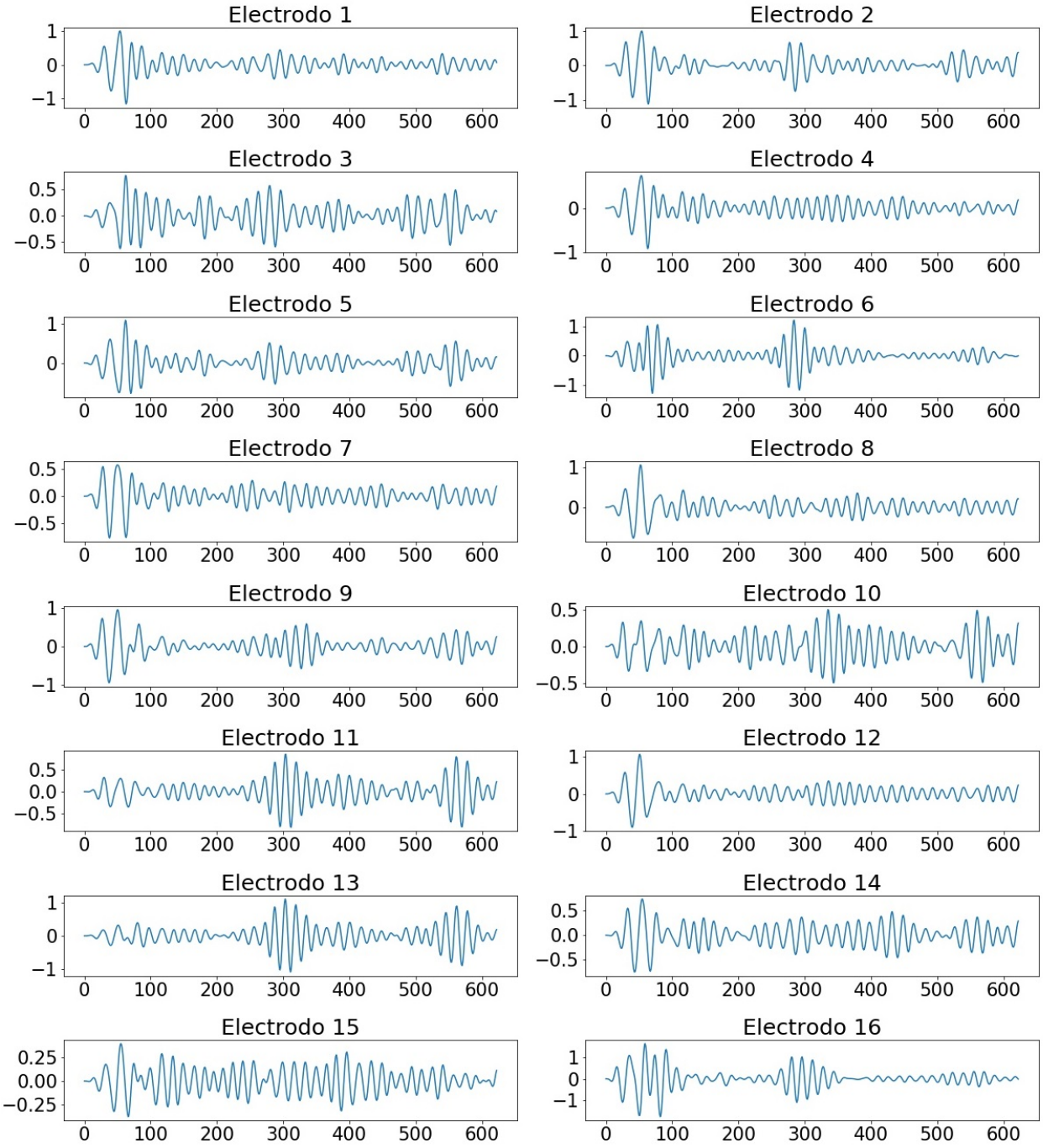
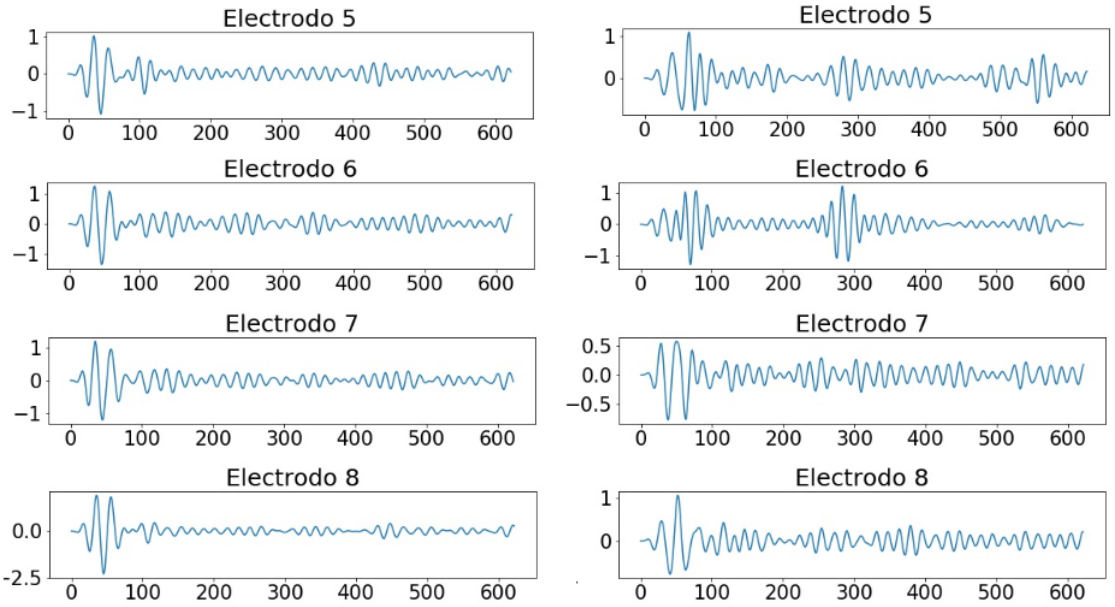



**Figura 3-6**.: Señales capturadas con la disposición sugerida para el proyecto con base en la teoría filtradas para que se observa la banda $\alpha$ (8-13 Hz), con el eje $y$ en micro voltios y el eje $x$ en número de muestra




(a) Señales con la configuración del fabricante (b) Señales con la configuración sugerida para el proyecto

**Figura 3-7**.: Comparación de las señales obtenidas con los electrodos que se cambiaron de posición, donde la imagen (a) corresponde a los canales 5, 6, 7 y 8 de la configuración sugerida por OpenBCI, mientras que la imagen (b) corresponde a los mismos canales pero con la configuración extraída del estudio teórico. La posición de los canales según el estándar 10/20 se muestra en la tabla **3-1**.

# 4. Análisis exploratorio de los datos EEG

## 4.1. Filtrado de los datos

En los adultos, se encuentra que las señales EEG pueden ser divididas en cinco ritmos diferentes distribuidos entre distintos rangos de frecuencias. Estas bandas nombradas desde las menores frecuencias a las mayores son: delta ($\delta$), theta ($\theta$), alpha ($\alpha$), beta ($\beta$), y gamma($\gamma$). [Saeid Sanei, 2007].

- Las ondas delta se ubican entre los 0.5-4 Hz. Estas ondas son asociadas principalmente con el sueño profundo, aunque también se encuentran presentes en el estado consiente y despierto. [Saeid Sanei, 2007, Arriola, 2016].
- Las ondas theta se ubican entre los 4-7.5 Hz. Estas ondas están asociadas con la transición entre la consciencia y la somnolencia además de estar profundamente relacionadas con el acceso al inconsciente, la inspiración creativa y la meditación profunda.
- Las ondas alpha se ubican entre los 8-13 Hz. Estas ondas son asociadas con estados de conciencia tranquila pero no necesariamente de concentración.[Saeid Sanei, 2007, Arriola, 2016].
- Las ondas beta se ubican entre los 14-26 Hz. Estas ondas son asociadas con el pensamiento activo, la atención, la concentración, la conciencia del mundo circundante, y la resolución de problemas concretos.[Saeid Sanei, 2007, Arriola, 2016].
- Las ondas gamma se ubican entre los 30-80 Hz. Estas ondas se han asociado a movimientos motores. Sin embargo la amplitud de estas ondas suele ser pequeña, y no ocurren con frecuencia por lo que su estudio y localización a menudo se utilizan para confirmar ciertas neuropatías o enfermedades mentales.[Arriola, 2016].

La figura **4-1**, pertenece a una grabación de 4 segundos, de amplitud en micro voltios y obtenida por el electrodo ubicado en C3 cuando la persona levanta la mano derecha, donde se puede aprecian los ritmos cerebrales por separados de una señal EEG. La gráfica de la parte superior muestra la señal pura sin haber pasado por algún filtro, de la que se desprenden el resto mostrando los diferentes ritmos. En la figura **4-1**, se aprecia la diferencia de actividad entre bandas, mostrando que el ritmo delta, poco tiene que ver con la acción motora realizada, el ritmo theta, alpha y beta muestran un incremento en la amplitud en los

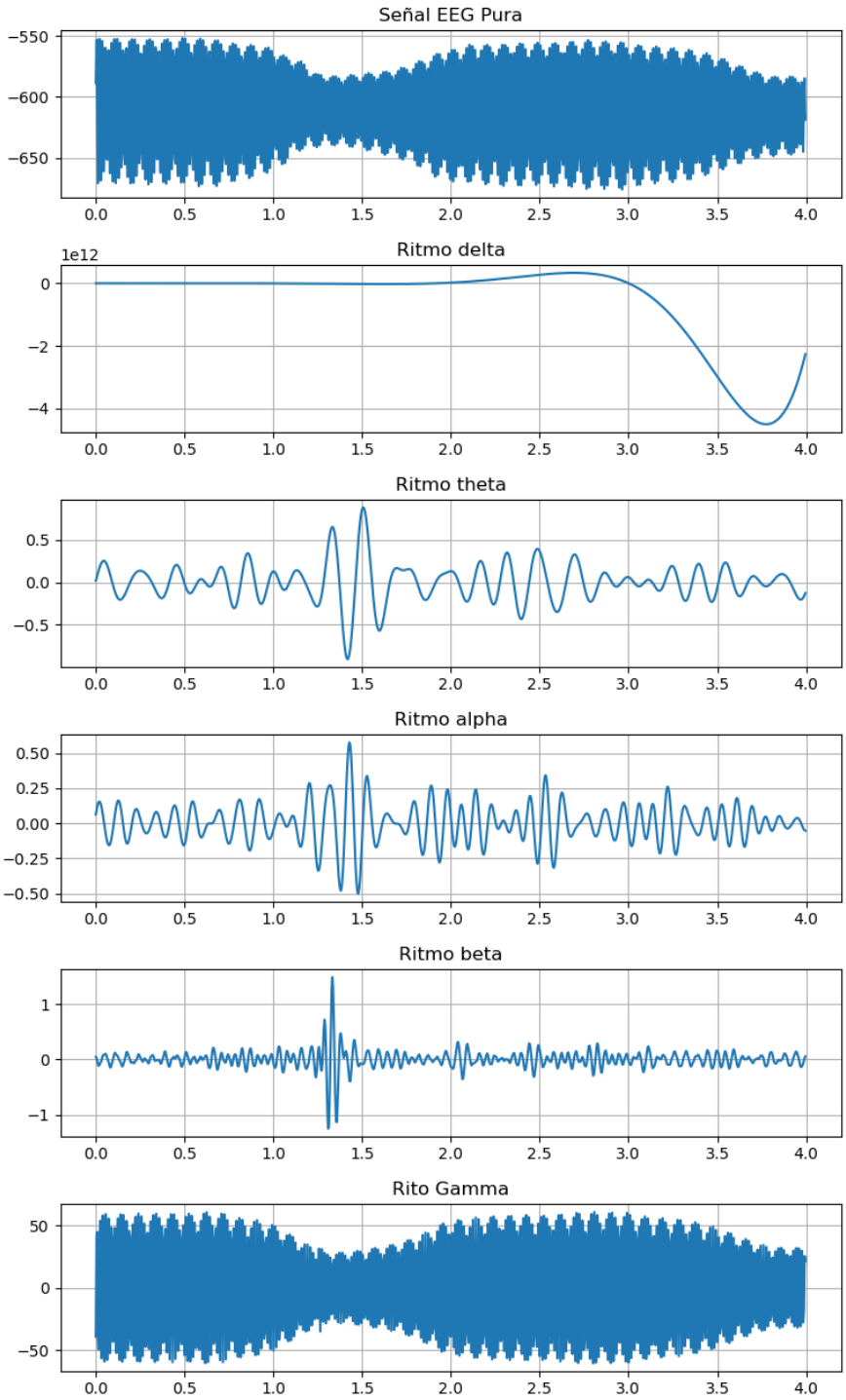



**Figura 4-1**.: Señal EEG del levantamiento de la mano derecha tomada por el electrodo ubicado en C3 donde la gráfica superior es la señal captada sin filtrar seguida por las señales filtradas en orden de menor rango de frecuencias a mayor rango de frecuencias

momentos cercanos al movimiento, mientras que el ritmo gamma muestra una disminución de la amplitud en este intervalo comportándose de una forma muy similar a la señal pura debido a tener un mayor porcentaje del espectro de la señal original.

Siguiendo la teoría, las ondas alpha son las más recomendadas para el estudió, y como se evidencia en la figura **4-1**, estas reaccionan a los movimientos conscientes por lo que se usará esta banda de frecuencias para el desarrollo del proyecto; La figura **4-1** muestra que las ondas beta tienen un tipo de reacción similar a las alpha pero en una amplitud menor y en momentos más específicos, pero como la teoría dice que esta es una banda relacionada con la concentración, tal vez su uso sea más conveniente en aplicaciones relacionadas con la imaginación motora.

## 4.2. Preprocesamiento de los datos

Conociendo la banda que se va a usar, para hacer un análisis en general de las señales, se requiere que todas estén en un rango de valores de amplitud similares, buscando facilitarle las cosas al paradigma de inteligencia computacional encargado de clasificar ya que normalmente muchos algoritmos de machine learning mejoran el rendimiento cuando el vector de características se modifica para que los estimadores posteriores muestren una mejor respuesta a valores atípicos. [sci, 2020]. En el presente trabajo se prueban tres procesamientos, la estandarización de los datos, el escalado de los datos dentro de un rango y la normalización, todos se encuentran en la librería de *scikit learn*.

Ya que muchos de los clasificadores podrían comportarse mal si algunos de los datos no se parecen al conjunto de datos distribuidos, el proceso de estandarización consiste en hacer una distribución Gausiana con una cero de media y unidad de varianza. [sci, 2020]. La ecuación (4-1) muestra el estandarizado,donde $\mu$ es el valor medio calculado en la ecuación (4-2) y $\sigma$ es la desviación estándar calculada en la ecuación (4-3).

$$z = \frac{x - \mu}{\sigma} \tag{4-1}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{4-2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{4-3}$$

Otra alternativa es escalar los datos para que se encuentren entre un valor máximo y uno mínimo determinado (normalmente entre uno y cero) haciendo que los valores absolutos de cada dato se ajusten al tamaño de la unidad [sci, 2020]. La ecuación (4-4) muestra la forma de escalar un conjunto de datos.

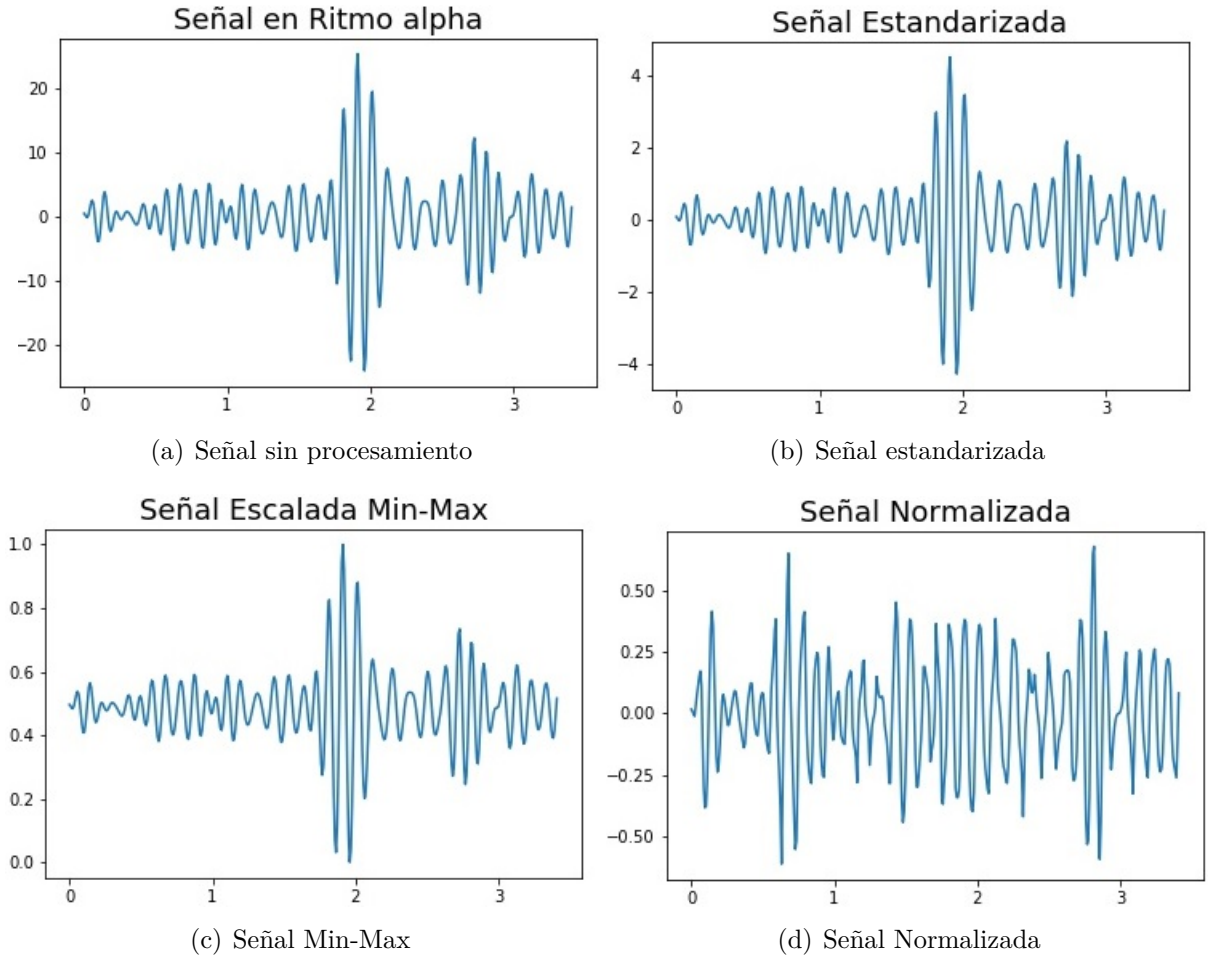
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4-4}$$

El último método utilizado es la normalización de los datos. Esto consiste en escalar individualmente las muestras para tener una norma unitaria. El modulo usado realiza la misma operación que en el estandarizado pero tratando los datos que no son cero de manera individual según *scikit learn* [sci, 2020].

La figura **4-2**, muestra los métodos sugeridos previamente aplicados a una señal EEG, donde se puede apreciar particularidades en cada caso. En el caso de la señal estandarizada y la señal escalada ((b) y (c) respectivamente), se puede apreciar que la forma de la señal se mantiene aunque la varianza entre los valores se reduce significativamente. Sin embargo, la señal estandarizada mantiene valores negativos mientras que la señal escalada no lo hace, lo que se traduce a una varianza de datos todavía menor al escalar los datos que al estandarizarlos. El caso de la normalización resalta por cambiar la forma de la señal, aunque la variación de los datos se mantiene en un rango similar al del escalado.

Entre las ventajas que se pueden obtener al implementar estas formas de procesamiento, se encuentra una inmunidad a la intensidad con la que se ejerza el gesto, ya que la amplitud de la onda cambia dependiendo de la fuerza con la que este se realice. Además la red neuronal al tener valores de datos más cercanos restara importancia a la diferencia de estos dando relevancia a otras características de la señal que diferencien los gestos produciendo una mejor respuesta ante datos desconocidos.

Para determinar que método funciona mejor, se entrena la misma red varias veces con la misma base de datos sin procesamiento y luego con los diferentes tratamientos mostrados anteriormente, se comparan los resultados y se decide teniendo como criterio la respuesta a datos desconocidos. (El capitulo: *Resultados* muestra lo obtenido).




(a) Señal sin procesamiento

(b) Señal estandarizada

(c) Señal Min-Max

(d) Señal Normalizada

**Figura 4-2**.: Señal EEG filtrada en la banda $\alpha$ perteneciente al levantamiento de un brazo captada por el electrodo 3 (C3). La imagen (a) es la señal filtrada que no se ha sometido a ningún procesamiento. La imagen (b) se obtiene tras estandarizar la señal de (a). La imagen (c) se obtiene tras escalar a un rango específico la señal de (a). Por ultimo la imagen (d) se obtiene tras Normalizar la señal de (a)

# 5. Desarrollo del modelo de inteligencia computacional

Para desarrollar un modelo de machine lerning se decide seguir los pasos recomendados por Victor Romman, los cuales se resumen en :"*Definir adecuadamente el problema (objetivo, salidas deseadas ...), recopilar datos, elegir una medida o indicador de éxito, establecer un protocolo de evaluación y los diferentes protocolos disponibles, preparar los datos (tratar con campos vacios, con valores categóricos...), dividir correctamente los datos, diferenciar entre sobre y sub-ajustes (over y underfitting en inglés), desarrollar un modelo de referencia y elegir un modelo adecuado y ajustarlo para conseguir el mejor rendimiento posible*".[Roman, 2019].

## 5.1. Definición

EL principal objetivo del paradigma es clasificar diferentes acciones motoras (Gestos) mediante señales electroencefalográficas en tiempo real, siendo cada gesto una salida del clasificador por lo que este no es un clasificador binario. Las entradas que tendrá la red son series de tiempo obtenidas por los electrodos, con una cantidad de muestras de entre 250 y 700, ya que para que el procesamiento en tiempo real sea posible las series de tiempo no deben ser muy largas para no generar mucho delay. Adicionalmente se prefiere que en caso de clasificar gestos de manera errónea, los asocie como si no se hubiese realizado un gesto.

## 5.2. Levantamiento de datos

Este paso es crítico para el proyecto, ya que entre mejor sea la calidad de los datos, mejores resultados se van a obtener. En capítulos pasados se mencionaron los procesos y las regiones de la corteza cerebral necesarios para mejorar la calidad de los datos adquiridos; a continuación se mostrarán los gestos que entran al clasificador, y como se toman.

### 5.2.1. Selección de gestos

En este punto es necesario recordar el objetivo principal del proyecto, el cual es controlar un drone usando señales EEG. Este vehículo cuenta con seis grados de libertad, con lo cual permite los siguientes movimientos: subir, bajar, ir adelante, ir atrás, ir derecha, ir izquierda,

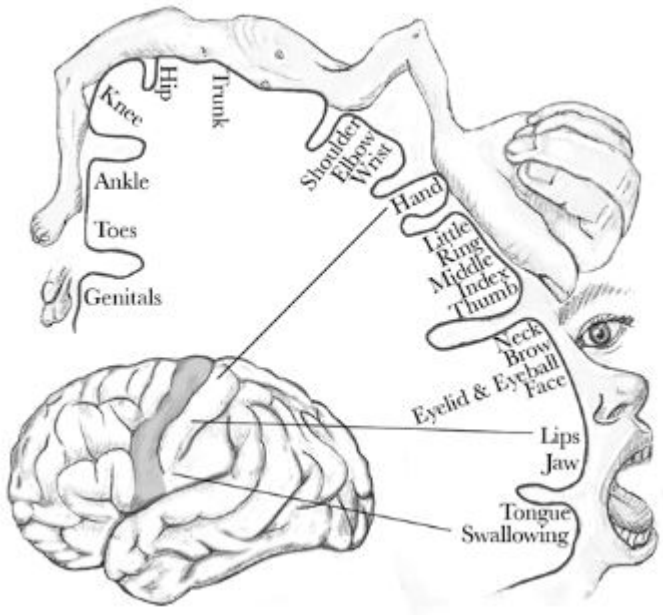



**Figura 5-1**.: Homúnculo motor de Penfield. Imagen tomada de [Oscar, 2019]

pitch, yaw y roll.

Para ejecutar las anteriores acciones se tienen diferentes gestos, los cuales se eligen teniendo como referencia observaciones propias al ejecutarlos (como son los parpadeos, los cuales se aprecian con facilidad en los electrodos 1 y 2) y el homúnculo motor de Penfield el cual muestra como la corteza cerebral administra las diferentes partes móviles del cuerpo humano. La imagen **5-1**, muestra el homúnculo que representa las partes del cerebro que tienen relación directa con las diversas partes del cuerpo.

Al observar la figura **5-1**, se aprecia que una gran parte de la corteza cerebral motora se dedica a realizar movimientos faciales, y manuales, por lo que los gestos se escogen teniendo en cuenta esta basta distribución con el fin de obtener la mayor cantidad de información relevante y asociada a la parte del cuerpo.

Con la información anterior se eligen los siguientes gestos para el controlar el drone:

- Parpadeo del ojo izquierdo
- Parpadeo del ojo derecho
- Levantamiento de cejas
- Presión de la mandíbula mediante una mordida

- Levantamiento de la mano izquierda
- Levantamiento de la mano derecha

Además de éstos, es necesario que el clasificador sea capaz de detectar cuando hay un orden y cuando no, por lo que se agrega una clase adicional en el cual la persona no realiza ninguna de las acciones anteriores, con el objetivo que se interprete como ninguna orden.

### 5.2.2. Toma de señales

La toma de señales se realiza mediante un GUI (Interfaz gráfica de usuario) diseñado por miembros[1] del grupo de investigación LASER el cual permite no solo capturar señales EEG con la tarjeta Cyton y Deisy, sino que permite captar señales EMG con el brazalete MYO.

Los datos, se tomaron en 15 personas que accedieron a participar; se tomaba una grabación de cinco segundos de duración y se les pidió que al ver en la pantalla que habían pasado 2 segundos ejecutaran la respectiva acción. Cada acción se grabó 10 veces por persona, para obtener un total de 1050 grabaciones. Adicionalmente, ciertas personas accedieron a repetir la grabaciones varias veces, lo que incremento el número de muestras a las 1330.[2]

## 5.3. Indicador de éxito y protocolo de evaluación

Como indicador de éxito se optó por elegir la exactitud (accuracy), debido a que es una medida usada normalmente en clasificadores.

Para el evaluar la red usada, se dividen los datos de la siguiente manera: 10 % para prueba, 20 % validación y 70 % para entrenamiento. El protocolo elegido es el k-fold cross validation, debido a su amplio uso a nivel general y a que mejora la distribución de los datos en el entrenamiento y la validación.

## 5.4. Preparación de los datos

Al realizar las grabaciones no todas quedan con el mismo número de datos, por lo que se recortan al final de la señal, para que en total sean 624 muestras por grabación, con lo cual

[1] Los miembros que desarrollaron el GUI son Fabian Barreto, Nicolas Rodriguez y Cristian Sanchez; este se puede encontrar en la siguiente dirección perteneciente a Github: `https://github.com/fabianbarreto02/Open_BCI_and_MYO_UD/tree/master/GUI_Adquisicion`

[2] Las grabaciones se pueden encontrar en la siguiente dirección URL: `https://github.com/IvanDanielVillegas/Data-Base-EEG-face-und-arms-Movement`

en algunos casos se pierde una muestra; pérdida que no resulta relevante al no ser una disminución significativa de la señal.

Debido a la construcción del casco Ultracortex, no todas las grabaciones tomadas capturaron información útil para el clasificador, por lo que las grabaciones se revisaron para eliminar aquellas que puedan introducir error al clasificador,entre las que se encuentran todas las relacionadas con los gestos de levantamiento de las manos, ya que los datos obtenidos no eran representativos de estos gestos.

Entre las grabaciones que presentan información relevante, hay casos en los que algún electrodo sufrió alguna desconexión, por tal motivo no muestra información, aunque otros electrodos si lo hagan. Estos datos se introducen al entrenamiento ya que estos valores están dentro del rango medio de los datos, por lo que no hay necesidad de tratarlos.

### 5.4.1. Datos categóricos

Buscando características que ingresar a la red, se calculan valores en el tiempo sugeridos por [Boris Medina, 2018], como la media de la señal, la varianza, la energía de la señal, los cambios de pendiente, la longitud de onda y los parámetros Hjort.

La figura **5-2** muestra el comportamiento de estas características al evaluarse en ventanas de 90 muestras con pasos de un dato. Se aprecia que el valor RMS, la energía de la señal y la amplitud Willison presentan un comportamiento similar en el cual es fácilmente apreciable el momento donde se realiza el gesto. Los cambios de pendiente muestran un comportamiento inverso al de las características anteriores, ya que hay una disminución en los cambios cuando se realiza el gesto, sin embargo este cambio es menos evidente que en las características anteriores. Por ultimo la figura **5-2** muestra un parametro denominado longitud de onda el cual compara los valores entre muestras consecutivas; si bien es evidente el momento en que se llevan a cabo las actividades, el resultado es una señal más compleja incluso que la EEG normalizada.

Tanto [Boris Medina, 2018] como [Hernández, 2016] proponen usar los denominados parámetros Hjorth: actividad, movilidad y complejidad, los cuales representan la energía promedio, la frecuencia media y el ancho de banda de la señal respectivamente. Al igual que con las características anteriores se utiliza una ventana de 90 muestras con pasos de 1 para observar su comportamiento a lo largo de la señal dando como resultado lo que se observa en la figura **5-3**. Al analizar la figura se aprecia que la movilidad tiene el mismo comportamiento de la energía, lo cual era de esperar debido a la naturaleza del parámetro. La movilidad presenta un comportamiento similar al de los cambios de pendiente en el sentido en que es inverso al comportamiento de la actividad, y que de la misma forma es más confuso encontrar una rela-

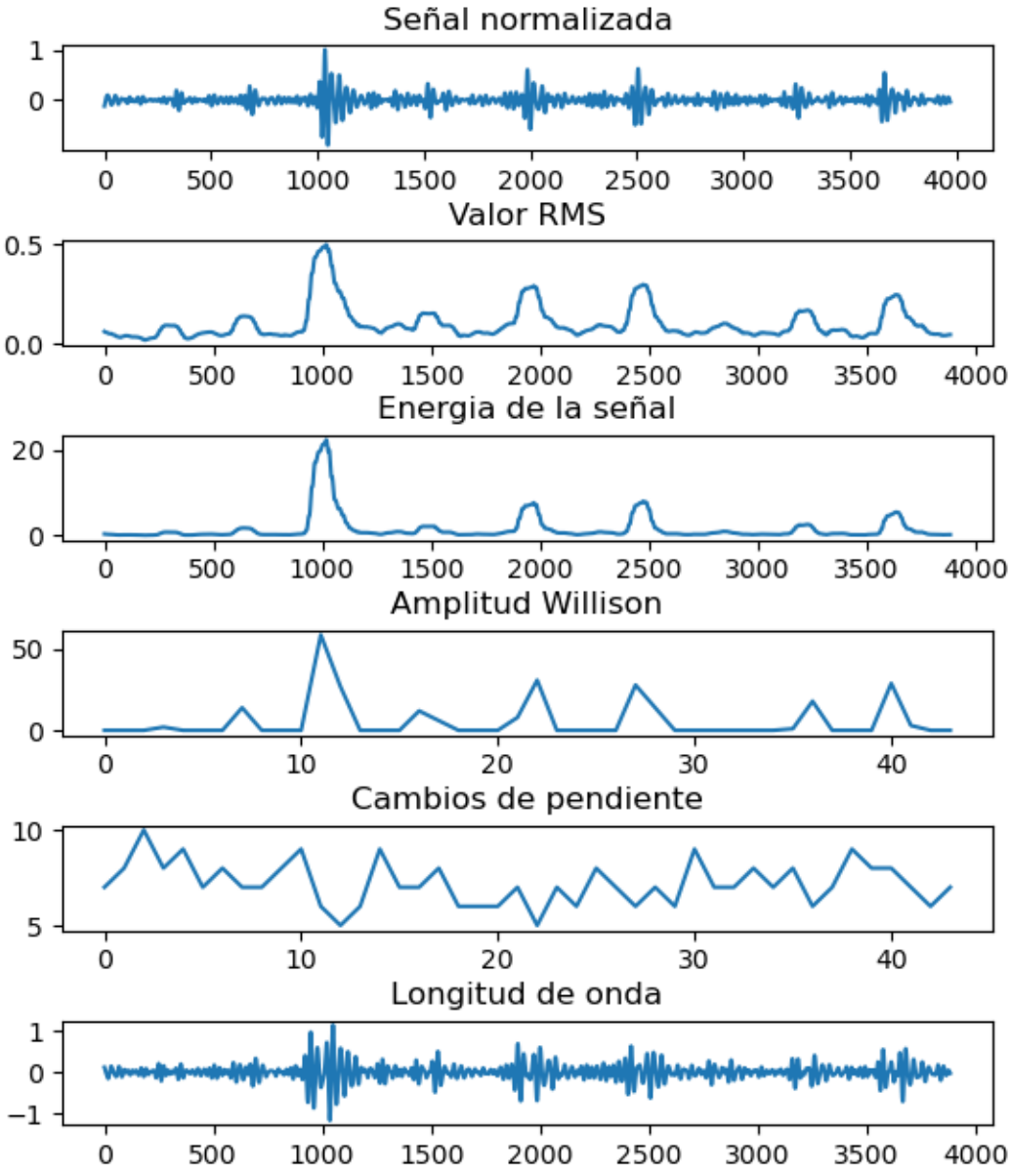



**Figura 5-2**.: Características en el tiempo de la señal EEG tomada por el electrodo 1 perteneciente a cinco parpadeos del ojo derecho

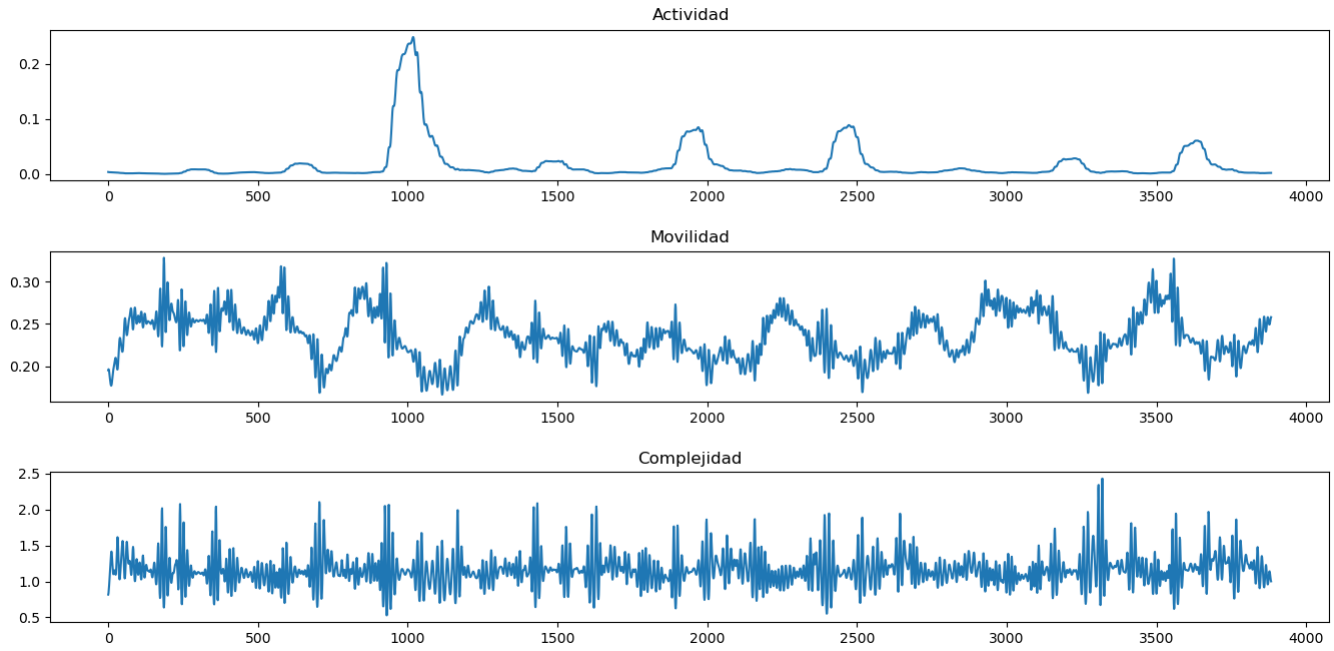
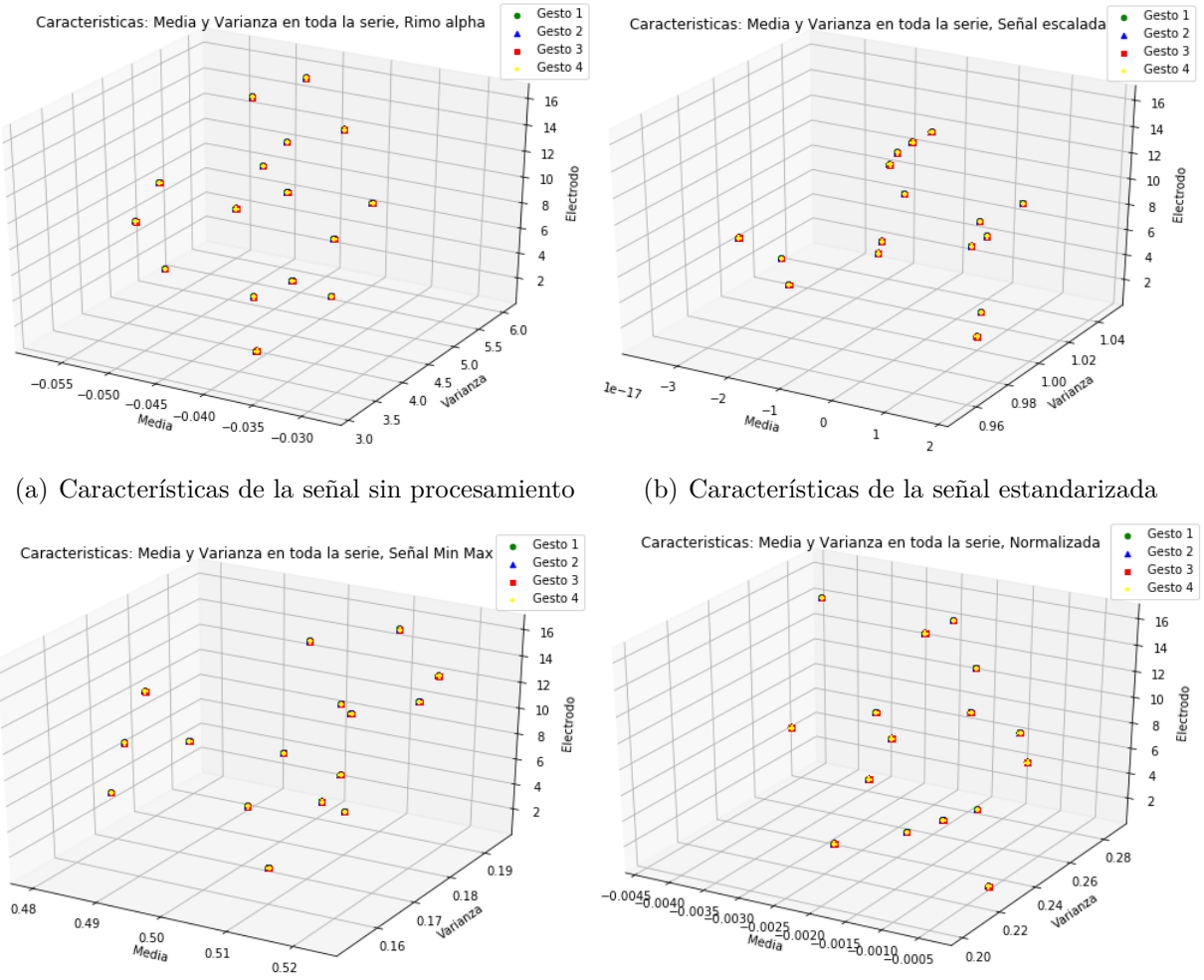



**Figura 5-3**.: Parámetros de Hjorth de la señal EEG tomada por el electrodo 1 perteneciente a cinco parpadeos del ojo derecho

ción entre los valores y la actividad cerebral; además la movilidad y los cambios de pendiente se relacionan por ser medidas asociadas a la frecuencia. Por último se tiene la complejidad, de la cual no se puede apreciar ninguna aparente relación a simple vista, probablemente, debido a los pasos con lo que se evalúa el comportamiento.

Para evaluar la eficiencia de las características, estas se presentan en una gráfica de cuatro dimensiones, donde el eje x muestra la media, el eje y la varianza, el eje z es el electrodo y el delta de color representa el gesto. Los resultados se muestran en la figura **5-4**, donde se observa que sin importar que método de procesamiento se use, los gestos se sobreponen entre si, mostrando los valores de la media y la varianza son muy similares independientemente del gesto, haciendo imposible clasificar de manera adecuada con estas características.

De igual manera evaluando la eficiencia de los parámetros Hjort, estos se presentan en una gráfica de cuatro dimensiones, donde el eje x muestra la actividad, el eje y la movilidad, el eje z es la complejidad y el delta de color representa el gesto. Los resultados se muestran en la figura **5-4**, donde se observa que sin importar que método de procesamiento se use, los gestos se sobreponen entre si, mostrando que los valores de los parámetros Hjorth son muy similares independientemente del gesto, haciendo imposible clasificar de manera adecuada con estas características.



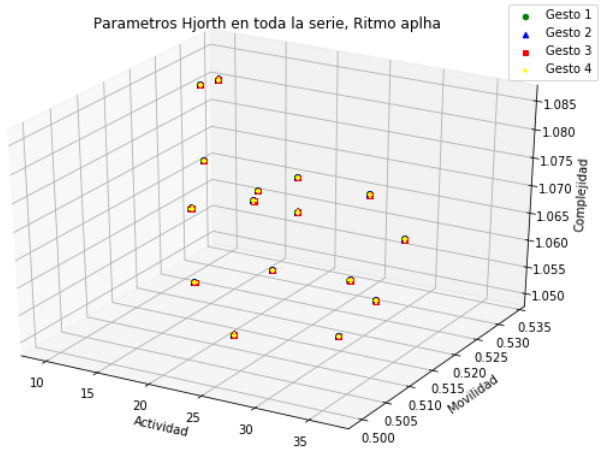

(a) Características de la señal sin procesamiento

(b) Características de la señal estandarizada

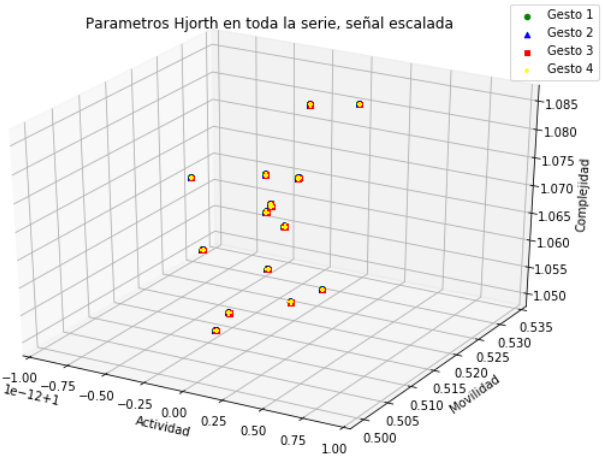
(c) Características de la señal Min-Max

(d) Características de la señal Normalizada

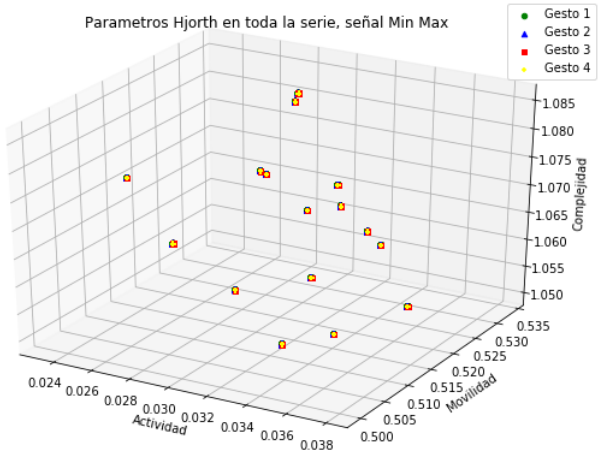
**Figura 5-4**.: Gráficas de la media y varianza de la señales captadas para entrenar el modelo con los diferentes pre-procesamientos propuestos. La imagen (a) son las características de la señal filtrada que no se ha sometido a ningún procesamiento. La imagen (b) muestra las características tras estandarizar la señal de (a). La imagen (c) se obtiene de las características tras escalar a un rango especifico la señal de (a). Por último la imagen (d) presenta las características tras Normalizar la señal de (a)



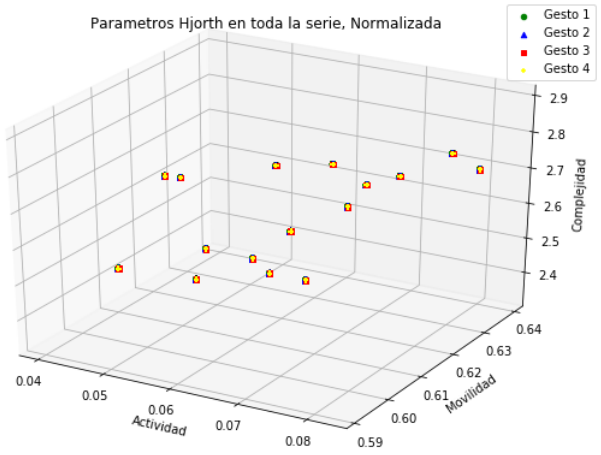

(a) Parámetros Hjorth de la señal sin procesamiento




(b) Parámetros Hjorth de la señal estandarizada




(c) Parámetros Hjorth de la señal Min-Max




(d) Parámetros Hjorth de la señal Normalizada

**Figura 5-5**.: Gráficas de los parámetros Hjorth de la señales captadas para entrenar el modelo con los diferentes pre-procesamientos propuestos. La imagen (a) son las parámetros de la señal filtrada que no se ha sometido a ningún procesamiento. La imagen (b) muestra los parámetros tras estandarizar la señal de (a). La imagen (c) se obtiene de las parámetros tras escalar a un rango especifico la señal de (a). Por último la imagen (d) presenta las características tras Normalizar la señal de (a)

La ineficiencia de las características para diferenciar los gestos puede que se deba a que se calcularon para toda la serie, siendo la mayor parte de la grabación: actividad cerebral que no pertenecía a los gestos que se realizaban en un tiempo cercano al de un segundo, por lo que la información que suministra la acción se pierde ante el ruido del resto de la grabación.

## 5.5. Desarrollo del modelo base

En el desarrollo del modelo se consideran clasificadores RF (Random Forest) y KNN (K-Nearest Neighbor). El RF es un clasificador que utiliza una combinación de árboles predictores, cada árbol depende de los valores de un vector aleatorio de manera independiente, pero con la misma distribución en cada uno. Mientras que el algoritmo KNN clasifica las entradas dependiendo de la clase que tenga más cercana estadísticamente a la que se quiere ubicar. Estos modelos se entrenan varias veces, cambiando los datos de entrenamiento entre las series de tiempo filtradas en el ritmo alpha con los diferentes procesamientos relacionados con la estandarización, sin embargo, las características mencionadas anteriormente aplicadas a toda la serie no son usadas debido a que como se evidencia en las figuras **5-4** y **5-5** no es fácil diferenciar los gestos con esos datos.

Para mostrar los resultados se utiliza un diagrama de caja y bigotes mostrando diferentes métricas. *Precision* hace referencia a la cantidad de verdaderos positivos frente a falsos positivos, es decir evalúa verdaderamente cuantas de las predicciones de una determinada clase, pertenecen a dicha clase. *Recall* compara la cantidad de verdaderos positivos con los falsos negativos, es decir, de los datos de una clase, cuantos se clasifican como pertenecientes a la verdadera clase y cuantos al resto.[Shmueli, 2019a]. El *F1-score* es una métrica que hace un promedio entre ambas mientras que el *overall* es parecido al *Recall* aunque teniendo en cuenta los datos usados en cada clase.[Shmueli, 2019b].

La figura **5-6** muestra los resultados de los entrenamientos realizados con los modelos base, comparando los diferentes modos de pre-procesamiento enunciados en el capitulo anterior. Se puede ver que la mayor exactitud se obtiene con el RF usando el pre-procesamiento del escalado entre valores fijos, con máximo en 1 y mínimo en 0, mientras que el resto de métricas alcanzan su mayor valor en RF solo con el filtrado en el ritmo $\alpha$. Sin embargo, ninguno de los valores muestra un buen resultado, pues solo el mejor de todos supera un valor superior a 0.5.

La figura **5-7** muestra los resultados de los entrenamientos realizados con los modelos base, comparando los diferentes modos de pre-procesamiento ya mencionados anteriormente. Se puede ver que la mayor exactitud se obtiene con el RF con el pre-procesamiento del escalado, al igual que el resto de métricas que alcanzan su mayor valor en RF ya sea con el pre-procesamiento del escalado normal o entre valores fijos, con máximo en 1 y mínimo en 0. Se aprecia que al disminuir la cantidad de gestos, el valor de los parámetros aumenta hasta

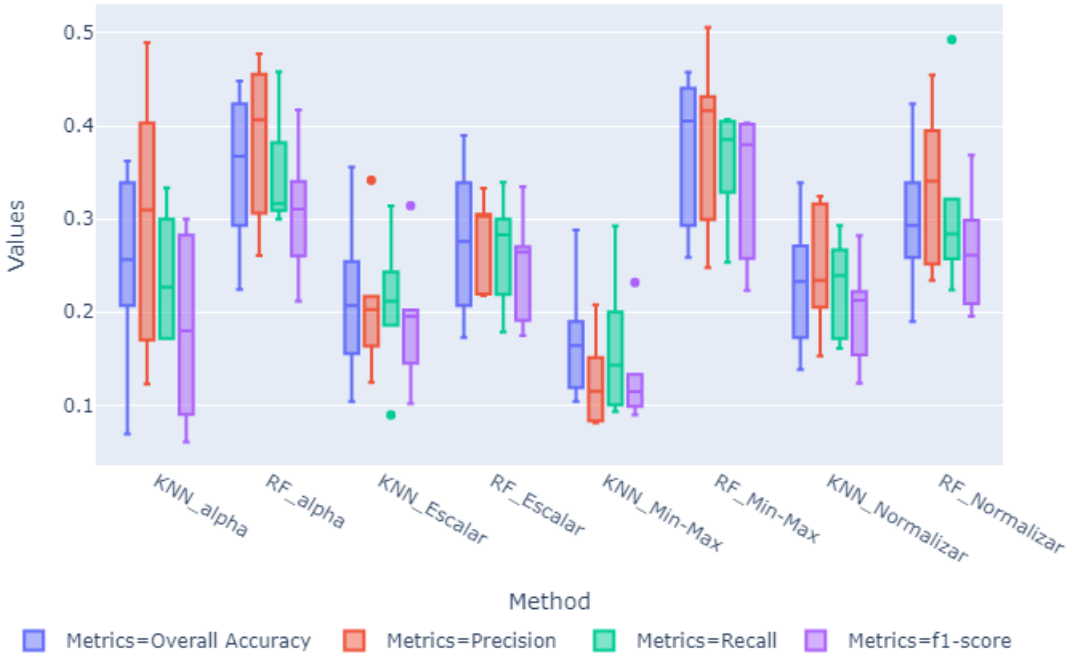



**Figura 5-6**.: Resultados de las pruebas con las redes Base con siete gestos además de la inactividad

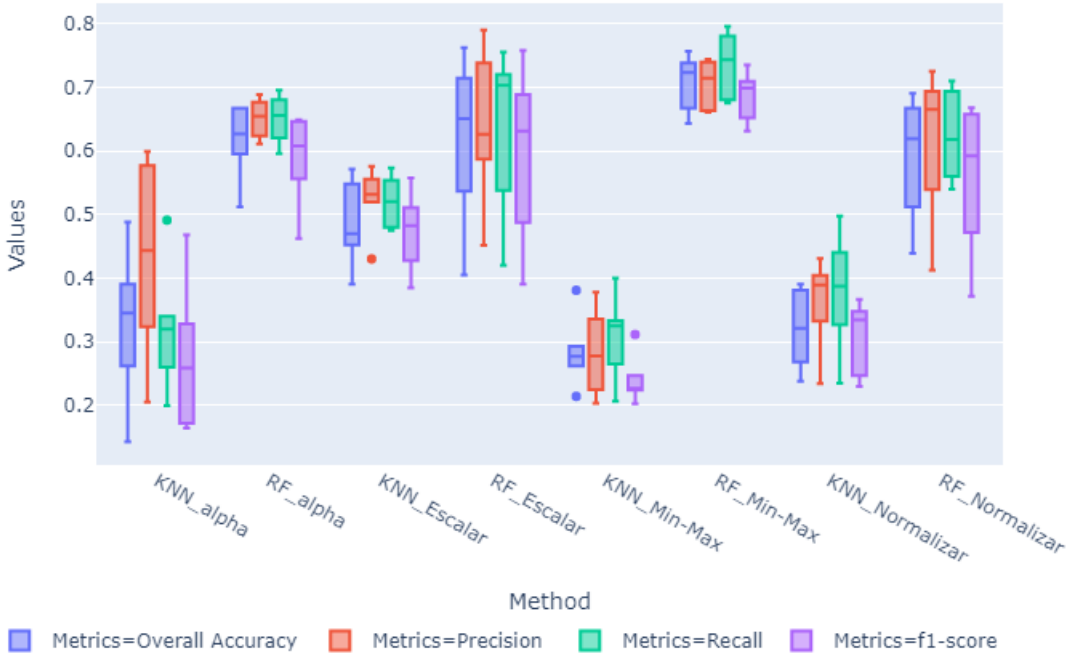



**Figura 5-7**.: Resultados de las pruebas con las redes Base con cuatro gestos además de la inactividad

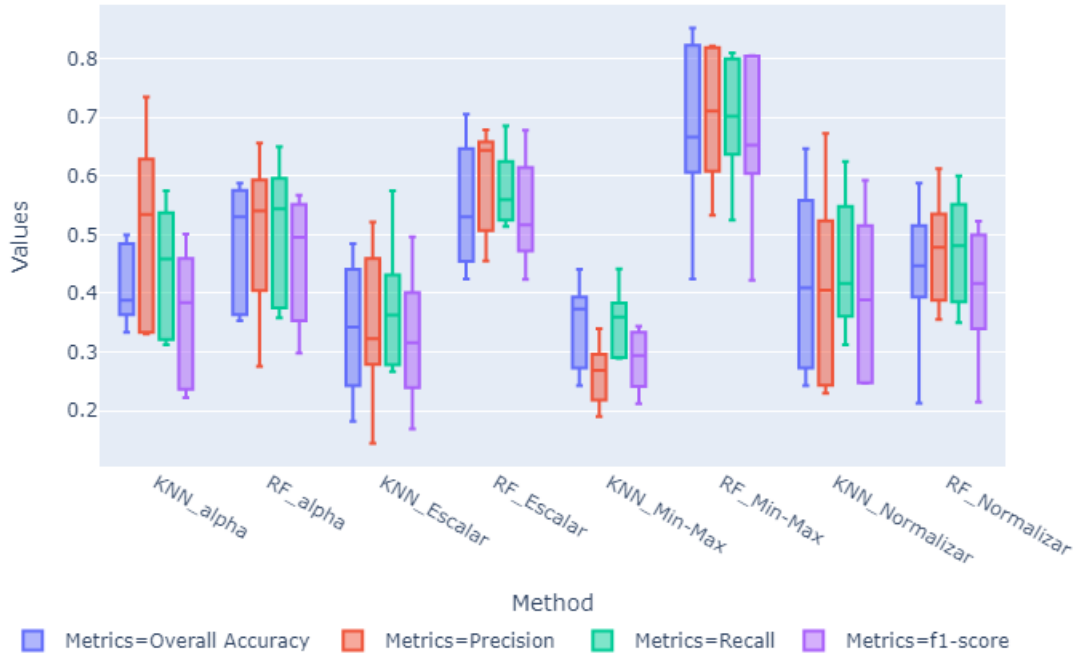



**Figura 5-8**.: Resultados de las pruebas con las redes Base con tres gestos además de la inactividad

acercarse al 0.8.

La figura **5-9** muestra los resultados de los entrenamientos realizados con los modelos base, comparando los diferentes modos de pre-procesamiento de la misma forma que en los casos anteriores. Se puede ver que los mayores valores de evaluación se obtienen en RF con el pre-procesamiento del escalado entre valores fijos, con máximo en 1 y mínimo en 0, donde estos superan el 0.8.

Es válido aclarar que en las pruebas anteriores, se usaron para verificar la eficiencia de los modelos clasificando, no como responden ante datos nuevos, ya que para validar y testear se usaron los mismos datos con los que se entrenaron los modelos. Lo cual en parte explica el buen desarrollo que tienen estos.

Lo mencionado anteriormente se hace evidente en la imagen **5-9**, donde se usaron otros datos para entrenar el modelo, lo cuales no fueron incluidos en los entrenamientos anteriores. Se puede ver que la mayor exactitud se obtiene con el KNN sin pre-procesamiento, solamente con las señales filtradas en el ritmo $\alpha$,mientras el resto de métricas que alcanzan su mayor valor en RF con el pre-procesamiento del escalado entre valores fijos, con máximo en 1 y mínimo en 0, lo cual era lo esperado, aunque en este caso los valores del RF no sea acercan al 0.8 como antes. Por este motivo es necesario buscar un modelo que presente una mejor

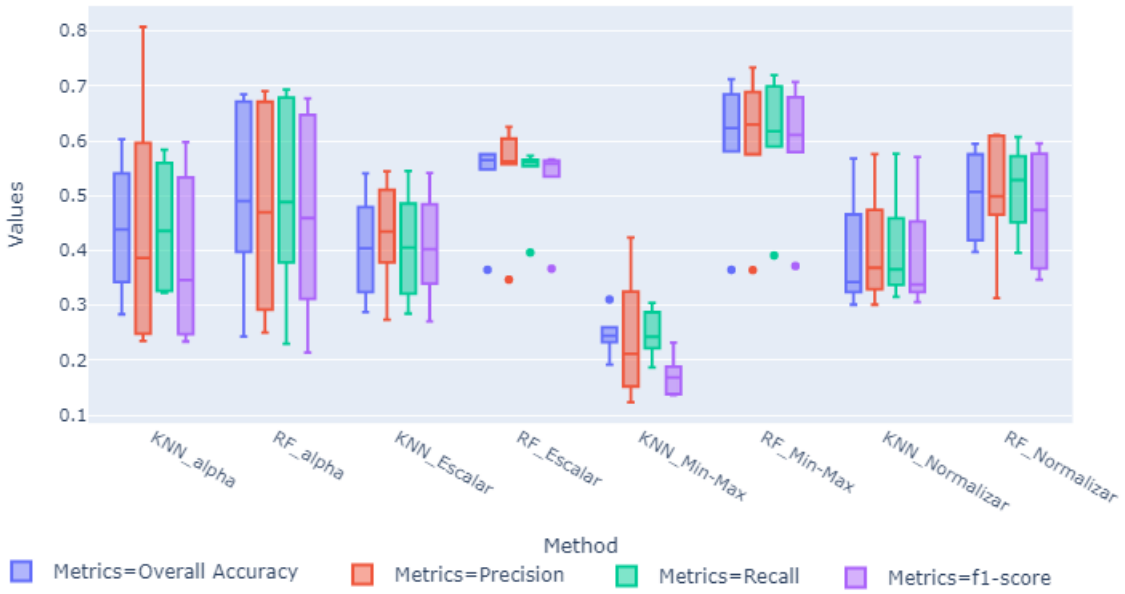



**Figura 5-9**.: Resultados de las pruebas con las redes Base con tres gestos además de la inactividad con datos diferentes

respuesta ante datos desconocidos.

## 5.6. Desarrollo de un modelo mejor y más ajustado

De la misma forma que se probaron los modelos RF y KNN se evaluó el modelo implementando CNN (Convolutional Neural Network) con las series de tiempo filtradas y con la estandarización. Al igual que con el modelo base, las características no se tuvieron en cuenta, al no dar una buena información.

Las redes convolucionales se utilizan para la clasificación en imágenes; es de interés el hecho que detecten algo en específico en cualquier parte de la imagen, sin importar la posición, lo cual para este proyecto es relevante debido a la necesidad de implementar la red en tiempo real, ya que se requiere que la red sea capaz de identificar el gesto en cualquier momento de la grabación sin importar cuando ocurra y evitando que suceda lo mismo que con las características en donde el ruido de la grabación cobra más importancia que el gesto realizado.

Con lo anterior, teniendo como referencia una red que clasifica fotos correspondientes al lenguaje de señas, se adapta el modelo para que funcione en una dimensión con 16 canales (cada uno de los electrodos); para definir los valores de los parámetros a usar en la red, se realizan varios entrenamientos, con 200 épocas cada uno y con la exactitud de validación como monitor, variando la configuración en diferentes capas (Los resultados de estos expe-

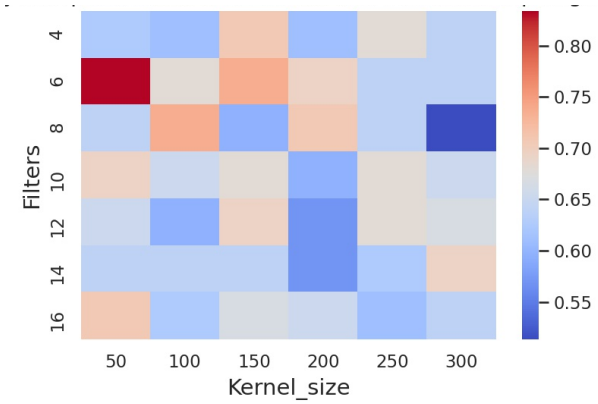
rimentos se muestran en los mapas de calor en la figura **5-10**), el resultado final es la tabla que se muestra en la figura **5-11**.

La figura **5-11** muestra un mapa de calor, en donde los colores más cercanos al azul intenso representan un valor de exactitud alcanzado bajo, mientras que los valores altos se representan por colores cercanos al rojo. En el caso de la imagen **5-11**-(a) se muestra información sobre la primera capa convolucional, donde la mayoría de las combinaciones dan como resultado una exactitud por debajo del 70 %, siendo bastante evidente que la mejor combinación para la primera capa es con un *kernel size* de 50 y seis filtros. Pasando por la imagen **5-11**-(b) se evidencian más casos con valores por encima del 70 %, sin embargo se presentan casos en el que se obtienen valores inferiores a los menores valores obtenidos en el mapa anterior; de igual manera se encuentra también una combinación que permite resultados por encima del 80 %, la cual se da al usar 16 filtros y un *kernel size* de 250 en la segunda capa convolucional. Pasando a las capas densas, se evalúa la cantidad de neuronas a usar y el *drop out*, el mapa de calor **5-11**-(c) muestra que al usar 250 neuronas en la capa densa se obtienen los valores de exactitud más altos, entre las que se encuentran dos posibles valores de *drop out* (0.4 y 0.6).

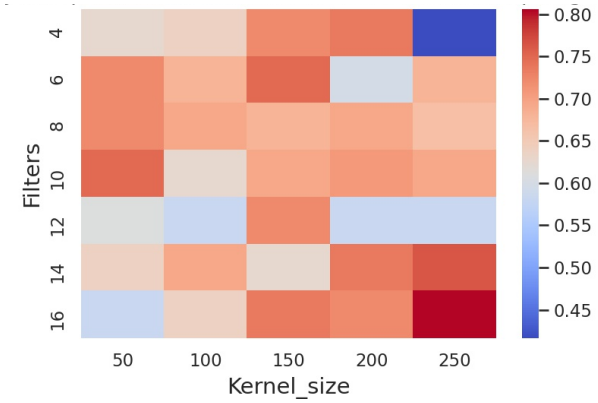
Debido a lo anteriormente descrito el modelo resultante mostrado en la figura **5-11** está compuesto por: una capa de entrada con capacidad para 16 canales cada uno con 625 muestras, a continuación una capa convolucional en una dimensión que reduce la salida a 6 canales cada uno con 575 datos a la que le sigue un *max pooling* que se encarga de reducir las muestras a 288 manteniendo los seis canales; nuevamente se agrega una capa convolucional que baja la cantidad de muestras a 39, sin embargo esta aumenta el número de canales nuevamente a 16, y como en las capas anteriores se agrega un *max pooling* que mantiene el número de canales pero reduce los datos en estos a 20. Se continua con un *flatten* el cual se encarga de pasar todos los datos a una sola dimensión, dando como salida la multiplicación de las muestras por los canales. En la capa densa, como se mencionó anteriormente se tienen 250 neuronas y un *dropout* de 0.4 que termina por llegar a una capa densa con cuatro neuronas como salida (En el modelo final). Como el resultado es un vector de probabilidades se utiliza el comando *np.argmax* para determinar cuál neurona es la que posee la mayor probabilidad, para determinar con facilidad la clasificación de la red.

### 5.6.1. Modelo para seis gestos e inactividad

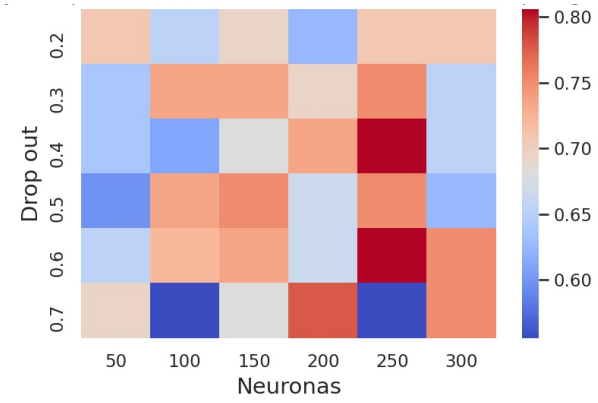
Considerando los modelos base, estos exponen un comportamiento regular, se decide probar el modelo CNN con todos los gestos para ver si era posible obtener un mejor resultado. Aun así, como muestran las figura **5-12** los resultados son incluso peores que los obtenidos con los modelos base, en los que incluso se presenta el caso que sin importar que tipo de pre procesamiento se aplica a los datos, ninguno es destacable.




(a) Mapa de calor de los parámetros en la primera capa convolucional

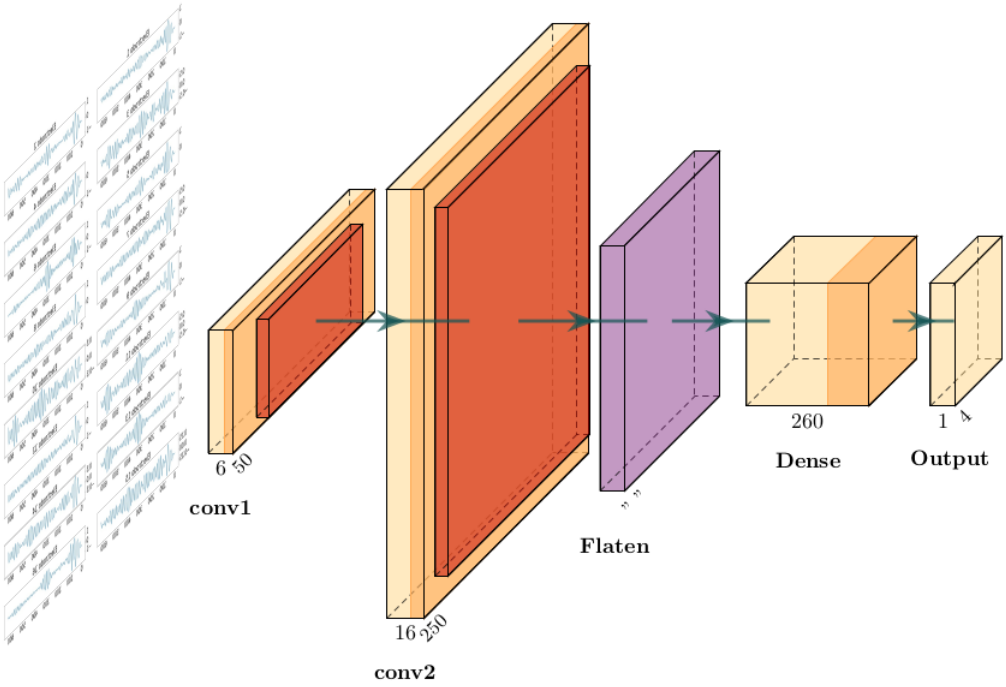



(b) Mapa de calor de los parámetros en la segunda capa convolucional




(c) Mapa de calor de los parámetros en las

**Figura 5-10**.: Mapas de calor que muestran las pruebas realizadas para determinar los valores de la topología de la red con mejores resultados en la exactitud




(a) Diagrama de bloques de la red CNN implementada

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_187 (InputLayer) | [(None, 625, 16)] | 0 |
| conv1d_372 (Conv1D) | (None, 576, 6) | 4806 |
| max_pooling1d_370 (MaxPoolin | (None, 288, 6) | 0 |
| conv1d_373 (Conv1D) | (None, 39, 16) | 24016 |
| max_pooling1d_371 (MaxPoolin | (None, 20, 16) | 0 |
| flatten_184 (Flatten) | (None, 320) | 0 |
| dense_368 (Dense) | (None, 250) | 80250 |
| dropout_184 (Dropout) | (None, 250) | 0 |
| dense_369 (Dense) | (None, 4) | 1004 |

Total params: 110,076
Trainable params: 110,076
Non-trainable params: 0

(b) Valores de salida y parámetros de la red CNN implementada

**Figura 5-11**.: Red CNN implementada

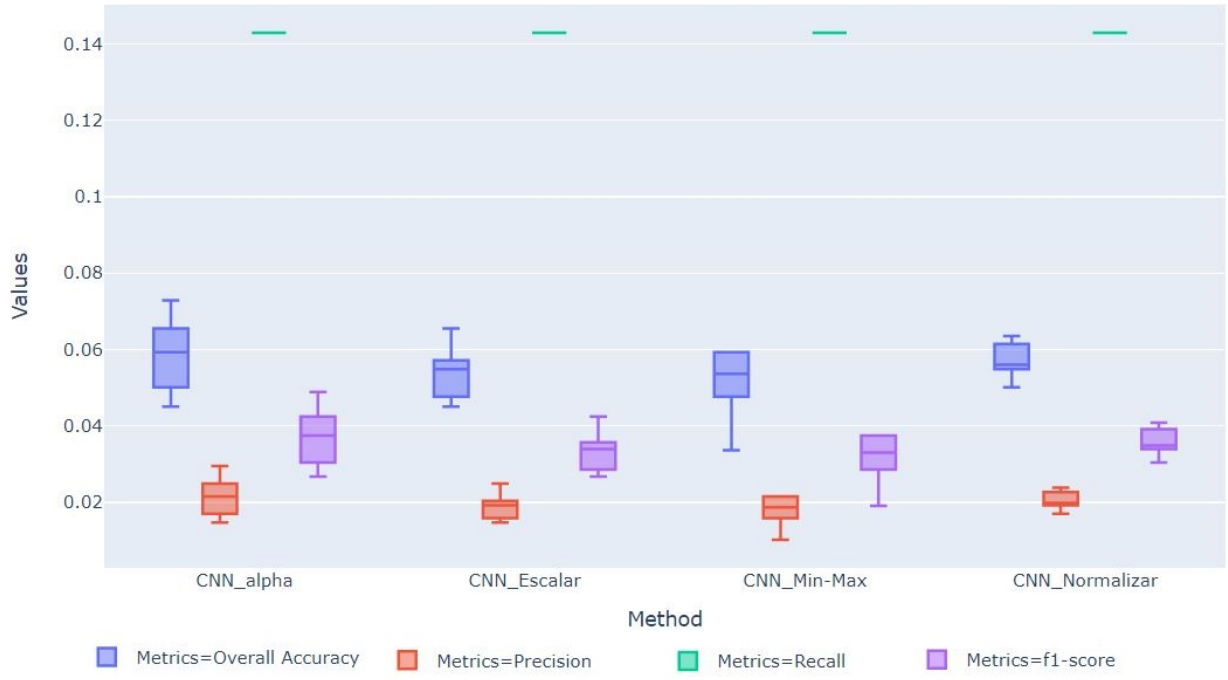



**Figura 5-12**.: Resultados de las pruebas de validación con la red CNN con seis gestos además de la inactividad

### 5.6.2. Modelo para cuatro gestos e inactividad

Continuando con el procedimiento, se eliminan los gestos relacionados con el movimiento de los brazos al igual que se hizo en los modelos base, y se entrena la red con los demás gestos obteniendo los resultados mostrados en la figura **5-13**.

En la figura **5-13** se aprecia una clara mejora frente a el modelo de los siete gestos expuesto en la figura **5-12** en la mayoría de los casos a excepción de la red entrenada con los datos escalados entre valores fijos la cual sigue con métricas muy bajas. Destacan entre las pruebas las relacionadas que pertenecen a los datos sin pre-procesamiento y los datos normalizados. En el caso de los datos sin pre-procesar las pruebas alcanzan los valores más altos sin embargo sus resultados varían bastante por lo que su promedio es bajo. Respecto al otro caso que resalta (el de los datos normalizados) los valores máximos alcanzados no son tan altos, sin embargo, al tener una desviación estándar menor, el promedio es más alto.

### 5.6.3. Modelo para tres gestos e inactividad

De todas formas el rendimiento alcanzado no es adecuado y es inferior al obtenido con las redes RF y KNN por lo que al igual que se hizo con los modelos base, se retira un gesto más con ánimo de ganar exactitud, y se entrena la red con los conjuntos de datos ya mencionados.

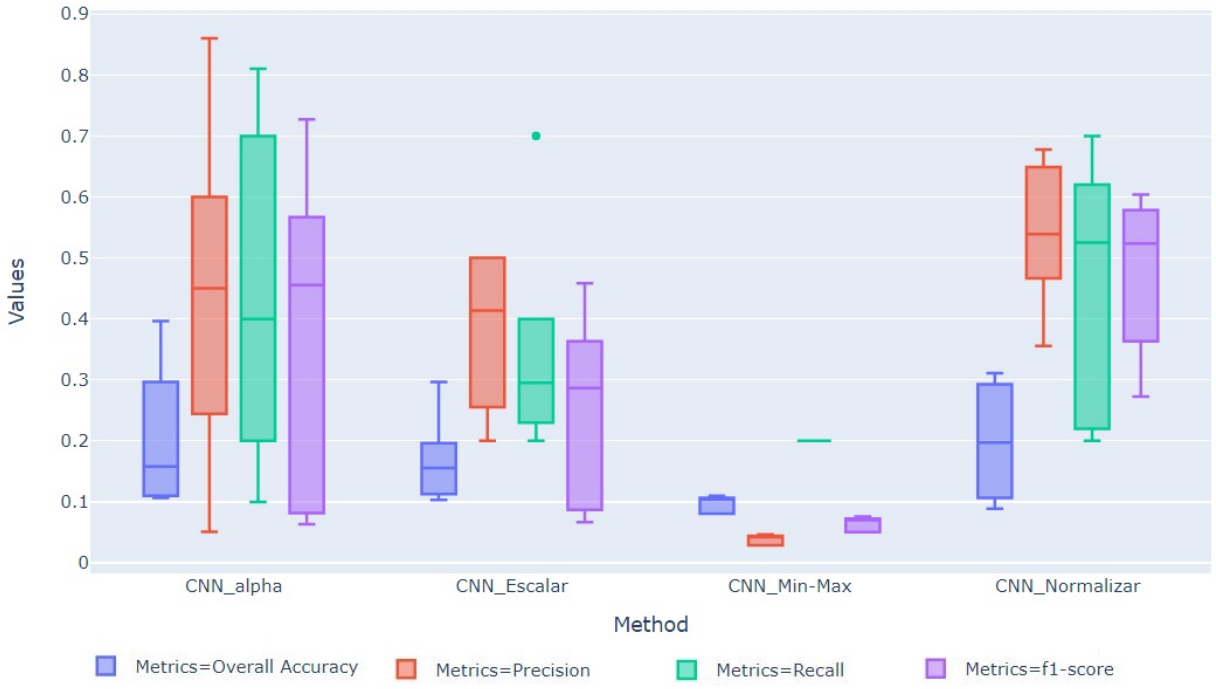



**Figura 5-13**.: Resultados de las pruebas de validación con la red CNN con cuatro gestos además de la inactividad

Al evaluar la red con solo tres gestos además de la inactividad se obtienen resultados excelentes en dos casos en concreto. Como se aprecia en la figura **5-14** la red tiene un desempeño excelente en la prueba con los datos sin pre-procesamiento y con los datos normalizados se obtienen los valores más altos en las diferentes métricas superiores en la mayoría de los casos superior al 80 %, con una desviación estándar baja, adecuada para el proyecto que se quiere tratar. El caso de los datos escalados presenta una mejora en algunos casos insuficiente, mientras que los datos escalados entre los valores 0 y 1 no presenta ninguna mejora.

Siguiendo el procedimiento realizado con los modelos base, se realizan pruebas con datos que nunca han sido vistos por las redes, para evaluar el desempeño con datos totalmente desconocidos, obteniendo los resultados mostrados en la figura **5-15**, donde al igual que con los modelos base se observa un decremento en los valores de las métricas, sin embargo, en el caso de los datos solamente filtrados en el ritmo $\alpha$ el promedio sigue estando por encima del 80 %. El caso de los datos normalizados, aunque continua como la segunda opción, los resultados están alrededor del 70 % con una desviación estándar baja en sus métricas a excepción de *Precision*.

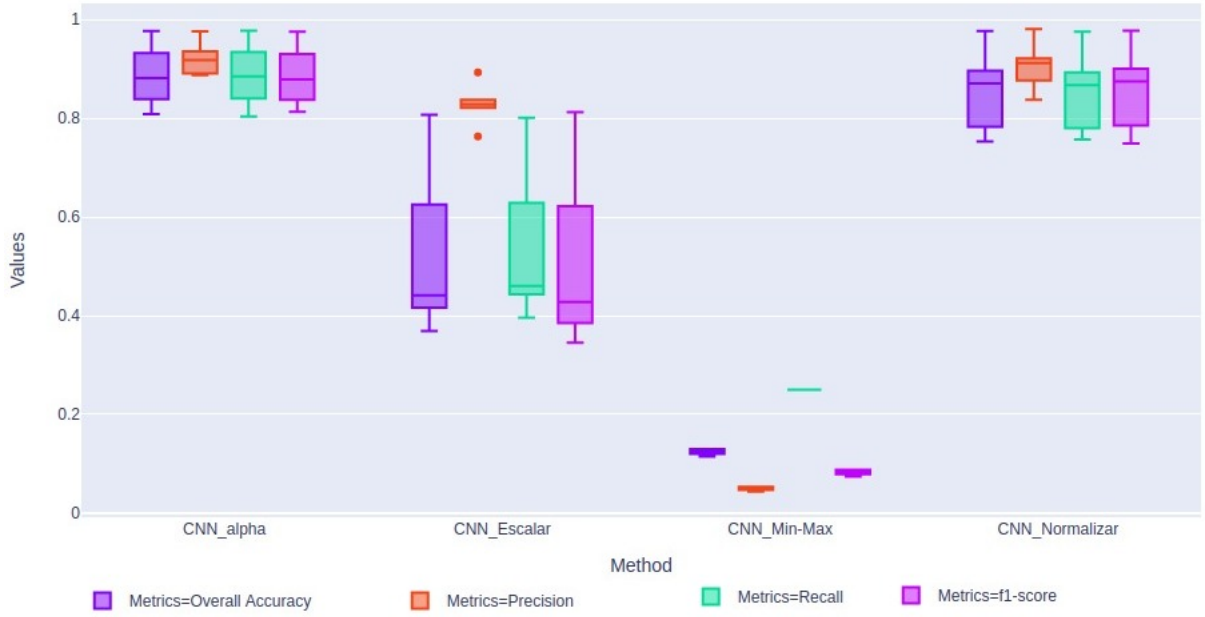



**Figura 5-14**.: Resultados de las pruebas con la red CNN con tres gestos además de la inactividad

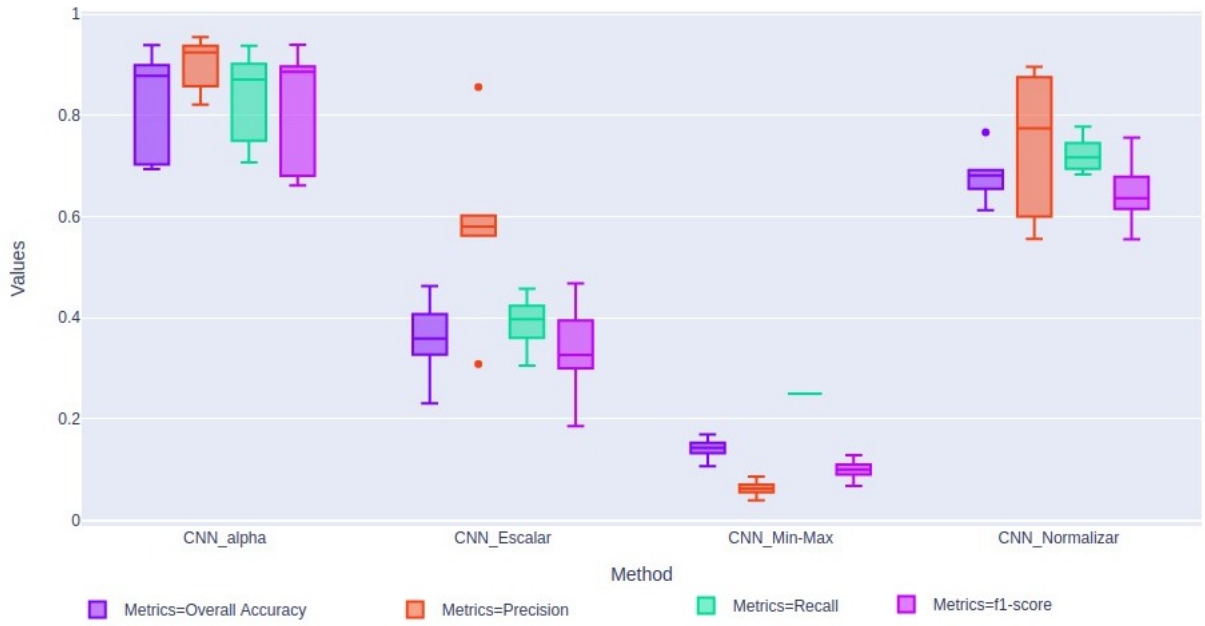
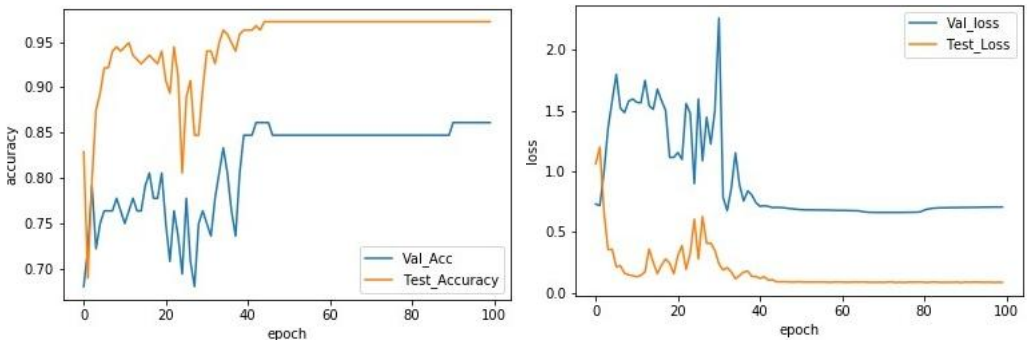



**Figura 5-15**.: Resultados de las pruebas con la red CNN con tres gestos además de la inactividad usando datos nunca vistos por el clasificador

### ¿Que gesto usar entre la mordida y el movimiento de cejas?

Como se muestra en la sección anterior, cuando se clasifican tres gestos se obtiene el mejor desempeño de la red. Determinar cual gesto entre la mordida y el movimiento de las cejas se realiza teniendo en cuenta su desempeño en con datos desconocidos, debido a que los mejores valores de exactitud con datos de validación obtenidos en los entrenamientos son similares, 0.8611 para la mordida (figura **5-16**) y 0.8666 para el levantamiento de las cejas.

En la figura **5-16** se pueden ver dos gráficos donde se expone el desempeño de la red durante el entrenamiento. Las curvas en amarillo representan el desempeño con los datos de entrenamiento, mientras que las que se muestran en azul hacen referencia al resultado de la red con los datos de validación. La imagen **5-16**-(a) evidencia el desempeño de la exactitud, esto representa el porcentaje de los datos clasificados correctamente; como se puede ver la tendencia a lo largo de las épocas es que la exactitud vaya aumentando hasta su punto más alto con un valor superior al 85 % con los datos de validación. En el caso de la imagen **5-16**-(b) se muestra la pérdida, la cual muestra la sumatoria de las distancias a las que se encuentran las predicciones de los valores verdaderos. La pérdida es un monitor inverso a la exactitud por lo que lo ideal es que su comportamiento sea inverso, tal como se ve en la figura **5-16**.




(a) Resultados de la exactitud en el entrenamiento y validación

(b) Resultados de la pérdida en el entrenamiento y validación

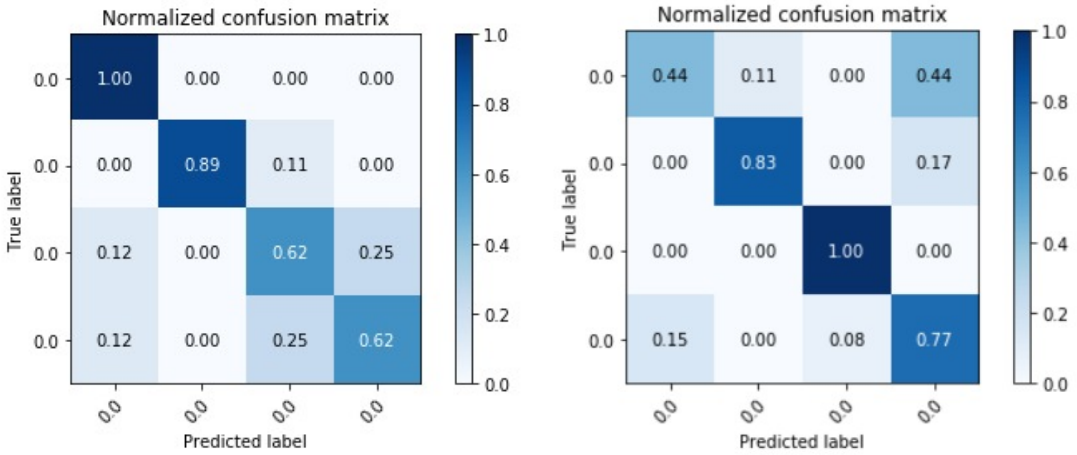
**Figura 5-16**.: Gráficas de los clasificadores CNN únicamente con el filtrado en el ritmo $\alpha$, en la imagen (a) se evidencia la exactitud alcanzada superior al 85 % mientras que en (b) se muestra los resultados de la pérdida. El resultado se obtiene al entrenar con el gesto de la mordida, sin embargo, al entrenar con el gesto de las cejas el resultado es bastante similar.

Para mostrar los resultados obtenidos se utilizan gráficas de los valores de los evaluadores exactitud y pérdidas, a lo largo del entrenamiento, mostrando el valor del entrenamiento y el de la validación; En la figura **5-16** se muestra en la imagen (b) que tras la época 40 la pérdida baja y después vuelve a subir levemente, como la pérdida de validación nunca toca

el de train pareciera que estuviera sub ajustado (underfit) pero al final se aleja del valor de train por lo que el modelo se comienza a sobre ajustar (overfit); para seleccionar el conjunto de pesos se elige el de mayor exactitud con la menor pérdida esto corresponde a uno entre la época 40 y 50. Para la respuesta ante datos desconocidos se muestra la matriz de confusión [3], donde las filas representan la verdadera etiqueta de los gestos y las columnas la clase donde el clasificador ubica los datos. De esta manera si se recorre la matriz a lo largo de las filas, se puede observar el porcentaje de los datos pertenecientes a una clase en particular, ubicados según el clasificador. Mientras que, al recorrer la matriz a lo largo de una columna, se puede observar la verdadera clase a la que pertenece, lo que el clasificador determina como una clase en específico. Con esto en mente un clasificador ideal, debería tener la diagonal principal con el 100 % de los datos, mostrando que el clasificador ubica en la clase correcta todos los datos. Para este trabajo las columnas y filas pertenecen a las clases mostradas en la tabla **B-1**, excepto en los resultados de tres gestos e inactividad, donde la columna y fila cuatro pertenecen a la inactividad en vez de a la mordida.

Como muestra la figura **5-17**, la respuesta ante datos totalmente desconocidos es buena en ambos casos en el caso de la mordida, el 76 % de los casos se clasifican de manera correcta, mientras que para las cejas 78.5 % de los datos. Como las cejas tienen el valor, estas son las elegidas para ser usadas en el modelo en tiempo real.

[3] Para hacer las matrices se utilizo el código para python suministrado en [sci, 2020]




(a) Matriz de confusión donde la tercera fila y columna pertenece al gesto de las cejas.

(b) Matriz de confusión donde la tercera fila y columna pertenece al gesto de la mordida.

**Figura 5-17**.: Matrices de confusión de los clasificadores CNN únicamente con el filtrado en el ritmo $\alpha$, donde de arriba hacia abajo y de izquierda a derecha la primera fila y columna pertenece al parpadeo con el ojo izquierdo, la segunda fila y columna es el parpadeo con el ojo derecho y la cuarta fila y columna es la inactividad y donde el gesto 3 se cambia, en la imagen (a) este gesto pertenece a el movimiento de las cejas, mientras que en (b) el gesto pertenece a la mordida

# 6. Manejo del Drone

A continuación, se muestra el proceso para adaptar el control de un cuadricoptero, para que funcione con señales EEG. Teniendo en cuenta que todo el proceso hasta ahora se ha desarrollado en python, se elige un drone, que sea compatible con este lenguaje, como lo es el Parrot Mambo a través de la librería pyparrot.

## 6.1. Parrot Drone Mambo

El Parrot Mambo es un cuadricoptero con un acelerómetro de tres ejes y un giroscopio de tres ejes, cuenta también con un sensor de ultrasonido y una cámara, el alcance varia entre los 20 y 100 metros dependiendo del dispositivo de control. Con una batería de 660mAh tiene una autonomía de aproximadamente 10 minutos sin accesorios y su carga tarda 30 minutos con un cargador de 2.1A. Vale la pena aclarar que no es un drone profesional, sus medidas son de 18x18 cuando tiene puestas las protecciones, siendo este vehículo, más parecido a un juguete.[SANTAMARIA, 2018].

## 6.2. Algoritmo diseñado para el manejo

Debido a que el clasificador obtiene su mejor desempeño con tres gestos, aparte de la inactividad, no se puede asignar un gesto específico a cada dirección de movimiento. Para garantizar que el drone se pueda mover en las tres dimensiones, se opta por una estrategia, en la que un gesto actúa como un interruptor con el cual se selecciona el eje en el que se quiere desplazar, mientras que, los otros dos gestos se encargan de dar la orden al drone de moverse hacia la dirección positiva o negativa.

El algoritmo comienza asumiendo que tanto el ultracortex como el drone, están correctamente conectados al computador [1], la figura **6-1** muestra el inicio del proceso, en el que se inicializan las variables que se van a usar, se solicitan los datos del clasificador y la altura, para posteriormente evaluar si el gesto pertenece al interruptor o a alguno de dirección.
Dado el caso que el gesto pertenezca al asignado como interruptor, se verifica el estado en el

[1] Para obtener la información de como conectar estos artículos al computador se sugiere ir a las paginas oficiales [OpenBCI, 2020] y [McGovern, 2018]

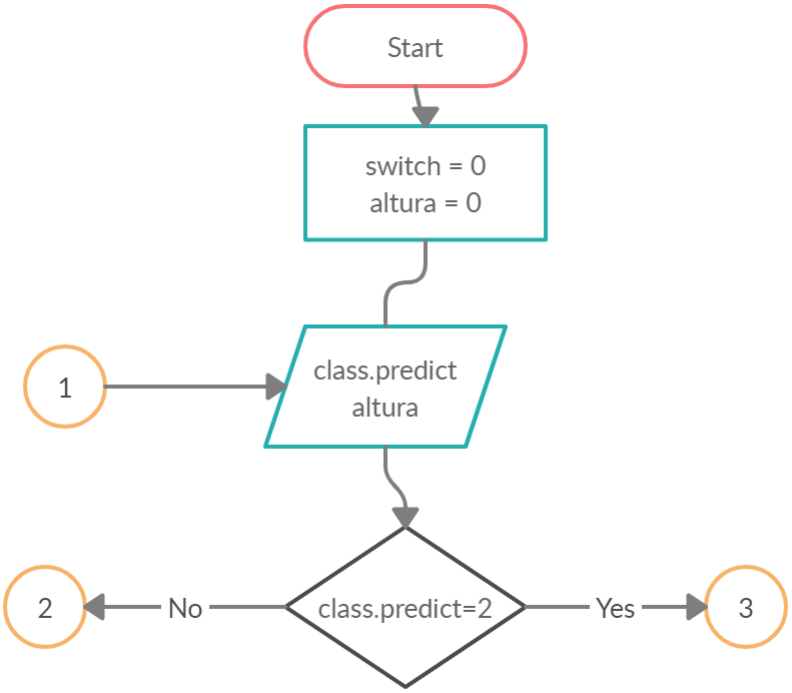



**Figura 6-1**.: Inicio del algoritmo

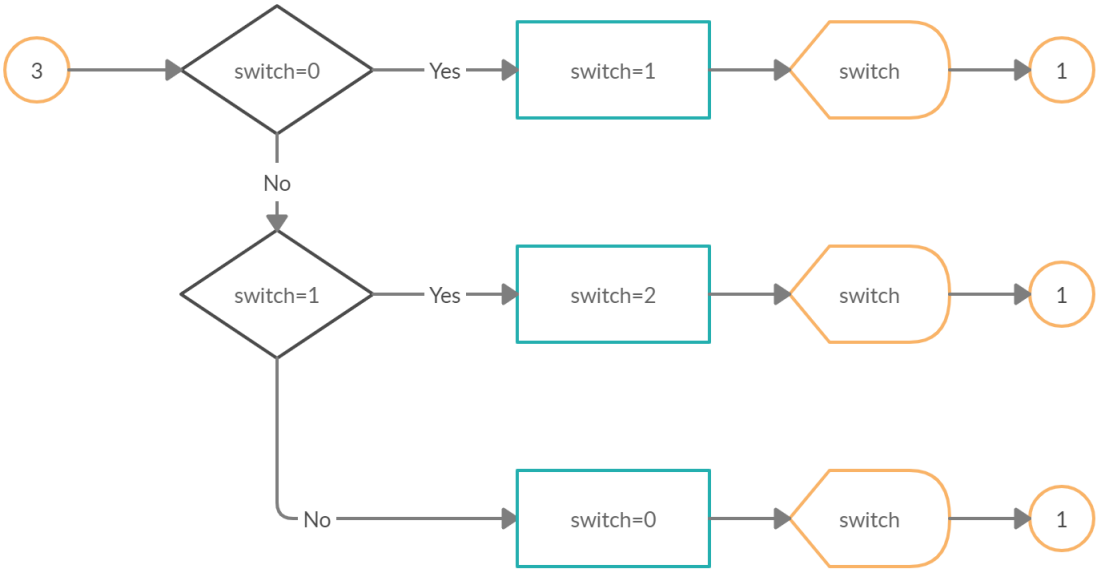



**Figura 6-2**.: Algoritmo del interruptor

que este se encuentra y lo cambia al siguiente mostrando qué movimiento se ha de habilitar para volver a preguntar por el siguiente gesto en llegar por el clasificador (figura **6-2**).

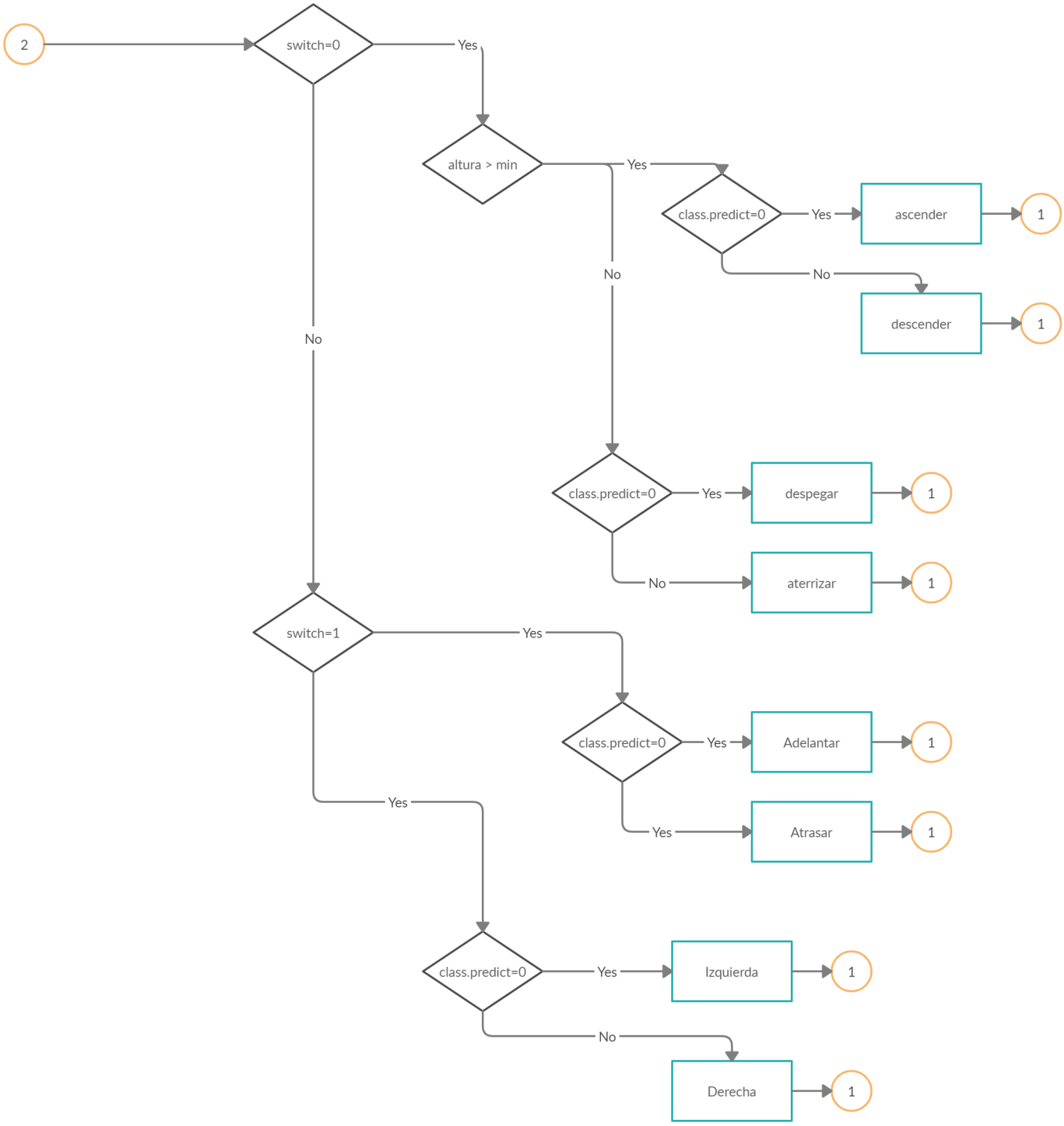



**Figura 6-3**.: Algoritmo de dirección

Por el contrario (figura **6-3**), si el gesto hace parte a uno perteneciente a la dirección, se pregunta primero por el estado del interruptor, para saber en qué eje se debe mover el drone y posteriormente preguntar por el gesto para ejecutar el movimiento correspondiente. En caso que el interruptor se encuentre en el eje vertical, se tienen dos opciones que se ejecutarán dependiendo de la altura a la que se encuentre el drone, despegar y aterrizar si se está cerca del piso, o, ascender y descender si por el contrario se está a cierta distancia del piso.

## 6.3. Implementación del algoritmo

Para unir el ultracortex, el clasificador y el algoritmo de control del drone, se crea un GUI (figura **6-4**), basado en el que se utiliza para tomar las señales del entrenamiento. La principal diferencia en la estructura del código, radica en que se elimina todo lo relacionado con las gráficas de las señales EEG para reemplazarlo por las clasificaciones de las señales y dar las ordenes al drone. Naturalmente, todo lo relacionado con guardar datos de las señales EEG y las EMG no es usado, por ende se elimina.

El GUI se diseña utilizando *QtDesigner*, lo que genera un archivo con la extensión "*.ui*", y posteriormente se pasa a Python ("*.py*") usando las funciones que trae consigo la librería PyQt5. Además de eso, también es necesario pasar las figuras del formato "*.qrc.*[a]l usado en python utilizando otra función de la misma librería ya nombrada. Para hacer esa conversión, se abre la terminal en la carpeta donde están los archivos y posteriormente se ejecutan los siguientes comandos: *pyuic5 [nombre del archivo].ui -o [nombre del archivo].py* para la conversión del archivo de todo el GUI, y *pyrcc5 -o [nombre del archivo]_rc.py [nombre del archivo].qrc* para los archivos generados para las imágenes.

Una vez se tiene el archivo en python funcionando de manera correcta, sobre este se comienza a desarrollar el resto de funciones implementadas para que funcione como se desea. Lo primero en implementarse son las funciones necesarias para conectar el ultracortex (función que también está implementada en la interfaz de la toma de datos) además, de las variables universales que son necesarias para su correcto funcionamiento. Se crea la red CNN en el código y se cargan los pesos con mejor desempeño en el entrenamiento. El modelo también se define como global. El botón se utiliza para ejecutar la conexión del drone, y la función *updater* del código referencia, se altera, como se menciono anteriormente, para que en vez de mostrar las gráficas, se encargue de ejecutar el algoritmo explicado en la sección anterior.

## 6.4. Ejecución del GUI

Uno de los principales problemas para el correcto desempeño de la aplicación, recae en la toma y predicción de datos en tiempo real, ya que se corre el riesgo que se realice el gesto

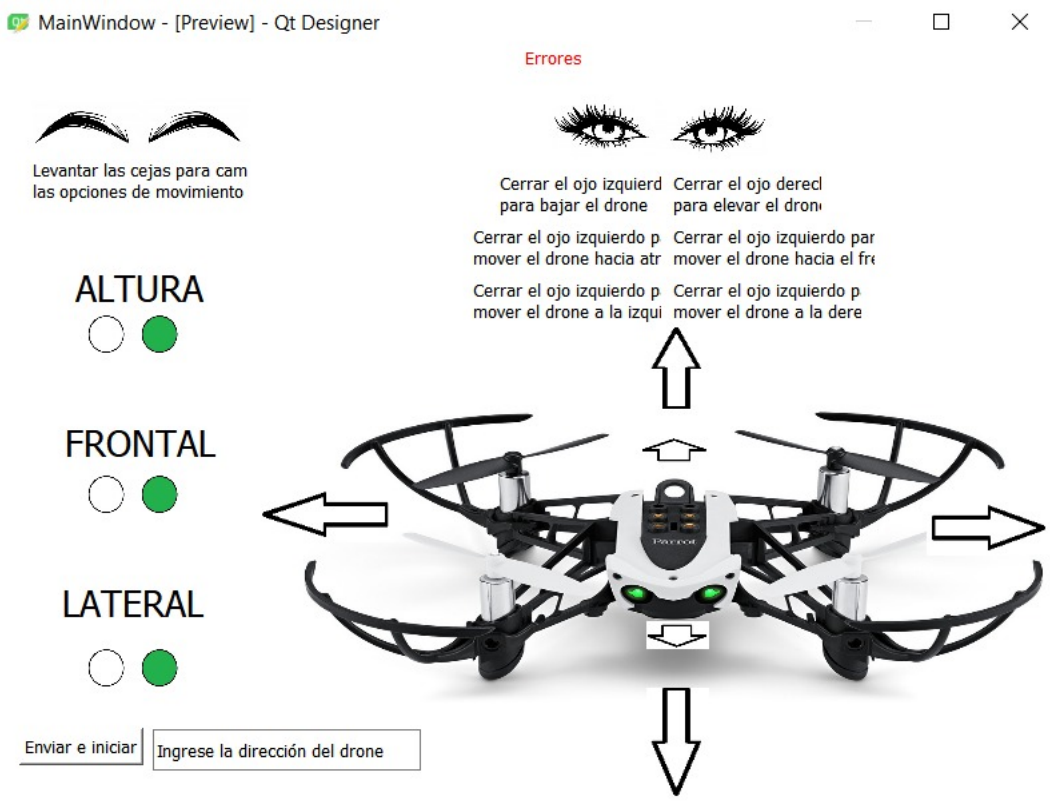



**Figura 6-4**.: Interfaz gráfica de usuario para el control del drone con señales EEG

al final del tiempo que analiza la red convolucional y parte de la información se corte dando una mala predicción. Para solucionar ese problema se decide tomar datos en un determinado tiempo, avisando al usuario mediante una bandera la cual se puede ver mediante un mensaje en la consola. El tiempo estipulado es de dos segundos, pero la variable guarda los últimos ocho segundos, luego se filtran digitalmente en el ritmo $\alpha$, y se elimina el equivalente a los primeros tres segundos de la variable, para evitar que el tiempo de establecimiento del filtro afecte la red. Las 625 muestras restantes en la variable son las que se ingresan al clasificador.

Otro inconveniente, es el hecho que al ejecutar el GUI en Windows, el drone deja de ejecutar órdenes apenas se establece la comunicación con las tarjetas Cyton y Deisy. Al desconocer la razón de esto, se opta por probar con dos diferentes opciones, la opción uno es ejecutar la predicción y la conexión con el casco como un subproceso de un GUI que controle el drone (opción que busca solucionar el problema en Windows); la segunda opción se resume en probar el código original en un sistema operativo basado en Linux. [2]

### 6.4.1. Subproceso en Windows

Para ejecutar la clasificación de gestos como un subproceso en Windows, se tiene un GUI para controlar el drone desde el computador, desde esa interfaz se inicia la comunicación con el casco y el clasificador en segundo plano, este último escribe los resultados en una variable o en un *.txt* que es leído por el código del manejo del drone y posteriormente ejecutado.

Desafortunadamente este método no presenta ninguna mejora respecto al problema inicial, de hecho en ocasiones presenta más inconvenientes debido a que cerrar la aplicación en segundo plano no siempre se realiza de la manera correcta, lo que termina ocupando espacio en la memoria RAM.

### 6.4.2. Ejecución en Linux

Para el correcto funcionamiento en Ubuntu 20.04 (sistema operativo basado en Linux) del GUI es necesario cambiar el código de conexión del casco. Dado que en Windows se utiliza: *board = OpenBCICyton(port='COM*')* el cual es cambiado por:*board = OpenBCICyton(port='/dev/ttyUSB*')*.

Además, antes de ejecutar el programa en Linux es necesario habilitar los puertos para que se puedan usar, para esto en el terminal se ejecuta el comando *lsusb** el cual muestra una lista de dispositivos entre los que es necesario identificar el Dongle de las tarjetas Cyton y Deisy. Una vez identificado, se ejecuta el comando *sudo modprobe usbserial vendor=0xXXXX*

[2]El GUI se puede encontrar para descargar en la siguiente URL: `https://github.com/IvanDanielVillegas/EEG_Drone_Control`

*product=0xXXX* reemplazando las XXXX de *vendor* por los primeros cuatro números del ID que se mostraron en la lista del comando anterior y las XXXX de *product* por los últimos cuatro números del ID. Después se ejecuta el comando *ls -l /dev/ttyUSB** el cual permite ver el ttyUSB para introducir en la linea de codigo reemplazada, pero para habilitar el puerto es necesario ejecutar el comando *udo chmod 777 /dev/ttyUSBX* donde la X se reemplaza por el número del puerto que haya mostrado el comando anterior. A continuación un ejemplo:

```
$ lsusb*
Bus 002 Device 003: ID 84d0:0052 [Nombre del dispositivo (DONGLE)]
.
.
.
$ sudo modprobe usbserial vendor=0x84d0 product=0x0052
$ ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 may 29 09:17 /dev/ttyUSB
$ sudo chmod 777 /dev/ttyUSB0
$ ls -l /dev/ttyUSB*
crwxrwxrwx 1 root dialout 188, 0 may 29 09:23 /dev/ttyUSB0
.
```

Al probar con Ubuntu, se evidencia un comportamiento similar al obtenido al ejecutar el código en Windows. Aunque el drone responde a ciertas órdenes cuando se realiza la conexión con las tarjetas, no siempre las realiza de la manera indicada, y en ocasiones mantiene comunicación únicamente con uno de los elementos, dejando totalmente de lado el otro. Sin embargo, si se ejecuta primero un programa para controlar el drone desde el computador para hacer que el drone levante vuelo, y luego el GUI desarrollado, el drone parece responder a algunos comandos enviados desde el GUI.

Para solucionar estos problemas, en el GUI se hace que apenas este se empieza a ejecutar el drone levante vuelo, y tras ajustar los tiempos de los hilos que se encargan de realizar las predicciones de los gestos tomados por el ultracortex y del hilo de comunicación con el drone (cada 4 segundos y cada dos segundos respectivamente) el parrot mambo responde a las órdenes que se le son asignadas dependiendo de las predicciones de las señales capturadas por el módulos BCI.

# 7. Análisis de resultados

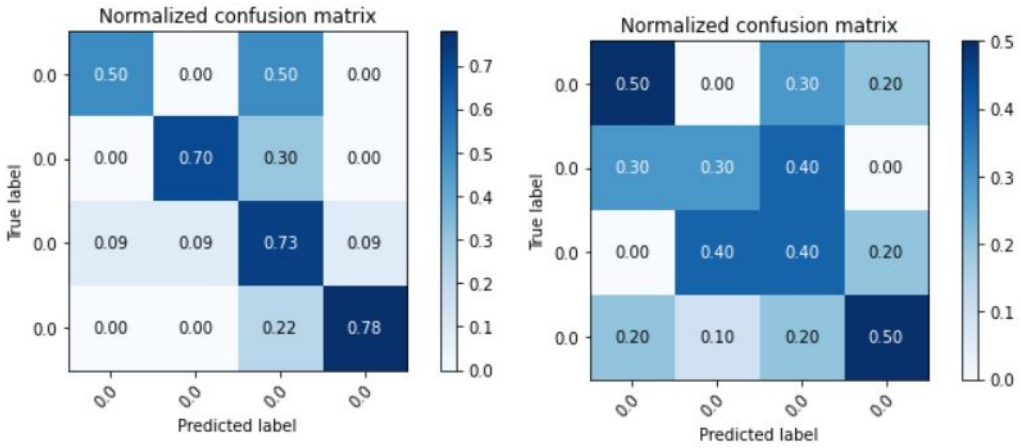
El objetivo del cambio en la distribución de los electrodos en la cabeza, es abarcar la mayor cantidad de información referente a las acciones motoras, aprovechando una característica de la electroencefalografía relacionada con la superposición de las señales EEG en los electrodos debido a que estos están sobre la superficie del cuero cabelludo y no directamente sobre la corteza cerebral. Al rodear todas las zonas relacionadas con el movimiento con electrodos, la posibilidad de conseguir información relevante de una región adyacente aumenta. Sin embargo, no todos los electrodos contienen datos útiles para el desarrollo del proyecto, por lo que se puede inferir que tal cantidad de electrodos no es necesaria. Al reducir la cantidad de electrodos empleados se puede simplificar el clasificador, y puede dar lugar a un dispositivo de electroencefalografía más cómodo, si se piensa en una posterior aplicación comercial.

Aunque la literatura recomienda una serie de métodos para extraer características de las señales EEG que se pueden usar en la imaginación motora, los datos categóricos en el dominio del tiempo no son suficientemente diferentes como para ayudar a diferenciar entre gestos, aplicando los métodos mencionados a toda una señal EEG filtrada en la banda $\alpha$. La razón de la poca fiabilidad de estas características puede radicar en que han sido aplicados a toda una grabación en la que la mayoría del tiempo había inactividad motora, tardando un gesto un tiempo máximo de 1.5 segundos en ser realizado, mientras que la grabación total dura cinco segundos. Seleccionando los intervalos relacionados exclusivamente a los gestos motores puede que se logre encontrar utilidad en los datos categóricos en el tiempo.

Aunque los procesos de estandarización han servido en el caso de RF para mejorar el desempeño de las redes, en el CNN su uso disminuye la exactitud. Esto puede deberse a la arquitectura de la red convolucional, donde se utiliza un maxpool después de los filtros, el cual toma los datos de mayor valor y los pasa a la siguiente capa. Al dejar todos los valores de los diferentes electrodos en intervalos muy cercanos, la red tiende a confundir los gestos puesto que hay regiones del encéfalo que son más activas eléctricamente que otras cuando se realizan ciertos gestos.

Al observar el desempeño del clasificador en tiempo real se aprecia un decremento en su confiabilidad respecto a las pruebas realizadas con el conjunto de datos utilizados para el test durante el entrenamiento como se puede apreciar en las figuras **7-1** y **7-2**. Durante el proceso de entrenamiento de la red, se guardaron los pesos de los dos mejores resultados

supervisando que el valor de *val_acc* fuera superior al 80 % y luego se observo su comportamiento con los datos desconocidos para la red, el resultado se muestra en las imágenes (a) de las figuras **7-1** y **7-2**, donde si bien el resultado no es perfecto, es adecuado para implementar en algoritmo para enviar las órdenes al drone. Los pesos guardados se cargan en el GUI que permite evaluar el desempeño en tiempo real, donde se realizan sucesivamente la ejecución de gestos y se observa la salida del clasificador, los resultados de estas pruebas se muestran en las imágenes (b) de las figuras **7-1** y **7-2**.

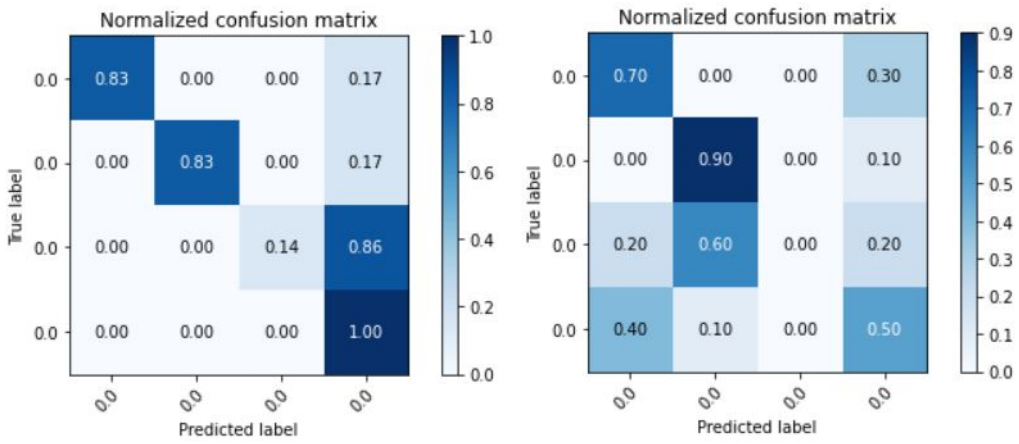



(a) Matriz de confusión con pruebas de la red off-line

(b) Matriz de confusión con pruebas de la red on-line

**Figura 7-1**.: Comparación de las matrices de confusión con el conjunto de pesos 1 obtenidos. La imagen (a) pertenece a la evaluación utilizando los datos de test de la base de datos, mientras que (b) se obtiene de pruebas en tiempo real.

De las figuras **7-1** y **7-2** se extrae la información mostrada en la tabla **7-1**, donde se comparan las métricas de los experimentos con dos pesos diferentes en su desempeño tanto en tiempo real como con datos grabados. Como se hace evidente los resultados son mejores con el segundo conjunto de pesos, teniendo un problema para la implementación en tiempo real, ya que como muestra la métrica *recall* es muy baja en un caso en particular; los levantamientos de cejas, mostrados como el gesto 3, el cual se encarga de variar el eje de desplazamiento siempre es un falso negativo, es decir que siempre es confundido con otro gesto, y como la *precision* del mismo gesto es tan alta, lo cual significa que ningún otro gesto se clasifica en esta clase, resulta en que sea casi imposible cambiar el eje de movimiento en tiempo real. Este problema también se evidencia en las pruebas off-line, sin embargo, en estas aún parece posible reconocer el gesto en algunas ocasiones. El principal problema de el primer conjunto de pesos, resulta ser el bajo valor de sus métricas las cuales hacen que sea difícil controlar el drone, ya que las órdenes que este reciba van a ser incorrectas en más de la mitad de los casos.




(a) Matriz de confusión con pruebas de la red off-line

(b) Matriz de confusión con pruebas de la red on-line

**Figura 7-2**.: Comparación de las matrices de confusión con el conjunto de pesos 2 obtenidos. La imagen (a) pertenece a la evaluación utilizando los datos de test de la base de datos, mientras que (b) se obtiene de pruebas en tiempo real.

| | Off-Line | | | | On-Line | | | |
|---|---|---|---|---|---|---|---|---|
| Pesos | Metricas | Presicion | Recall | F1-Score | Metricas | Presicion | Recall | F1-Score |
| 1 | Gesto 1 | 0,847 | 0,50 | 0,629 | Gesto 1 | 0,500 | 0,500 | 0,500 |
| | Gesto 2 | 0,89 | 0,7 | 0,782 | Gesto 2 | 0,375 | 0,300 | 0,333 |
| | Gesto 3 | 0,417 | 0,730 | 0,531 | Gesto 3 | 0,308 | 0,400 | 0,348 |
| | Gesto 4 | 0,897 | 0,780 | 0,834 | Gesto 4 | 0,556 | 0,500 | 0,526 |
| | Promedio | 0,762 | 0,678 | 0,694 | Promedio | 0,435 | 0,425 | 0,427 |
| 2 | Gesto 1 | 1 | 0,83 | 0,907 | Gesto 1 | 0,538 | 0,7 | 0,609 |
| | Gesto 2 | 1 | 0,83 | 0,907 | Gesto 2 | 0,5625 | 0,9 | 0,692 |
| | Gesto 3 | 1 | 0,14 | 0,246 | Gesto 3 | 1 | 0 | 0 |
| | Gesto 4 | 0,455 | 1 | 0,625 | Gesto 4 | 0,455 | 0,5 | 0,476 |
| | Promedio | 0,864 | 0,700 | 0,671 | Promedio | 0,639 | 0,525 | 0,445 |

**Tabla 7-1**.: Métricas obtenidas de las matrices de confusión de las figuras **7-1** y **7-2**, donde se comparan los valores de las pruebas realizada con grabaciones de la base de datos con las pruebas realizadas en tiempo real obtenidos con dos diferentes conjuntos de pesos.

Con el 42.5 % de los datos con conjunto del primer conjunto de pesos y el 52.5 % de los datos con el segundo conjunto de pesos clasificados correctamente, el clasificador RF parece una opción a probar en tiempo real. Otra alternativa para mejorar el desempeño en tiempo real, es hacer más robusta la red CNN donde se agreguen más capas.

Para conocer qué electrodos brindan mayor información a la red se adecua el clasificador convolucional para funcionar con dos electrodos menos. El procedimiento a seguir es eliminar pares de electrodos del entrenamiento para observar con cuales se obtiene el mayor desempeño, siendo estos los que menos información útil aportan. La tabla **C-1** (esta se encuentra en el anexo C) muestra los resultados obtenidos al quitar los respectivos pares de electrodos. La tabla **7-2** muestra el tratamiento estadístico de los datos de la tabla **C-1**, mostrando el promedio y la desviación estándar; al observar el promedio de la exactitud se puede determinar cual par de electrodos aportan una mayor cantidad de información, estos se muestran en la tabla **7-2** de mayor a menor información suministrada acorde con la exactitud obtenida. Debido a que el clasificador se configura para que su referencia sea la exactitud, tiene sentido que la desviación estándar de esta sea muy baja, contrario de lo que sucede en la pérdida cuya desviación estándar es mayor.

| Electrodos Retirados | Promedio de la exactitud [%/100] | Desviación estándar de la exactitud | Promedio de la perdida | Desviación estándar de la perdida |
|---|---|---|---|---|
| Fp1-Fp2 | 0,524 | 0,045 | 50,793 | 91,180 |
| Cp1-Cp2 | 0,597 | 0,060 | 7,109 | 3,948 |
| C3-C4 | 0,614 | 0,031 | 5,893 | 1,795 |
| T3-T4 | 0,614 | 0,068 | 5,121 | 2,527 |
| F7-F8 | 0,618 | 0,026 | 4,143 | 3,062 |
| Cp5-Cp6 | 0,642 | 0,055 | 4,099 | 3,478 |
| Fc1-Fc2 | 0,666 | 0,071 | 3,155 | 1,939 |
| Fc5-Fc6 | 0,670 | 0,038 | 4,321 | 2,131 |

**Tabla 7-2**.: Promedio y desviación estándar de los datos mostradas en la tabla **C-1** obtenidos para determinar la importancia de cada par de electrodos en el modelo de clasificador organizados en orden de importancia según la información que brindan.

La unificación del control del drone con la clasificación en tiempo real no ha sido posible en el sistema operativo Windows; en Ubuntu un sistema operativo basado en Linux es posible conectar las tarjetas Cyton, Deisy y el drone parrot mambo a través del computador usando la interfaz desarrollada en este proyecto donde cada cuatro segundos se pregunta por el gesto que se desea realizar y se da un tiempo de dos segundos para realizarlo.

# 8. Conclusiones

La distribución de electrodos que se sugiere para el presente proyecto, donde se ubican 12 electrodos alrededor de la región motora de la corteza cerebral y 4 cerca a la frente, cumple con el objetivo por el que se sugiere, ya que permite detectar la información necesaria para clasificar con cierta precisión gestos motores relacionados con los movimientos de los órganos faciales, como se muestra en la tabla **7-2** al no haber ningún caso en que se alcance un valor de precisión significativamente alto, evidencia que todos los electrodos aportan información; sin embargo al estar los electrodos tan juntos puede darse el caso de redundancia en la información, por lo que se abre la posibilidad que no sean necesarios todos los electrodos para clasificar los gestos que se trabajan en este documento, esto se dice basado en que hay ciertos pares de electrodos que aportan más información que otros como lo son los ubicados en Fp1, Fp2, Cp1, Cp2, C3, C4, T3 y T4. Vale la pena agregar que los cuatro electrodos que se cambiaron de posición, Cp5, Cp6, Fc5 y Fc6, están ubicados en los que menos información aportan, por lo que el cambio puede ser poco significativo para la red CNN.

Aunque no se realizaron pruebas filtrando la señal en otro rango de frecuencias diferente al de las ondas $\alpha$, si se considera necesario realizar el filtrado de la señal para que el clasificador realice de manera adecuada su trabajo. Como muestran los resultados del clasificador CNN las señales filtradas son suficientes para realizar de manera adecuada la clasificación, pero no se considera que sea algo que se puede obviar, debido a que en los otros clasificadores que se probaron, los pre-procesamientos para dejar las señales en un rango de valores similar son necesarios para mejorar el desempeño de la red, sin un filtrado adecuado la red va a ser incapaz de diferenciar los electrodos donde hay actividad cerebral de aquellos en los que solo hay ruido, ya que todos los canales tienen valores similares, además facilita diferenciar las señales EEG de las señales EMG y evita que estas sean confundidas entre sí.

Realizar un proceso donde los valores de las señales presentan valores cercanos (especialmente entre los valores fijos cero y uno) presenta mejoras en los clasificadores RF y KNN, donde se puede apreciar mejoras respecto a la señal que solo ha sido filtrada. No así en el clasificador CNN donde el hecho que las señales tengan valores similares empeora el rendimiento de la red.

Entre los paradigmas evaluados, los que mejor comportamiento tienen son el RF (con los datos filtrados en el ritmo $\alpha$ y posteriormente escalados entre valores de cero y uno) donde se alcanza una precisión cercana al 85 % y la red CNN (Con los datos filtrados en el rit-

mo $\alpha$) donde se alcanza una precisión del 86,66 %, la diferencia entres los modelos radica principalmente en su respuesta ante datos desconocidos, donde la red CNN tiene un mejor desempeño logrando un 78.25 % de precisión con datos totalmente desconocidos.

Para implementar el clasificador en tiempo real, es necesario indicar al usuario en que intervalo de tiempo puede realizar el gesto, esto debido a que pueden realizarse varios gestos en el mismo intervalo de tiempo que confundan la red, lo que reduce su desempeño además esto evita que el usuario se confunda debido al retardo entre sus órdenes y lo mostrado en el GUI. Sin embargo es importante mencionar que el clasificador presenta una baja en su rendimiento respecto a lo mostrado en el entrenamiento y testeo del mismo en tiempo real, situando el clasificador RF y CNN con datos normalizados como opciones a tener en cuenta debido a que durante el entrenamiento este mostró un buen desempeño y valdría la pena evaluar su comportamiento en tiempo real.

El desarrollo del GUI facilita verificar que el algoritmo y el clasificador funcionan de manera aceptable, ya que la pantalla permite ver las acciones que se deberían realizar, las cuales en la mayoría de los casos son las deseadas, y los casos en donde no, son tomadas como inactividad o como un cambio en el eje del movimiento, lo cual no implica movimiento del drone y no pone en peligro la integridad del dispositivo, sin embargo esto no siempre ocurre debido a la disminución en la precisión del clasificador el parpadeo del ojo izquierdo y el levantamiento de cejas aveces son confundidos con el parpadeo derecho.

Aunque no se logra que el drone tenga los seis grados de libertad que tiene por defecto, debido a que el “yaw” no se ha implementado, el drone se puede mover a lo largo de las tres dimensiones usando el algoritmo creado el cual aprovecha el pitch y roll para el desplazamiento a lo largo de todo el espacio.

# A. Anexo: Base de datos usada para el entrenamiento de los clasificadores.

https://github.com/IvanDanielVillegas/Data-Base-EEG-face-und-arms-Movement

El anterior link conduce a la base de datos usada para el entrenamiento de una red convulucional que clasifique entre gestos motores realizados por diferentes personas. La base de datos no está limpia, por lo que hay grabaciones que pueden no contener información útil, además todas las grabaciones se encuentran filtradas entre los 8 y 13 Hz (ritmo alpha). Los gestos son parpadeo del ojo izquierdo en las carpetas llamadas gesto 1, parpadeo del ojo derecho en las carpetas gesto 2, levantamiento de cejas en gesto 3, mordida en gesto 4, inactividad motora en gesto 5, levantamiento del brazo izquierdo en gesto 6, levantamiento del brazo derecho en gesto 7.

# B. Anexo: Gráficas de precisión y perdida, además de las matrices de confusión de las pruebas realizadas con las redes CNN

## B.1. Gestos en la matriz de confusión

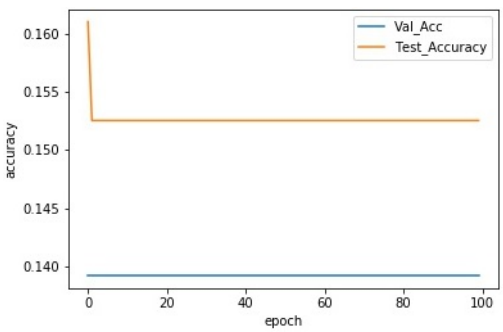
La tabla **B-1** muestra a que pertenece cada fila y columna de las matrices de confusión. En el caso de los resultados con tres gestos, el gesto cuatro es reemplazado por el cinco, esto significa que la cuarta fila y columna no pertenecen a la mordida sino a la inactividad.

| Filas y columnas | Gesto |
|---|---|
| 1 | Parpadeo con el ojo izquierdo |
| 2 | Parpadeo con el ojo derecho |
| 3 | Levantamiento de cejas |
| 4 | Mordida |
| 5 | Inactividad |
| 6 | Levantamiento de la mano izquierda |
| 7 | Levantamiento de la mano derecha |

**Tabla B-1**.: Distribución de las clases en las matrices de confusión

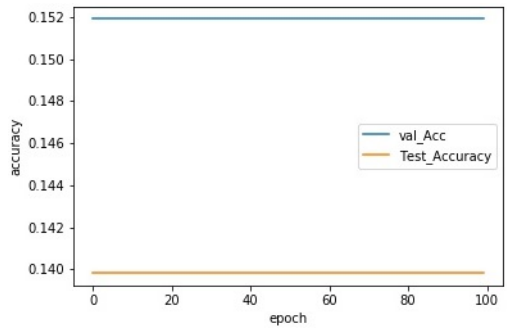
## B.2. Modelo para seis gestos e inactividad

**B-1**,**B-2**,**B-3** y **B-4**, la red fue incapaz de clasificar de manera correcta, ni siquiera se presentaron cambios en la precisión a lo largo del entrenamiento, y como muestran las matrices de confusión, los datos desconocidos los etiqueta todos en una sola clase.



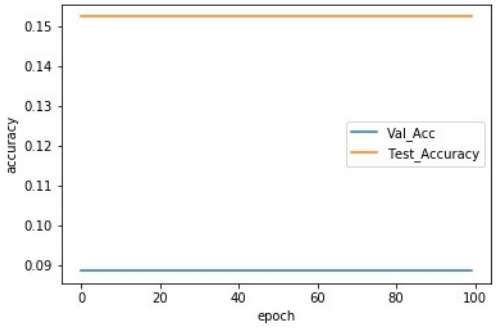

(a) Gráfica de la precisión (b) Matriz de confusión

**Figura B-1**.: Resultados del entrenamiento de la red CNN con seis gestos e inactividad sin pre-procesamiento, únicamente con el filtrado en el ritmo $\alpha$




(a) Gráfica de la precisión (b) Matriz de confusión

**Figura B-2**.: Resultados del entrenamiento de la red CNN con seis gestos e inactividad con los datos escalados

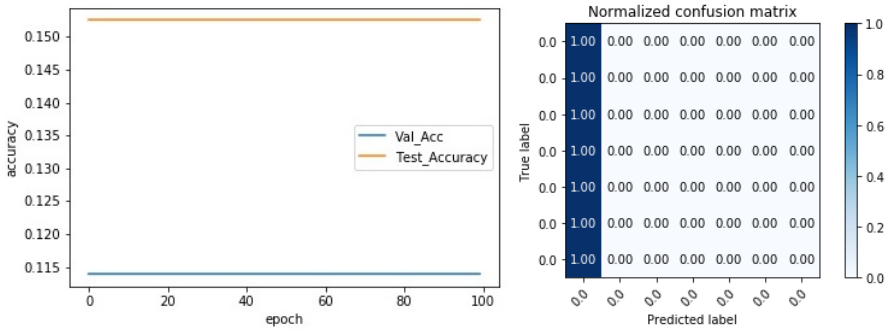



(a) Gráfica de la precisión (b) Matriz de confusión

**Figura B-3**.: Resultados del entrenamiento de la red CNN con seis gestos e inactividad con los datos escalados entre valores fijos 0 y 1

## B.3. Modelo para cuatro gestos e inactividad

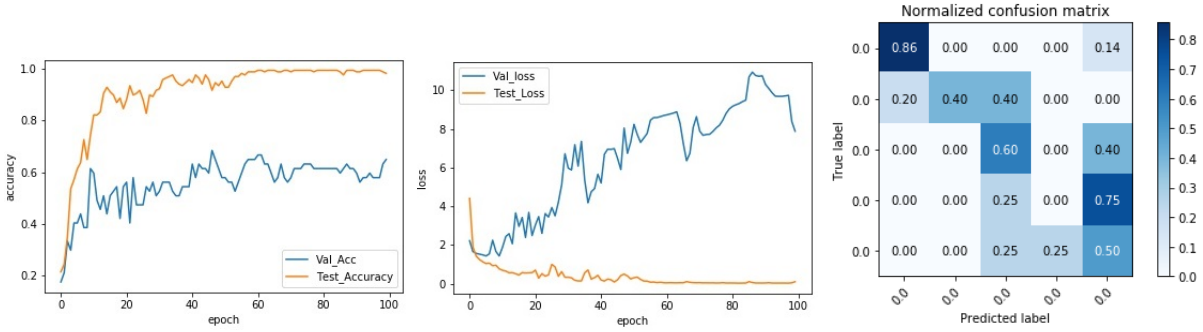
Al eliminar los gestos seis y siete, los cuales corresponden al levantamiento de los brazos, y se entrena la red con los demás gestos obteniendo mejores resultados. Esto se muestra en las




(a) Gráfica de la precisión (b) Matriz de confusión

**Figura B-4**.: Resultados del entrenamiento de la red CNN con seis gestos e inactividad con los datos Normalizados

graficas **B-5**, **B-6**, **B-7** y **B-8**.

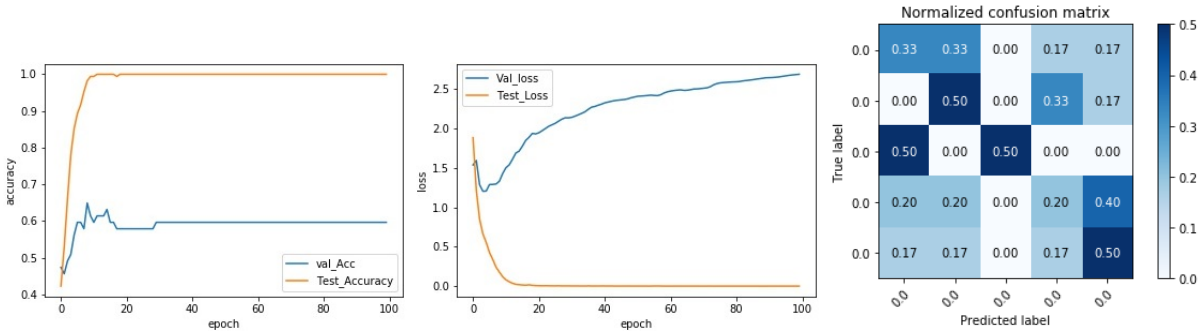



(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-5**.: Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad sin pre-procesamiento, únicamente con el filtrado en el ritmo $\alpha$

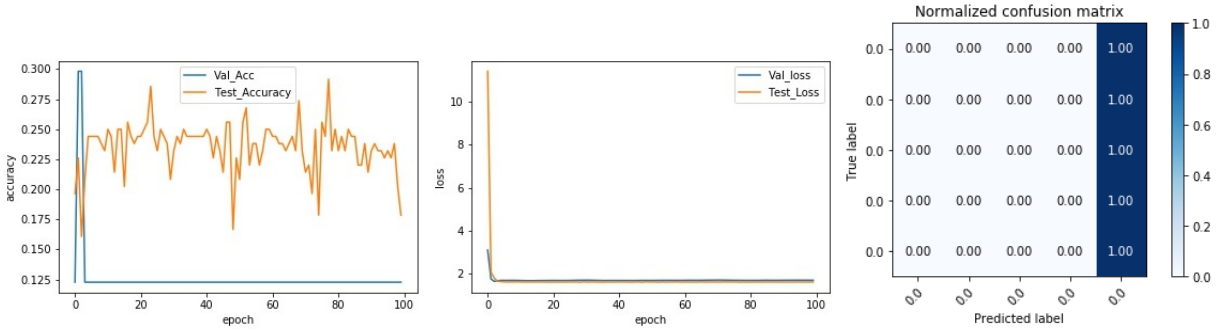
## B.4. Modelo para tres gestos e inactividad

De igual forma que se hizo con los modelos base, se retira un gesto más con ánimo de ganar precisión, y se entrena la red con los conjuntos de datos ya mencionados. Los resultados se muestran en las figuras **B-9**, **B-10**, **B-11** y **B-12**.



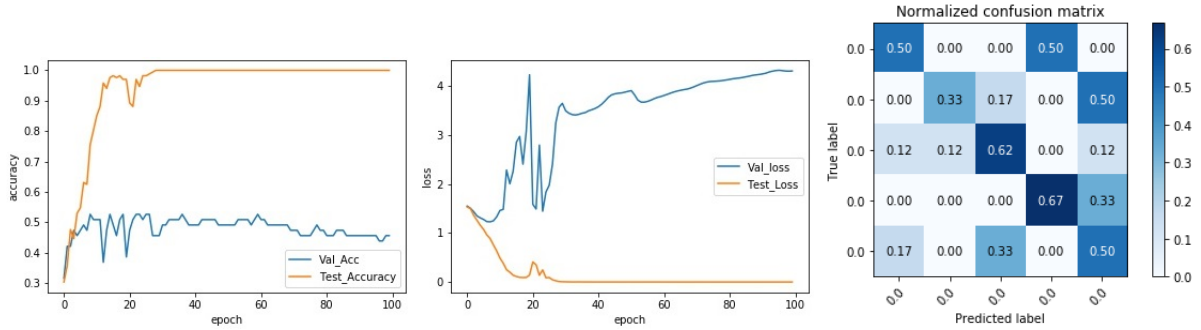

(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-6**.: Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad con los datos escalados



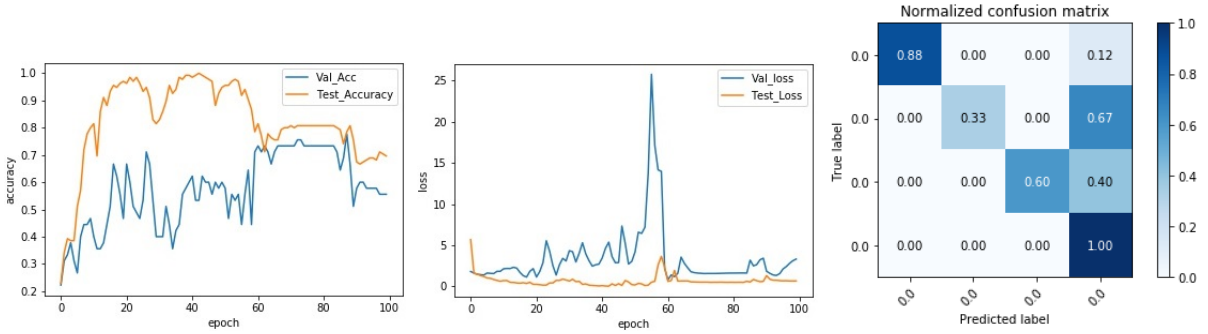

(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-7**.: Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad con los datos escalados entre valores fijos 0 y 1



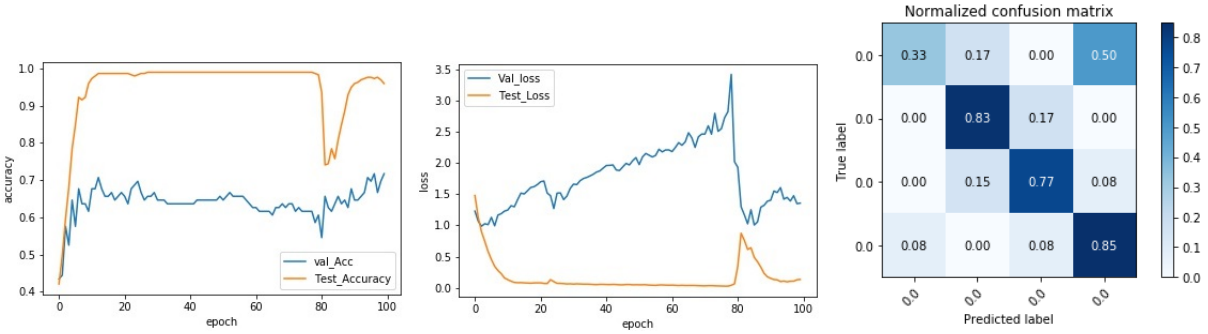

(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-8**.: Resultados del entrenamiento de la red CNN con cuatro gestos e inactividad con los datos Normalizados



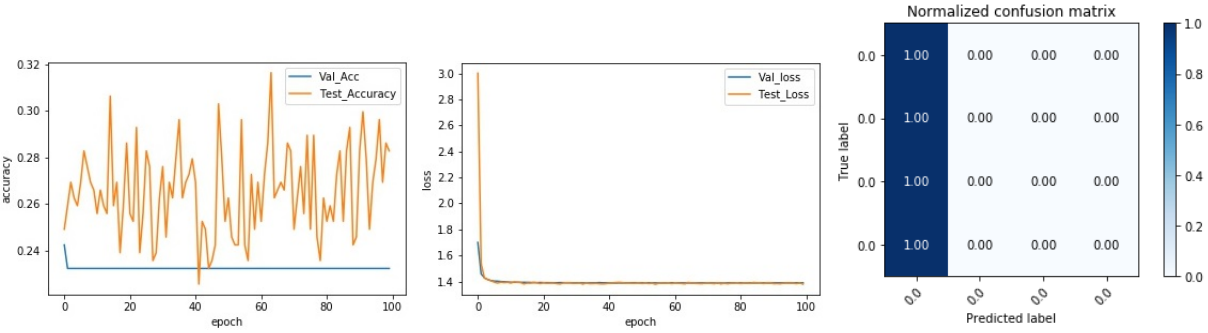

(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-9**.: Resultados del entrenamiento de la red CNN con tres gestos e inactividad sin pre-procesamiento, únicamente con el filtrado en el ritmo $\alpha$



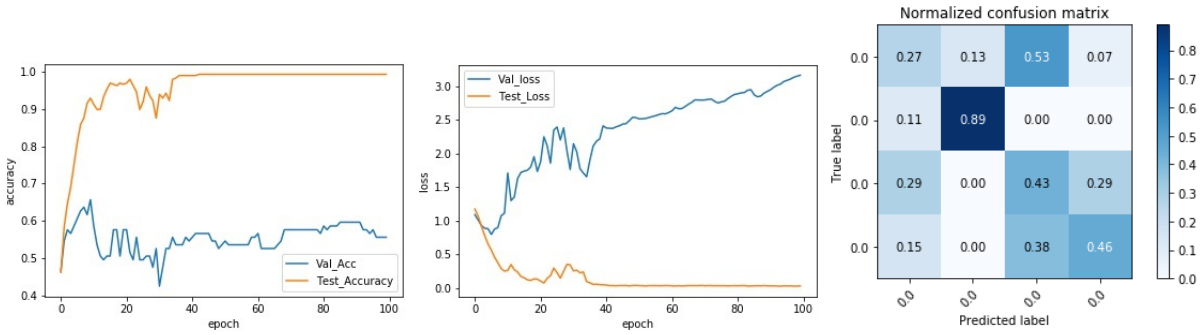

(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-10**.: Resultados del entrenamiento de la red CNN con tres gestos e inactividad con los datos escalados




(a) Gráfica de la precisión (b) Gráfica de la perdida estandarizada (c) Matriz de confusión

**Figura B-11**.: Resultados del entrenamiento de la red CNN con tres gestos e inactividad con los datos escalados entre valores fijos 0 y 1




(a) Gráfica de la precisión

(b) Gráfica de la perdida estandarizada

(c) Matriz de confusión

**Figura B-12**.: Resultados del entrenamiento de la red CNN con tres gestos e inactividad con los datos Normalizados

# C. Anexo: Pruebas realizadas para determinar la información suministrada por cada par de electrodos

En la tabla se muestran las pruebas para determinar la información suministrada por cada par de electrodos, retirando dos canales de entrada de la red, los cuales pertenecen a unos determinados pares de electrodos según muestra la tabla **3-1**.

| Canales retirados | Precisión de validación | Perdida de Validación |
|---|---|---|
| 1-2 | 0.5833 | 7.4569 |
| | 0.5139 | 18.7542 |
| | 0.5278 | 4.6776 |
| | 0.4722 | 3.4962 |
| 3-4 | 0.5833 | 5.1458 |
| | 0.5972 | 8.4581 |
| | 0.6250 | 5.6490 |
| | 0.6528 | 4.3208 |
| 5-6 | 0.5972 | 3.0954 |
| | 0.6111 | 9.2392 |
| | 0.6389 | 1.6309 |
| | 0.7222 | 2.4311 |
| 7-8 | 0.7083 | 6.0715 |
| | 0.6250 | 1.2554 |
| | 0.6528 | 4.5905 |
| | 0.6944 | 5.3698 |
| 9-10 | 0.6389 | 8.6608 |
| | 0.5833 | 2.1598 |
| | 0.6111 | 3.4214 |
| | 0.6389 | 2.3338 |
| 11-12 | 0.6667 | 1.4219 |
| | 0.6944 | 3.6871 |
| | 0.5694 | 5.6635 |
| | 0.7361 | 1.8495 |
| 13-14 | 0.6250 | 3.7090 |
| | 0.6944 | 8.8985 |
| | 0.6111 | 4.1640 |
| | 0.5278 | 3.7135 |
| 15-16 | 0.6250 | 6.6572 |
| | 0.5139 | 5.3988 |
| | 0.5972 | 12.7358 |
| | 0.6528 | 3.6467 |

**Tabla C-1**.: Resultados de las pruebas realizadas para obtener el par de electrodos que brinda más información.

# D. Anexo: Vídeo evidencia del manejo del drone con señales EEG

A continuación se anexa un link a un vídeo en youtube, donde se puede ver el control del drone a través de señales EEG.
`https://www.youtube.com/watch?v=Vi2YhFofQYw&t=1s`

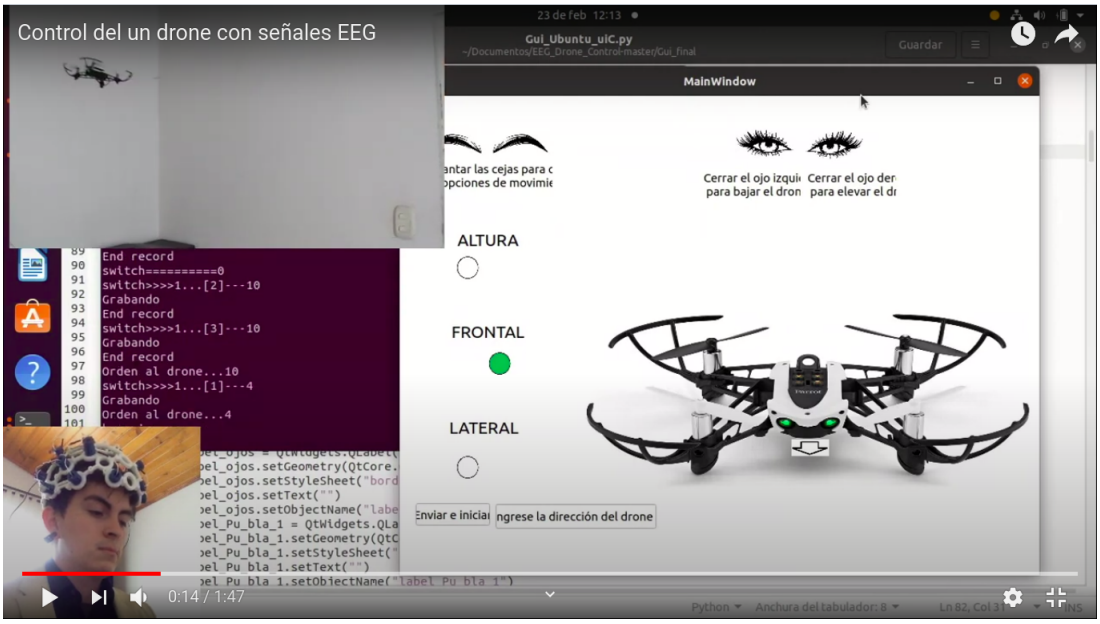



**Figura D-1**.: Captura del vídeo evidencia al que dirige el link.

# Bibliografía


[sci, 2020] (2020). *Preprocessing data*. Scikit learn, Accedido en 23-08-2020 https://scikit-learn.org/stable/modules/preprocessing.html.

[Arriola, 2016] Arriola, J. M. (2016). Representación matemática de ondas cerebrales. Master's thesis, UNIVERSIDAD NACIONAL DEL SUR.

[Boris Medina, 2018] Boris Medina, Javier E. Sierra, A. B. U. (2018). Técnicas de extracción de características de señales eeg en la imaginación de movimiento para sistemas bci. *Espacios*, 39.

[Bratislava, 2002] Bratislava, S. (2002). Fundamentals of eeg measurement. *MEASUREMENT SCIENCE REVIEW*, 2.

[DANE, 2010] DANE (2010). Departamento administrativo nacional de estadística, población con registro para la localización y caracterización de las personas con discapacidad – informe de investigación @ONLINE.

[Hernández, 2016] Hernández, F. O. R. (2016). Sistema clasificador para ondas cerebrales de acciones simples para la interacción con robots. Master's thesis, Universidad Veracruzana.

[McGovern, 2018] McGovern, A. (2018). *pyparrot's documentation*. scikit learn.

[OMS, 2011] OMS (2011). Organización mundial de la salud, resumen informe mundial sobre la discapacidad @ONLINE.

[OpenBCI, 2020] OpenBCI (2020). Openbci: Documentation, tipo @ONLINE. Accedido en 14-08-2020 a https://docs.openbci.com/docs/Welcome.htmldocs-website-update.

[Oscar, 2019] Oscar (2019). El homúnculo de penfield. Technical report, Colegios Montessori Bellavista.

[Purves, 2007] Purves, Augustine, F. H. L. A. L. M. W. (2007). *Neurociencia*. Editorial médica Panamericana, tercera edición edition.

[Python, 2020] Python (2020). python tm: About, tipo @ONLINE. Accedido en 14-08-2020 a https://www.python.org/.

[ROBERTO HORNERO, 2012] ROBERTO HORNERO, REBECA CORRALEJO, D. (2012). Brain-computer interface (bci) aplicado al entrenamiento cognitivo y control domótico para prevenir los efectos del envejecimiento. *Lychnos*, 08.

[Roman, 2019] Roman, V. (2019). Machine learning: Cómo desarrollar un modelo desde cero, tipo @ONLINE. *Ciencia y Datos, Accedido en 21-08-2020 a https://medium.com/datos-y-ciencia/machine-learning-c %C3 %B3mo-desarrollar-un-modelo-desde-cero-cc17654f0d48.*

[Saeid Sanei, 2007] Saeid Sanei, J. A. C. (2007). *EEG Signal Processing*. John Wiley and Sons Ltd, primera edición edition.

[SANTAMARIA, 2018] SANTAMARIA, P. (2018). Xataka: Parrot mambo fpv, análisis: el dron más divertido para volar sin miedo, tipo @ONLINE. Accedido en 14-08-2020 a https://www.xataka.com/analisis/parrot-mambo-fpv-analisis-el-dron-mas-divertido-para-volar-sin-miedo.

[Shmueli, 2019a] Shmueli, B. (2019a). Multi-class metrics made simple, part i: Precision and recall, tipo @ONLINE. *Towards Data Science, Accedido en 21-01-2021 a https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2.*

[Shmueli, 2019b] Shmueli, B. (2019b). Multi-class metrics made simple, part ii: the f1-score, tipo @ONLINE. *Towards Data Science, Accedido en 21-01-2021 a https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1.*

[Universidad a Distancia de Madrid, 201] Universidad a Distancia de Madrid, U. (201). Youtube: La corteza cerebral, tipo @ONLINE. Accedido en 14-08-2020 a https://www.youtube.com/watch?v=XTWTo8jB$_z$8.

[Álvaro González Velasco, 2012] Álvaro González Velasco (2012). El movimiento humano. Technical report, Universidad de Salamanca.